# Wastewater Surveillance for Emerging Infectious Diseases: Optimization and Implementation of Methods

Sarah Elizabeth Philo

A dissertation

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

University of Washington

2023

Reading Committee:

John Scott Meschke, Chair

Erica R. Fuhrmeister

Ricardo Santos

Program Authorized to Offer Degree:

Environmental and Occupational Health Sciences

University of Washington

## Abstract

Wastewater Surveillance for Emerging Infectious Diseases: Optimization and Implementation of Methods

Sarah Elizabeth Philo

Chair of the Supervisory Committee:

John Scott Meschke

Department of Environmental and Occupational Health Sciences

Wastewater surveillance has long been used to aid the global public health effort to eradicate poliovirus. More recently, wastewater surveillance has been implemented to help understand the SARS-CoV-2 pandemic. Shortly after it was discovered to be shed in stools, the virus was detected in sewage around the world. While there is no evidence of SARS-CoV-2 transmission through contact with feces or sewage, wastewater surveillance has helped the public health response. It has been used to study trends, emergence into a community, and the specific variants spreading. The rapid expansion of wastewater surveillance for SARS-CoV-2 has led to numerous calls to include

additional targets such as influenza viruses and antimicrobial-resistant bacteria. Antimicrobial resistance is of added concern because it is not well known how the SARS-CoV-2 pandemic itself contributed to rising resistance. The wastewater surveillance methods being used for SARS-CoV-2 and antimicrobial resistance are not well characterized for these targets because they were developed for enteric pathogens. Additionally, there are no standardized data reporting formats to provide those results to public health officials. This has resulted in confusion about how to do wastewater surveillance and what results from those projects mean in the larger context of public health. This dissertation aims to validate and optimize wastewater surveillance methods for SARS-CoV-2 and antimicrobial resistance. While not all methods will work for all targets, the methods laid out in this project can help validate protocols for additional pathogens.

**Acknowledgments**

My project could not have been completed without the guidance of my PhD advisor, Scott Meschke, whose advice was always welcome. Additional thanks to the members of my dissertation committee. I am a better scientist because of your input and support. This project could also not have been possible without our collaborators at the King County Wastewater treatment division, who were always willing to provide me samples and data when requested. A very special thanks to the Laboratório Análises do Instituto Superior Técnico, who hosted me for a year and helped with every aspect of my project in Portugal. I am also extremely grateful to have been funded by different organizations throughout my time at the University of Washington, including the Department of Environmental and Occupational Health Sciences, the Biostatistics, Epidemiologic, and Bioinformatics Training in Environmental Health (BEBTEH) fellowship, the Northwest Center for Occupational Health and Safety, and the US Fulbright Student Program.

Thank you also to all members, past and present, of the Environmental and Occupational Health Microbiology Lab. Your camaraderie through long hours in the lab is greatly made my time at the University of Washington infinitely more enjoyable. Many thanks also to my fellow graduate students throughout the department for their moral support. Finally, I would be remiss if I did not mention the support of my friends and family, particularly my partner Trey, who constantly believed in me and supported me throughout this long and stressful process. I would not have been able to do this without them all.

# Table of Contents

# 1 Literature Review

## 1.1 History of Environmental Surveillance for Infectious Diseases

Environmental surveillance (ES) has most notably been used for the control and elimination of poliovirus. Poliovirus is a particularly good target for ES for several reasons. First, it is secreted in the feces of almost all infected individuals within two weeks of infection and can continue for more than a month after symptom onset (Melnick *et al.*, 1946, Metcalf *et al.*, 1995, Centers for Disease Control and Prevention, 2021). Additionally, excreted virus remains infectious in feces and the environment and is transmitted fecal-orally after contact with fecally contaminated food or water (Murray *et al.*, 2013, Asghar *et al.*, 2014, Centers for Disease Control and Prevention, 2021). Even more importantly, most individuals who are infected with poliovirus do not show any symptoms, with only roughly one percent of infections resulting in acute flaccid paralysis (AFP) (Murray *et al.*, 2013, Centers for Disease Control and Prevention, 2021). This “silent” transmission makes traditional clinical surveillance for AFP difficult in resource limited settings. Finally, there is an important public health utility to ES: detection of poliovirus in the environment can target specific populations for vaccination campaigns (Asghar *et al.*, 2014).

ES for poliovirus has been a crucial tool in the progress towards eradication of the virus. The Global Poliovirus Eradication Initiative (GPEI) was launched at the World Health Assembly in 1988 and sought to eradicate poliovirus by the year 2000 (Hovi *et al.*, 2012). While the year 2000 eradication goal was not met, wild-type poliovirus type 2 was declared eradicated in September 2015, and type 3 was declared eradicated in October 2019 (Global Polio Eradication Initiative). The World Health Organization (WHO) published guidelines for poliovirus ES in 2003, formally acknowledging its role in eradication (The World Health Organization, 2003). These guidelines still recognize AFP clinical surveillance as the gold standard (Hovi *et al.*, 2012), but note that ES is crucial where clinical surveillance is difficult and in locations where the virus is

endemic or has frequent re-introduction events (The World Health Organization, 2003). Routine ES is carried out in Pakistan and Afghanistan, the only two countries in the world with active community transmission. In Pakistan, wild-type poliovirus has been detected in the environment in regions without any AFP cases (Asghar *et al.*, 2014). Using the WHO method, both wild-type poliovirus and vaccine derived poliovirus (viruses from the live-attenuated vaccines that regained the ability to cause neurological symptom) can be detected, crucial for identifying the source of the viruses (Dowdle *et al.*, 2003). More recently, poliovirus was detected in wastewater in New York (Ryerson *et al.*, 2022) and London (Hill & Pollard, 2022). In both instances, the detected virus was linked to international travel.

## 1.2 Environmental Surveillance for Emerging Infectious Diseases

There has been a greater focus in the last few decades to better understand newly emerging pathogens. It is estimated that roughly 60 percent of existing human pathogens and 75 percent of emerging pathogens are zoonotic (Taylor *et al.*, 2001, Parvez & Parveen, 2017). Fortunately, not all emerging zoonotic pathogens are transmissible from person to person, as humans are frequently the dead end host (Woolhouse & Gowtage-Sequeria, 2005). There are a number of factors that make existing or emerging pathogens more likely to cause a pandemic. First, rapid spread from person to person, with an emphasis on respiratory transmission, is crucial (Hunter, 2007, Adalja *et al.*, 2018). Additionally, the pathogen must be virulent and result in a high mortality rate (Hunter, 2007). It is also essential that there be a large number of immunologically naive people (Adalja *et al.*, 2018). The growing human population is encroaching on wild areas, leading to increased contact with wild animals and their diseases (Woolhouse & Gowtage-Sequeria, 2005, Parvez & Parveen, 2017). Finally, with globalization, it is easier for diseases to spread around the globe (Woolhouse & Gowtage-Sequeria, 2005, Parvez & Parveen, 2017).

One suggested solution to better understanding emerging infectious diseases is searching for them in settings with a high risk for transmission. One Health has been proposed as a potential framework to search for emerging infectious diseases. One Health is the idea that human, animal, and environmental health is intertwined and research projects and public health programs should aim to incorporate all three sectors (Centers for Disease Control and Prevention, 2018). The One Health framework has been implemented to search for respiratory and diarrheal pathogens in live bird markets and swine farms in China (Anderson *et al.*, 2018, Borkenhagen *et al.*, 2018, Wang *et al.*, 2020), as well as in North Carolina USA (Bailey *et al.*, 2020). While molecular assays routinely detect potentially zoonotic viruses, it is often difficult to isolate live viruses from environmental samples (Bailey *et al.*, 2020). Nevertheless, environmental sampling can shed light on high risk areas for infectious disease transmission.

### 1.2.1 Pandemic Coronaviruses

Coronaviruses have long been considered viruses with the potential to cause pandemics. Because they have an RNA genome, they are more mutable than DNA viruses and can more easily adapt to new hosts (Adalja *et al.*, 2018). SARS-CoV-1 is one of the more well-known recently emerged coronaviruses. It emerged in autumn 2001 in Guangdong Province, China, and eventually spread to 29 countries, resulting in over 8,000 cases and more than 900 deaths (Cherry & Krogstad, 2004). One of the highly published outbreaks occurred at the Amoy Gardens apartment complex in Hong Kong (Cherry & Krogstad, 2004, McKinney *et al.*, 2006). A man with diarrhea (later a confirmed SARS-CoV-1 case) was visiting his brother who lived in one of the apartment blocks. Ultimately, 321 Amoy Gardens residents were infected, with 41% of cases in the same apartment block. It was later determined that the u-traps in the building's plumbing were not properly sealed with water (McKinney *et al.*, 2006); when toilets flushed in the apartment block, the contaminated fecal matter aerosolized and spread to other units (Cherry & Krogstad, 2004, McKinney *et al.*, 2006). Additionally, the building's ventilation caused the aerosols to travel to other buildings, resulting in cases in other buildings. This outbreak

indicates that SARS-CoV-1 was viable in feces and it spread via aerosolized stool (McKinney *et al.*, 2006).

SARS-CoV-2, the causative agent of COVID-19, was first reported in December of 2019 in the Hubei Province of China, and was later declared a pandemic on March 11, 2020 (World Health Organization, 2020, World Health Organization, 2020, March 11). In addition to respiratory and febrile symptoms, symptoms of infection may include diarrhea (Zhang *et al.*, 2020). Even though SARS-CoV-2 is not transmitted fecal-orally, it was shown that the virus could be shed in infected individuals' stool (Wang *et al.*, 2020). Given the symptoms, detection in stool, and the documented SARS-CoV-1 outbreak associated with fecal material, research groups around the world started doing ES for SARS-CoV-2. Early adopters of wastewater surveillance were located in the Netherlands (Medema *et al.*, 2020), Australia (Ahmed *et al.*, 2020), and the United States (Peccia *et al.*, 2020). Since then, ES for SARS-CoV-2 has expanded to over 200 universities in more than 50 countries (Naughton *et al.*, 2021). It was initially proposed that ES for SARS-CoV-2 could serve as an early warning of emergence or re-emergence in a community, provide information about the size of the infected population, or confirm the absence of circulation (Medema *et al.*, 2020). More recently, wastewater surveillance has been used to give information about infection trends and the variants circulating in a community (Water Research Foundation, 2020, Yu *et al.*, 2021, Solis-Moreira, 2022).

Despite the wide adoption of wastewater surveillance for SARS-CoV-2, the methods being used are diverse and data reporting is not standardized. Methods include PEG precipitation (Zhang *et al.*, 2020), ultrafiltration (Medema *et al.*, 2020, Bertrand *et al.*, 2021), membrane filtration (Ahmed *et al.*, 2020, Gonzalez *et al.*, 2020, Rimoldi *et al.*, 2020), hollow-fiber filtration (Gerrity *et al.*, 2021, Monteiro *et al.*, 2022), and two-phase separation (La Rosa *et al.*, 2020), among others (Ahmed *et al.*, 2020). A group of experts created a list of information that should be reported with SARS-CoV-2 wastewater data to aid with interpretation and

comparability (McClary-Gutierrez *et al.*, 2021). Information such as structure of the wastewater treatment system, how the samples were collected and processed, recovery controls, and target quantification are all included in this list because each of these factors can affect the interpretation and representativeness of the collected sample (Medema *et al.*, 2020). But, according to a review of studies of SARS-CoV-2 wastewater surveillance, most papers do not report quality control data such as these (Ahmed *et al.*, 2020). Not reporting this data ultimately affects the public health utility. Without quality control data, applying results to the community or comparing results between different laboratories can be challenging (Ahmed *et al.*, 2020). Although numerous platforms have been developed to facilitate discussion (Bivins *et al.*, 2020, Naughton *et al.*, 2021), there is no binding agreement on reporting guidelines.

## 1.3 Antimicrobial Resistance

Antimicrobial resistance (AMR) is proving to be one of the largest threats to modern medicine (Adalja *et al.*, 2018). In the year 2015 alone, there were an estimated 671,689 AMR infections total, with 427,277 nosocomial infections resulting in an estimated 33,110 deaths (Cassini *et al.*, 2019). The number of deaths around the world each year could reach as high as 10 million if no substantial actions are pursued (Interagency Coordination Group on Antimicrobial Resistance, 2019). When antibiotics were first discovered, they were seen as a miracle of modern medicine. Penicillin was discovered in 1928 and widely used during WWII, but, within a few years, penicillin resistant *Staphylococcus aureus* species were isolated (Saga & Tamaguchi, 2009, Hutchings *et al.*, 2019). This pattern repeated itself with other antibiotics. Methicillin was first used clinically in 1960 and by 1961 MRSA appeared (Saga & Tamaguchi, 2009). As first-generation antibiotics show increased resistance, patients must be treated with second- and third- generation antibiotics. This is problematic because the rate of discovery of new antibiotics has drastically slowed (Saga & Tamaguchi, 2009, Hutchings *et al.*, 2019). There have been five antibiotics approved for clinical use since 2000, but many of those were

discovered decades ago. Because of how long it takes to bring a discovered compound to the market for clinical use, it could be years or decades for new antibiotics to be widely available (Hutchings *et al.*, 2019). Because of this, the world needs to re-think how antibiotics are discovered, used, and surveilled. ES can help provide a global picture of AMR.

### 1.3.1 Antimicrobial Resistance and the Environment

The environment is mostly not included in surveillance programs for AMR. The first international AMR surveillance program, the European Antimicrobial Resistance Surveillance network (EARS-Net), only included data from clinical labs (Aarestrup & Woolhouse, 2020). Later the WHO Global Antimicrobial Resistance Surveillance System standardized surveillance in clinical settings but did not include standards or guidance for environmental surveillance (Tornimbene *et al.*, 2018). One of the more recent surveillance programs to include the environment, the Global Tricycle Surveillance project, only focuses on extended-spectrum beta lactam (ESBL) resistant *E. coli* (World Health Organization, 2021). It is crucial to include human-impacted environments in AMR surveillance because places such as wastewater treatment plants (WWTPs) may help better understand AMR in clinical settings. WWTPs are recognized as hotspots of antimicrobial resistance because antibacterial-resistance genes (ARGs) are frequently detected at high numbers (Rizzo *et al.*, 2013). Additionally, mobile genetic elements (MGEs) are routinely identified in WWTPs, suggesting they facilitate transfer of ARGs between bacteria (Rizzo *et al.*, 2013, Guo *et al.*, 2017). Horizontal transfer of genes facilitated by MGEs is of particular concern in WWTPs, as naturally occurring ARGs may easily move into clinically relevant bacteria (Huijbers *et al.*, 2019). It is crucial to understand how ARGs move between environmental and clinical bacteria, and how these two different communities are related (Huijbers *et al.*, 2019).

As was previously discussed, substantial work has been done to characterize ES for SARS-CoV-2. There is currently a strong push to take the lessons learned with SARS-CoV-2 to new topics, including AMR (Pruden *et al.*, 2021). Guidelines have recently been published that

lay out method options for different AMR research goals (Liguori *et al.*, 2022). Refocusing the ES skills and knowledge obtained during the SARS-CoV-2 pandemic on AMR can help fill the gaps to better characterize the fate and transport of AMR.

### 1.3.2 Antimicrobial Resistance and SARS-CoV-2

As SARS-CoV-2 first emerged, patients were frequently prescribed antibiotics because there were no established treatment regimens. Even though antibiotics were being investigated for clinical efficacy against SARS-COV-2 infection (Chibber *et al.*, 2020), they had not been shown to be effective. Despite a lack of efficacy data, 85% of a cohort of over 50,000 hospitalized patients from February to June 2020 were treated with antibacterial medications (Russell *et al.*, 2021) even though secondary bacterial infections were low, with just over 1,000 patients having a COVID-19 related bacterial infection (Russell *et al.*, 2021). In a survey carried out with 480 randomly selected pharmacists, 67% prescribed antibiotics for COVID-19 symptoms and 82% provided antibiotics upon a physician's prescription (Elsayed *et al.*, 2021). All doctors surveyed in North America and roughly 60% of doctors in Portugal prescribed antibiotics to hospitalized COVID-19 patients (Beović *et al.*, 2020). Because the use of antibiotics leads to increased resistance (Hutchings *et al.*, 2019), it is likely that prescription of antibiotics in the SARS-CoV-2 pandemic increased AMR. Almost every organism tracked by the US Centers for Disease Control and Prevention (CDC) increased in resistance from 2019 to 2022 (Centers for Disease Control and Prevention *et al.*, 2022). However, data reporting is incomplete due to the pandemic. Wastewater surveillance for AMR may help fill some of the holes in clinical data.

There are very few studies attempting to understand if antibiotic use in the COVID-19 pandemic resulted in detectable changes in ARG abundance in wastewater. One study carried out in Nevada collected samples in late 2020 and early 2021 during a surge in SARS-CoV-2 cases (Harrington *et al.*, 2022). They were able to show that environmental AMR was statistically higher during SARS-CoV-2 peaks compared to periods with fewer cases, and the

ARGs detected coincided with resistance in clinical settings and the antibiotics used to treat infections (Harrington *et al.*, 2022). In the absence of clinical data, projects like this can supplement traditional surveillance programs. However, the methods being used must be validated to carry out surveillance for the specific target of interest, either AMR bacteria or ARGs themselves.

# 2 SARS-CoV-2 Wastewater Surveillance

## 2.1 A comparison of SARS-CoV-2 wastewater concentration methods for environmental surveillance

As published: **Philo SE**, Keim EK, Swanstrom R, Ong AQW, Burnor EA, Kossik AL, Harrison JC, Demeke BA, Zhou NA, Beck NK, Shirai JH, Meschke JS. A comparison of SARS-CoV-2 wastewater concentration methods for environmental surveillance. *Science of The Total Environment*. 2021;760:144215. doi:10.1016/j.scitotenv.2020.144215 PMID – 33340739

### 2.1.1 Abstract

Wastewater surveillance of SARS-CoV-2 may be a useful supplement to clinical surveillance as it is shed in feces, there are many asymptomatic cases, and diagnostic testing can have capacity limitations and extended time to results. Although numerous studies have utilized wastewater surveillance for SARS-CoV-2, the methods used were developed and/or standardized for other pathogens. This study evaluates multiple methods for concentration and recovery of SARS-CoV-2 and seeded human coronavirus OC43 from municipal primary wastewater and/or sludge from the Greater Seattle Area (March-July 2020). Methods evaluated include the bag-mediated filtration system (BMFS), with and without Vertrel$^{TM}$ extraction, skimmed milk flocculation, with and without Vertrel$^{TM}$ extraction, polyethylene glycol (PEG) precipitation, ultrafiltration, and sludge extraction. Total RNA was extracted from wastewater concentrates and analyzed for SARS-CoV-2 and OC43 with RT-qPCR. Skimmed milk flocculation without Vertrel$^{TM}$ extraction performed consistently over time and between treatment plants in Seattle-area wastewater with the lowest average OC43 $C_q$ value and smallest variability (24.3; 95% CI: 23.8-24.9), most frequent SARS-CoV-2 detection (48.8% of sampling events), and highest average OC43 percent recovery (9.1%; 95% CI: 6.2-11.9%). Skimmed milk flocculation is also beneficial because it is feasible in low-resource

settings. While the BMFS had the highest average volume assayed of 11.9mL (95% CI: 10.7-13.1mL), the average OC43 percent recovery was low (0.7%; 95% CI: 0.4-1.0%). Ultrafiltration and PEG precipitation had low average OC43 percent recoveries of 1.0% (95% CI: 0.5-1.6%) and 3.2% (95% CI: 1.3-5.1%), respectively. The slopes and efficiency for the SARS-CoV-2 standard curves were not consistent over time, confirming the need to include a standard curve each run rather than using a single curve for multiple plates. Results suggest that the concentration and detection methods used must be validated for the specific water matrix using a recovery control to assess performance over time.

### 2.1.2 Introduction

In December 2019, an outbreak of pneumonia of unknown etiology associated with the live animal market in the Hubei Province of China was first reported to the World Health Organization (WHO) (World Health Organization, 2020). This pneumonia would later be classified as COVID-19, the disease caused by the novel coronavirus SARS-CoV-2. SARS-CoV-2 is a novel coronavirus like those that caused severe acute respiratory syndrome (SARS) in 2003 and Middle East respiratory syndrome (MERS) in 2012 (Fani *et al.*, 2020). SARS-CoV-2 spread quickly throughout the world and was declared a pandemic by the WHO on March 11, 2020 (World Health Organization, 2020, March 11).

Symptoms of COVID-19 may include fever, dry cough, tiredness, sore throat, body aches, and diarrhea (Zhang *et al.*, 2020). Because viral shedding can occur before an individual becomes symptomatic, the rapid worldwide spread can partially be attributed to transmission by pre-symptomatic and asymptomatic individuals (He *et al.*,

2020). Additionally, individuals infected with SARS-CoV-2 shed the virus in their stool (Wang *et al.*, 2020). Together, this suggests that wastewater can be easily collected to conduct disease surveillance in low prevalence areas or before clinical cases are identified (Mallapaty, 2020, Randazzo *et al.*, 2020).

Public Health departments and research groups around the world are conducting wastewater surveillance for SARS-CoV-2 using various methods, including ultrafiltration, polyethylene glycol (PEG) precipitation, direct sludge extraction, and skimmed milk flocculation (Ahmed *et al.*, 2020, Medema *et al.*, 2020, Peccia *et al.*, 2020). However, many of the methods being used have been optimized for non-enveloped enteric viruses. Because SARS-CoV-2 is an enveloped virus, the existing methods are not optimized for its surveillance. Additionally, wastewater profiles can vary greatly due to geography, population, treatment processes at the plant, and where the sample is collected. A comparison of concentration methods is necessary to develop effective environmental wastewater surveillance and to standardize the methods being used across communities. To compare and optimize methods currently being used around the world, the following methods were carried out on primary wastewater from three wastewater treatment plants (WWTPs) in the Seattle area: the Bag-Mediated Filtration System (BMFS) (Zhou *et al.*, 2019), modified skimmed milk flocculation from Calgua *et al.* (2008) (Falman *et al.*, 2019), polyethylene glycol precipitation (PEG) (Lewis & Metcalf, 1988, Falman *et al.*, 2019), and ultrafiltration via Millipore filtration concentration (Ahmed *et al.*, 2020). Direct sludge extraction was also carried out on primary sludge samples from the same WWTPs. This methods comparison evolved as the pandemic evolved and was not designed to result in statistical comparisons, but

rather to describe how the methods perform using a recovery control. This is the first SARS-CoV-2 wastewater concentration methods comparison performed in the United States to date and provides useful guidance for monitoring wastewater for SARS-CoV-2.

## 2.1.3 Methods

### 2.1.3.1 Generation of OC43 Stock

Human coronavirus OC43 (OC43; ATCC VR-1558) stocks were prepared by infecting confluent HCT-8 cells (ATCC CCL-244) in flasks at 33°C in RPMI-1640, 2% fetal bovine serum, and vancomycin/gentamycin. Six days post infection, virus was harvested by freeze/thawing. Lysates were clarified by centrifugation 2500xg for 20 minutes at 4°C. Supernatant was collected and stored at -80°C or was further concentrated by PEG precipitation. Further PEG precipitation included adding 9% PEG-8000 (Sigma-Aldrich, St. Louis, MO, USA) and 0.5 M NaCl (ThermoFisher Scientific, Waltham, MA, USA) to supernatant and shaking overnight at 5°C. PEG slurry was centrifuged at 5,500x g for 30 minutes at 4°C. Virus pellets were resuspended for a 10-fold concentration in PBS and stored at -80°C.

Stocks of OC43 were quantified using a modified version of a previously described immunoperoxidase assay by fixing cells 4 dpi using 2% paraformaldehyde (Lambert *et al.*, 2008). Fixed cells were permeabilized using PBS with 0.5% TritonX-100 and 20mM Glycine. Cells were labeled and stained using Anti-Coronavirus OC43 nucleoprotein monoclonal mouse antibody at 1:900 (MilliporeSigma, MAB9013), IgG (H+L) Cross-absorbed Goat anti-Mouse HRP at 1:1000 (Invitrogen, G21040) and

Thermo-Scientific Pierce DAB substrate kit. Infected wells were counted, and TCID50/ml was calculated using the Spearman & Kärber algorithm.

#### 2.1.3.2 Wastewater Concentration

Primary wastewater was grab sampled weekly from three Seattle area wastewater treatment plants from late-March to July 2020 (Appendices Table 2.5.1). All wastewater was stored at 4°C and used within one week of collection. For 1.0L grab samples collected over the course of a single day, they were composited and mixed prior to seeding with OC43. OC43 was seeded at a concentration of $3.3x10^4$ $TCID_{50}$/L of wastewater.

Seeded wastewater samples were aliquoted and concentrated using four methods: the BMFS (with and without Vertrel™ extraction), skimmed-milk flocculation (with and without Vertrel™ extraction), PEG precipitation, or ultrafiltration (Table 2.1.1) Full protocols are included in appendices section 2.5.1.2. The BMFS concentrated an average of 2.63L (Table 2.1.1) of primary wastewater using previously published methods, including filtration, elution, and secondary concentration with a two-hour

**Table 2.1.1**: Number of total samples for each concentration method and average volume sampled across all weeks.

| | | Plant A | | Plant B | | Plant C | |
|---|---|---|---|---|---|---|---|
| | | *Number* | *Average Volume (L)* | *Number* | *Average Volume (L)* | *Number* | *Average Volume (L)* |
| **BMFS** | Vertrel | 15 | 1.23 | 15 | 1.23 | 15 | 2.63 |
| | No Vertrel | 3 | | 3 | | 3 | |
| **Skimmed Milk** | Vertrel | 0.25L = 15<br>0.5L = 1<br>0.05L = 1 | 0.25 | 0.25L = 15<br>0.5L = 1<br>0.05L = 1 | 0.25 | 0.25L = 15<br>0.5L = 1<br>0.05L = 1 | 0.25 |
| | No Vertrel | 0.25L = 15<br>0.5L = 1<br>0.05L = 1 | 0.25 | 0.25L = 15<br>0.5L = 1<br>0.05L = 1 | 0.25 | 0.25L = 15<br>0.5L = 1<br>0.05L = 1 | 0.25 |
| **Polyethylene Glycol Precipitation** | | 0 | - | 0 | - | 4 | 0.50 |
| **Ultrafiltration** | | 7 | 0.1 | 7 | 0.1 | 8 | 0.1 |
| **Sludge Extraction** | | 4 | 0.0025 | 6 | 0.0025 | 6 | 0.0025 |

shake at 200 RPM and pellet resuspension in 4mL of sterile PBS (pH 7.4) (Fagnant *et al.*, 2018, Falman *et al.*, 2019, Zhou *et al.*, 2019). Skimmed milk flocculation was also carried out on primary wastewater samples in volumes of 0.1L, 0.5L, and 1.0L following the protocol used during BMFS processing. Sample volumes varied to try to optimize the effective volume assayed and processing time, 0.5L was used most often because it best balanced these two priorities. Skimmed milk flocculation pellets were resuspended in either 4mL or 6mL of sterile PBS (pH = 7.4). Initial volumes of 0.1L were resuspended in 4mL, and initial volumes of 0.5L and 1.0L were resuspended in 6mL. Resuspensions of both BMFS and skimmed milk samples were then divided into two volumes: one for RNA extraction and the other to separate the viruses from the solids with Vertrel XF$^{TM}$ (Miller-Stephenson, Inc., Danbury, CT, USA) (Falman *et al.*, 2019). PEG precipitation was carried out on 0.5L of primary wastewater by shaking for four hours (4°C at 200RPM) and resuspended in 10mL of sterile PBS (pH = 7.4) following previously published methods (Falman *et al.*, 2019). There were a limited number of replicates of this method due to the time required for processing and restrictions on people and laboratory space. Ultrafiltration with Centricon Plus-70 centrifugal filter devices (MilliporeSigma, Burlington, MA, USA) were used to concentrate 0.1L of composite wastewater following manufacturer instructions. The number of ultrafiltration replicates was limited by supply chain constraints. Prior to ultrafiltration, 0.1L of wastewater was centrifuged at 6800 X G and 4°C for 30 minutes to pellet out the solids. The supernatant was applied to the filter device, discarded after filtration (repeated once to allow full volume to pass through), and the retentate retained for detection (Ahmed *et al.*, 2020).

Resuspension volumes for all methods are reported in supplemental material Figure 2.5.1. All concentrates were stored at -80°C for RNA extraction.

#### 2.1.3.3 RNA Extraction

RNA extraction was carried out on all concentrated wastewater in duplicate using the QIAamp Viral RNA Mini Kit (QIAGEN, Germantown, MD, USA). For liquid samples, the input volume was 280uL with the exception of ultrafiltration, which had an input volume of 140uL due to the small retentate volume generated. The ultrafiltration pellet from 50mL of sewage entered the kit via resuspension in Buffer AVL. Each sample was eluted in 60uL and duplicates were combined before being re-aliquoted into 60uL aliquots and frozen at -20°C.

#### 2.1.3.4 Sludge Extraction

Primary composite sludge was collected on the same day as primary wastewater and was not seeded with OC43 prior to extraction following published methods by Peccia, et al., 2020 (Peccia *et al.*, 2020). Briefly, viral RNA was extracted using the QIAGEN RNeasy PowerSoil Total RNA kit by adding 2.5mL of well mixed sludge directly to the commercial kit with an elution volume of 100 µL.

#### 2.1.3.5 RT-qPCR

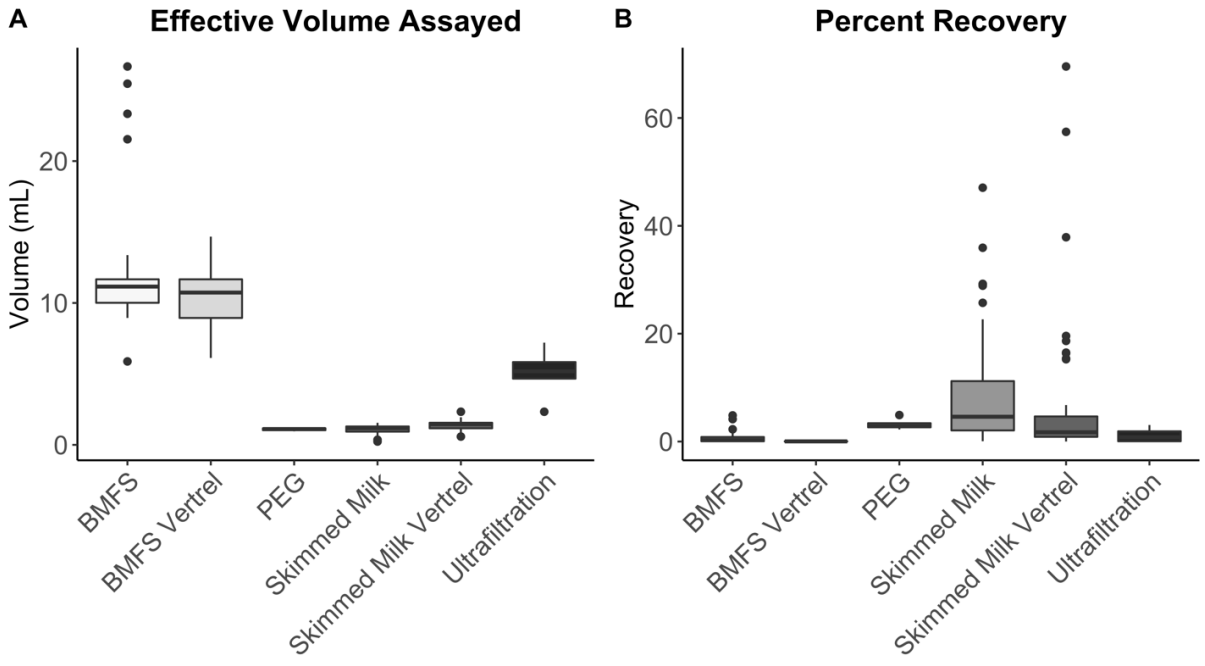
Reverse-transcription qPCR (RT-qPCR) for OC43 and SARS-CoV-2 was carried out on all RNA extracts using the iTaq Universal Probes One-Step Kit (Bio-Rad Laboratories, Hercules, CA, USA) with a total reaction volume of 20uL. All samples were run with both undiluted RNA extracts and $10^{-1}$ dilutions of the RNA extracts. OC43 was detected using a previously published protocol with 0.3uM of primers and 0.2uM of the FAM probe targeting the M protein, the membrane glycoprotein (Vijgen *et al.*, 2005). The US Centers for Disease Control and Prevention SARS-CoV-2 research-use only

detection kit provided by IDT (Integrated DNA Technologies, Inc., Coralville, IA, USA) was used targeting three regions of the N gene (US Centers for Disease Control and Prevention, 2020). RT-qPCR cycling conditions for both assays were 50°C for 10 minutes, 95°C for three minutes, and 45 cycles of 95°C for 15 seconds and 60°C for 30 seconds. Positive control standard curves for both viruses were done in duplicate using serial 10-fold dilutions in nuclease free water. Positive control standard curves for the N1, N2, and N3 assays were carried out using a plasmid control containing the target genes for SARS-CoV-2 (Integrated DNA Technologies, Inc., Coralville, IA, USA). Positive control standard curves for OC43 were generated from extractions of serially diluted human coronavirus OC43 (OC43; ATCC VR-1558) enumerated by $TCID_{50}$ on HCT-8 cells (ATCC CCL-244). Negative controls were nuclease free water. No negative controls had detections for either OC43 or SARS-CoV-2. An off-target control was used for each assay, with SARS-CoV-2 serving as this control for OC43 and OC43 serving as this control for the SARS-CoV-2 targets. There was no cross-reactivity in the assays. Limits of detection (LOD) and quantification (LOQ) were determined in nuclease free water and in RNA extracted skimmed milk flocculation concentrated wastewater using standard curves with ten replicates at each concentration. The LOD was determined as the concentration below which less than 90% of replicates are detected as positive (Burd, 2010). The LOQ is the lowest concentration with a coefficient of variation below 35% (Klymus *et al.*, 2020).

#### 2.1.3.6 Data Analysis

Bio-Rad CFX Maestro for Mac (Bio-Rad Laboratories, Hercules, CA, USA) was used to analyze all RT-qPCR tests, and data were collated and managed using Microsoft Excel (Microsoft Corp., Redmond, WA, USA). Cycle threshold levels were

manually set at the point where the positive controls start exponentially multiplying. Samples with non-exponential multiplication were considered false positives, and samples with reduced fluorescence as evident in the qPCR curves were considered inhibited. All samples with a Cq larger than 40 in both the undiluted and the $10^{-1}$ dilution




| C | BMFS | BMFS - Vertrel | PEG | Skimmed Milk | Skimmed Milk - Vertrel | Ultrafiltration |
|---|---|---|---|---|---|---|
| **Maximum** | 4.85 | 0.09 | 4.91 | 47.07 | 69.55 | 3.07 |
| **3rd Quartile** | 0.84 | 0.044 | 3.40 | 11.20 | 4.66 | 1.90 |
| **Median** | 0.29 | 0.030 | 2.83 | 4.60 | 1.72 | 0.81 |
| **1st Quartile** | 0.053 | 0.0078 | 2.64 | 2.06 | 0.87 | 0.051 |
| **Minimum** | 4.57E-5 | 6.01E-4 | 2.23 | 0.057 | 0.0016 | 0.0063 |
| **Mean** | 0.69 | 0.038 | 3.20 | 9.05 | 6.54 | 1.02 |
| **Non-Detect** | 5 | 0 | 0 | 3 | 3 | 5 |

**Figure 2.1.1**: Effective volume assayed and percent recovery for each method. A) The effective volume assayed is the proportion of the original wastewater sample assayed by RT-qPCR. The BMFS, with and without Vertrel, have the largest average effective volume assayed per reaction. PEG precipitation and skimmed milk flocculation, with and without Vertrel, have the smallest average effective volume assayed per reaction. B) The percent recovery is calculated using the standard curves for the RT-qPCR assay generated with each experimental run and the Cq value for each undiluted sample. Skimmed milk flocculation has the highest average percent recovery, followed by skimmed milk flocculation with Vertrel extraction. Both BMFS methods had the two lowest average percent recoveries. All methods tested had some non-detections in the undiluted sample except PEG precipitation and BMFS with Vertrel extraction. C) Percent recovery descriptive statistics by method.

were considered negative. All figures were generated using RStudio (RStudio, PBC, Boston, MA, USA).

## 2.1.4 Results and Discussion

### 2.1.4.1 Method Comparison using OC43 as a Surrogate Coronavirus

Multiple factors were considered when using OC43 to compare concentration methods including the effective volume assayed, OC43 percent recovery, detection frequency, detection consistency, and concentration. The total volume assayed for each method was assessed by calculating the proportion of the initial volume of concentrated wastewater that was assayed by RT-qPCR. The BMFS method, with and without Vertrel$^{TM}$ extraction, had the highest average effective volume assayed (11.90 and 10.41mL, respectively) when compared to the other four methods (Figure 2.1.1, Appendices Table 2.5.2). PEG precipitation and skimmed milk flocculation, with and without Vertrel$^{TM}$ extraction, had the three lowest average effective volume assayed of the methods of 1.09mL, 1.10mL, and 1.37mL, respectively (Figure 2.1.1, Appendices Table 2.5.2). Ultrafiltration had an average effective volume assayed of 5.31mL (Figure 2.1.1, Appendices Table 2.5.2).

The percent recovery for the spiked OC43 recovery control was calculated for each method using the standard curves generated for each RT-qPCR assay and the estimated spiked concentration (Equation 1, 2, 3).

$$Equation\ 1: Slope = \frac{\Delta C_q}{\Delta C_{OC43}}$$

$Where$: $C_q$ is the cycle quotient as determined using standard curves generated from OC43 stock solutions, $C_{OC43}$ is $\log concentration\ in \frac{TCID_{50}}{mL}$

$$Equation\ 2: C_{sample} = \log\left(OC43(TCID_{50}/mL)\right) = 10^{\frac{Cq-Intercept}{Slope}}$$

$$Equation\ 3: \%\ Recovery = \left(\frac{C_{sample}V_{sample}}{C_{inoc}V_{inoc}}\right)100$$

$Where$: $V_{sample}$ is volume of sample adjusted for amount entering qPCR,

$V_{inoc}$ is fraction of volume assayed relative to volume processed

The highest average percent recoveries across all methods were obtained for skimmed milk flocculation with Vertrel$^{TM}$ extraction (9%) and without (6%) (Figure 2.1.1B, C). Ultrafiltration and the BMFS, with and without Vertrel$^{TM}$ extraction, had the three lowest average percent recoveries across all methods compared (1.0%, 0.04%, and 0.7%, respectively) (Figure 2.1.1C). The only two methods that did not have non-detections in the undiluted sample were PEG precipitation and BMFS with Vertrel$^{TM}$ extraction (Figure 2.1.1C). However, these two methods had fewer total replicates compared to the other methods (Table 2.1.1) and are therefore not directly comparable.

Because the BMFS concentrates and assays the largest volumes of sewage of the methods tested, it is likely that the percent recovery is reduced due to inhibitors concentrated with the wastewater as larger volumes of concentrated water exhibit reduced detection (Loge *et al.*, 2002). Additionally, because all of the $10^{-1}$ diluted samples detected OC43 (data not presented), it is probable that inhibitors present in the un-diluted samples reduced or blocked detection. Our recovery values are similar to those reported for Bovine-Coronavirus (BCoV) (Jafferali *et al.*, 2020), but substantially lower than recoveries reported for murine hepatitis virus (MHV) (Ahmed *et al.*, 2020). MHV is an enteric virus and, therefore, potentially persists longer in sewage than OC43 or BCoV. These results emphasize the importance of using seeded recovery controls to correct for matrix effects and inhibition when selecting a method for SARS-CoV-2 wastewater surveillance.

To compare the efficiency of the methods across treatment plants and time, control charts were generated for each method (Figure 2.1.2). Upper and lower warning limits (UWL and LWL) were set to one standard deviation from the mean Cq value for

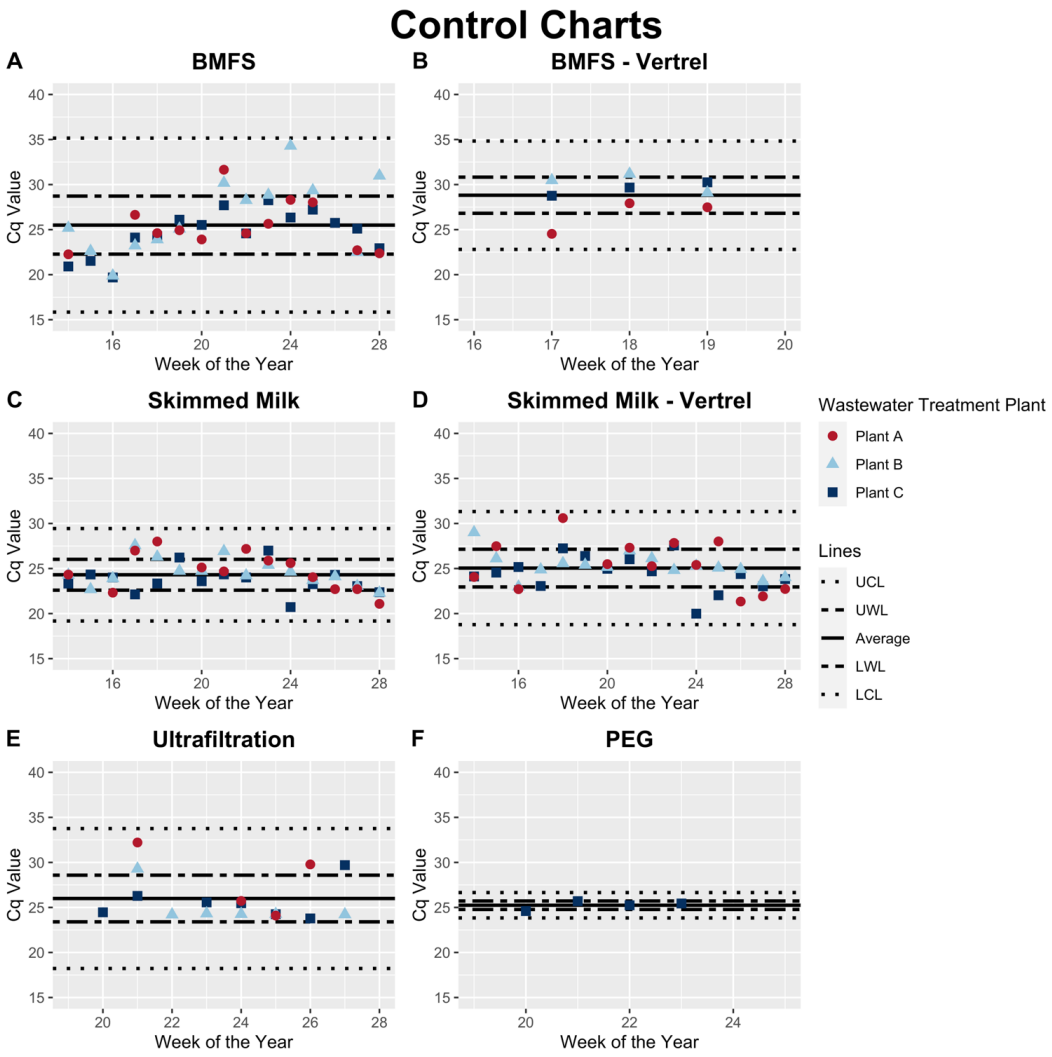



**Figure 2.1.3**: OC43 RT-qPCR control charts. Average Cq values were calculated for each method by averaging across treatment plants and time. The UWL and LWL, or upper and lower warning limits, for each method were calculated by adding or subtracting, respectively, the standard deviation from the average Cq. The UCL and LCL, or the upper and lower control limits, for each method were calculated by adding or subtracting, respectively, three times the standard deviation from the average Cq. Anything detected at or above a Cq of 40 was considered a non-detection. All samples that had non-detections by RT-qPCR in the undiluted samples reported here had detection in the $10^{-1}$ dilution. BMFS without Vertrel extraction (A) has a lower average Cq compared to BMFS with Vertrel extraction (B), but has a substantially larger range of data. Skimmed milk flocculation without Vertrel extraction (C) and with Vertrel extraction (D) have similar average Cq's, control limits, and warning limits. Ultrafiltration (E) had a similar average Cq to both skimmed milk methods, but had a larger variability in the data and fewer detections in the undiluted samples. PEG precipitation (F) had a low average Cq and variability around the mean, but only one treatment plant was tested with this method and it is therefore not directly comparable.

that concentration method. Upper and lower control limits (UCL and LCL) were set to three standard deviations from the mean Cq value for that concentration method.

The BMFS without Vertrel$^{TM}$ extraction has a lower mean Cq (25.6) compared to BMFS with Vertrel$^{TM}$ extraction (28.8) (Figure 2.1.2 A, B, Table 2.1.2). Both skimmed milk with and without Vertrel$^{TM}$ extraction have similar mean Cq values of 24.3 and 25.1, respectively (Figure 2.1.2 C, D, Table 2.1.2). However, skimmed milk with Vertrel$^{TM}$ extraction has a higher standard deviation and variance compared to skimmed milk without Vertrel$^{TM}$ extraction (Table 2.1.2). Ultrafiltration has a comparable mean Cq value (26.0) to the skimmed milk methods, but it has a much higher standard deviation and variance around the mean compared to both skimmed milk methods (Table 2.1.2).

The only two methods that detected OC43 from all undiluted samples were the BMFS with Vertrel$^{TM}$ extraction and PEG precipitation. However, these have fewer samples and are therefore not directly comparable. BMFS with Vertrel$^{TM}$ extraction and PEG precipitation were dropped early in the methods comparison because they were time consuming and did not add any additional detection power for SARS-CoV-2. Both skimmed milk flocculation methods had three non-detected samples by RT-qPCR in the undiluted reaction (all detected in $10^{-1}$ diluted reaction), and ultrafiltration had five non-detected samples in the undiluted reaction (all detected in $10^{-1}$ diluted reaction). The

**Table 2.1.3**: Descriptive statistics of OC43 Cq values by RT-qPCR for each method.

| | Average | Standard Deviation | Variance | Max | Min | Non-Detection | Number |
|---|---|---|---|---|---|---|---|
| **BMFS** | 25.5 | 3.2 | 10.4 | 34.3 | 19.7 | 5 | 15 |
| **BMFS - Vertrel** | 28.8 | 2.0 | 4.0 | 31.2 | 24.5 | 0 | 9 |
| **Skimmed Milk** | 24.3 | 1.7 | 2.9 | 28.0 | 20.7 | 2 | 15 |
| **Skimmed Milk - Vertrel** | 25.1 | 2.1 | 4.4 | 30.6 | 20.0 | 2 | 15 |
| **Ultrafiltration** | 26.0 | 2.6 | 6.7 | 32.2 | 23.8 | 7 | 22 |
| **PEG** | 25.2 | 0.5 | 0.2 | 25.7 | 24.6 | 0 | 4 |

BMFS had the highest number of non-detected samples with six samples being non-detected in the undiluted RT-qPCR reaction (Table 2.1.2). Therefore, all concentration methods were susceptible to inhibition.

Taken together, these results suggest that, of the evaluated methods, skimmed milk without Vertrel$^{TM}$ extraction performs well for detection of OC43. Skimmed milk flocculation without Vertrel$^{TM}$ extraction had the highest mean percent recovery (Figure 2.1.1B), consistent performance over time and across treatment plants (Figure 2.1.2C),

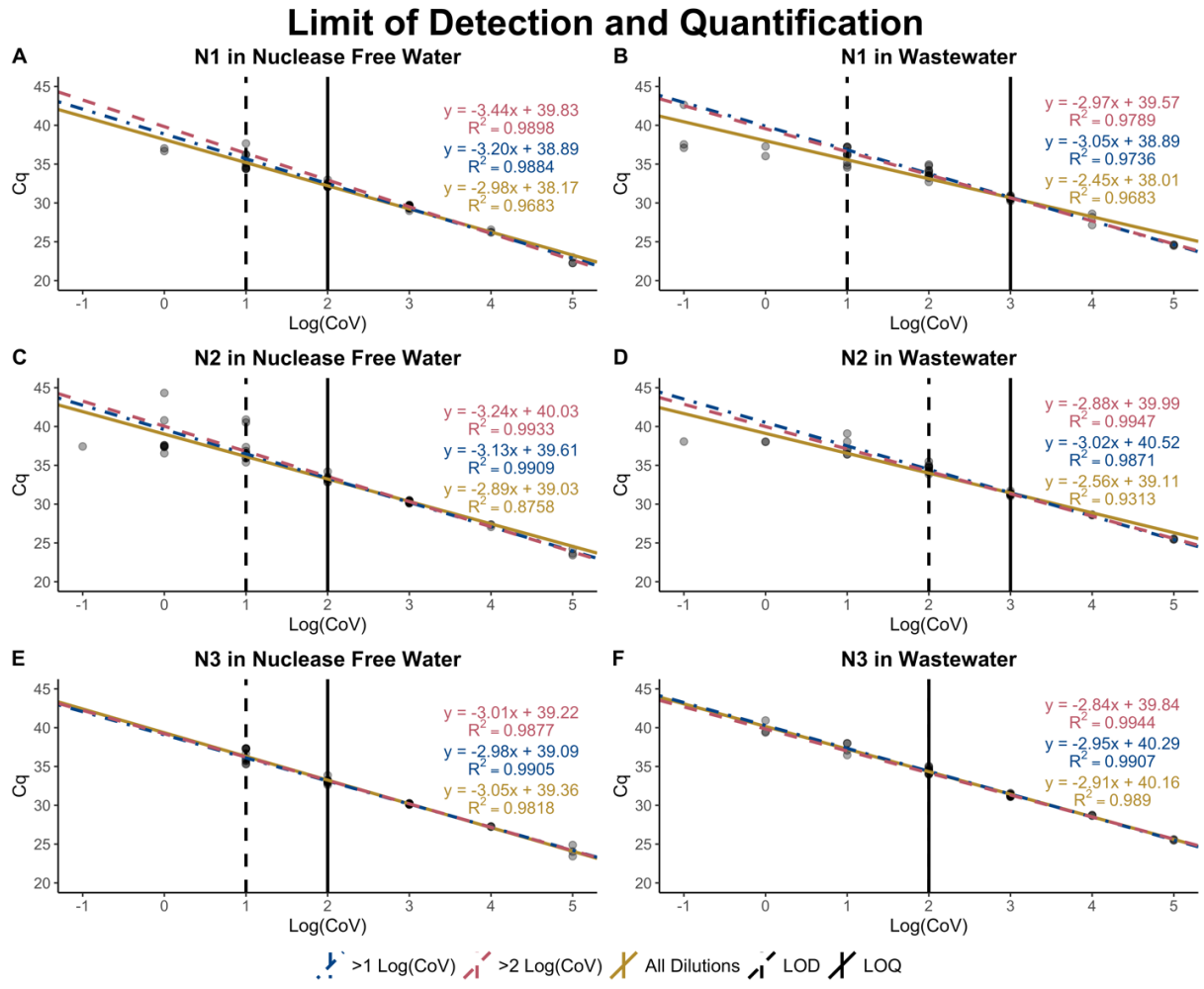



**Figure 2.1.5**: Limits of detection and quantification for SARS-CoV-2 assays. The limit of detection and limit of quantification for each SARS-CoV-2 assay is dependent on the water matrix. Greater inhibition of the assays were seen when standard curves were prepared using skimmed milk wastewater extracts as the diluent (B, D, F) than when standard curves were prepared using nuclease free water as the diluent (A, C, E), as seen by decreased efficiency and higher limits of detection. In panels D and F, the single vertical line indicates the LOD and LOQ are on top of each other.

lowest mean Cq for OC43 detection (Table 2.1.2), and lowest variability around the mean (Table 2.1.2). Although Vertrel$^{TM}$ extraction on the skimmed milk resuspensions performed similarly to the non- Vertrel$^{TM}$ extracted samples, Vertrel$^{TM}$ extraction is not an optimal step for human coronavirus detection as it was developed for non-enveloped viruses and added substantial processing time (Mendez *et al.*, 2000). Because human coronaviruses are enveloped, Vertrel$^{TM}$ extraction is not an ideal method to separate residual solids from the virions after pellet resuspension.

#### 2.1.4.2 RT-qPCR efficiency

By graphing all of the standard curves in nuclease free water for a single assay (OC43, N1, N2, or N3) on the same plot (Appendices Figures 2.5.2, 2.5.3) and extracting the slopes and intercepts for the standards when including all dilutions or the $10^{-1}$ through $10^{-3}$ dilutions, substantial variation in the standard curves for the three SARS-CoV2 assays is observed. This is evidenced by variable slopes and intercepts for each assay. OC43 standard curves in nuclease free water were much more consistent as the slopes and intercepts were less variable between assays. Variability in SARS-CoV-2 standard curves makes sample quantification difficult because of its poor-performance assay to assay.

The LOD for all three SARS-CoV-2 assays in nuclease free water is 10 gene copies (Figure 2.1.3), while the LOQ in nuclease free water is 100 gene copies. While the LOD for N1 in Seattle wastewater is the same as in nuclease free water (10 gene copies), the LOD is increased for N2 and N3 to 100 gene copies, likely due to inhibitors present in the water matrix (Loge *et al.*, 2002). Additionally, all three assays have reduced efficiency in wastewater compared to nuclease free water. The LOQ for N1 and N2 in Seattle wastewater is 1000 gene copies, and the LOQ for N3 in Seattle

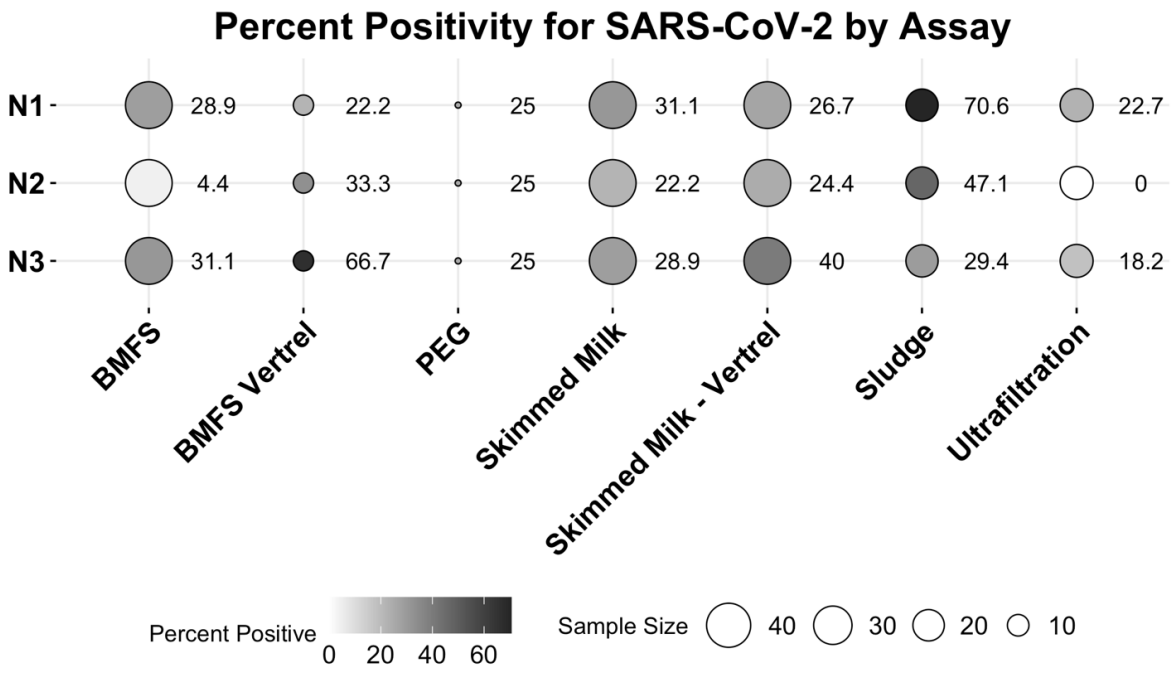



**Figure 2.1.7**: Percent positivity for SARS-CoV-2. Samples were considered positive if the $C_q \leq 40$. Direct sludge extraction had highest percent positive for N1 and N2 but had fewer samples. N3 had the highest percent positive in skimmed milk flocculation with Vertrel$^{TM}$ extraction. All three assays had roughly 30% positive with skimmed milk flocculation.

wastewater is 100 gene copies. This highlights the importance of validating assays in different water matrices and using recovery controls to quantify inhibition. The LOD and LOQ were not determined for the OC43 assay because all samples were detected within the ranges of standard curve and there was little variability in the standard curves over time.

#### 2.1.4.3 SARS-CoV-2 Results

Samples positive for SARS-CoV-2 were identified using the following criteria:

- Detection for at least one of the three SARS-CoV-2 assays (N1, N2, or N3) AND
- Amplification below $C_q = 40$ in non-diluted reaction OR
- Amplification below $C_q = 40$ in $10^{-1}$ diluted reaction.

| Presumptive Positive for SARS-CoV-2 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BMFS | | BMFS-V | | Skimmed Milk | | Skimmed Milk - V | | Sludge | | PEG | | Ultrafiltration | |
| | % | n | % | n | % | n | % | n | % | n | % | n | % | n |
| Undiluted | 33.3 | 45 | 33.3 | 9 | 48.9 | 45 | 51.1 | 45 | 70.6 | 21 | 25 | 4 | 27.3 | 22 |
| 1:10 Dilution | 46.7 | 45 | 46.7 | 9 | 57.8 | 45 | 57.8 | 45 | 76.5 | 21 | 25 | 4 | 68.2 | 22 |

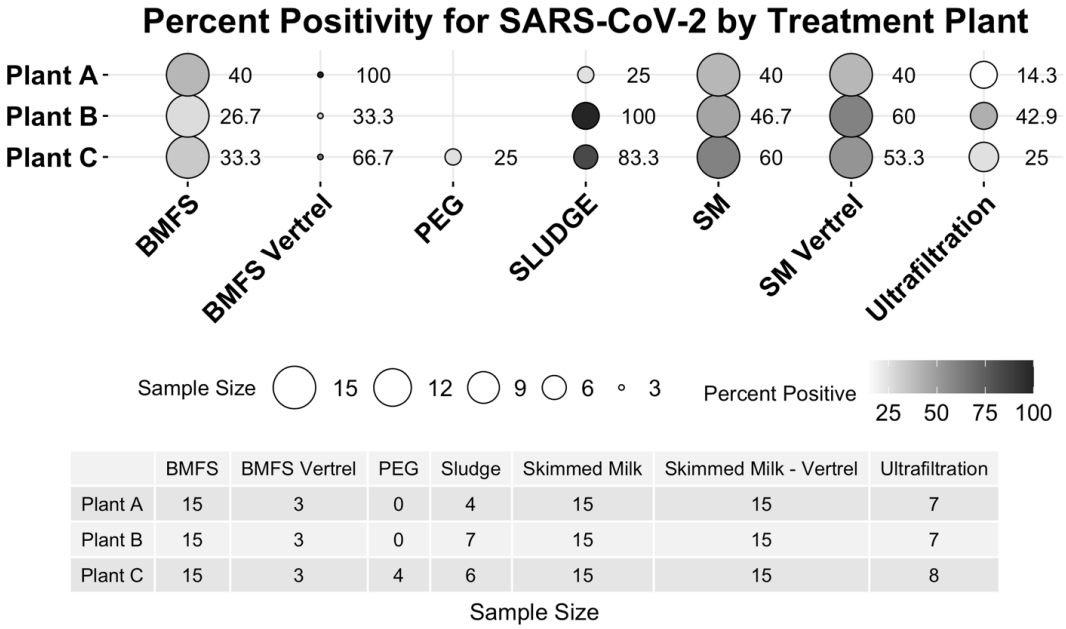
**Figure 2.1.9**: Presumptive positive for SARS-CoV-2 by method. A sample was considered "presumptive positive" if at least one of the SARS-CoV-2 assays had a $C_q \leq 40$ in either the non-diluted or $10^{-1}$ dilution reaction. Darker colors indicate higher positivity for SARS-CoV-2.

Many of the samples across methods amplified at ranges that were unquantifiable at a genome copy level but were considered positive if they amplified below $C_q$ of 40. The performance of each assay varied by method (Figure 2.1.4), with N1 and N2 having the highest detection in direct sludge extraction, but N3 having the highest detection in skimmed milk – Vertrel$^{TM}$. Across all methods looking at wastewater, skimmed milk flocculation provided more positive samples in undiluted assays (48.9%, n=45) (Figure 2.1.5) with lower $C_q$ values than other methods surveying wastewater, despite the lower effective volume assayed (Figure 2.1.1A). Surveying sludge/biosolids yielded the lowest $C_q$ values for SARS-CoV-2 (Appendices Figure 2.5.4), demonstrating its ability to concentrate viral particles. However, it is not as comparable to water-based samples and methods because treatment plants handle biosolids differently. Each WWTP has a different number of settling tanks, the holding time in these tanks is dependent on the flow and what flocculants used, if any, and the sample is not a composite from all clarifiers. Therefore, a sludge sample does not reflect an influent sample and is variable and difficult to interpret.

Detection of SARS-CoV-2 differed across wastewater treatment plants and between samples. For example, Plant A had less detection using direct sludge

extraction compared to skimmed milk flocculation (Figure 2.1.6), while the other two treatment plants had more detection for sludge extraction. This indicates that the between-plant differences in handling of influent wastewater and biosolids can substantially affect detection of SARS-CoV-2. Additionally, investigators noted differences in wastewater consistency and free suspended solids in samples from different treatment plants. This demonstrates that differences in influent and variability in SARS-CoV-2 detection cannot be differentiated.

Across the three SARS-CoV-2 assays, there was a substantial amount of variability in detection. While the N1 assay had fewer non-detects (Figure 2.1.4) compared N2 and N3, the mean $C_q$ value for all three assays is similar (37.0, 37.1, and 36.8, respectively) (Appendices Figure 2.5.5) indicating some variability of performance amongst the assays or degradation in the target genetic material. Although, several samples (weeks 19, 23, 24, 27 and 28) amplified across all three assays at quantifiable




| | BMFS | BMFS Vertrel | PEG | Sludge | Skimmed Milk | Skimmed Milk - Vertrel | Ultrafiltration |
|---|---|---|---|---|---|---|---|
| Plant A | 15 | 3 | 0 | 4 | 15 | 15 | 7 |
| Plant B | 15 | 3 | 0 | 7 | 15 | 15 | 7 |
| Plant C | 15 | 3 | 4 | 6 | 15 | 15 | 8 |

Sample Size

**Figure 2.1.11:** SARS-CoV-2 Detection by wastewater treatment plant. Percent positivity includes undiluted and $10^{-1}$ diluted RT-qPCR assays for each method and treatment plant. Detection of SARS-CoV-2 varied by method for each treatment plant.

ranges using a standard curve, many samples are non-quantifiable. This indicates that SARS-CoV-2 is at or near detectable levels in the majority of these samples due to dilution in the wastewater, low persistence, and/or the presence of inhibitors.

Since these WWTPs serve hundreds of thousands of people in King County, most of detectable SARS-CoV-2 samples were often out of quantifiable ranges, and template degradation is likely, it is not feasible to use this data as precise measure of infections in the community. However, these results serve as an indicator of wide-spread community transmission and can track trends in infections at the community level. Additionally, these results can be used to determine effective sampling and processing techniques that allow for the detection of endemic SARS-CoV-2 on a smaller scale such as in wastewater conveyance systems, pump stations, or residential communities.

### 2.1.5 Conclusion

These results stress the importance of validating sampling and concentration methods in different water matrices using seeded recovery controls. Other studies have utilized surrogate viruses to control for recovery and compare methods (Ahmed *et al.*, 2020, Jafferali *et al.*, 2020), but this is the first study to conduct a methods comparison of SARS-CoV-2 at multiple treatment plants for consecutive weeks. Additionally, this is the first methods comparison published using another human respiratory coronavirus as the recovery control organism for SARS-CoV-2. Although all methods tested here and previously published were shown effective at detecting SARS-CoV-2 and the seeded recovery control (Ahmed *et al.*, 2020, Jafferali *et al.*, 2020), a method that is feasible in one setting and environment may not be feasible elsewhere. Skimmed milk flocculation

was chosen for continued surveillance in our lab because of its detection consistency and simplicity. Because skimmed milk flocculation does not require extensive laboratory resources, it is a promising method for wastewater surveillance in resource limited settings. It does not rely on hard to acquire consumables and can therefore allow groups to conduct uninterrupted surveillance. For wastewater surveillance to effectively supplement clinical surveillance, methods must be validated and selected using both logistical and performance considerations. Given the national shortage in clinical SARS-CoV-2 tests, wastewater surveillance has the potential to effectively prevent new outbreaks in communities without cases, as was reported at a residential hall at the University of Arizona (Peiser, 2020). Additionally, wastewater surveillance may help understand changes in pandemic trends in the water catchment area, such as reductions or increases in cases (Water Research Foundation, 2020). Using effective sampling methods is critical for wastewater surveillance to serve as a leading indicator of clinical infection and to accurately describe community transmission of SARS-CoV-2.

### 2.1.6 Acknowledgments:

Thanks to our collaborators at the wastewater treatment plants who helped facilitate sample collection, particularly the West Point Process lab; the South Treatment Plant collection team including Wieslawa Wakulak, Jason Karlstrom, Rebecca Hee, Neila Glidden, and Rachael Dyda; and the Brightwater Operations Staff.

### 2.1.7 Funding

This work was supported by the University of Washington Population Health Initiative. This project was also supported by funds provided by the Department of Environmental & Occupational Health Sciences at the University of Washington.

## 2.2 Development and validation of the skimmed milk pellet extraction protocol for SARS-CoV-2 wastewater surveillance

As published: **Philo SE**, Ong AQW, Keim EK, Swanstrom R, Kossik AL, Zhou NA, Beck NK, Meschke JS. Development and Validation of the Skimmed Milk Pellet Extraction Protocol for SARS-CoV-2 Wastewater Surveillance. *Food and Environmental Virology*. 2022 Feb 10; 1-9. doi: 10.1007/s12560-022-09512-5 PMID – 35143035.

### 2.2.1 Abstract

Wastewater surveillance for SARS-CoV-2 may serve as a useful source of data for public health departments as the virus is shed in the stool of infected individuals. However, for wastewater data to be actionable, wastewater must be collected, concentrated, and analyzed in a timely manner. This manuscript presents modifications on a skimmed milk concentration protocol to reduce processing time, increase the number of samples that can be processed at once, and enable use in resource-limited settings. Wastewater seeded with Human coronavirus OC43 (OC43) was concentrated using a skimmed milk flocculation protocol, and then pellets were directly extracted with the QIAamp Viral RNA Mini kit. This protocol has a higher average effective volume assayed (6.35mL) than skimmed milk concentration methods, with and without Vertrel XF™, that involve resuspension of the pellets in PBS extraction prior to nucleic acid extraction (1.28mL, 1.44mL, respectively). OC43 was selected as a recovery control organism because both it and SARS-CoV-2 are enveloped respiratory viruses that primarily infect humans resulting in respiratory symptoms. The OC43 percent recovery for the direct extraction protocol (3.4%) is comparable to that of skimmed milk concentration with and without Vertrel XF™ extraction (4.0%, 2.6%, respectively). When comparing SARS-CoV-2 detection using McNemar's chi-square test, the pellet extraction method is not statistically different from skimmed milk concentration, with and without Vertrel XF™ extraction. This suggests that the method performs equally as well

as existing methods. Added benefits include reduced time spent per sample and the ability to process more samples at a single time. Direct extraction of skimmed milk pellets is a viable method for quick turnaround of wastewater data for public health interventions.

### 2.2.2 Introduction

SARS-CoV-2 emerged in late 2019 in Wuhan, China, and within a few months it was declared a pandemic by the WHO (World Health Organization, 2020, March 11). Infection with SARS-CoV-2 primarily results in respiratory symptoms, but can also cause gastrointestinal disease in a fraction of infected individuals (Jones *et al.*, 2020). Additionally, even if individuals do not show gastrointestinal symptoms, they can shed the virus in their stool for weeks after infection (Wang *et al.*, 2020, Wu *et al.*, 2020, Xiao *et al.*, 2020), which led research groups around the world to begin analyzing wastewater and stool samples for the virus (Ahmed *et al.*, 2020, Medema *et al.*, 2020, Peccia *et al.*, 2020).

Although the infectivity of virus detected in wastewater and stool is still being understood, presence of the virus in wastewater can serve as a useful source of information for public health departments. Three potential use cases for wastewater surveillance include: detecting an introduction of new cases to a water catchment area, monitoring changes in trends of infection, and estimating community prevalence (Water Research Foundation, 2020). Research has shown that an outbreak was averted in a university dorm in Arizona, USA because the virus was detected in wastewater before positive clinical tests, giving time for residents to be tested and isolated (Peiser, 2020). Additionally, detection in wastewater has been shown to precede the first positive

clinical tests by weeks in Brisbane, Australia (Ahmed *et al.*, 2021), and also frequently matches community infection trends (Gonzalez *et al.*, 2020, Peccia *et al.*, 2020).

However, for wastewater surveillance to be a useful tool for public health officials, results must be available as quickly as possible. Many of the existing concentration methods take hours or are dependent on unreliable supply chains. Here we present a modified version of the skimmed milk flocculation method (Philo *et al.*, 2021), the skimmed milk pellet extraction protocol. Skimmed milk flocculation was originally developed to concentrate viruses from coastal waters (Calgua *et al.*, 2008), and was previously used by this group to further concentrate the beef extract eluate off of positively charged ViroCap filters (Scientific Methods, Granger, IN, USA) (Falman *et al.*, 2019). However, in all those variations on the method, the skimmed milk pellets were resuspended in PBS with a fraction of the PBS resuspension being used for RNA and/or DNA extraction. In this optimized method, the skimmed milk pellets are resuspended in the lysis buffer of the RNA extraction kit. This allows the entire initially concentrated volume of wastewater to be extracted and results in larger effective volumes assayed. It also takes less time to concentrate and extract the genomic material than other methods.

The skimmed milk flocculation method was chosen for optimization because it performed well for recovery of human coronavirus OC43 (OC43, used to determine method recovery efficiencies) and SARS-CoV-2 detection (Philo *et al.*, 2021). Additionally, it is more feasible in low resource settings because it does not require expensive or hard to obtain supplies and the entire protocol can be completed in a single day. The method was validated using wastewater collected from three Seattle-area wastewater treatment plants (WWTPs) over fifteen weeks. This method is a viable

alternative for public health departments and research groups to conduct timely SARS-CoV-2 wastewater surveillance and successfully detects SARS-CoV-2 in Seattle-area wastewaters.

## 2.2.3 Methods

### 2.2.3.1 Sample Collection and Seeding

Primary wastewater was grab sampled from three Seattle-area wastewater treatment plants (WWTP) from October 2020 through March 2021 (Appendices Table 2.5.3) All wastewater was transported in coolers with ice packs and stored at 4°C for no longer than 3 days. Grab samples collected from a single WWTP over the course of the day were composited and mixed before being processed. All wastewater samples were seeded with OC43 at a concentration of $3.3 \times 10^4$ $TCID_{50}$/L of wastewater before concentration (OC43 ATCC VR-1558). OC43 viral stocks were produced following previously published protocols (Philo *et al.*, 2021).

### 2.2.3.2 Skimmed Milk Concentration

Skimmed milk flocculation with and without Vertrel XF™ (Miller-Stephenson, Inc., Danbury, CT, USA) extraction was carried out as previously published (Philo *et al.*, 2021). Briefly, a 5% skimmed milk solution was added to 0.6L of seeded wastewater (1% v/V final). The pH of the wastewater with skimmed milk was dropped to 3-4 using 5M HCl, shaken at 200RPM for two hours, and then divided into 50mL conicals and centrifuged at 3500G for 30 minutes at 4°C. The supernatant was then poured off and the pellets were used for different resuspension and RNA extraction protocols.

#### 2.2.3.3 Resuspension and RNA Extraction

##### *2.2.3.3.1 Resuspension in PBS*

The equivalent of 500mL of wastewater (ten 50mL pellets) were resuspended in 6.0mL of sterile PBS (pH = 7.4). This resuspension volume was then split into two equal volumes: one for RNA extraction and the other for Vertrel XF$^{TM}$ extraction to separate the viruses from the solids following previously published protocols (Philo *et al.*, 2021). RNA Extraction was carried out on both resuspensions using the QIAamp ViralRNA Mini Kit (QIAGEN, Germantown, MD, USA). Each sample was extracted in duplicate with a doubled input volume of 280uL for a total input extraction volume of 560uL. Each column was eluted in 60uL of Buffer AVE, for a total eluate volume of 120uL. Duplicate extracts were combined and re-aliquoted before being stored at -20°C.

##### *2.2.3.3.2 Resuspension in Lysis Buffer*

Buffer AVL from the QIAamp Viral RNA Mini Kit was prepared with carrier RNA according to the manufacturer's protocol. Two skimmed milk pellets (100mL wastewater equivalent) from each WWTP were resuspended in 560uL each of the Buffer AVL/carrier RNA mixture. This resuspended pellet was then transferred to a PowerBead Pro tube (Cat. No. 19301, QIAGEN, Germantown, MD, USA). The PowerBead pro tubes were then vortexed using a horizontal adaptor at maximum speed for ten minutes. They were then centrifuged at 15,000g for one minute to pellet the solids. The supernatant was transferred without disturbing the solids or beads to a sterile 1.5mL flip cap tube, one tube for each initial pellet. 560uL of 100% ethanol was then added to the supernatant and mixed by pulse vortexing for 15 seconds.

Starting with step six of the manufacturer's instructions, 630uL was applied to the column and samples were spun at 15,000G for five minutes. The extraction then

followed the manufacturer's instructions until the final elution step. At this point, 40uL of Buffer AVE was added to each column and incubated for one minute. After elution, the same 40uL was applied back onto the column for a second elution. The elution from replicate columns was then mixed and re-aliquoted for immediate molecular processing or storage at -20°C, for a total eluate volume of 80uL. The complete method is available for viewing online at YouTube (https://youtu.be/Xyb8CScM0Ko).

#### 2.2.3.4 Generation of Synthetic OC43 RNA Control

As OC43 RNA extracts were enumerated using $TCID_{50}$, an OC43 RNA control (cRNA) was made using published constructed RNA methods for use in RT-qPCR standard curves (Fronhoffs *et al.*, 2002, Vijgen *et al.*, 2005). Briefly, an OC43 DNA amplicon was constructed from extracted RNA from cell culture using the iTaq Universal SYBR Green one-step kit (Cat. No. 172-5151, Bio-Rad Laboratories, Hercules, CA, USA). Reactions contained 0.3uM of the OC43-FPT7 and OC43-RP primers from Vijgen *et al,* 2005 (Table 2.2.1). Temperatures followed the Bio-Rad mastermix instructions, with an annealing temperature of 60°C. DNA amplicon concentration was measured using a Nanodrop. The MEGAshortscript T7 transcription kit (Cat. No. AM1354, Invitrogen Corp., Waltham, MA, USA) was used to transcribe 84 ng of

**Table 2.2.1**: Sequences and concentrations of each primer and probe used in RT-qPCR for the Skimmed Milk - Direct Pellet extraction method

| Target | Name | Sequence (5' – 3') | Final Conc. | Reference |
|---|---|---|---|---|
| *OC43* | OC43-FP | ATGTTAGGCCGATAATTGAGGACTAT | 0.3uM | Vijgen *et al.* (2005) |
| | OC43-RP | AATGTAAAGATGGCCGCGTATT | 0.3uM | |
| | OC43-TP | FAM-CATACTCTGACGGTCACAAT-NFQ-MGB | 0.2uM | |
| | OC43-FPT7 | TAATACGACTCACTATAGGGAGGATGTTAGGCCGATAATTGAGGACTAT | 0.3uM | |
| *SARS-CoV-2 N1* | N1-F | GACCCCAAAATCAGCGAAAT | 0.2uM | US Centers for Disease Control and Prevention (2020) |
| | N1-R | TCTGGTTACTGCCAGTTGAATCTG | 0.2uM | |
| | N1-P | FAM-ACCCCGCAT-ZEN-TACGTTTGGTGGACC-3IABkFQ | 0.2uM | |
| *SARS-CoV-2 N2* | N2-F | TTACAAACATTGGCCGCAAA | 0.2uM | |
| | N2-R | GCGCGACATTCCGAAGAA | 0.2uM | |
| | N2-P | HEX-ACAATTTGC-ZEN-CCCCAGCGCTTCAG-3IABkFQ | 0.2uM | |

amplicon into constructed RNA of 73bp in three reaction volumes according to the manufacturer's instructions. DNA was then digested using 3µl of TURBO dnase, incubated for 15 minutes at 37°C, and stopped by adding 15ul ammonium acetate stop solution and 115 µl of molecular grade water. RNA precipitation was performed on cRNA by mixing in two volumes of 200 proof ethanol and chilling at -20°C for 15 minutes. Chilled cRNA was then spun at 21,000g for 15 min at 4°C. Supernatant was removed and cRNA pellet was suspended in AE buffer (10mM Tris-Cl, 0.5mM EDTA) resulting in $3x10^{13}$ genome copies per µl as determined by nanodrop. Aliquots of diluted cRNA were stored at -80°C.

#### 2.2.3.5 RT-qPCR

Reverse-transcription real-time PCR (RT-qPCR) for OC43 and SARS-CoV-2 was carried out on all wastewater RNA extracts using the iTaq Universal Probes One-Step Kit and the BioRad CFX qPCR systems (Bio-Rad Laboratories, Hercules, CA, USA). All samples were run in duplicate, with duplicate 10-fold dilutions to control for inhibition. All primers and probes were obtained from IDT (Integrated DNA Technologies, Inc., Coralville, IA, USA). Their sequences and final concentrations are in Table 2.2.1. RT-qPCR cycling conditions for both assays were 50°C for 10 minutes, 95°C for three minutes, and 45 cycles of 95°C for 15 seconds and 60°C for 30 seconds.

The OC43 assay targeting the M protein was carried out as previously published (Vijgen *et al.*, 2005, Philo *et al.*, 2021). Positive control standard curves were generated by making 10-fold serial dilutions in nuclease-free water either by using RNA extracts from viral stocks or the above-mentioned synthetic RNA. Negative controls were nuclease free water, and SARS-CoV-2 was included as an off-template control to ensure there was no cross-reactivity.

RT-qPCR for SARS-CoV-2 was carried out using the US CDC targets for the N1 and N2 genes in a total reaction volume of 15uL (US Centers for Disease Control and Prevention, 2020). Samples positive for SARS-CoV-2 were identified if either replicate for N1 or N2 had amplification with Cq < 40 in either the non-diluted or $10^{-1}$ diluted RT-qPCR reactions. To reduce the time to results, the assay was multiplexed by tagging the N1 probe with FAM and the N2 probe with HEX (Table 2.2.1). Positive control standard curves were generated by making 10-fold serial dilutions of a plasmid control containing the target genes for SARS-CoV-2 (Integrated DNA Technologies, Inc., Coralville, IA, USA) or RNA extract obtained from BEI (Item No. NR-52285, BEI Resources, Manassas, VA, USA). RNA extract was used because it resulted in more consistent standard curves than the plasmid positive control. Negative controls were nuclease free water, and OC43 was included as an off-template control to ensure there was no cross-reactivity.

#### 2.2.3.6 Recovery Calculations

Fractional recovery was calculated differently depending on the type of positive control used to generate the standard curve. The actual OC43 assayed for each sample was calculated using Equation 1. $C_q$ is the cycle quotient as determined using the standard curves, and the intercept and slope are the values obtained from the standard curve.

Equation 1: $$OC43_{Actual} = 10^{\frac{C_q - Intercept}{Slope}}$$

To calculate the theoretical OC43 assayed, the volume adjusted OC43 that we would have detected had 100% of it been recovered was calculated using Equation 2.

$C_{inoculated}$ is the concentration of OC43 inoculated into the sample and $V_{inoculated}$ is the effective volume assayed.

Equation 2: $$OC43_{Theoretical} = C_{inoculated}V_{inoculated}$$

Percent recovery was calculated using Equation 3.

Equation 3: $$Percent\ Recovery = \left(\frac{OC43_{Actual}}{OC43_{Theoretical}}\right) \times 100$$

#### 2.2.3.7 Data Analysis

All RT-qPCR data were analyzed using the Bio-Rad CFX Maestro for Mac program (Bio-Rad Laboratories, Hercules, CA, USA). All data were collated and managed using Microsoft Excel (Microsoft Corp., Redmond, WA, USA) and REDCap electronic data capture tools hosted at the University of Washington, Seattle (Harris *et al.*, 2009, Harris *et al.*, 2019). Statistical comparisons, calculations, and figures were generated using RStudio (2019 RStudio: Integrated Development for R. RStudio, Inc., Boston, MA, URL http://www.rstudio.com/).

To understand method performance over time, control charts were generated for each method using OC43 Cq values from the RT-qPCR assay and paired t-tests were conducted. The average value is the OC43 Cq value averaged across treatment plants and time for each method. Upper warning limits (UWL) and lower warning limits (LWL) for each method were calculated by adding or subtracting, respectively, one standard deviation from the mean Cq. Upper confidence limits (UCL) and lower confidence limits (LCL) for each method were calculated by adding or subtracting three times the standard deviation from the mean Cq. These were then plotted against the average Cq value for each sample for a visual representation of variability over time.

To compare SARS-CoV-2 detection between concentration and extraction methods, two-by-two contingency tables were generated comparing the direct extraction

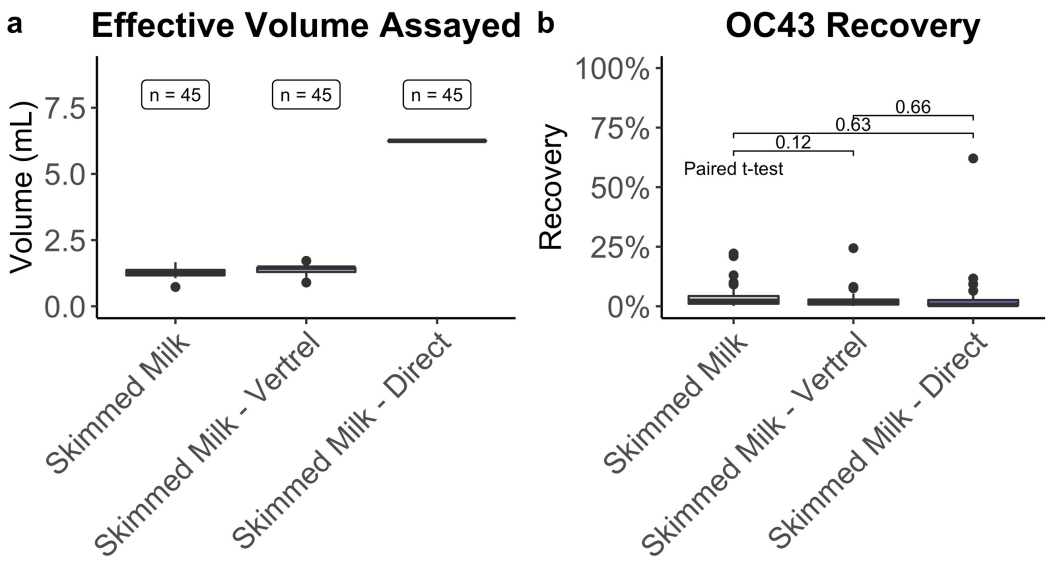



**Figure 2.2.1**: Effective volume assayed and percent recovery for each method. a) The effective volume assayed is the proportional volume of the original wastewater sample assayed by RT-qPCR. The skimmed milk direct pellet extraction has the highest volume assayed per reaction (6.25mL). Because it is not dependent on variable resuspension volumes, the effective volume assayed for each sample is the same, unlike skimmed milk flocculation with or without Vertrel XF$^{TM}$ extraction. b) The OC43 recovery is the volume adjusted recovery using the RT-qPCR standard curves. The mean recoveries for all three compared methods are comparable.

method to both skimmed milk methods. Comparisons were made for both undiluted assays and $10^{-1}$ diluted assays. A positive sample for SARS-CoV-2 was determined by detection of N1 or N2 target with a representative sigmoidal curve and a Cq of less that 40 in either the undiluted or ten-fold diluted sample. McNemar's Chi-Square test for count data was carried out on the tables using the built in R function to assess differences in SARS-CoV-2 detection between methods. McNemar's test was chosen because it assesses concordant and discordant pairs of paired data (McNemar, 1947, Smith & Ruxton, 2020).

## 2.2.4 Results and Discussion

### 2.2.4.1 OC43 Surrogate Comparisons

To understand how the three tested methods compared, the effective volume assayed, OC43 percent recovery, and performance over time were all considered. The effective volume assayed was calculated as the proportion of the initial volume of

**Table 2.2.2**: Descriptive statistics for OC43 comparisons. A) Effective volume assayed by method. B) Percent recovery by method.

| | A) Effective Volume Assayed | | | B) Percent Recovery | | |
|---|---|---|---|---|---|---|
| | *Skimmed Milk* | *Skimmed Milk - Vertrel* | *Skimmed Milk - Direct* | *Skimmed Milk* | *Skimmed Milk - Vertrel* | *Skimmed Milk - Direct* |
| **Max** | 1.67 | 1.72 | 6.25 | 22.16 | 24.38 | 62.06 |
| **75th Quant.** | 1.37 | 1.46 | 6.25 | 4.29 | 2.81 | 2.62 |
| **Median** | 1.3 | 1.46 | 6.25 | 2.29 | 1.38 | 0.57 |
| **25th Quant.** | 1.17 | 1.3 | 6.25 | 1.09 | 0.73 | 0.31 |
| **Minimum** | 0.73 | 0.9 | 6.25 | 0.12 | 0.13 | 0 |
| **Mean** | 1.28 | 1.44 | 6.25 | 4.03 | 2.63 | 3.36 |

concentrated wastewater that was assayed by RT-qPCR. The mean effective volume assayed for skimmed milk flocculation with and without Vertrel XF$^{TM}$ extraction is 1.28mL and 1.44mL, respectively (Table 2.2.2A, Figure 2.2.1A). Because the direct pellet extraction method does not rely on variable resuspension volumes of the pellets, the effective volume assayed for each sample is 6.25mL. This consistency in volume assayed has a substantial benefit because it reduces variability between samples. Variability in SARS-CoV-2 detection cannot be attributed to variable effective volumes assayed because this is a constant number with the direct pellet extraction.

When comparing OC43 percent recoveries, all three methods perform similarly. The average OC43 percent recovery for the direct pellet extraction method is 3.36% compared to 4.03% and 2.63% for skimmed milk flocculation with and without Vertrel XF$^{TM}$ extraction, respectively (Table 2.2.2B, Figure 2.2.1B). Paired t-tests were run comparing the percent recoveries, and there were no statistically significant differences present in the distributions (Figure 2.2.1B).

These data suggest that there is no loss in performance relative to either of the prior skimmed milk flocculation methods when using this novel extraction method. Although the OC43 recovery values in this study are low, they are similar to those from previous work by the authors (Philo *et al.*, 2021) as well as recoveries obtained for other viruses after wastewater concentration, including <10% recovery for porcine

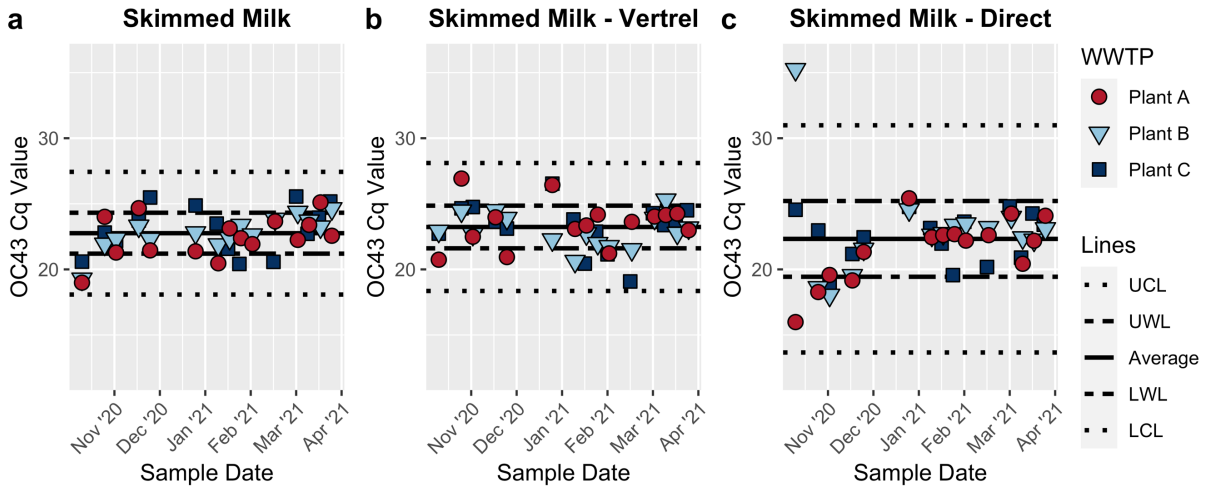



**Figure 2.2.2**: Control charts of the OC43 RT-qPCR Cq values for a) skimmed milk flocculation, b) skimmed milk flocculation with Vertrel XF$^{TM}$ extraction, and c) direct skimmed milk pellet extraction. Average values were calculated by averaging across treatment plant and time. The upper warning limit (UWL) and lower warning limit (LWL) for each method were calculated by adding or subtracting, respectively, the standard deviation from the average Cq. The upper confidence limit (UCL) and lower confidence limit (LCL) were calculated by adding or subtracting, respectively, three times the standard deviation from the average Cq.

reproductive and respiratory syndrome virus (PRRSV) (Farkas *et al.*, 2021) and approximately 5% for bovine coronavirus (BCoV) (Gonzalez *et al.*, 2020). Recovery is low for several potential reasons. Concentration and recovery methods focus on viral particles, while detection methods focus on viral RNA. It is not possible to distinguish between virus clusters and individual virus particles with cell culture methods in the way that RT-qPCR differentiates genome copies. Additionally, because OC43, PRRSV, BCoV, and SARS-CoV-2 are enveloped respiratory viruses, they could be less stable in the wastewater matrix and degrade quickly compared to non-enveloped enteric viruses (Ye *et al.*, 2016, Bogler *et al.*, 2020).

Previous work has indicated that larger effective volumes assayed can reduce OC43 recovery due to inhibitors concentrated with the wastewater. The bag-mediated filtration system had an average effective volume assayed of 11.90mL through RT-qPCR, but had an average OC43 percent recovery of 0.69% (Philo *et al.*, 2021). Since the direct pellet extraction method does not have the same relationship, it can be

inferred that this method better controls for the presence of inhibitors and recovers more OC43 despite the large volume assayed.

To look at the methods' performance over time, control charts were developed using the Cq values obtained from the OC43 RT-qPCR assays (Figure 2.2.2). The mean Cq values for all three methods are very similar (Table 2.2.3). However, the variability around the mean Cq value for the direct pellet extraction method is substantially larger than for both of the other skimmed milk methods (Table 2.2.3, Figure 2.2.2). This is likely due to the method having both the lowest minimum Cq value and the highest maximum Cq value (Table 2.2.3). While the large maximum detection value is of concern, the low minimum Cq value suggests that direct pellet extraction can recover more of the seeded OC43 than the other two methods. Additionally, there is one direct pellet extraction method sample that has a Cq greater than the UWL level compared to seven with the skimmed milk flocculation method and four with the skimmed milk flocculation with Vertrel method.

OC43 was chosen as the surrogate organism in this study because both OC43 and SARS-CoV-2 are beta-coronaviruses that cause respiratory infections in humans and are spread via droplet and/or aerosol transmission (Harrison *et al.*, 2020, Greenhalgh *et al.*, 2021, Liu *et al.*, 2021). If the purpose of using a surrogate organism is to understand how a method performs over time in the given water matrix, as with this

**Table 2.2.3**: Descriptive statistics for OC43 RT-qPCR values used to generate control charts.

| | Skimmed Milk | Skimmed Milk - Vertrel | Skimmed Milk - Direct |
|---|---|---|---|
| **Mean** | 22.8 | 23.2 | 22.3 |
| **Std. Dev.** | 1.6 | 1.6 | 2.9 |
| **Variance** | 2.4 | 2.6 | 8.3 |
| **Max** | 25.6 | 26.9 | 35.2 |
| **Min** | 19 | 19.1 | 16 |
| *Non. Detect.* | *0* | *0* | *0* |

study, how the surrogate behaves in relation to SARS-CoV-2 may not be as important. However, if the purpose is to adjust SARS-CoV-2 values using the recovery of the surrogate, it is crucial that the surrogate behaves similarly to SARS-CoV-2. Adjusting SARS-CoV-2 values using recovery values of an organism without direct comparisons to SARS-CoV-2 adds additional uncertainty to the results.

#### 2.2.4.2 SARS-CoV-2 Detection

McNemar's chi-squared tests were run on two-by-two contingency tables assessing detection of SARS-CoV-2 using criteria established in section 2.2.3.7. There were no statistically significant differences in detection between the direct pellet extraction method and skimmed milk flocculation with and without Vertrel XF$^{TM}$ extraction in both the non-diluted and $10^{-1}$ diluted RT-qPCR reactions (Table 2.2.4). Although it is difficult to prove statistical similarity, the inability to reject the null hypothesis that the row and column marginal frequencies are equal (Smith & Ruxton, 2020) indicates that these methods result in similar SARS-CoV-2 detection.

Balloon plot heat maps were also generated to visualize differences in detection between assays and methods (Figure 2.2.3). When looking at the non-diluted RT-qPCR reactions (Figure 2.2.3A), all three methods had similar percent positivity for the individual N1 and N2 reactions and SARS-CoV-2. Because the values for N1 and SARS-CoV2 are the same or very similar (Figure 2.2.3A), it can be inferred that most

**Table 2.2.4**: Results of McNemar's chi-squared test for count data analyzing SARS-CoV-2 detection in A) non-diluted RT-qPCR reactions and B) $10^{-1}$ RT-qPCR reactions (n = 45). There is no statistically significant difference in SARS-CoV-2 detection between the direct pellet extraction method and skimmed milk flocculation with and without Vertrel XF$^{TM}$ extraction in both the A) non-diluted reactions and B) $10^{-1}$ reactions.

| | | Method 1 | Method 2 | McNemar's $X^2$ | Degrees of Freedom | p-value |
|---|---|---|---|---|---|---|
| **A)** | **Non-Diluted** | *Skimmed Milk – Direct* | Skimmed Milk | 0.23 | 1 | 0.63 |
| | | | Skimmed Milk - Vertrel | 0.06 | 1 | 0.80 |
| **B)** | **$10^{-1}$ Dilution** | *Skimmed Milk - Direct* | Skimmed Milk | 1.78 | 1 | 0.18 |
| | | | Skimmed Milk - Vertrel | 0.90 | 1 | 0.34 |

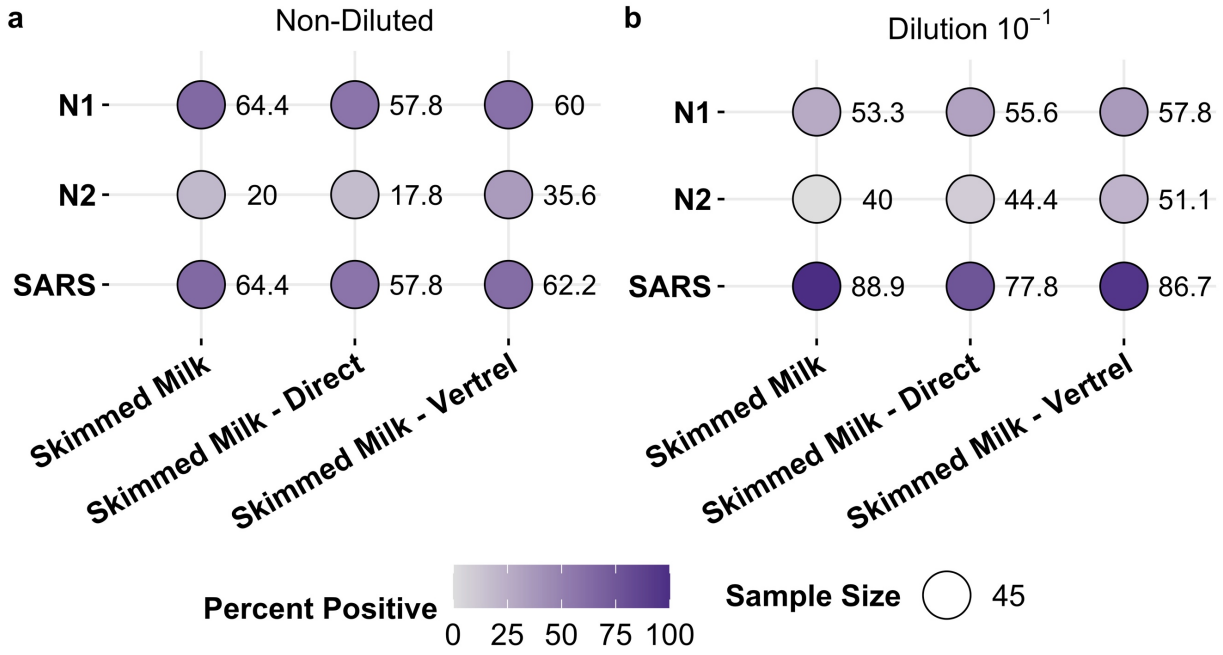



**Figure 2.2.3**: Percent positivity for SARS-CoV-2 by assay in either a) non-diluted or b) $10^{-1}$ diluted RT-qPCR reactions. Darker colors indicate more detection. There is higher percent positivity for the N1 assay compared to N2 across all the methods, suggesting N1 contributes more to SARS-CoV-2 detection. SARS-CoV-2 detection is higher in the $10^{-1}$ diluted RT-qPCR reactions than the non-diluted reactions, indicating there are residual effects of inhibitors on the RT-qPCR reactions.

SARS-CoV-2 detection in the undiluted reactions is due to N1 detection and not N2 detection. Frequency of SARS-CoV-2 detection in the $10^{-1}$ reactions (Figure 2.2.3B) is higher for all three methods than in the non-diluted reactions (Figure 2.2.3A). Additionally, the proportion of samples positive for N2 are higher in the diluted reactions compared to the non-diluted reactions (Figure 2.2.3). These results together indicate that there are inhibitors from the wastewater present in the RNA extracts, and that N2 is more susceptible to inhibitors than N1. Previous work indicates the limit of detection (LOD) is higher for N2 (100 gene copies) in wastewater than N1 (10 gene copies), likely due to the presence of inhibitors (Philo *et al.*, 2021). Because the Cq values for these samples are at or near the LOD in wastewater (Appendices Figure 2.5.6), ten-fold dilutions could be diluting past the point at which we expect detection in each sample. Additionally, other research suggests the N1 primer and probe set is more sensitive

than the N2 primer and probe set in nasopharyngeal samples (Vogels *et al.*, 2020). This effect could be amplified in wastewater samples like those collected and processed for this study. To control for inhibitors, all RT-qPCR reactions for these methods should be run using ten-fold dilutions in addition to undiluted samples (Loge *et al.*, 2002).

This manuscript describes the development and validation of a modification to existing skimmed milk flocculation protocols for surveillance of SARS-CoV-2 in Seattle-area wastewater. Currently, these data are not compared to the clinical case trends in King County, WA. Even though these samples were collected during a large peak in cases over the holiday period (Appendices Table 2.5.3), SARS-CoV-2 was not detected in every sample (Figure 2.2.3). Because the wastewater is well mixed by the time it reaches the treatment plants, SARS-CoV-2 should have been detectable during the entire sampling period. It is likely that the SARS-CoV-2 signal is diluted beyond the limit of detection before collection. The treatment plants where samples were collected process tens of millions of gallons of wastewater a day, with up to twice as much processed daily during the winter rainy season. For this reason, in locations that have clear wet and dry seasons, taking flow adjusted composite samples and correcting for flow in the molecular results may better represent local infection trends.

This method was developed because other existing methods require more active time in a laboratory, rely on access to expensive and often difficult to obtain consumable materials, or have limits on daily samples processed. Since the PBS resuspension step is not included in the method described here compared to other uses of skimmed milk flocculation (Calgua *et al.*, 2008, Falman *et al.*, 2019, Philo *et al.*, 2021), samples can be concentrated, extracted, and analyzed for SARS-CoV-2 in a single day. The skimmed milk pellet extraction method requires three to four hours of active time and

can process up to 12 samples with a single microcentrifuge, ensuring data can be reported in a timely manner. Finally, methods such as membrane filtration and ultrafiltration require expensive materials and/or advanced laboratory facilities. The direct pellet extraction method can be carried out in more resource-limited settings, making it more feasible in low- and middle- income countries (LMICs).

Despite the success of the skimmed milk direct pellet extraction at detecting SARS-CoV-2, this method can be further optimized. It is unclear where the OC43 RT-qPCR signal is being lost to result in the low mean percent recovery (Table 2.2.2B). This suggests that different methods need to be tested to identify one that performs better. Additionally, recovery studies should be carried out assessing loss at each step of the method and with different extraction kits. The QIAamp Viral RNA Mini kit was selected to maintain consistency across methods and because it was suggested by the company in early/mid-2020. However, it is likely that different protocols work better because there are still some effects of inhibitors on RT-qPCR with this extraction kit (Figure 2.2.3). Additionally, many other groups are using digital droplet or digital partitioning PCR systems for molecular detection. Optimizing the workflow for these systems will likely increase fractional recovery because they better control for inhibitors than RT-qPCR. Finally, how samples are collected can be better optimized. Understanding how frequently to take samples for a given population, whether to use grab or composite samples, and at what time of day to collect can result in more representative and useful information. Environmental and wastewater surveillance for SARS-CoV-2 will be most useful for public health officials if it provides insight beyond what can be obtained with clinical surveillance. Methods being used must be optimized and performance validated for SARS-CoV-2 in the specific catchment area for this to be

true. Although the skimmed milk pellet extraction method could still be improved, it has been shown to be fit for the purpose of detecting SARS-CoV-2 in wastewater.

## 2.3 Comparing four commercially available RNA extraction protocols for SARS-CoV-2 wastewater surveillance

### 2.3.1 Abstract

Wastewater surveillance is being widely used to supplement clinical surveillance for SARS-CoV-2. A previously optimized wastewater concentration method for SARS-CoV-2, direct extraction of skimmed milk flocculation pellets, had low OC43 recovery values with the ViralRNA mini kit. This manuscript compares the use of three commercially available RNA extraction kits, the RNeasy PowerBiofilm kit, the NucleoSpin RNA Stool Kit, and the NucleoMag DNA/RNA Water kit, against the previously used method. Primary influent wastewater was collected from three wastewater treatment plants throughout 2021, seeded with human coronavirus OC43 to serve as a recovery control, and concentrated using skimmed milk flocculation. The skimmed milk pellets were extracted using these four kits. All three newly tested kits resulted in higher OC43 method recovery than with the ViralRNA kit, but none of the kits removed all the inhibitors present in the wastewater. The NucleoMag kit best removed RT-qPCR inhibitors. SARS-CoV-2 was detected in almost all samples. Nevertheless, diluted RT-qPCR reactions were more likely to be positive for SARS-CoV-2 than undiluted reactions. These results indicate the importance of including dilutions in RT-qPCR wastewater assays to enumerate and control for residual inhibitors in the RNA extracts.

### 2.3.2 Introduction

Shortly after SARS-CoV-2, the causative virus of the COVID-19 pandemic, emerged in late 2019, it was detected in the stool of infected individuals and in wastewater all around the world (Ahmed *et al.*, 2020, Medema *et al.*, 2020, Peccia *et*

*al.*, 2020, Wang *et al.*, 2020). This discovery led to the widespread implementation of wastewater surveillance for SARS-CoV-2. Since then, over 200 universities in more than 50 countries have implemented environmental surveillance (ES) programs for SARS-CoV-2 (Naughton *et al.*, 2021). There are a number of potential use-cases for SARS-CoV-2 ES. These include serving as an early warning for emergence or re-emergence in a community (Medema *et al.*, 2020, Bibby *et al.*, 2021). This use-case is evidenced with an outbreak prevention in a university dorm when wastewater detection allowed for the testing and isolation of infected individuals before they showed symptoms (Betancourt *et al.*, 2021). Additionally, ES could provide information about infection trends in the community (Water Research Foundation, 2020). Finally, as new variants of concern have been identified, wastewater surveillance has been used to detect the variants in the sewershed before clinical sample sequencing results are available (Yu *et al.*, 2021). Because each variant of concern has different transmission rates and risks of severe infection (Lin *et al.*, 2021, Wolter *et al.*, 2022), understanding which variants are circulating in a community can help local health providers prepare. Additionally, sequencing wastewater samples allows public health officials to sequence the infections of an entire community instead of single individuals to provide a bigger picture on the variants.

Wastewater data must be collected and reported accurately to help inform the public health response for these use cases. However, a recent review noted that many published papers on SARS-CoV-2 ES do not report data in a usable format (Ahmed *et al.*, 2020). Publications often lack detailed methodology, preventing replicability and comparability, and data on recovery efficiency of the PCR assay performance are regularly omitted (Ahmed *et al.*, 2020). These data, along with information about the

volume of sample collected, when it was collected, and how frequently may affect result interpretation (Medema *et al.*, 2020). Although there have been collaborations established with the goal of facilitating discussion and knowledge sharing (Bivins *et al.*, 2020, Naughton *et al.*, 2021), there are no agreed upon standards for data reporting or method validation. This is of particular concern when groups conducting SARS-CoV-2 ES switch methods or PCR platforms. A lack of descriptive or quality control data in these instances increase the difficulty of interpreting data. This, ultimately, may limit the reliability and actionability of wastewater data.

There are numerous published papers comparing wastewater concentration methods (Ahmed *et al.*, 2021, Forés *et al.*, 2021, Gerrity *et al.*, 2021, Perez-Cataluna *et al.*, 2021, Philo *et al.*, 2021), but the same thorough evaluations have not been carried out for RNA extraction protocols. For SARS-CoV-2 in wastewater, an RNA extraction comparison was carried out in Porto, Portugal by assessing how much total RNA the kits extracted (Tomasino *et al.*, 2021). However, the comparison did not use a seeded recovery control. When SARS-CoV-2 was not detected, there was no molecular comparison to assess the kits' performance. While some groups conduct RNA extraction comparisons as part of a larger concentration methods comparison (Perez-Cataluna *et al.*, 2021), the multiple comparisons make it difficult to interpret if increased efficiency is due to the concentration method or the extraction protocol.

This study compares four commercially available extraction kits using a previously validated concentration method: direct extraction of skimmed milk flocculation pellets (Philo *et al.*, 2022). The goal was to improve OC43 recovery and reduce RT-qPCR inhibitors for skimmed milk flocculation. The comparison is carried out on primary influent wastewater collected from the Seattle area throughout 2021 seeded

with OC43 as a recovery control. The RNeasy PowerBiofilm Kit (QIAGEN, Hilden, Germany), the NucleoSpin RNA Stool Kit (Macherey-Nagel, Düren, Germany), and the NucleoMag DNA/RNA Water kit (Macherey-Nagel, Düren, Germany) will be compared against the previously validated QIAamp Viral RNA Mini Kit (QIAGEN, Hilden, Germany) (Philo *et al.*, 2021, Philo *et al.*, 2022). The choice of extraction kit used affects the detection of SARS-CoV-2 and the method recovery of the seeded recovery control (OC43). This manuscript provides a blueprint that other groups can follow to validate a new RNA extraction protocol, or to select the protocol that best controls for inhibition in their wastewater matrix.

## 2.3.3 Experimental Methods

### 2.3.3.1 Wastewater Collection and Seeding

Primary influent wastewater was grab sampled from three wastewater treatment plants (WWTPs) in the Seattle area in 2021. One of the WWTPs collected one grab samples every hour for 3 hours. These were composited before being concentrated. All wastewater was transported to the lab in coolers with ice packs on the same day it was collected and stored at 4°C. Wastewater was processed within three days of being received. Additionally, before concentration, all samples were seeded with human coronavirus OC43 (OC43) at a final concentration of $3.3x10^4$ $TCID_{50}$/L in wastewater (OC43, ATCC VR-1558). OC43 stocks were produced following a previously published protocol (Philo *et al.*, 2021) and were stored at -80°C until use.

### 2.3.3.2 Skimmed Milk Flocculation

Skimmed milk flocculation was carried out on 100mL of wastewater from each WWTP following previously published protocols (Philo *et al.*, 2021, Philo *et al.*, 2022). In brief, a 5% skimmed milk solution was added to 0.1L of wastewater (1% v/V final

concentration). Next, the pH of the wastewater was dropped to 3 to 4 using 5M HCl and pH strips. Then, the wastewater was shaken for two hours at 200RPM and room temperature. The wastewater was then divided into 50mL conicals, centrifuged at 4°C and 3500G for 30 minutes. Finally, the supernatant was poured off and the pellets used in the extraction protocols.

#### 2.3.3.3 RNA Extraction Protocols

##### *2.3.3.3.1 ViralRNA Mini Kit*

The protocol for the QIAamp ViralRNA Mini kit (ViralRNA) was previously optimized and is available to watch online at YouTube (https://youtu.be/Xyb8CScM0Ko) (Philo *et al.*, 2022). Because it was previously validated to extract skimmed milk pellets, the ViralRNA mini kit served as the control kit against which the others were compared. Briefly, two pellets from each WWTP (100mL equivalent) were resuspended in 560uL each of Buffer AVL prepared according to the manufacturer's protocol. The resuspension was then transferred to a PowerBead Pro tube (Cat. No. 19301, QIAGEN, Germantown, MD, USA), horizontally vortexed at maximum speed for 10 minutes, centrifuged at 15,000G for one minute, and the supernatant transferred to a clean 1.5mL tube. Ethanol (560uL of >99%) was next added to the supernatant and mixed. The RNA was then bound to the columns by spinning at 15,000G for 5 minutes. The rest of the extraction followed the manufacturer's protocol until the elution step. To elute the RNA, 40uL of buffer AVE was added to the column, eluted, and the same 40uL added back onto the column for a second elution. Elution from the two columns was then mixed for a total elution volume of 80uL and aliquoted for storage at -20°C or immediate molecular analysis.

#### *2.3.3.3.2 RNEasy Power Biofilm Kit*

First, the MBL solution from the RNEasy Power Biofilm kit (PowerBiofilm) was prepared with beta-mercaptoethanol according to the manufacturer's protocol (MBL-βME). Two pellets from each WWTP (100mL equivalent) were resuspended in 350uL of solution MBL-βME each and transferred to the included PowerBiofilm bead tube. The extraction continued according to the manufacturer's instructions. Each column was eluted in 50uL of RNase-free water (total elution volume of 100uL), combined, and re-aliquoted for immediate molecular processing or storage at -20°C.

#### *2.3.3.3.3 NucleoSpin RNA Stool Kit*

One pellet from each WWTP (50mL equivalent) was resuspended in 200uL of NucleoZOL and 660uL of Buffer RST1 from the NucleoSpin RNA Stool kit (NucleoSpin). The resuspension was transferred to the NucleoSpin Bead Tube Type A. The extraction then continued following the manufacturer's protocol with the DNA digestion step included. The RNA was eluted with 100uL of RNase-free water and stored at -20°C until molecular processing

#### *2.3.3.3.4 NucleoMag DNA/RNA Water Kit*

The methods for the NucleoMag DNA/RNA Water kit (NucleoMag) followed the RNA extraction protocol publicly available online on protocols.io (Rasile & Maas, 2021). The following alterations were made to the protocol. Two pellets from each WWTP (100mL equivalent) were resuspended in 500uL each of buffer MWA1 and then transferred to PowerBead Pro tubes. The bead tubes were then horizontally vortexed at maximum speed for 10 minutes, centrifuged at 15,000G for one minute, and 450uL of supernatant transferred to a clean 1.5mL flip cap tube. To this tube, 475uL of Buffer MWA1 and 25uL of NucleoMag B-Beads were added and then the extraction continues

according to step 12.1 of the online protocol. For the final elution step off the NucleoMag B-Beads, 50uL RNase-free water was added to each tube, shaken at 56°C and 500RPM for five minutes, removed from the magnetic beads, and then combined for a total elution volume of 100uL. The eluate was then re-aliquoted and stored at -20°C for molecular processing.

#### 2.3.3.4 RT-qPCR

Reverse-transcription quantitative PCR (RT-qPCR) was conducted on the RNA extracts for both OC43 and SARS-CoV-2 using the iTaq Universal Probes One-Step kit and the BioRad CFX qPCR system (Bio-Rad Laboratories, Hercules, CA, USA). RNA assays for both targets were run in duplicate with duplicate 10-fold dilutions to control for inhibitors in the wastewater. Primers and probes were obtained from IDT (Integrated DNA Technologies, Inc., Coralville, IA, USA). Primer and probe sequences and their final concentration are listed in Table 2.3.1. RT-qPCR reaction conditions for both OC43 and SARS-CoV-2 were 50°C for 10 minutes, 95°C for three minutes, and 45 cycles of 95°C for 15 seconds and 60°C for 30 seconds.

RT-qPCR for the OC43 M protein was carried out as previously published (Vijgen *et al.*, 2005, Philo *et al.*, 2021), with standard curves generated using 10-fold serial

**Table 2.3.1**: Sequences and final concentrations for primers and probes used in RT-qPCR to compare extraction kits.

| Target | Name | Sequence (5' – 3') | Final Conc. | Reference |
|---|---|---|---|---|
| *OC43* | OC43-FP | ATGTTAGGCCGATAATTGAGGACTAT | 0.3uM | Vijgen *et al.* (2005) |
| | OC43-RP | AATGTAAAGATGGCCGCGTATT | 0.3uM | |
| | OC43-TP | FAM-CATACTCTGACGGTCACAAT-NFQ-MGB | 0.2uM | |
| | OC43-FPT7 | TAATACGACTCACTATAGGGAGGATGTTAGGCCGATAATTGAGGACTAT | 0.3uM | |
| *SARS-CoV-2 N1* | N1-F | GACCCCAAAATCAGCGAAAT | 0.2uM | US Centers for Disease Control and Prevention (2020) |
| | N1-R | TCTGGTTACTGCCAGTTGAATCTG | 0.2uM | |
| | N1-P | FAM-ACCCCGCAT-ZEN-TACGTTTGGTGGACC-3IABkFQ | 0.2uM | |
| *SARS-CoV-2 N2* | N2-F | TTACAAACATTGGCCGCAAA | 0.2uM | |
| | N2-R | GCGCGACATTCCGAAGAA | 0.2uM | |
| | N2-P | HEX-ACAATTTGC-ZEN-CCCCAGCGCTTCAG-3IABkFQ | 0.2uM | |

dilutions of a synthetic RNA construct of the M-gene previously developed (Philo *et al.*, 2022). The negative control was the nuclease-free water used to generate the standard curves and make sample dilutions, and SARS-CoV-2 RNA was included as an off-template control to ensure there was no cross-reactivity in the two assays. Multiplexed RT-qPCR for SARS-CoV-2 using the US CDC targets for the N1 and N2 genes was carried out using a previously published assay (Philo *et al.*, 2022). Samples were run in duplicate with duplicate 10-fold dilutions. Samples were considered positive for SARS-CoV-2 if either duplicate in either the non-diluted or 1:10 diluted RT-qPCR reactions for N1 or N2 had amplification with a Cq < 40. Standard curves for SARS-CoV-2 were generated by making 10-fold serial dilutions of either a plasmid control containing the N1 and N2 genes (Integrated DNA Technologies, Inc., Coralville, IA, USA) or RNA extract obtained from BEI (Item No. NR-52285, BEI Resources, Manassas, VA, USA). Nuclease free water was used as a negative control, and OC43 RNA was included as an off-template control to ensure there was no cross-reactivity in the assays.

#### 2.3.3.5 Data Analysis

All RT-qPCR data were analyzed using the Bio-Rad CFX Maestro for Mac Program (Bio-Rad Laboratories, Hercules, CA, USA). Results and sample data were managed using Microsoft Excel (Microsoft Corp., Redmond, WA, USA) and REDCap electronic data capture tools hosted at the University of Washington, Seattle (Harris *et al.*, 2009, Harris *et al.*, 2019). Statistics, data manipulation, and figures were generated using RStudio and associated packages (2019 RStudio: Integrated Development for R. RStudio, Inc., Boston, MA, URL http://www.rstudio.com/) (Xu *et al.*, 2021).

The effective volume assayed, the volume adjusted amount of initial wastewater assayed in each PCR reaction, for each extraction kit was calculated for each method.

The fractional recovery of OC43 was calculated using the sample Cq values and the standard curve as previously published (Philo *et al.*, 2022). Equation 1 was used to obtain the actual OC43 recovered and assayed in the sample. Cq values are the average cycle quotient from the two RT-qPCR replicates, and the intercept and slope are obtained from the standard curve.

$$Equation\ 1: OC43_{actual} = 10^{\frac{C_q - Intercept}{Slope}}$$

Theoretical OC43 assayed was calculated by estimating the volume adjusted OC43 genome copies that we would have detected if 100% of it had been recovered (Equation 2). $C_{inoculated}$ is the concentration of OC43 inoculated into the sample and $V_{inoculated}$ is the effective volume assayed in the RT-qPCR reaction, as previously calculated. Percent recovery (Equation 3) is estimated using the theoretical and actual OC43 assayed calculations. To statistically compare percent recoveries in extractions performed on paired samples with different kits, Wilcoxon signed-rank tests were carried out using the built-in R function (wilcox.test(paired = TRUE)).

$$Equation\ 2: OC43_{Theoretical} = C_{inoculated} V_{inoculated}$$

$$Equation\ 3: Percent\ Recovery = \left(\frac{OC43_{Actual}}{OC43_{Theoretical}}\right) \times 100$$

Inhibition from the RNA extracts was assessed by comparing the difference in average Cq values between the undiluted and 1:10 diluted reactions for OC43 and the SARS-CoV-2 N1 target (Equation 4) with the slope of the standard curves.

$$Equation\ 4: Difference = Cq_{undiluted} - Cq_{1:10\ diluted}$$

Samples without any inhibition were expected to have a difference in Cq values equal to the slope of the standard curve, while samples with inhibition were expected to have differences less than the slope. Outliers for inhibition were assessed using normal

probability plots (NPPs) and Teitjen-Moore test for outliers following the methods and R-code specified in the National Institute of Standards and Technology Engineering Science Handbook (NIST/SEMATECH, 2012). If outliers were detected, these severely inhibited samples (difference between diluted and undiluted >0) were assessed separately. Ideal differences in RT-qPCR dilutions (the slope value) were compared to observed differences using the Wilcoxon signed-rank test.

It was not possible to assign gene copies to SARS-CoV-2 Cq values because detection was frequently near or at the limit of detection (Philo *et al.*, 2021). To compare SARS-CoV-2 detection achieved by different extraction kits, two-by-two contingency tables were generated for both the undiluted and 1:10 diluted RT-qPCR results. Results were all compared against the ViralRNA kit. McNemar's Chi-Square test for count data was carried out on the tables using the built in R function (mcnemar.test()) to determine differences in SARS-CoV-2 detection between extraction kits in the two-by-two contingency tables (McNemar, 1947, Smith & Ruxton, 2020).

## 2.3.4 Results and Discussion

### 2.3.4.1 Recovery Comparisons

The effective volume assayed was first calculated for each extraction kit used on the skimmed milk pellets (Figure 2.3.1). There is no intra-kit variability in effective volume assayed as with previous methods tested (Philo *et al.*, 2021). The ViralRNA mini kit had the highest effective volume assayed of the four extraction kits tested (6.25mL). The PowerBiofilm kit had the next highest effective volume assayed with 5.0mL. Both Macherey-Nagel kits had effective volumes assayed of less than 5.0 mL, with the NucleoMag kit assaying 4.5mL and the NucleoSpin kit assaying 2.5mL.

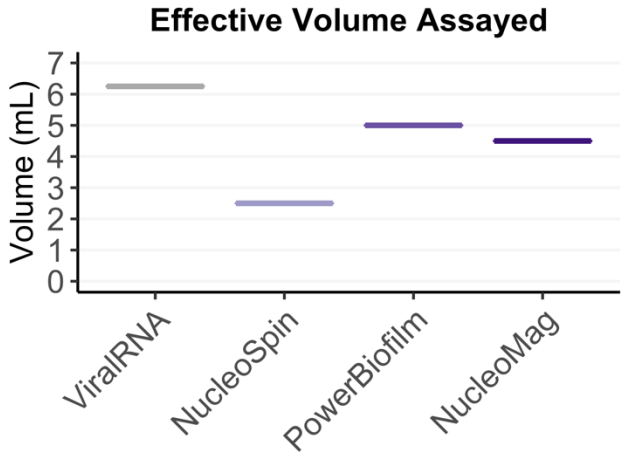



**Figure 2.3.1**: Effective volume assayed of each extraction kit. The effective volume assayed is the volume of initially sampled wastewater assayed in each RT-qPCR reaction.

The effective volume assayed is not commonly reported for methods to detect SARS-CoV-2 in wastewater. For papers that report these data, it ranges from less than one milliliter of wastewater assayed in RT-qPCR to over 50 milliliters (Medema *et al.*, 2020, Forés *et al.*, 2021, Gerrity *et al.*, 2021, Philo *et al.*, 2021). If the volume assayed is small and SARS-CoV-2 concentrations are low, the probability of detection will be affected because detection is a function of both effective volume assayed and method efficiency.

The OC43 recovery was assessed between extraction kits (Figure 2.3.2, Appendices Table 2.5.4). The ViralRNA mini kit had the lowest average OC43 recovery of the extraction kits compared (1.1%, Appendices Table 2.5.4). The extraction kit with the highest mean recovery is the NucleoMag kit, with a mean recovery of 11.9% (Figure 2.3.2C, Appendices Table 2.5.4). Wilcoxon signed-rank tests were conducted on sample pairs for the ViralRNA and PowerBiofilm kits (Figure 2.3.2A), for the ViralRNA and NucleoSpin kits (Figure 2.3.2B), and for the ViralRNA and NucleoMag kits (Figure 2.3.2C). All three extraction kits statistically significantly improve on the OC43 percent

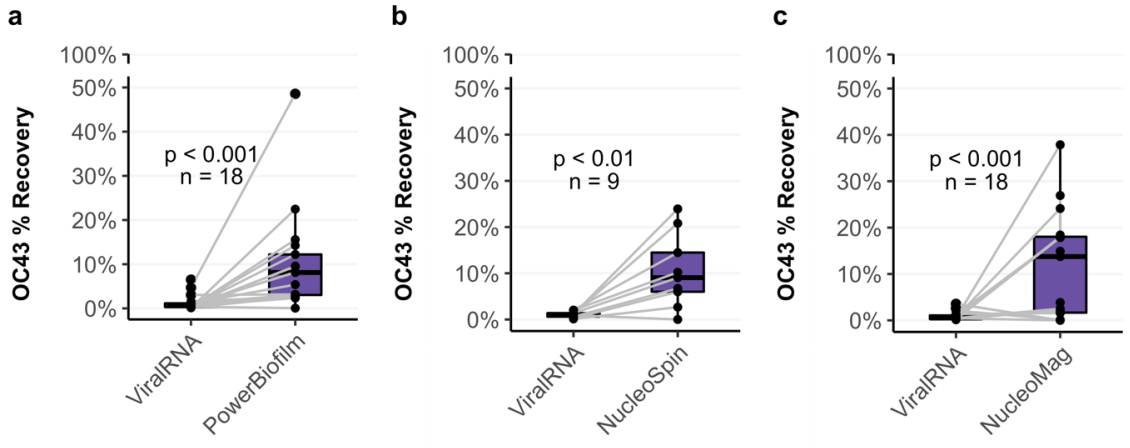



**Figure 2.3.2**: Paired OC43 method recovery comparing each extraction kit to the ViralRNA mini kit on skimmed milk flocculation pellets. OC43 percent recovery is the actual amount of OC43 that was detected by RT-qPCR compared to what would have theoretically been recovered if 100% had been recovered from the skimmed milk pellet extraction method. Wilcoxon signed-rank tests for paired samples were run comparing the **a)** PowerBiofilm ($p < 0.001$), **b)** NucleoSpin ($p < 0.01$), and **c)** NucleoMag ($p < 0.001$) kits with the ViralRNA kit. All three extraction kits significantly improve on the OC43 skimmed milk pellet extraction method recovery when compared to the ViralRNA kit.

recovery compared to the ViralRNA mini kit used in conjunction with the skim milk pellet method. However, when comparing 95% confidence intervals (Appendices table 2.5.4), the recoveries for all three newly tested kits exhibited greater variability for OC43 recovery compared to the ViralRNA Mini Kit. Mean recovery is just one parameter to consider when selecting an extraction kit. Precision and variability are also important factors to understand.

#### 2.3.4.2 Inhibitor Assessment

The presence of inhibitors in RNA extracts was assessed by comparing the observed difference between the undiluted and the tenfold diluted OC43 RT-qPCR reactions with the slopes of the standard curves (Figure 2.3.3). Samples that were severely inhibited as determined with the Teitjen-Moore test were assessed separately (Appendices Figure 2.5.7). Because both the standard curve and sample dilutions were made with tenfold serial dilutions, RNA extracts without inhibition should have a difference similar to the slope of the standard curves. The slope of a standard curve

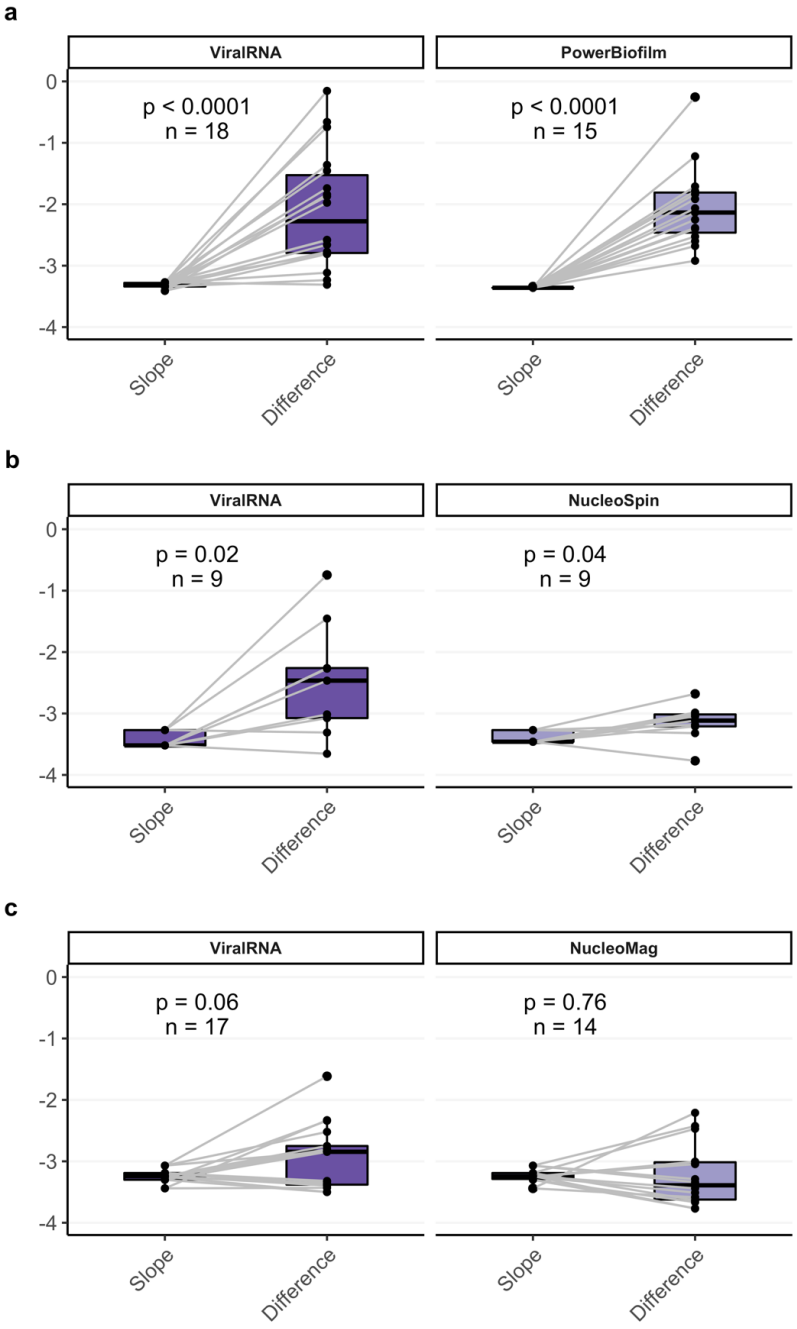



**Figure 2.3.3**: Differences between the slope values and undiluted vs 1:10 diluted OC43 Cq values to assess for inhibition using samples with a difference <0. RNA extraction kits that perfectly control for inhibition should have similar differences to the slope values of the standard curves. Comparisons on wastewater extracted using the ViralRNA and the **a)** PowerBiofilm, **b)** NucleoSpin, and **c)** the NucleoMag kits. The differences in Cq values are significantly different than the slope values using the Wilcoxon signed-rank test for all kits except in panel c.

with 90-100% efficiency should be between -3.6 and -3.3 (Svec *et al.*, 2015). The mean slope for the OC43 standard curves is -3.32 (95%CI: -3.34 to -3.29, Table 2.3.2), suggesting almost 100% standard curve efficiency.

Of the three new kits tested, the PowerBiofilm kit performed the worst for inhibition, with a mean difference between the undiluted and tenfold diluted of -2.04 (95% CI: -2.41 to -1.68) (Figure 2.3.3A, Table 2.3.2). Two samples were severely inhibited, and one sample did not have any detection in the undiluted value (Appendices Figure 2.5.7a). The ViralRNA kit on the same wastewater samples had a similar mean difference (-2.09, 95%CI: -2.56 to -1.63).

The samples processed by the NucleoSpin kit have a mean difference between undiluted and tenfold diluted OC43 reactions of -3.15 (95% CI: -3.37 to -2.92, Table 2.3.2, Figure 2.3.3b). The differences for the NucleoSpin kit were significantly different than the slope values (Figure 2.3.3b, p = 0.04). Additionally, of the three new kits tested, the NucleoSpin was the only kit that did not have any severely inhibited samples. Even though the NucleoSpin differences are statistically different from the slopes, that difference is small and the kit performs better than the ViralRNA kit on the same wastewater (Table 2.3.2).

After removing three severely inhibited samples (Appendices Figure 2.5.7b), the NucleoMag kit is the only new kit that is not statistically significantly different than the slope values (Figure 2.3.3c, p = 0.76). The mean difference is -3.22 (95% CI: -3.52 to -2.92), compared to -3.24 for the mean slope values (95%CI: 3.30 to -3.18) (Table

**Table 2.3.2**: Descriptive data for the differences between average Cq values of undiluted and 1:10 diluted RT-qPCR reactions and their respective slope values for samples with a difference <0.

| | | Difference | | | Standard Curve Slope | | | |
|---|---|---|---|---|---|---|---|---|
| *Kit Comparisons* | *Extraction Kit* | *Median* | *Mean* | *95% CI* | *Median* | *Mean* | *95% CI* | *n* |
| | ViralRNA | -2.28 | -2.09 | -2.56 to -1.63 | -3.30 | -3.32 | -3.34 to -3.29 | 18 |
| | PowerBiofilm | -2.13 | -2.04 | -2.41 to -1.68 | -3.36 | -3.36 | -3.36 to -3.35 | 15 |
| | ViralRNA | -2.46 | -2.47 | -3.18 to -1.76 | -3.46 | -3.41 | -3.46 to -3.36 | 9 |
| | NucleoSpin | -3.12 | -3.15 | -3.37 to -2.92 | | | | 9 |
| | ViralRNA | -2.84 | -2.95 | -3.22 to -2.67 | -3.24 | -3.24 | -3.30 to -3.18 | 17 |
| | NucleoMag | -3.39 | -3.22 | -3.53 to -2.92 | -3.24 | -3.24 | -3.30 to -3.18 | 14 |

2.3.2). The ViralRNA kit did not have a statistically significant difference between the slope values and the Cq differences in these samples ($p = 0.06$, mean = -2.95; 95%CI: -3.22 to -2.67) (Table 2.3.2, Figure 2.3.3c). Together, this suggests that these wastewater samples had less inhibition than others. However, one of the RT-qPCR reactions in the severely inhibited samples showed up 15 cycles earlier than the diluted reaction, indicating the NucleoMag kit does not remove all inhibitors.

The three newly tested extraction kits (PowerBiofilm, NucleoSpin, and NucleoMag) better remove inhibitors than the ViralRNA kit. It is likely that the design and chemistry of the three newly tested kits affects inhibition removal from wastewater samples. All three newly tested kits have steps to remove environmental inhibitors while the ViralRNA kit does not. Inhibitor removal from RNA extracts is particularly important for SARS-CoV-2 wastewater surveillance. Wastewater samples, like the ones collected from Seattle, contain numerous PCR inhibitory substances, not limited to human waste (Schrader *et al.*, 2012, Ahmed *et al.*, 2022). The presence of inhibitors can lead to false negatives for SARS-CoV-2 assays (Ahmed *et al.*, 2022). RNA extraction kits and protocols specifically designed to control for various environmental inhibitors are crucial for producing actionable data. Data presented here indicate that the ViralRNA mini kit is not as well suited for wastewater surveillance. Additionally, because severe inhibition was still present in kits designed for environmental samples, all RT-qPCR reactions for recovery controls should be carried out with ten-fold dilutions.

#### 2.3.4.3 SARS-CoV-2 Detection

Two-by-two contingency tables were generated for SARS-CoV-2 detection using criteria described in Section 2.3.3.4 (Appendices Table 2.5.5). It was not possible to run McNemar's tests on three of the six tables because the of the high SARS-CoV-2

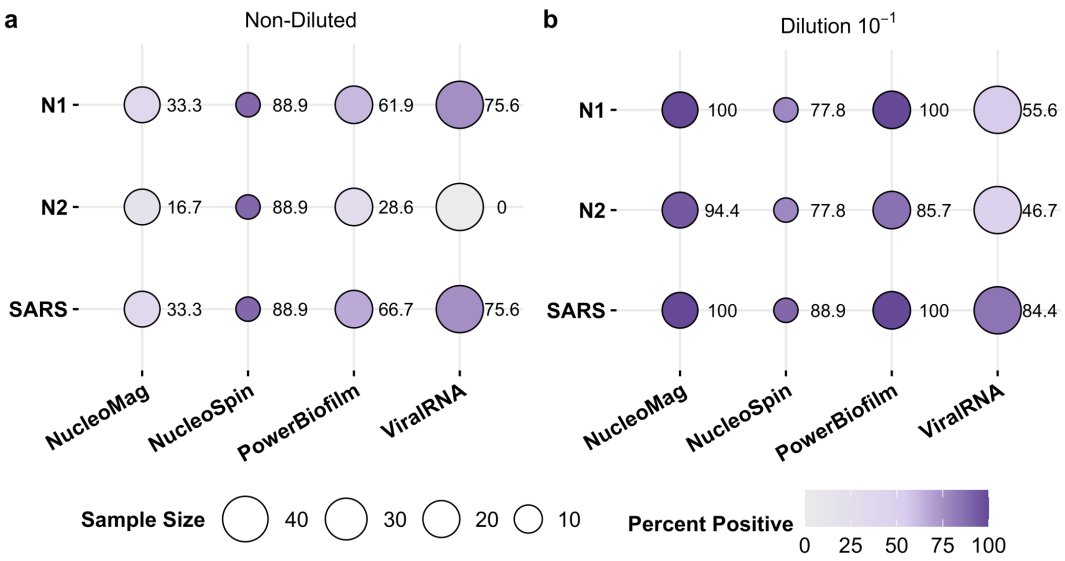



**Figure 2.3.4**: Balloon plots assessing SARS-CoV-2 percent positivity by RT-qPCR assay and RNA extraction kit in either a) the non-diluted reactions or b) the 1:10 diluted reactions. Darker colors indicate higher detection and larger circles indicate more samples were assayed by that kit. There is higher N1 detection than N2 in the non-diluted reactions, but they are similar in the 1:10 diluted reactions. SARS-CoV-2 detection is higher in the 1:10 diluted reactions.

prevalence in the wastewater during the testing period (Table 2.3.3). Three kit comparisons exhibited almost 100% SARS-CoV-2 detection (Appendices Table 2.5.5). In the undiluted RT-qPCR reactions, the ViralRNA and PowerBiofilm kits do not significantly differ in their SARS-CoV-2 detection (Table 2.3.3). However, in the tenfold diluted RT-qPCR reactions comparing the ViralRNA and PowerBiofilm kits, the PowerBiofilm kit detected SARS-CoV-2 in six samples that the ViralRNA kit did not (Table 2.3.3, p-value: 0.04). These data indicate the PowerBiofilm kit performed better only when the wastewater was diluted (Appendices Table 2.5.5).

**Table 2.3.3**: Results from McNemar's chi-squared test on contingency tables of extraction comparisons for which it was possible to run statistical tests.

| | | Extraction Kit 1 | Extraction Kit 2 | McNemar's $X^2$ | Degrees of Freedom | p-value |
|---|---|---|---|---|---|---|
| **A)** | *Non-Diluted* | Viral RNA | PowerBiofilm | 0.44 | 1 | 0.50 |
| | | | NucleoSpin | - | - | - |
| | | | NucleoMag | 7.69 | 1 | 0.006 |
| **B)** | *1:10 Dilution* | Viral RNA | PowerBiofilm | 4.17 | 1 | 0.04 |
| | | | NucleoSpin | - | - | - |
| | | | NucleoMag | - | - | - |

The ViralRNA kit detected SARS-CoV-2 in 12 undiluted reactions that the NucleoMag kit did not detect (Appendices Table 2.5.5), suggesting it performed better than the NucleoMag kit in the undiluted RT-qPCR assays (Table 2.3.3, p-value: 0.006). However, in the 1:10 diluted RT-qPCR assays comparing the NucleoMag and ViralRNA kits, the NucleoMag kit detected SARS-CoV-2 in all samples and the ViralRNA kit detected SARS-CoV-2 in all samples but one. These data suggest that there were frequently false-negatives in the undiluted SARS-CoV-2 assays across all RNA extraction kits, further highlighting the necessity of dilution to control inhibitors (Ahmed *et al.*, 2022).

Balloon plot heat maps were generated to visualize differences in SARS-CoV-2 detection between extraction kits and RT-qPCR assays (Figure 2.3.4). For the non-diluted reactions, N1 has more detection than N2 (Figure 2.3.4a), consistent with previous work by this group (Philo *et al.*, 2022). The NucleoMag extraction kit had the lowest detection of SARS-CoV-2 of the extraction kits in the undiluted reaction (33%, Figure 2.3.4a). Both the NucleoMag and PowerBiofilm kits detected SARS-CoV-2 in all samples in tenfold diluted RT-qPCR reactions (Figure 2.3.4b). This provides additional evidence for the use of dilutions when assaying wastewater by RT-qPCR to prevent false negatives. The NucleoSpin kit has less detection for both N1 and N2 in the 1:10 diluted reactions compared to the undiluted reactions (Figure 2.3.4b), suggesting the SARS-CoV-2 RNA might be diluted out beyond what is detectable by RT-qPCR.

### 2.3.5 Conclusions

This manuscript describes an RNA extraction comparison carried out on wastewater collected from the Seattle area in 2021. Because all the extraction kits were used on the same type of wastewater concentrate, skimmed milk pellets, it can be

assumed that the main differences between the kits is how well they recover OC43 RNA seeded into the unconcentrated wastewater, remove inhibitors present in the wastewater, and produce precise measurements. Previous work had identified skimmed milk flocculation as a viable method for detecting SARS-CoV-2 without depending on expensive and hard-to-obtain reagents (Philo *et al.*, 2021, Philo *et al.*, 2022). Although, OC43 recovery was still remarkably low, with the direct pellet extraction resulting in a mean OC43 recovery of 3.36%. Data in this manuscript suggest that the chosen RNA extraction protocol is responsible for much of the loss in recovery, as all three RNA extraction kits tested had significantly higher OC43 recovery than the ViralRNA kit (Figure 2.3.2).

Despite the clear improvement in OC43 method recovery with the newly tested extraction kits (Figure 2.3.2), it is apparent that skimmed milk concentration results in substantial RT-qPCR inhibition regardless of the extraction protocol. SARS-CoV-2 detection was higher in the 10-fold diluted RT-qPCR reactions for all kits except the NucleoSpin kit (Figure 2.3.4), indicating residual inhibition. Additional clean up steps could be added to these protocols to reduce inhibition. Two options include washing the extract with lithium chloride (Vennapusa *et al.*, 2020) or guanidinium isothiocyanate (Shieh *et al.*, 1995), both of which have been shown to effectively reduce the effects of inhibitors on RT-qPCR. However, adding extra washing steps may result in further RNA loss, so whether steps are necessary or not these must be considered for each use case. In situations when only presence absence is needed, like detection in a long-term care facility, these steps are not needed. If trend analysis is the goal, extra clean-up steps may increase efficiency and allow quantification. A protocol must be designed with the specific use-case in mind.

Research groups frequently use wastewater markers like pepper mild mottle virus (PMMoV) to adjust their SARS-CoV-2 data. When SARS-CoV-2 wastewater values are adjusted for PMMoV, many groups report that the wastewater data correlate better with either clinical cases or trends in clinical cases after adjusting for PMMoV (Wu *et al.*, 2020, Bivins *et al.*, 2021, D'Aoust *et al.*, 2021). Additionally, recovery of a control organisms like bovine coronavirus (Gerrity *et al.*, 2021) or bovine respiratory syncytial virus (Bivins *et al.*, 2021) are often used to adjust SARS-CoV-2. Poor performance of methods using recovery organisms introduces additional variability. Data presented here show OC43 recovery variability for each extraction kit. This study does not use recovery to adjust SARS-CoV-2 results but would ultimately result in added variability to the data.

This study is limited in that the extraction kits were only tested on a single concentration method, direct extraction of skimmed milk flocculation pellets. Additionally, because OC43 was seeded at the beginning of the protocol before wastewater concentration, it is not possible to understand how much of the loss in recovery is due to the method or the extraction kits. However, it is clear different extraction kits drastically improve the overall method recovery. Future work should attempt to elucidate which steps of the method result in reduced recovery of OC43. Additionally, even with different extraction kits, RT-qPCR inhibition is still a substantial problem with the skimmed milk concentration method. More work must be done to identify a method that detects SARS-CoV-2 consistently and precisely. Finally, because SARS-CoV-2 was prevalent in almost all wastewater samples, statistical comparisons are not possible using natively present SARS-CoV-2. Experiments using wastewater negative for SARS-CoV-2 and seeded with inactivated virus should be done to quantify

recovery and detection of SARS-CoV-2 throughout the concentration and extraction process relative to recovery control viruses.

### 2.3.6 Acknowledgment

This work was carried out with the support of Cynthia Ripoll, PhD, and Macherey-Nagel, Inc., who provided extraction kits for experimental comparisons. Our collaborators at the wastewater treatment plants, the West Point Process Lab, the South Treatment Plant collection team, and the Brightwater Operations staff, facilitated sample collection. Without them, this research would not have been possible.

### 2.3.7 Funding Sources and Conflicts of Interest

Research reported in this publication was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under award number T32ES015459. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. The authors declare no competing financial interest.

## 2.4 Systematically comparing additional wastewater concentration methods for SARS-CoV-2 environmental surveillance

### 2.4.1 Abstract

Magnetic beads have been used routinely to assist with genomic extraction and next generation sequencing, but they have only recently been developed to separate respiratory viruses from different matrices, including wastewater. In this study, a methods comparison was carried out to concentrate wastewater using Nanotrap® Magnetic Virus Particles, membrane filtration, and skimmed milk flocculation pellet extraction. Samples were collected from three wastewater treatment plants in the Seattle-area from May to September 2021. Because the skimmed milk pellet extraction method was previously validated, both membrane filtration and the Nanotrap ® beads were compared against skimmed milk. The Nanotrap ® beads were extracted with both the ViralRNA Mini kit and the NucleoMag DNA/RNA water kit to understand which step resulted in added efficiency. Samples were seeded with OC43 to serve as a method recovery control. The Nanotrap® beads significantly improved on OC43 recovery with both extraction kits tested compared to the skimmed milk direct extraction with the ViralRNA kit. Concentration with the Nanotrap® beads and extraction with the NucleoMag kit had higher SARS-CoV-2 detection in both undiluted and ten-fold diluted RT-qPCR reactions compared to skimmed milk flocculation, but extraction with the ViralRNA kit had low SARS-CoV-2 detection in the undiluted RT-qPCR reactions. Membrane filtration only detected OC43 in one of the 15 samples and did not detect SARS-CoV-2 in any of the samples in undiluted RT-qPCR reactions. Concentration with the Nanobeads® and extraction with the NucleoMag kit successfully improved upon

OC43 recovery and SARS-CoV-2 detection in these samples and can be easily automated, increasing sample throughput.

### 2.4.2 Introduction

SARS-CoV-2 emerged in late 2019 and was quickly detected in wastewater and stools of infected individuals (Medema *et al.*, 2020, Peccia *et al.*, 2020, Wang *et al.*, 2020). This led to a massive expansion of wastewater surveillance, with over 200 universities in more than 50 countries conducting wastewater surveillance of SARS-CoV-2 as of December 2022 (Naughton *et al.*, 2021). Early efforts focused on whether wastewater surveillance worked or not, and if it could be used as an early warning for emergence in a community (Medema *et al.*, 2020, Bibby *et al.*, 2021). More recent projects have switched their focus to looking for variants and studying changes in wastewater levels (Yu *et al.*, 2021, Centers for Disease Control and Prevention, 2022, Solis-Moreira, 2022). It has been suggested that variants can be detected in wastewater up to two weeks before detection in clinical samples (Solis-Moreira, 2022). In order for these goals to be realized, sampling and concentration methods must be chosen that are sensitive and provide results quickly.

Concentrations of SARS-CoV-2 in wastewater are low and often at the limit of detection because it is a respiratory virus with variable shedding in stool (Medema *et al.*, 2020, Miura *et al.*, 2021, Philo *et al.*, 2021). This necessitates a wastewater concentration step before analyzing a sample, but concentration steps introduce additional variability to the data, with virus recoveries being highly variable between methods and research groups (Pecson *et al.*, 2021). At this point in the pandemic, there are numerous publications discussing methods comparisons and optimizations for SARS-CoV-2 wastewater surveillance (Ahmed *et al.*, 2020, Dumke *et al.*, 2021, Pecson

*et al.*, 2021, Philo *et al.*, 2021, Ahmed *et al.*, 2022, Ahmed *et al.*, 2023). The broad conclusion from these papers is that many methods will work to detect SARS-CoV-2. However, as new technologies are developed and implemented, more evaluations must be carried out. A relatively new and promising technique for wastewater surveillance involves using magnetic beads to bind SARS-CoV-2 and separate the virions from wastewater. Wastewater concentration with CeresNano Nanotrap ® Magnetic Virus Particles (SKU# 44202, CeresNano, Manassas, VA, USA) has been successfully implemented in San Diego, California (Karthikeyan *et al.*, 2021), but it has not undergone the same systematic validations as many existing methods.

A recent study compared the magnetic Nanotrap® beads with membrane filtration using negatively charged filters (Ahmed *et al.*, 2023). Ahmed *et al.* (2023) found that concentration with the Nanotrap® beads resulted in higher concentrations of PMMoV but it was highly sensitive to wastewater turbidity. Additionally, SARS-CoV-2 concentrations were higher with membrane filtration than with the Nanotrap® beads. In the current study, we aimed to compare concentration with both membrane filtration and the Nanotrap® beads to a previously validated skimmed milk flocculation protocol (Philo *et al.*, 2022). Concentration with membrane filtration, Nanotrap magnetic beads using an existing protocol (Rasile & Maas, 2021), and skimmed milk flocculation were compared by assessing recovery of OC43 seeded into the samples and detection of SARS-CoV-2 in the wastewater. Samples were collected from three different wastewater treatment plants in the Seattle area. The Nanotrap magnetic beads were extracted with both the NucleoMag DNA/RNA water kit (Macherey-Nagel, Düren, Germany) and the ViralRNA Mini kit (QIAGEN, Germantown, MD, USA) to determine if added efficiency is due to concentration or extraction steps.

## 2.4.3 Experimental Methods

### 2.4.3.1 Wastewater Sampling

Primary wastewater was grab sampled weekly from three Seattle area wastewater treatment plants from May to September 2021 (Appendices Table 2.5.6). All wastewater was transported to the Environmental and Occupational Health Sciences Lab (EOHML) on ice and was stored at 4°C. All samples were processed within four days of collection. Grab samples collected over the course of a single day were composited and mixed. Wastewater was seeded with human coronavirus OC43 (OC43) at a final concentration of $3.3x10^4$ $TCID_{50}$/L (OC43, ATCC VR-1558). Viral stocks were produced following a previously published protocol and were stored at -80°C until use (Section 2.1.3.1, Philo *et al.* (2021)).

### 2.4.3.2 Skimmed Milk Flocculation Pellet Extraction

Skimmed milk flocculation was carried out on 100mL of wastewater as previously discussed (Section 2.2, Philo *et al.* (2022)). In short, a 5% skimmed milk solution was added to 100mL of seeded wastewater (1% v/V final) and the pH of the wastewater was dropped to 3 – 4 using 5M HCl. The skimmed milk and wastewater solution was then shaken at 200RPM for two hours, divided into 50mL conicals, and centrifuged at 3500G and 4°C for 30 minutes. The supernatant was poured off and pellets resuspended in 560uL Buffer AVL from the QIAamp Viral RNA Mini Kit prepared with carrier RNA according to the manufacturer's protocol (QIAGEN, Germantown, MD, USA). The resuspended pellet was transferred to a PowerBead Pro tube (Cat. No. 19301, QIAGEN, Germantown, MD, USA). The bead tubes were next horizontally vortexed at maximum speed for 10 minutes and centrifuged at 15,000G for one minute. The supernatant was then transferred to a new 1.5mL flip cap tube (one tube per pellet)

without disturbing the solids or beads and 560uL of 100% ethanol added to the supernatant.

The genomic material was then bound to the columns (one column per pellet) by spinning 630uL at 15,000G for five minutes until the entire mixture had passed through the columns. The extraction then followed the manufacturer's instructions starting with step eight. For elution, 40uL of buffer AVE was added to each column and incubated for one minute. After elution, the same 40uL was applied to the column again for a second elution. The elution from replicate columns was combined and re-aliquoted for immediate molecular processing or storage at -20°C.

#### 2.4.3.3 Membrane Filtration

Membrane filtration was adapted from existing protocols (Ahmed *et al.*, 2015, Ahmed *et al.*, 2020) and was carried out using mixed cellulose ester membrane filters with a 0.45um pore size (47mm, HAWG047S6 Millipore Membrane Filter, MilliporeSigma, Burlington, MA, USA). A step-by-step protocol is included in the appendices in section 2.5.4. Prior to filtration, 2.5mL of a 1M $MgCl_2$ solution was added to 100mL of wastewater (25mM final concentration). The pH of the wastewater was adjusted to 3 – 4 using 5M HCl and pH strips. Wastewater was filtered 50mL at a time for up to one hour. Filters were added to 5.0mL screwcap tubes with 0.7mm garnet beads (one per filter). Lysis Buffer MWA1 from the NucleoMag DNA/RNA Water kit (Macherey-Nagel, Düren, Germany) was added to each tube (900uL/tube) and the samples were vortexed horizontally at maximum speed for 10 minutes. After vortexing, 450uL of lysate was transferred to a sterile 1.5mL flip cap tube, and extraction continued following step 3 of the manufacturer's protocol. Each tube was eluted with 50uL of nuclease free water for a total elution volume of 100uL per sample. Eluate from

replicate tubes was combined and re-aliquoted for immediate molecular processing or storage at -20°C.

#### 2.4.3.4 CeresNano Nanotrap Beads Concentration

Wastewater was concentrated using CeresNano Nanotrap® Magnetic Virus Particles (SKU# 44202, CeresNano, Manassas, VA, USA) following a publicly available protocol on protocols.io (Rasile & Maas, 2021). The full protocol is available in the appendices in section 2.5.4. Briefly, 40mL of seeded wastewater was combined with 600uL of Nanotrap® Magnetic Virus Particles and incubated at room temperature for 20 minutes. The beads were separated using a 50mL magnetic rack and the supernatant gently removed using serological pipettes. Genetic material was then extracted using either the NucleoMag DNA/RNA Water kit or the ViralRNA Mini kit.

##### *2.4.3.4.1 Extraction with the NucleoMag DNA/RNA Water Kit*

The beads were resuspended in 500uL of buffer MWA1 from the Macherey-Nagel NucleoMag DNA/RNA Water kit. This solution was incubated at room temperature for 10 minutes, and the Nanotrap beads again separated on a 50mL magnetic rack. Next, 450uL of lysate was transferred to a sterile 1.5mL flip cap tube and the extraction continued starting with step 12 of the protocol on protocols.io. RNA was eluted in 60uL of RNase-free water. Eluate was ready for immediate molecular processing or was stored at -20°C until further use.

##### *2.4.3.4.2 Extraction with the ViralRNA Mini Kit*

Wastewater concentrated using Nanotrap® Magnetic Virus Particles was also extracted using the ViralRNA Mini Kit. After concentration, beads were resuspended in 560uL of Buffer AVL prepared with carrier RNA according to the manufacturer's protocols. The resuspension was incubated at room temperature for 10 minutes, and

the beads again separated on a 50mL conical magnetic rack. Supernatant was then transferred to a sterile 1.5mL flip cap tube, and the extraction continued following step five of the manufacturer's spin protocol. Samples were eluted in 60uL of Buffer AVE for immediate molecular processing or storage at -20°C until future use.

#### 2.4.3.5 RT-qPCR

Reverse-transcription quantitative PCR (RT-qPCR) was carried out for both OC43 and the SARS-CoV-2 N1 and N2 genes on the extracts as previously published (Philo *et al.*, 2022). The iTaq Universal Probes One-Step kit was used with the BioRad CFX qPCR systems (Bio-Rad Laboratories, Hercules, CA, USA). Each sample was run in duplicate with duplicate 10-fold dilutions. Primers and probes were obtained from IDT (Integrated DNA Technologies, Inc., Coralville, IA, USA). The sequences and final concentrations are listed in Table 2.4.1. Cycling conditions for both the OC43 and SARS-CoV-2 assay were 50°C for 10 minutes, 95°C for three minutes, and 45 cycles of 95°C for 15 seconds and 60°C for 30 seconds. The negative control for both reactions was nuclease-free water. Standard curves for the OC43 assay were generated using ten-fold dilutions of a synthetic RNA construct previously developed (Philo *et al.*, 2022). Standard curves for the multiplexed SARS-CoV-2 assay was RNA extract obtained from

**Table 2.4.1**: Primer and probe sequences and final concentrations for OC43 and SARS-CoV-2 RT-qPCR.

| Target | Name | Sequence (5' – 3') | Final Conc. | Reference |
|---|---|---|---|---|
| OC43 | OC43-FP | ATGTTAGGCCGATAATTGAGGACTAT | 0.3uM | Vijgen *et al.* (2005) |
| | OC43-RP | AATGTAAAGATGGCCGCGTATT | 0.3uM | |
| | OC43-TP | FAM-CATACTCTGACGGTCACAAT-NFQ-MGB | 0.2uM | |
| | OC43-FPT7 | TAATACGACTCACTATAGGGAGGATGTTAGGCCGATAATTGAGGACTAT | 0.3uM | |
| SARS-CoV-2 N1 | N1-F | GACCCCAAAATCAGCGAAAT | 0.2uM | US Centers for Disease Control and Prevention (2020) |
| | N1-R | TCTGGTTACTGCCAGTTGAATCTG | 0.2uM | |
| | N1-P | FAM-ACCCCGCAT-ZEN-TACGTTTGGTGGACC-3IABkFQ | 0.2uM | |
| SARS-CoV-2 N2 | N2-F | TTACAAACATTGGCCGCAAA | 0.2uM | |
| | N2-R | GCGCGACATTCCGAAGAA | 0.2uM | |
| | N2-P | HEX-ACAATTTGC-ZEN-CCCCAGCGCTTCAG-3IABkFQ | 0.2uM | |

BEI (Item No. NR-52285, BEI Resources, Manassas, VA, USA). Standard curves for both assays were run in duplicate.

#### 2.4.3.6 Data Analyses

RT-qPCR data were analyzed using the Bio-Rad CFX Maestro for Mac Program (Bio-Rad Laboratories, Hercules, CA, USA). Assay results and sample data were maintained using Microsoft Excel (Microsoft Corp., Redmond, WA, USA) and REDCap electronic data capture tools hosted at the University of Washington, Seattle (Harris *et al.*, 2009, Harris *et al.*, 2019). Data manipulation and statistics were carried out using RStudio and associated packages (2019 RStudio: Integrated Development for R. RStudio, Inc., Boston, MA, URL http://www.rstudio.com/) (Xu *et al.*, 2021).

The volume adjusted amount of initial wastewater assayed in each RT-qPCR reaction (effective volume assayed) was calculated for each concentration method. Fractional OC43 recovery was calculated using average sample Cq values and associated standard curves as previously discussed in Section 2.3 (Philo *et al.*, 2021, Philo *et al.*, 2022). Wilcoxon signed-rank tests were carried out on the fractional OC43 recovery with paired wastewater samples for each concentration method. Membrane filtration and the Nanotrap ® beads were compared to the previously validated skimmed milk pellet extraction method presented in section 2.2 (Philo *et al.*, 2022). SARS-CoV-2 detection achieved by each method was compared using McNemar's Chi-Square test for count data for both diluted and ten-fold diluted reactions using the built-in R function (mcnemar.test()) (McNemar, 1947, Smith & Ruxton, 2020). Samples were considered positive for SARS-CoV-2 if either of the duplicate reactions for N1 or N2 had amplification with Cq < 40.

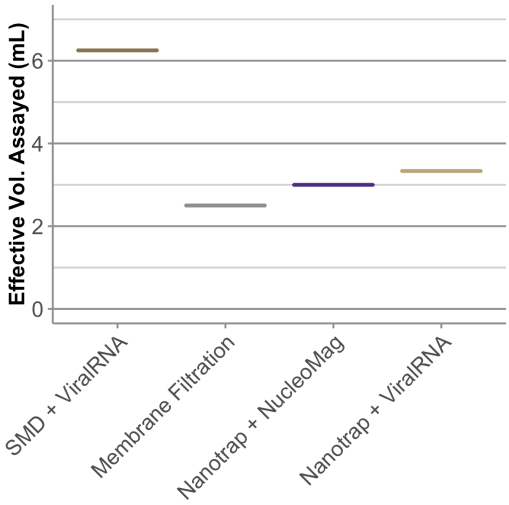



**Figure 2.4.1**: Effective volume assayed for each of the methods compared in this study. Effective volume assayed is the volume of initially sampled wastewater assayed in each RT-qPCR reaction.

## 2.4.4 Results and Discussion

### 2.4.4.1 Recovery Comparisons

The effective volume assayed was calculated for each method compared in the study (Figure 2.4.1). The skimmed – milk direct pellet concentration and extraction method had the highest effective volume assayed (6.25mL) and membrane filtration had the lowest effective volume assayed (2.5mL). For RT-qPCR data to be compared between sampling sites and process labs, those data must be converted to gene copies per volume of sampled wastewater from gene copies per PCR reaction volume. The US Centers for Disease Control and Prevention (CDC) does not allow data to be uploaded into the NWSS database without this adjustment (Centers for Disease Control and Prevention, 2022). However, effective volume assayed is not commonly reported in the scientific literature. Standard data collection and reporting procedures are needed to help public health officials incorporate wastewater data into their programs.

OC43 recovery was assessed for each of the new methods compared to the skimmed milk pellet extraction (Figure 2.4.2). Membrane filtration had the lowest

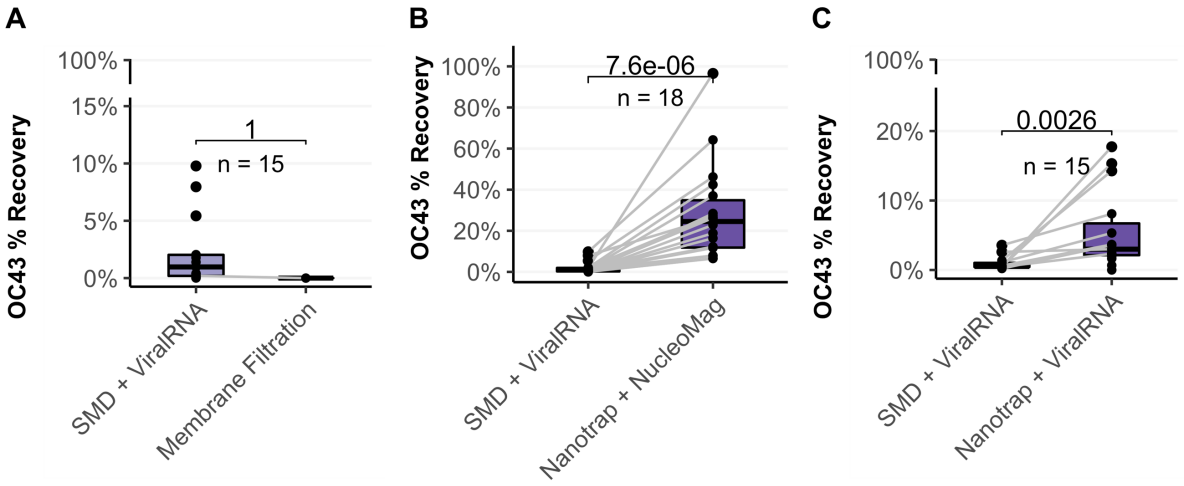



**Figure 2.4.2**: Paired OC43 method recovery efficiency with Wilcoxon signed-rank test p-values comparing the skimmed milk direct pellet extraction method with **A**) membrane filtration, **B**) the Nanotrap beads with the NucleoMag extraction kit and **C**) the Nanotrap beads with the ViralRNA extraction kit. Method recovery is the ratio of seeded OC43 detected by RT-qPCR to the theoretical amount that would have been detected with 100% recovery. Nanotrap bead extracted with both the NucleoMag kit and the ViralRNA kit significantly improve on the skimmed milk pellet extraction method (p-values < 0.001 and p = 0.0026, respectively). Membrane filtration performed very poorly for OC43 recovery, with OC43 detected in only one sample.

recovery, with only one reaction resulting in OC43 detection in the undiluted RT-qPCR reaction (Figure 2.4.2). It was not possible to statistically compare skimmed milk with membrane filtration because of membrane filtration's poor performance in this study. Despite these results with Seattle-area wastewater, membrane filtration has been successfully used around the world to detect SARS-CoV-2 in wastewater in places such as Louisiana and Virginia, USA (Gonzalez *et al.*, 2020, Sherchan *et al.*, 2020) and Brisbane, Australia (Ahmed *et al.*, 2020). However, in the past, membrane filtration has not been successfully implemented for other targets in Seattle wastewater (Zhou *et al.*, 2022). The specific wastewater matrix can have substantial effects on method performance. A study carried out with 32 different labs in the United States showed orders of magnitude difference in recovery of OC43 used as a matrix control organism between different lab groups (Pecson *et al.*, 2021). This highlights how strongly the specific wastewater can affect results.

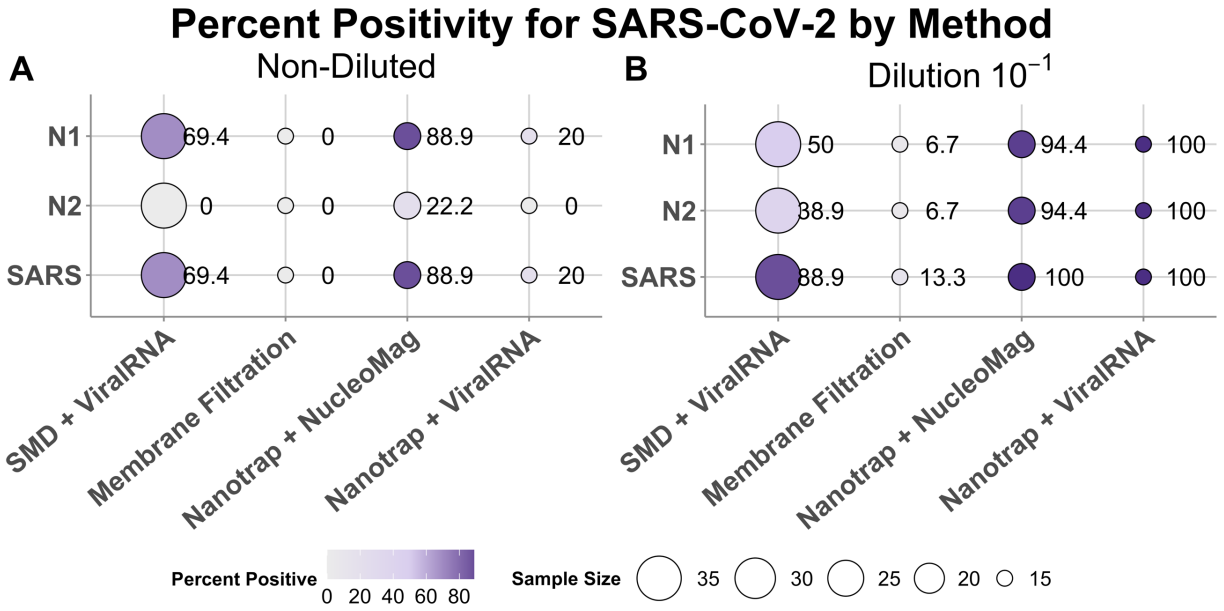



**Figure 2.4.3**: Balloon plots assessing SARS-CoV-2 percent positivity by RT-qPCR assay and method for either **A**) non-diluted or **B**) ten-fold diluted RT-qPCR reactions. Darker colors indicate higher detection and the size of circle is the number of samples assayed. The N1 assay has more detection in the undiluted reactions across methods. The Nanotrap ® beads with the NucleoMag kit has the highest SARS-CoV-2 detection in the undiluted reactions, and both Nanotrap methods have 100% SARS-CoV-2 detection in the ten-fold diluted reactions.

Both variations of the Nanotrap protocols significantly improved upon OC43 recovery compared to skimmed milk (Figure 2.4.2, Appendices table 2.5.7). The average OC43 recovery for the Nanotrap particles with the NucleoMag extraction kit is 28.1% (95% CI: 16.6 – 39.6) and with the ViralRNA kit it is 5.5% (95% CI: 2.3 – 8.6). Magnetic beads have long been used in genomic extraction and clean-up protocols, but they are only recently being used to separate infectious disease targets from wastewater (Karthikeyan *et al.*, 2021, Rasile & Maas, 2021, Ahmed *et al.*, 2023). The Nanotrap particles also successfully concentrate PMMoV, a marker of human fecal contamination that is commonly being used to normalize SARS-CoV-2 data (Ahmed *et al.*, 2023). There are numerous platforms that can be used to automate magnetic bead protocols. The Nanotrap ® particles can help scale up existing wastewater surveillance programs.

### 2.4.4.2 SARS-CoV-2 Detection

Two-by-two contingency tables were generated for SARS-CoV-2 detection using the criteria described in section 2.4.3.6 (Appendices Table 2.5.8). Statistical comparisons were not possible for the diluted Nanotrap particles extracted with the ViralRNA mini kit and skimmed milk because both methods detected SARS-CoV-2 in 100% of the samples. Using McNemar's Chi-Square test, skimmed milk detects SARS-CoV-2 significantly more than membrane filtration (Table 2.4.2), but both protocols with Nanotrap beads detected SARS-CoV-2 significantly more than skimmed milk in the undiluted RT-qPCR reactions (Table 2.4.2). In the ten-fold diluted RT-qPCR reactions, there is no difference in SARS-CoV-2 detection between skimmed milk and the Nanotrap® particles with the NucleoMag extraction kit (Table 2.4.2).

RT-qPCR for the N2 target is less likely to detect SARS-CoV-2 than the N1 target (Figure 2.4.3). Previous work also suggests that N2 is less sensitive to detect SARS-CoV-2 in Seattle wastewater (Philo *et al.*, 2021, Philo *et al.*, 2022). However, the ten-fold diluted RT-qPCR reactions for both Nanotrap extraction protocols detect N2 equally as well as N1 (Figure 2.4.3), suggesting that these protocols better clean up inhibitors of the N2 reaction. The Nanotrap particles with the NucleoMag extraction kit detects SARS-CoV-2 in almost as many undiluted reactions as ten-fold diluted reactions (Figure 2.4.3), suggesting there are fewer false negatives with this protocol. False negatives are important to understand in the context of wastewater surveillance but are not well

**Table 2.4.2**: Results from McNemar's chi-squared test on two-by-two contingency tables comparing membrane filtration, Nanotrap ® beads with the NucleoMag kit, and Nanotrap ® beads with the ViralRNA kit with skimmed milk flocculation for **A)** non-diluted and **B**) ten-fold diluted RT-qPCR reactions.

| | | Method 1 | Method 2 | McNemar's $X^2$ | p-value |
|---|---|---|---|---|---|
| **A)** | *Non-Diluted* | Skimmed Milk Direct | Membrane Filtration | 5.14 | 0.02 |
| | | | Nanotrap + NucleoMag | 4.9 | 0.03 |
| | | | Nanotrap + ViralRNA | 10.08 | 0.001 |
| **B)** | *1:10 Dilution* | Skimmed Milk Direct | Membrane Filtration | 6.75 | 0.009 |
| | | | Nanotrap + NucleoMag | 1.33 | 0.25 |
| | | | Nanotrap + ViralRNA | - | - |

reported (Ahmed *et al.*, 2022). False negatives or false positives may lead to improper choices by public officials. Reducing false negatives is crucial to implementing successful wastewater surveillance programs (Kumblathan *et al.*, 2022). The current study shows that running ten-fold dilutions with RT-qPCR or selecting a method that better removes inhibitors are both viable ways to reduce false negatives.

### 2.4.5 Conclusion

This manuscript presents a systematic methods comparison between membrane filtration and Nanotrap particles with the previously validated skimmed milk pellet extraction protocol (Philo *et al.*, 2022). The previous methods comparison carried out by this group was limited by supply availability and the difficulty of doing research while collecting data for public health officials (Philo *et al.*, 2021). The current methods comparison takes a much more systematic approach and ensures that there are enough replicates to make statistical comparisons. While prior research identified skimmed milk pellet extraction as a low-cost and consistent method to detect SARS-CoV-2 (Philo *et al.*, 2022), the Nanotrap particles perform significantly better than skimmed milk pellets and are more easily scaled up (Karthikeyan *et al.*, 2021, Ahmed *et al.*, 2023). This study is limited by the high prevalence of SARS-CoV-2 in the wastewater. Future work should aim to conduct studies seeded with SARS-CoV-2 to quantify recovery of the target organism rather than a surrogate. Much of the focus of SARS-CoV-2 wastewater surveillance has also switched to searching for different variants, with some research suggesting they can be detected in wastewater up to two weeks before clinical testing (Solis-Moreira, 2022) and that the variants detected are similar to those that are collected from clinical samples (Rios *et al.*, 2021). Understanding the circulating variants is crucial because later variants are more

resistant to certain antibody treatments (Aleem *et al.*, 2022, Singh *et al.*, 2022). High throughput wastewater surveillance for SARS-CoV-2 variants can help medical facilities prepare with the correct treatments. Concentration and extraction with the Nanotrap ® beads and the NucleoMag kit, respectively, should be tested with sequencing to help facilitate rapid detection of variants.

### 2.4.6 Acknowledgments and Funding

This work was carried out with the support of Cynthia Ripoll, PhD and Macherey-Nagel, Inc., who provided extraction kits for experimental comparisons, and Tara Jones-Roe and Ceres Nanosciences, Inc., who provided the Nanotrap particles to test. Our collaborators at the wastewater treatment plants, the West Point Process Lab, the South Treatment Plant collection team, and the Brightwater Operations staff, facilitated sample collection. Research reported in this publication was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under award number T32ES015459. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

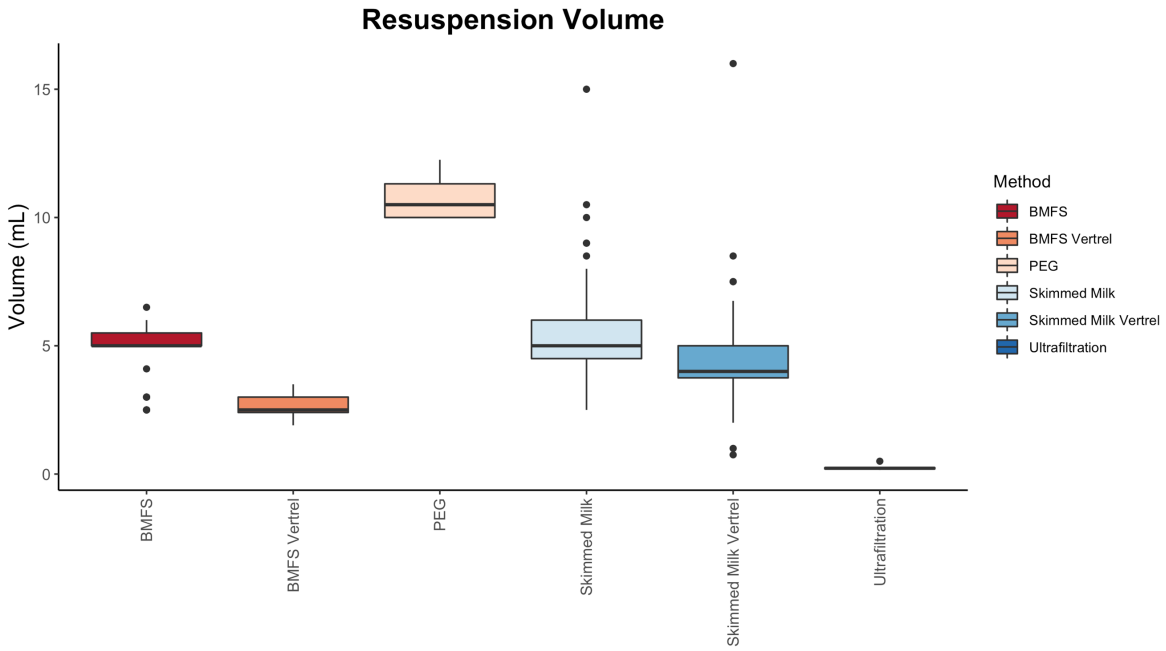
# 2.5 SARS-CoV-2 Appendices

## 2.5.1 First Methods Comparison

### 2.5.1.1 Supplementary Tables and Figures

**Table 2.5.1**: Dates and Volumes sampled for each wastewater treatment plant by week of the year.

| | Plant A | | Plant B | | Plant C | |
|---|---|---|---|---|---|---|
| ***Week of the year*** | *Date Sampled* | *Volume Sampled* | *Date Sampled* | *Volume Sampled* | *Date Sampled* | *Volume Sampled* |
| **Week 14** | Wednesday, April 1, 2020 | 2.66 L | Tuesday, March 31, 2020 | 2.86 L | Friday, April 3, 2020 | 2.78 L |
| **Week 15** | Wednesday, April 8, 2020 | 2.73 L | Tuesday, April 7, 2020 | 2.8 L | Friday, April 10, 2020 | 2.84 L |
| **Week 16** | Wednesday, April 15, 2020 | 2.94 L | Tuesday, April 14, 2020 | 2.86L | Friday, April 17, 2020 | 2.85 L |
| **Week 17** | Wednesday, April 22, 2020 | 3.0 L | Tuesday, April 21, 2020 | 2.8 L | Friday, April 24, 2020 | 3.0 L |
| **Week 18** | Wednesday, April 29, 2020 | 3.0 L | Tuesday, April 28, 2020 | 2.8 L | Friday, May 1, 2020 | 3.0 L |
| **Week 19** | Tuesday, May 5, 2020 | 3.99 L | Tuesday, May 5, 2020 | 4.0 L | Tuesday, May 5, 2020 | 3.0 L |
| **Week 20** | Tuesday, May 12, 2020 | 3.0 L | Tuesday, May 12, 2020 | 3.0 L | Tuesday, May 12, 2020 | 3.6 L |
| **Week 21** | Tuesday, May 19, 2020 | 3.1 L | Tuesday, May 19, 2020 | 3.1 L | Tuesday, May 19, 2020 | 3.6 L |
| **Week 22** | Tuesday, May 26, 2020 | 3.1 L | Tuesday, May 26, 2020 | 3.1 L | Tuesday, May 26, 2020 | 3.6 L |
| **Week 23** | Tuesday, June 2, 2020 | 3.1 L | Tuesday, June 2, 2020 | 3.1 L | Tuesday, June 2, 2020 | 7.1 L |
| **Week 24** | Tuesday, June 9, 2020 | 3.0 L | Tuesday, June 9, 2020 | 3.1 L | Tuesday, June 9, 2020 | 6.6 L |
| **Week 25** | Tuesday, June 16, 2020 | 3.0 L | Monday, June 15, 2020 | 3.1 L | Tuesday, June 16, 2020 | 6.6 L |
| **Week 26** | Tuesday, June 23, 2020 | 3.0 L | Tuesday, June 23, 2020 | 3.1 L | Tuesday, June 23, 2020 | 6.6 L |
| **Week 27** | Tuesday, June 30, 2020 | 3.0 L | Tuesday, June 30, 2020 | 3.1 L | Tuesday, June 30, 2020 | 3.1 L |
| **Week 28** | Tuesday, July 7, 2020 | 3.0 L | Wednesday, July 8, 2020 | 3.0 L | Wednesday, July 8, 2020 | 3.0 L |




| | BMFS | BMFS – Vertrel | PEG | Skimmed Milk | Skimmed Milk - Vertrel | Ultrafiltration |
|---|---|---|---|---|---|---|
| **Maximum** | 6.50 | 3.50 | 12.25 | 15.00 | 16.00 | 0.50 |
| **3rd Quartile** | 5.50 | 3.00 | 11.31 | 6.00 | 5.00 | 0.25 |
| **Median** | 5.00 | 2.50 | 10.50 | 5.00 | 4.00 | 0.23 |
| **1st Quartile** | 5.00 | 2.40 | 10.00 | 4.50 | 3.75 | 0.20 |
| **Minimum** | 2.50 | 1.90 | 10.00 | 2.50 | 0.75 | 0.16 |
| **Mean** | 4.78 | 2.69 | 10.81 | 5.60 | 4.53 | 0.23 |

**Figure 2.5.1**: Resuspension volume after wastewater concentration. BMFS is the Bag-Mediated Filtration System. PEG is polyethylene glycol precipitation. BMFS and skimmed milk pellets were resuspended in 4mL and 6mL of 1X PBS (pH=7.4), respectively. Resuspension volumes for Vertrel extracted samples is the supernatant volume after extraction. PEG precipitated pellets were resuspended in 6mL of 1X PBS (pH=7.4). The ultrafiltration resuspension volume is the volume of the retentate after filtering 100mL of wastewater.

**Table 2.5.1**: Effective volume assayed by method. All values are reported in mL.

| | BMFS | BMFS - Vertrel | PEG | Skimmed Milk | Skimmed Milk - Vertrel | Ultrafiltration |
|---|---|---|---|---|---|---|
| **Maximum** | 26.67 | 14.68 | 1.17 | 1.56 | 2.33 | 7.20 |
| **3rd Quartile** | 11.67 | 11.67 | 1.17 | 1.30 | 1.56 | 5.83 |
| **Median** | 11.15 | 10.73 | 1.11 | 1.17 | 1.46 | 5.19 |
| **1st Quartile** | 10.01 | 8.94 | 1.03 | 0.93 | 1.17 | 4.67 |
| **Minimum** | 5.88 | 6.13 | 0.95 | 0.23 | 0.58 | 2.33 |
| **Mean (mL)** | **11.90** | **10.41** | **1.09** | **1.10** | **1.37** | **5.31** |

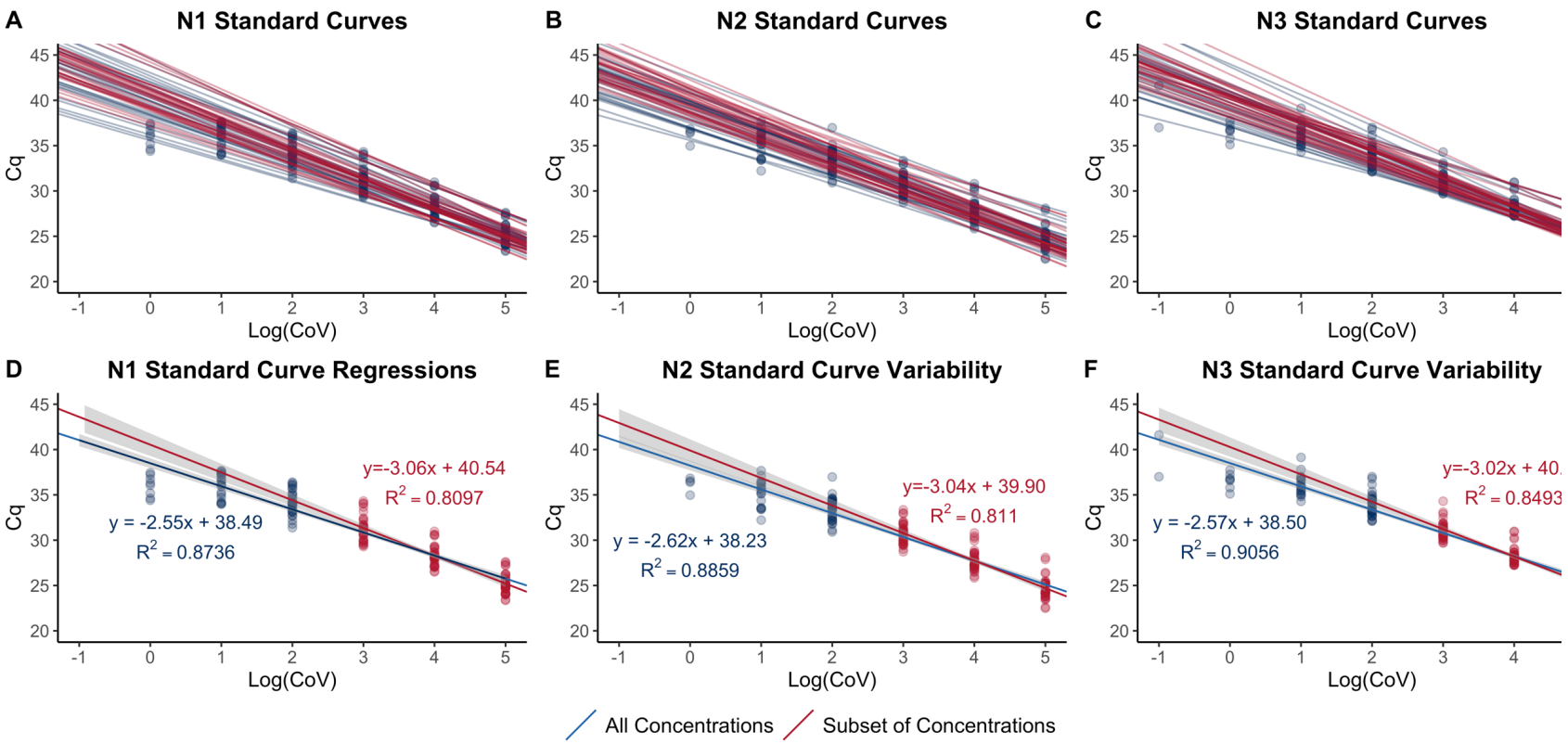



**Figure 2.5.2***:* Standard curves for each of the SARS-CoV-2 RT-qPCR assays. A-C) Slopes and intercepts were extracted for each standard curve for each assay run using all concentrations (blue) and log(5) through log(3) concentrations (red). For all three assays, there is considerable variability in the standard curves between each method. Additionally, the standard curves constructed using the first three dilutions are different than the curves using all the dilutions. D-F) Linear regression was run on all concentrations and on the highest concentrated standards. There is a considerable drop in linearity in more dilute positive control standards.

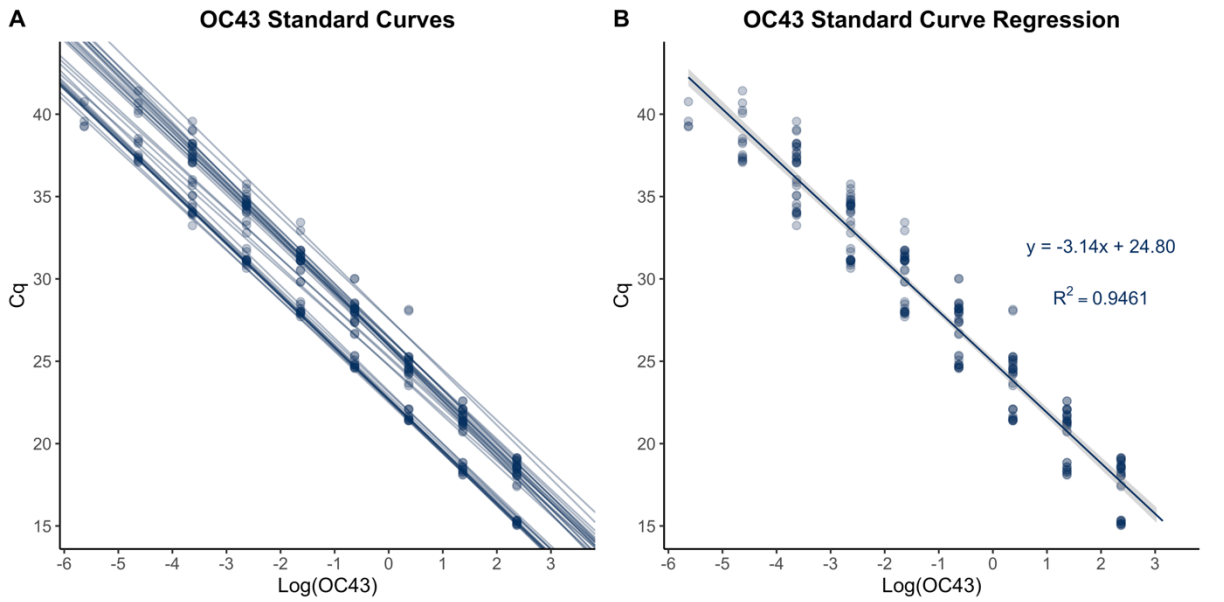



**Figure 2.5.3:** Standard curves for OC43 RT-qPCR assay. A) Slopes and intercepts were extracted for each standard curve for each assay run using all concentrations of OC43. While the intercept shifts left and right, the slope is relatively constant across assays. B) A linear regression of Cq against log(CoV) was run to produce a single standard curve line representative of all the data.

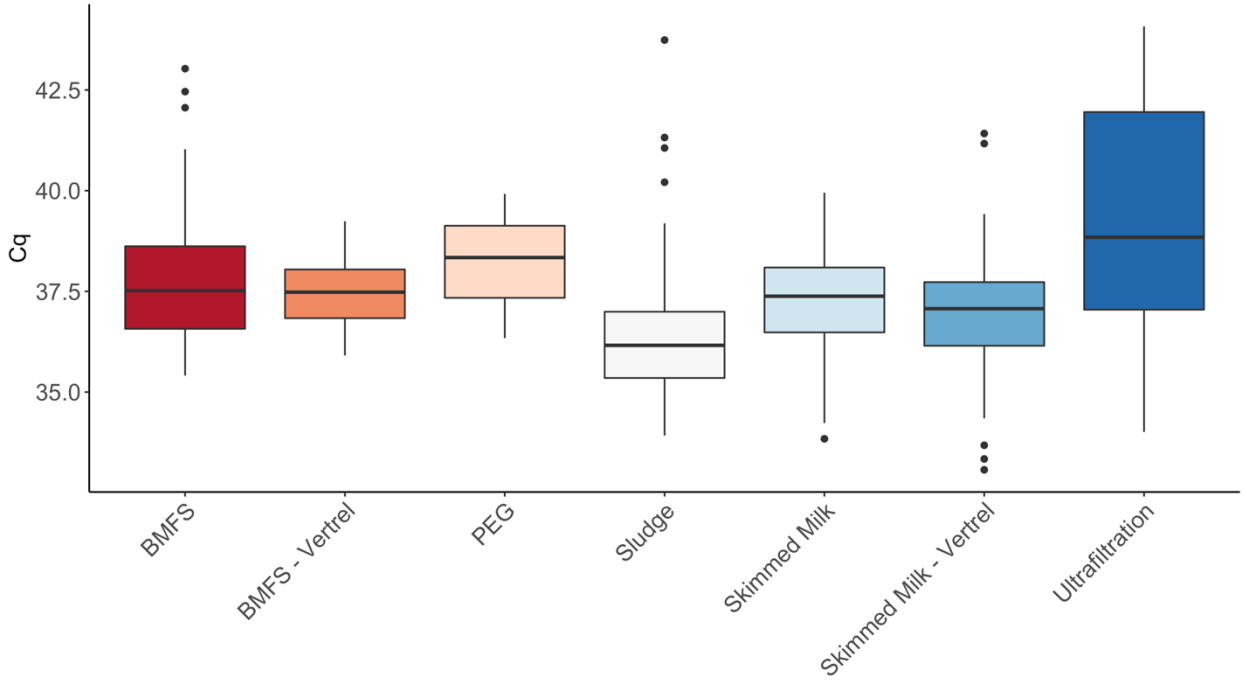



**Figure 2.5.4***: SARS-CoV-2 assay mean Cq by method.* BMFS is bag-mediated filtration system. PEG is polyethylene glycol.

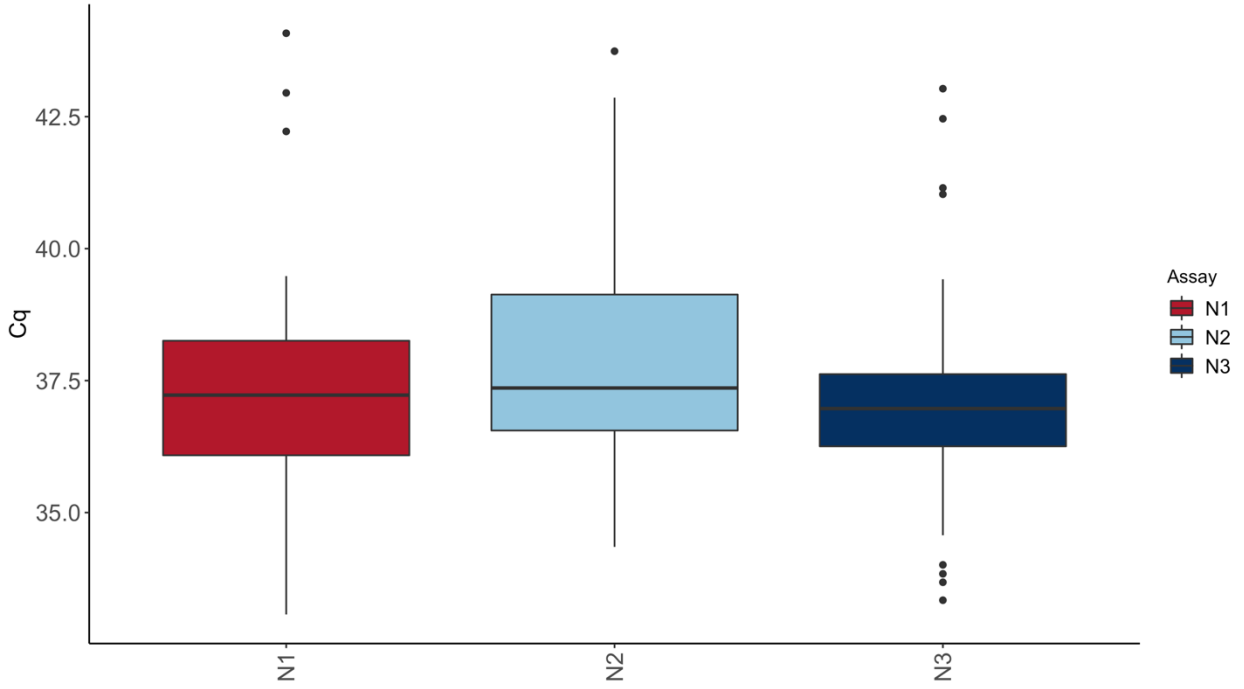



**Figure 2.5.5:** Mean Cq values for the N1, N2, N3 SARS-CoV-2 assays.

### 2.5.1.2 Supplementary Methods

#### *2.5.1.2.1 Bag-Mediated Filtration System*

Wastewater samples (Table 2.1.1) were concentrated by pumping the sample through a positively charged 2" ViroCap filter using a peristaltic pump (Scientific Methods, Granger, IN, USA) with an average volume of 2.63L. After filtration, the filters were eluted using a forward flush double elution of 150mL 1.5% beef extract and 0.05 M glycine solution (pH 9.5). The eluent was held in the filter housing unit for 15 minutes before flushing and a second elution for 15 minutes, for a total eluent volume of 300mL.

Secondary concentration was performed using a 5% skimmed milk solution. Three (3) mL of this skimmed milk solution was added to the 300mL eluent, and the pH was adjusted to 3.0-4.0 using 5M HCl. The pH-adjusted skimmed milk-added eluent was placed on a shaking table at room temperature (20-25°C) and 200RPM for two hours. After shaking, the sample was centrifuged (3500 x G, 4°C, 30 minutes). The

supernatant was then discarded, and the pellet was resuspended in 4mL of sterile PBS (pH = 7.4) by vortexing at maximum speed for 10 minutes. The sample was then either frozen at -80°C for RNA extraction or split into two volumes with one portion saved for Vertrel extraction and the other stored at -80°C for RNA extraction. The final resuspension and Vertrel extracted volumes are reported in supplemental information Figure A1.

#### *2.5.1.2.2 Skimmed-Milk Flocculation*

Composite wastewater samples in volumes of 0.1L, 0.5L, and 1.0L were concentrated using 1mL of a 5% skimmed milk solution per 100mL of wastewater. The pH of the skimmed-milk-wastewater solution was adjusted to a pH of 3.0-4.0 using 5M HCl. This was then placed on a shaking table at room temperature (20-25°C) and 200RPM for two hours. After shaking, the sample was centrifuged (3500 x G, 4°C, 30 minutes). The supernatant was discarded, and the pellets resuspended in either 10mL or 6mL of sterile PBS (pH = 7.4) by vortexing at maximum speed for 10 minutes. The volume of PBS used was dependent on the initial volume concentrated with 0.1L samples resuspended in 6mL and 0.5L and 1.0L samples in 10mL. Final resuspension volumes are reported in supplemental information Figure 2.5.1.

#### *2.5.1.2.3 Vertrel Extraction*

Vertrel XF (Miller-Stephenson, Inc., Danbury, CT, USA) was added to the aliquoted BMFS and skimmed-milk samples in a 1:5 ratio of Vertrel XF : sample. After adding Vertrel XF, the samples were parafilmed and vortexed for five minutes at maximum speed, placed on ice for three minutes, and then vortexed again for five minutes at maximum speed. The samples were then centrifuged (3000 x G, 4°C, 15

minutes). After centrifugation, the top layer was pipetted without disturbing the middle or bottom layers. This supernatant was then frozen at -80°C for RNA extraction.

#### *2.5.1.2.4 Polyethylene Glycol Precipitation*

Concentration of composite primary wastewater from Plant C was carried out with PEG/NaCl precipitation by adding polyethylene glycol (PEG) 8000 (14g/100mL) and NaCl (1.17g/100mL). After ensuring the PEG and NaCl was completely dissolved by vigorous shaking, the sample was placed on a shaking table (4 hours, 4°C, 200RPM). The samples were then centrifuged (6500 x G, 4°C, 30 minutes). The supernatant was removed, and the pellets were resuspended in 6mL of sterile PBS (pH = 7.4) by vortexing at maximum speed for 10 minutes. The resuspended pellets were then stored at -80°C for RNA extraction. Final resuspension volumes are reported in supplemental information Figure 2.5.1.

#### *2.5.1.2.5 Ultrafiltration*

Ultrafiltration with Centricon Plus-70 centrifugal filter devices (MilliporeSigma, Burlington, MA, USA) was used to concentrate 0.1L of composite wastewater. Prior to ultrafiltration, 0.1L of wastewater was centrifuged (6800 X G, 4°C, 30 minutes) to pellet out the solids. The supernatant was then collected, and the pellets stored at -80°C for RNA extraction. 50mL of the supernatant was spun in the centricons (3500 X G, 4°C, 30 minutes). After filtration, the filtrate was discarded and the remaining supernatant spun in the same conditions. The retentate was then eluted from the filter by spinning the concentration cup upside down on the sample filter cup at 1,000 X G for two minutes, yielding an average volume of 230µL per sample (Figure 2.5.1). The retentate was then stored at -80°C for RNA extraction.

## 2.5.2 Skimmed Milk Pellet Extraction Development and Validation Supplemental Information

### 2.5.2.1 Tables and Figures

**Table 2.5.3**: Dates and volumes concentrated for each week in the skimmed milk pellet extraction development and validation

| **Plant A** | | **Plant B** | | **Plant C** | |
|---|---|---|---|---|---|
| *Date* | *Volume (L)* | *Date* | *Volume (L)* | *Date* | *Volume (L)* |
| 10/27/20 | 0.6 | 10/27/20 | 0.6 | 10/27/20 | 0.6 |
| 11/10/20 | 0.6 | 11/10/20 | 0.6 | 11/10/20 | 0.6 |
| 11/17/20 | 0.6 | 11/17/20 | 0.6 | 11/17/20 | 0.6 |
| 12/1/20 | 0.6 | 12/1/20 | 0.6 | 12/1/20 | 0.6 |
| 12/8/20 | 0.6 | 12/8/20 | 0.6 | 12/8/20 | 0.6 |
| 1/5/21 | 0.6 | 1/5/21 | 0.6 | 1/5/21 | 0.6 |
| 1/19/21 | 0.6 | 1/19/21 | 0.6 | 1/18/21 | 0.6 |
| 1/26/21 | 0.6 | 1/26/21 | 0.6 | 1/25/21 | 0.6 |
| 2/2/21 | 0.6 | 2/2/21 | 0.6 | 2/1/21 | 0.6 |
| 2/9/21 | 0.6 | 2/9/21 | 0.6 | 2/8/21 | 0.6 |
| 2/23/21 | 0.7 | 2/23/21 | 0.7 | 2/22/21 | 0.7 |
| 3/9/21 | 0.7 | 3/9/21 | 0.7 | 3/8/21 | 0.7 |
| 3/16/21 | 0.7 | 3/16/21 | 0.7 | 3/15/21 | 0.7 |
| 3/23/21 | 0.7 | 3/23/21 | 0.7 | 3/22/21 | 0.7 |
| 3/30/21 | 0.7 | 3/30/21 | 0.7 | 3/29/21 | 0.7 |

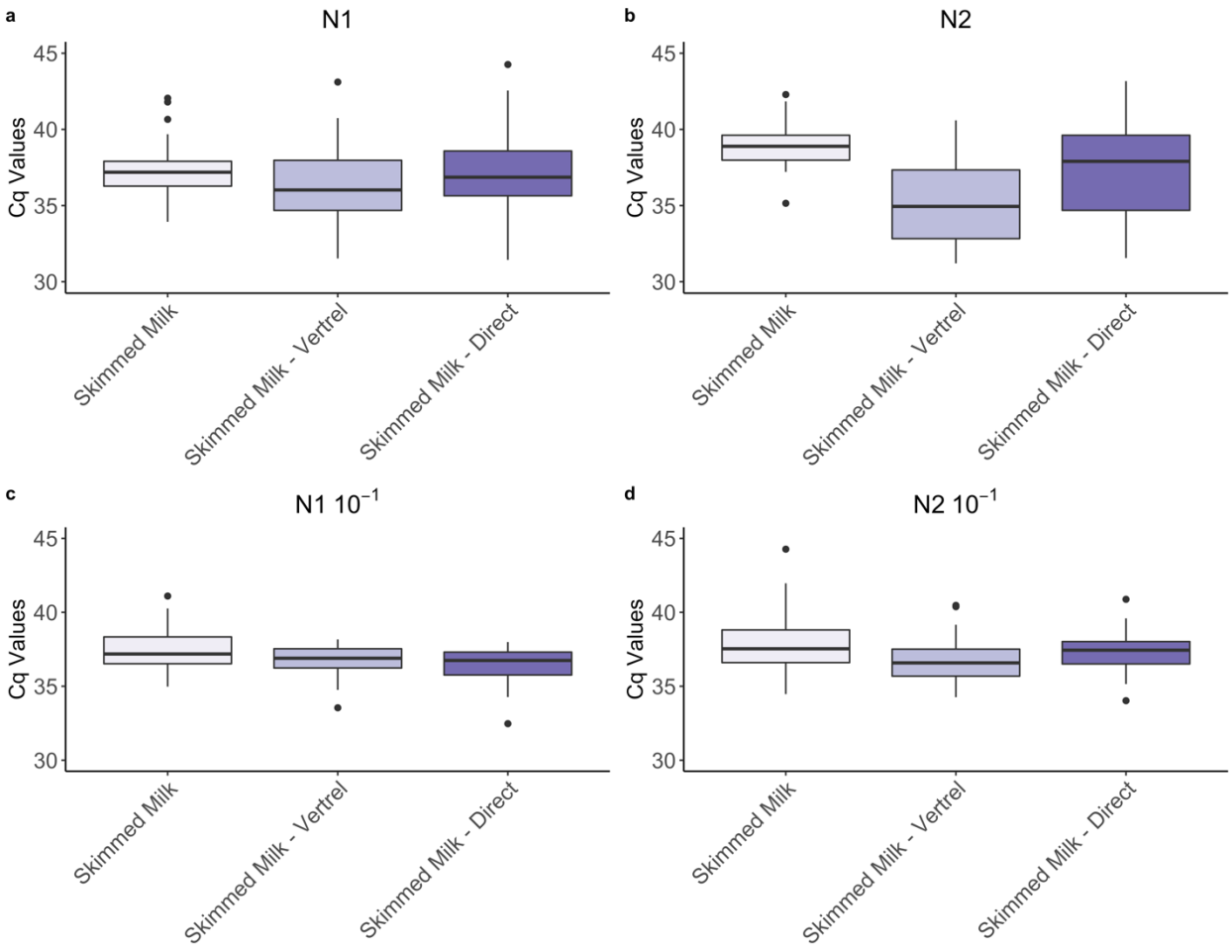



**Figure 2.5.6**: Boxplots indicating the average Cq values of the two replicate RT-qPCR assays for a) N1 and b) N2 and the ten-fold dilution RT-qPCR reactions for c) N1 and d) N2.

### 2.5.2.2 Skimmed Milk Pellet Extraction Protocol

## Day 1

## Preparation

1. Collect influent wastewater in 1-L polypropylene bottles. Store at 4-8°C for up to 4 days
2. Autoclave/ensure available:
    - 500mL polypropylene bottles
    - 10L carboys
    - 0.5L graduated cylinders
    - 50mL conicals
    - 25mL serological pipettes
    - 10mL serological pipettes

- pH strips
- Chlorine strips
- Print labels for 1.5mL tubes
- Viral RNA Mini Kit
- PowerBead Tubes

3. Prepare the following reagents for 3 skimmed milk flocculation samples:
    - 25mL 5% skimmed milk
    - 5M NaOH
    - 5M HCl

**Seeding HCoV-OC43**

1. Thaw viral stock on ice until completely thawed (approximately 30 minutes)
2. Combine WW from a single WWTP in a 10-L carboy using a graduated cylinder or peristaltic pump
3. Measure pH and chlorine content of WW using pH and chlorine strips
4. Place water sample on cart and mix by gently moving back and forth
5. Pump/pour 0.1L of wastewater sample using peristaltic pump into a 0.5L graduated cylinder and transfer 0.1L into a 0.25L or 0.5L bottle
6. Transfer ~30mL of mixed wastewater into a 50mL conical
7. Add 3.4uL of $10^6$ $TCID_{50}$ OC43 stock to the 50mL conical and vortex for 40 seconds for ongoing surveillance samples (3.4uL/100mL)
8. Transfer the seeded mixture to the bottle, close lid, and rock gently dispersing the virus into the WW

**Skimmed Milk Flocculation**

1. Add 1 mL of 5% skimmed milk solution to the 0.1-L sample
    a. To make skimmed milk: add 1.25g of skimmed milk powder to 25mL of DI water
    b. Autoclave for 15 minutes at 121°C
    c. Store at 4°C for up to two days before use
2. Adjust pH to 3.0-4.0 using 5 M HCl and pH strips.
3. Parafilm sample bottle.
4. Shake sample on a shaker for 2 hours at room temperature (20-25°C) at 200 RPM.
5. After shaking, transfer sample to centrifuge conicals (approximately 2-3 for 100ml).
    - Gently swirl sample thoroughly to evenly distribute flocs
6. Spin sample at 3500 x G, 4°C for 30 minutes.
7. Discard supernatant
8. Store skimmed milk pellets in fridge for up to 24 hours

**Direct Pellet Extraction**

1. Spin the PowerBead Pro tube to pull beads to the bottom
2. Add 560uL of AVL buffer + Carrier RNA from the QIAamp Viral RNA Mini Kit to each pellet
    - Transfer the sewage pellet and AVL + Carrier RNA to a PowerBead Pro tube
2. Attach the power bead tubes (balanced) to the horizontal vortex adaptor and vortex at maximum speed for 10 minutes
3. Centrifuge at 15,000g for 1 minute
4. Transfer the supernatant to a clean 1.5mL or 5.0mL flip-cap tube
5. Add 560uL of ethanol (96-100%) to the sample and mix by pulse vortexing for 15 seconds
6. Continue with Step 6 of the QIAamp Viral RNA Mini Kit spin protocol or Step 7 of the vacuum protocol
    - Speed for step 6: 15,000G for 5 minutes
7. Steps 8+, follow speeds/time of spin protocol.
8. Elution:
    - Add 40uL of Buffer AVE to each column and incubate for 1 minute
    - After centrifugation, apply the *same* 40uL to the column for a double elution
9. Combine the 40uL from the two columns into a single tube and ensure it is thoroughly mixed
10. Separate extractions back into 2 separate tubes
11. Store extracts at -20°C until RT-qPCR

## 2.5.3 Extraction Kit Comparison Supplemental Information

### 2.5.3.1 Tables and Figures

**Table 2.5.4**: Descriptive statistics for OC43 method recovery between extraction kit comparisons.

**Descriptive Statistics for OC43 % Recovery**

| | | *Extraction Kit* | *Minimum* | *1st. Quart.* | *Median* | *3rd Quart.* | *Maximum* | *Mean* | *95% CI* |
|---|---|---|---|---|---|---|---|---|---|
| *Kit Comparisons* | **A)** | ViralRNA (n = 18) | 0.13 | 0.32 | 0.56 | 1.14 | 6.5 | 1.28 | 0.42 to 2.15 |
| | | PowerBiofilm (n = 18) | 0.05 | 3.04 | 8.12 | 12.21 | 48.57 | 10.04 | 4.10 to 15.98 |
| | **B)** | ViralRNA (n = 9) | 0.13 | 0.60 | 1.12 | 1.36 | 2.04 | 1.08 | 0.56 to 1.61 |
| | | NucleoSpin (n = 9) | 1.47E-04 | 6.00 | 9.06 | 14.48 | 23.93 | 10.43 | 4.29 to 16.57 |
| | **C)** | ViralRNA (n = 18) | 0.14 | 0.28 | 0.49 | 1.06 | 3.56 | 0.89 | 0.42 to 1.37 |
| | | NucleoMag (n = 18) | 3.00E-05 | 1.66 | 13.75 | 18.02 | 37.87 | 11.86 | 5.98 to 17.74 |

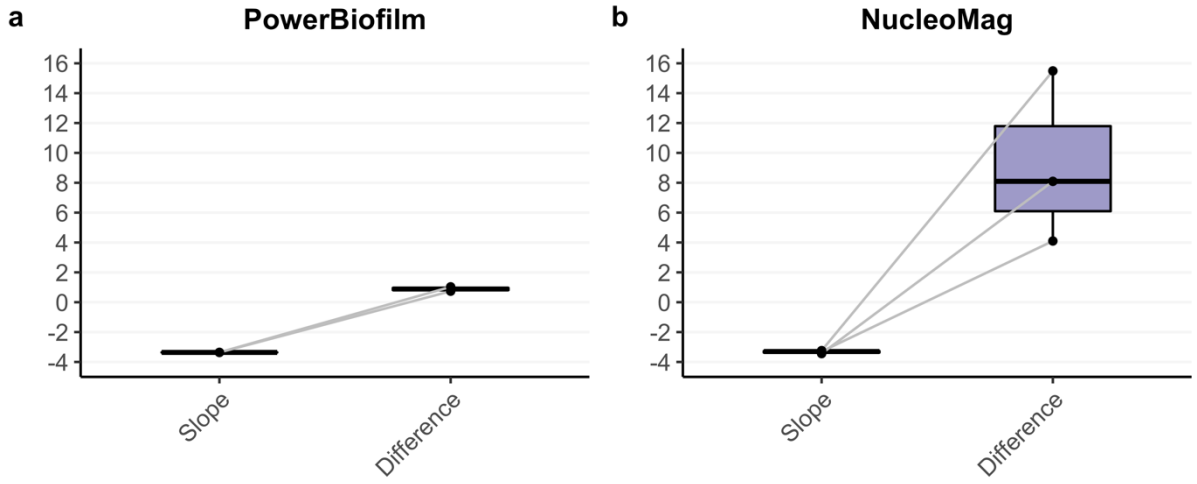



**Figure 2.5.7:** Differences between the slopes and undiluted vs diluted OC43 RT-qPCR Cq values of severely inhibited samples for the **a)** PowerBiofilm kit (n = 2) and **b)** the NucleoMag kit (n = 3). These samples all had earlier detection (lower Cq values) in the 1:10 diluted RT-qPCR reaction than the undiluted reaction, suggesting severe inhibition in the undiluted reactions.

**Table 2.5.5**: Two-by-two contingency tables for SARS-CoV-2 detection by RT-qPCR comparing the PowerBiofilm, NucleoSpin, and NucleoMag extraction kits to the ViralRNA extraction kit.

| | | | **Undiluted RT-qPCR** | | **1:10 Diluted RT-qPCR** | |
|---|---|---|---|---|---|---|
| | | | *Viral RNA SARS-CoV-2 Detection* | | *Viral RNA SARS-CoV-2 Detection* | |
| | | | No | Yes | No | Yes |
| **SARS-CoV-2 Detection** | *PowerBiofilm* | No | 4 | 3 | 0 | 0 |
| | | Yes | 6 | 5 | 6 | 12 |
| | *NucleoSpin* | No | 0 | 1 | 0 | 1 |
| | | Yes | 1 | 7 | 0 | 8 |
| | *NucleoMag* | No | 0 | 12 | 0 | 0 |
| | | Yes | 1 | 5 | 1 | 17 |

## 2.5.4 Second Methods Comparison Supplemental Information

### 2.5.4.1 Tables and Figures

**Table 2.5.6**: Sample dates and WWTP sampled for each of the methods tested in the second methods comparison.

| WWTP | Sample Date | Skimmed Milk Direct | Membrane Filtration | Nanotrap ® w/ NucleoMag | Nanotrap ® w/ Viral RNA |
|---|---|---|---|---|---|
| *C* | 5/17/21 | Y | - | Y | - |
| *A, B* | 5/18/21 | Y | - | Y | - |
| *C* | 5/24/21 | Y | Y | Y | - |
| *A, B* | 5/25/21 | Y | Y | Y | - |
| *C* | 5/31/21 | Y | Y | Y | - |
| *A, B* | 6/1/21 | Y | Y | Y | - |
| *C* | 6/12/21 | Y | Y | Y | - |
| *A, B* | 6/13/21 | Y | Y | Y | - |
| *C* | 6/21/21 | Y | Y | Y | - |
| *A, B* | 6/22/21 | Y | Y | Y | - |
| *C* | 6/27/21 | Y | Y | Y | - |
| *A, B* | 6/28/21 | Y | Y | Y | - |
| *A* | 7/20/21 | Y | - | - | Y |
| *C* | 8/23/21 | Y | - | - | Y |
| *A, B* | 8/24/21 | Y | - | - | Y |
| *C* | 8/30/21 | Y | - | - | Y |
| *A, B* | 8/31/21 | Y | - | - | Y |
| *A, B* | 9/7/21 | Y | - | - | Y |
| *C* | 9/8/21 | Y | - | - | Y |
| *C* | 9/13/21 | Y | - | - | Y |
| *A, B* | 9/14/21 | Y | - | - | Y |

**Table 2.5.7**: Descriptive statistics for OC43 method recovery for the second methods comparison.

| | **Comparisons** | | | | | |
|---|---|---|---|---|---|---|
| | *Skimmed Milk* | *Membrane Filtration* | *Skimmed Milk* | *Nanotrap + Nucleo Mag* | *Skimmed Milk* | *Nanotrap + ViralRNA* |
| Min. | 0.009 | NA | 0.009 | 6.50 | 0.21 | 0.0003 |
| Med. | 0.97 | NA | 0.97 | 24.52 | 0.51 | 3.00 |
| Max. | 9.78 | NA | 9.78 | 96.56 | 3.56 | 17.73 |
| 95% CI | 0.48 – 3.88 | NA | 0.60 – 3.40 | 16.57 – 39.61 | 0.42 – 1.46 | 2.33 – 8.64 |
| **Mean** | **2.18** | **NA** | **2.00** | **28.09** | **0.94** | **5.49** |

**Table 2.5.7**: Two-by-two contingency tables for SARS-CoV-2 detection between methods.

| | | | Undiluted RT-qPCR *Skimmed Milk Direct SARS-CoV-2 Detection* | | 1:10 Diluted RT-qPCR *Skimmed Milk Direct SARS-CoV-2 Detection* | |
|---|---|---|---|---|---|---|
| | | | No | Yes | No | Yes |
| **SARS-CoV-2 Detection** | *Membrane Filtration* | No | 8 | 7 | 2 | 11 |
| | | Yes | 0 | 0 | 1 | 1 |
| | *Nanotrap + NucleoMag* | No | 1 | 1 | 0 | 0 |
| | | Yes | 9 | 7 | 3 | 15 |
| | *Nanotrap + ViralRNA* | No | 0 | 12 | 0 | 0 |
| | | Yes | 0 | 3 | 0 | 15 |

### 2.5.4.2 Complete Protocols

#### *2.5.4.2.1 Membrane Filtration*

**Preparation – Day 1**

1. Collect influent wastewater in 1-L polypropylene bottles. Store at 4-8°C for up to 4 days.
2. Autoclave/ensure available:
    - 4 sets of tubing for peristaltic pump
    - 4 10L carboys
    - 1L polypropylene bottles
    - Membrane filtration cups with bases
        i. At least 2/sample
    - Electronegative membrane filters with 0.45um pore size (Millipore Cat. No. HAWG047S6)
        i. At least 2/sample
    - Vesphene
3. Set up membrane filtration units
4. Prepare the following reagents:
    - 5M HCl
    - 1M $MgCl_2$
5. Ensure other supplies available:
    - Serological pipettes
    - pH strips
    - Chlorine strips

- Labels for 1.5mL tubes

**Day 2 – Seeding HCoV-OC43**

1. Thaw viral stock on ice until completely thawed (approximately 30 minutes)
2. Combine WW from a single treatment plant in a 10L carboy using a graduated cylinder or peristaltic pump
3. Measure pH and chlorine content of WW using pH and chlorine strips
4. Pump wastewater sample using a peristaltic pump into graduated cylinders for each method
    - Record volume for each WWTP
    - Connect tubing to the 10-L PP sample carboy cap outlet, thread thru the peristaltic pump, and hold tubing outlet over graduated cylinder
    - Place the 10-L PP wastewater carboy in a large autoclave tray as secondary containment
    - Place the peristaltic pump on top of an upside-down autoclave tray to raise the height
    - Place graduated cylinder in a small autoclave tray as secondary containment
5. Place water sample on cart and mix by gently moving back and forth
6. Transfer ~30 mL of mixed wastewater into a 50-mL conical
7. Add ~$10^6$ TCID50 OC43 stock to the 50-mL conical and vortex 30 seconds (3.4uL per 100mL).
    - Add 3.4uL of $10^6$ TCID50 OC43 stock to the 50-mL conical and vortex 30 seconds, transfer seeded solution back to WW and mix thoroughly
8. Volumes:
    - HA filtration: pump 0.1L of seeded WW sample using a peristaltic pump into a 0.5L graduated cylinder and transfer 0.1L to a 0.25L bottle

**HA Filtration**

1. Add $MgCl_2$ to a final concentration of 25mM
    a. Add 2.5mL of a 1M $MgCl_2$ solution per 0.1L of WW for a 25mM concentration
2. Adjust pH of WW to 3.0-4.0 using 5M HCl and pH strips

3. Place 0.45um filter disc on the filtration unit using sterile tweezers (place in 200 proof EtOH and flame) and attach filtration unit cup
4. Pour 50mL of sample into filtration cup and turn on the vacuum to allow the sample to filter for up to 60 minutes
    - Record time of sample addition
    - If full volume passes through filter, add additional sample
5. After 60 minutes have passed, change the filter disc using sterile tweezers and place in Whirl-Pak bag
6. Repeat using more filters until whole volume has been filtered
7. Discard remaining sample
8. Disinfect the membrane filtration apparatus after use
9. Clean-up
    - Remove all cups and plastic inserts, close all switches for the three units
    - Wipe the outside with Vesphene (10-minute contact time) (or Oxivir?), followed by a rise with 70% EtOH
    - Starting with the unit farthest from the vacuum, fill the apparatus unit with DI water, then open the switch to flush the water through
    - Repeat DI rinse with other two units
    - Remove all tubing and add parafilm to the open end
    - Close all three switches and fill each unit with Vesphene
    - Open the switches to all three units so Vesphene fills main base
    - Close the switches to all three units, let stand for 30 minutes or O/N
    - After disinfection, remove parafilm and discard Vesphene by pouring into the sink
    - Close the switches and fill each apparatus to the top with 70% EtOH
    - Open the switches and flush out remaining Vesphene with EtOH, discard EtOH by pouring into the sink

**RNA extraction**

1. Add membrane to a bead tube
2. Add solution to the extraction kit

#### *2.5.4.2.2 CeresNano Magnetic Nanotrap Beads*

**Preparation – Day 1**

1. Collect influent wastewater in 1L polypropylene bottles. Store at 4-8°C for up to 4 days.
2. Autoclave/ensure available:
    - 3 sets of tubing for peristaltic pump
    - 3 10L carboys
    - 500mL polypropylene bottles
    - 100mL graduated cylinders
    - Magnetic racks (50mL, 1.5mL, 96-well plate)
    - 50mL conicals
    - Macherey-Nagel NucleoMag Kit
    - CeresNano Nanotrap beads
3. Ensure other supplies available:
    a. Serological pipettes
    b. pH strips
    c. Chlorine strips
    d. Labels for 1.5mL tubes

**Day 2 – Seeding HCoV-OC43 and Extraction**

1. Thaw viral stock on ice until completely thawed (approximately 30 minutes)
2. Combine WW from a single treatment plant into a 10L carboy
3. Measure pH and chlorine content of WW using pH and chlorine strips
4. Pump wastewater using a peristaltic pump into graduated cylinders and pour into a 100mL bottle
    - 100mL for Nanotrap Method
5. Transfer ~30mL of mixed wastewater into a 50mL conical
6. Add 3.4uL of $10^6$ TCID50 OC43 stock to the 50mL conical and vortex for 30 seconds, transfer seeded solution back to WW and mix thoroughly

7. Incubate spiked wastewater sample for a minimum of 10 minutes at room temperature to allow large aggregates to sediment at the bottom of the sample bottle
8. Using a serological pipette, transfer the top 40mL GENTLY to a new 50mL conical tube
    - Make sure to gently transfer from the top of the wastewater sample as to not disturb the sedimented particles at the bottom of the bottle
9. Add 600uL of Ceres Magnetic Nanotrap ® particles to the 40mL spiked wastewater aliquot
10. Invert the wastewater samples several times to incorporate the Magnetic Nanotrap ® particles for a minimum of 20 minutes at RT
    - Can shake if needed
11. Place conical tubes into magnetic racks and allow magnets to attract Magnetic Nanotrap particles for a minimum of 20 minutes at RT
    - The samples should be allowed to sit on the magnets long enough so that the supernatant is relatively clear. Some wastewater samples may be too turbid to reach this point
    - Red rust colored Magnetic Nanotrap particles have precipitated at the bottom near the magnets
12. Keeping the conicals fixed to the magnets, remove the supernatant using serological pipettes
    - Be careful not to disturb the pellet
13. Add 500uL of Buffer MWA1
    - Vortex to resuspend the Magnetic pellets in Buffer MWA1
14. Incubate samples at RT for 10 minutes
15. Place conical tubes on the custom magnet racks to separate the Magnetic Nanotrap particles
    - With the tubes on the magnets, transfer 450uL of lysate to a 1.5mL tubes
16. Add 475uL of Buffer MWA2 and 25uL of NucleoMag B-Beads to the lysate
    - Shake the samples at 1400RPM, 56°C for 5 minutes

17. After shaking, place the sample tubes on a magnet for at least 5 minutes to separate the NucleoMag B-Beads
    - Allow as much time as needed for the beads to completely precipitate
18. Remove the supernatant from each tube
    - If the pellet is disturbed, return to the 1.5mL tube and allow the beads to re-precipitate
19. Add 850uL of Buffer MWA3 to each tube
    - Shake the samples at 1400rpm, 56°C for 2 minutes
20. After shaking, place the sample tubes on a magnet for at least 2 minutes to separate the NucleoMag B-Beads
    - Allow as much time as needed for the beads to completely precipitate
21. Remove the supernatant from each tube
    - If the pellet is disturbed, return to the 1.5mL tube and allow the beads to re-precipitate
22. Add 850uL of Buffer MWA4 to each tube
    - Shake the samples at 1400rpm, 56°C for 2 minutes
23. After shaking, place the sample tubes on a magnet for at least 2 minutes to separate the NucleoMag B-Beads
    - Allow as much time as needed for the beads to completely precipitate
24. Remove the supernatant from each tube
    - Take extra care to remove as much supernatant as possible at this step. Carryover of the was buffer can cause molecular methods to fail
25. Place plate on shaker at 56°C for ~30 minutes to allow the beads to air dry
    - Ensure beads are thoroughly dry before proceeding
26. Add 60uL of RNase-free H2O to each tube
    - Shake the sample tubes at 500rpm, 56°C for 5 minutes
27. After shaking, place the sample tubes on a magnet for at least 2 minutes to separate the NuceloMag B-Beads
28. Transfer eluted RNA to a clean 1.5mL tube for downstream molecular processes. Samples can either be processed or frozen at -20°C until future use.

### 2.5.5 SARS-CoV-2 R Code

````
---
title: "OC43 Data"
author: "Sarah Philo"
date: "6/17/2020"
geometry: margin = 1.75cm
output: pdf_document
editor_options:
  chunk_output_type: console
---

##Set-up
```{r, tidy=TRUE}
knitr::opts_chunk$set(echo = TRUE)

setwd("~/OneDrive/Documents/UW/EOHML/CoV/Methods Comp
1/Figures")
```

##BMFS Graphing
```{r, tidy=TRUE}
library(readr)
Ct <- read_csv("Ct.csv")
mean <- mean(Ct$BMFS, na.rm=TRUE)
stdev <- sd(Ct$BMFS, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
bmfs <- ggplot(data=Ct, aes(x=Week, y=BMFS)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
````
````

```
  geom_point(aes(x=Week, y=BMFS, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(15, 40)+
  ggtitle("BMFS")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))
bmfs

ggsave("BMFScontrol.png", plot=bmfs, device="png", dpi=300,
limitsize=TRUE)

descriptivestats <- matrix(ncol=6, nrow=6)
colnames(descriptivestats) <- c("Average", "Standard Deviation",
"Variance", "Max", "Min", "Non-Detection")
rownames(descriptivestats) <- c("BMFS","BMFS - Vertrel",
"Skimmed Milk", "Skimmed Milk - Vertrel", "Ultrafiltration",
"PEG")
descriptivestats[1,1:6] <- c(mean, stdev, var(Ct$BMFS,
na.rm=TRUE), max(Ct$BMFS, na.rm=TRUE), min(Ct$BMFS, na.rm=TRUE),
sum(is.na(Ct$BMFS)))
```

## BMFS Vertrel Graphing

```{r, tidy=TRUE}
mean <- mean(Ct$BMFS_V, na.rm=TRUE)
stdev <- sd(Ct$BMFS_V, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
```

```
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
bmfs_v <- ggplot(data=Ct, aes(x=Week, y=Ct)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(x=Week, y=BMFS_V, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(15, 40)+
  xlim(16, 20)+
  ggtitle("BMFS - Vertrel")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))
bmfs_v

descriptivestats[2,1:6] <- c(mean, stdev, var(Ct$BMFS_V,
na.rm=TRUE), max(Ct$BMFS_V, na.rm=TRUE), min(Ct$BMFS_V,
na.rm=TRUE), sum(is.na(Ct$BMFS_V[Ct$Week>16&Ct$Week<20])))

ggsave("BMFSvertrelcontrol.png", plot=bmfs_v, device="png",
dpi=300, limitsize=TRUE)
```

## Skimmed milk Graphing

```{r, tidy=TRUE}
mean <- mean(Ct$SM, na.rm=TRUE)
stdev <- sd(Ct$SM, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
sm <- ggplot(data=Ct, aes(x=Week, y=SM)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(x=Week, y=SM, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(15, 40)+
  ggtitle("Skimmed Milk")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))
sm
```

```
ggsave("SMcontrol.png", plot=sm, device="png", dpi=300,
limitsize=TRUE)

descriptivestats[3,1:6] <- c(mean, stdev, var(Ct$SM,
na.rm=TRUE), max(Ct$SM, na.rm=TRUE), min(Ct$SM, na.rm=TRUE),
sum(is.na(Ct$SM)))
```


## Skimmed Milk Vertrel Graphing
```{r, tidy=TRUE}
mean <- mean(Ct$SM_V, na.rm=TRUE)
stdev <- sd(Ct$SM_V, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
sm_v <- ggplot(data=Ct, aes(x=Week, y=SM_V)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(x=Week, y=SM_V, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(15, 40)+
```

```
  ggtitle("Skimmed Milk - Vertrel")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))
sm_v

ggsave("SMvertrelcontrol.png", plot=sm_v, device="png", dpi=300,
limitsize=TRUE)

descriptivestats[4,1:6] <- c(mean, stdev, var(Ct$SM_V,
na.rm=TRUE), max(Ct$SM_V, na.rm=TRUE), min(Ct$SM_V, na.rm=TRUE),
sum(is.na(Ct$SM_V)))
```

```
##PEG Graphing
```{r, tidy=TRUE}
mean <- mean(Ct$PEG, na.rm=TRUE)
stdev <- sd(Ct$PEG, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
peg <- ggplot(data=Ct, aes(x=Week, y=PEG)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(x=Week, y=PEG, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
```
```

```
                      labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                      name = "Wastewater Treatment Plant",
                      labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(15, 40)+
  xlim(19, 25)+
  ggtitle("PEG")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))
peg

descriptivestats[6,1:6] <- c(mean, stdev, var(Ct$PEG,
na.rm=TRUE), max(Ct$PEG, na.rm=TRUE), min(Ct$PEG, na.rm=TRUE),
sum(is.na(Ct$PEG[Ct$WWTP=="westpoint"&Ct$Week>19&Ct$Week<24])))

ggsave("PEGcontrol.png", plot=peg, device="png", dpi=300,
limitsize=TRUE)
```

## UF Graphing

```{r, tidy=TRUE}
mean <- mean(Ct$UF, na.rm=TRUE)
stdev <- sd(Ct$UF, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
uf <- ggplot(data=Ct, aes(x=Week, y=UF)) +
 geom_hline(data = shewartlines,
            aes(yintercept = y, linetype = lines),
            size = 1)+
```

```
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(x=Week, y=UF, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(15, 40)+
  xlim(19, 28)+
  ggtitle("Ultrafiltration")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))
uf

ggsave("UFcontrol.png", plot=uf, device="png", dpi=300,
limitsize=TRUE)

descriptivestats[5,1:6] <- c(mean, stdev, var(Ct$UF,
na.rm=TRUE), max(Ct$UF, na.rm=TRUE), min(Ct$UF, na.rm=TRUE),
sum(is.na(Ct$UF[Ct$Week>19&Ct$Week<28])))
View(descriptivestats)

write.csv(descriptivestats, "DescriptiveStats.csv")
```

##UF Pellet Graphing
```{r, tidy=TRUE}
mean <- mean(Ct$UF_Pellet, na.rm=TRUE)
stdev <- sd(Ct$UF_Pellet, na.rm=TRUE)
lcl <- mean-3*stdev
ucl <- mean+3*stdev
```

```
lwl <- mean-stdev
uwl <- mean+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, mean, mean, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
ggplot(data=Ct, aes(x=Week, y=UF_Pellet)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(x=Week, y=UF_Pellet, shape=WWTP, color=WWTP),
size=2.5)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(10, 40)+
  xlim(20, 25)+
  ggtitle("Ultrafiltration Pellet Control Chart")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9))

ggsave("UFPelletcontrol.png", plot=last_plot(), device="png",
dpi=300, limitsize=TRUE)
```

## Control Chart Figure

```{r, tidy = True}
library(ggplot2)
library("ggpubr")

control_charts <- ggarrange(bmfs, bmfs_v, sm, sm_v, uf, peg,
                            ncol = 2,
                            nrow = 3,
                            common.legend = TRUE,
                            legend = "right",
                            labels = "AUTO")
control_charts
annotate_control <- annotate_figure(control_charts,
                                    top = text_grob("Control
Charts", face = "bold", size = 24))
annotate_control
ggsave(annotate_control, filename = "control_charts.tiff",
       units = 'in', width = 9, height = 9, dpi = 600)
```


##Effective Volume Assayed Graphs
```{r, tidy=TRUE}
Volume_Recovery <- read_csv("Volume Assayed and Recovery.csv")

volumestats <- matrix(ncol=6, nrow=6)
colnames(volumestats) <- c("Minimum", "1st Quartile", "Median",
"Mean", "3rd Quartile", "Maximum")
rownames(volumestats) <- c("BMFS", "BMFS - Vertrel", "Skimmed
Milk", "Skimmed Milk - Vertrel", "Ultrafiltration", "PEG")

volumestats[1,1:6] <-
summary(Volume_Recovery$EFFECTIVE_VOL[Volume_Recovery$METHOD=="B
MFS"])[1:6]
volumestats[2,1:6] <-
summary(Volume_Recovery$EFFECTIVE_VOL[Volume_Recovery$METHOD=="B
MFS Vertrel"])
volumestats[3,1:6] <-
summary(Volume_Recovery$EFFECTIVE_VOL[Volume_Recovery$METHOD=="S
kimmed Milk"])
volumestats[4,1:6] <-
summary(Volume_Recovery$EFFECTIVE_VOL[Volume_Recovery$METHOD=="S
kimmed Milk Vertrel"])
volumestats[5,1:6] <-
summary(Volume_Recovery$EFFECTIVE_VOL[Volume_Recovery$METHOD=="U
ltrafiltration"])
volumestats[6,1:6] <-
summary(Volume_Recovery$EFFECTIVE_VOL[Volume_Recovery$METHOD=="P
EG"])
```

```
write.csv(volumestats, "VolumeStats.csv")

library(RColorBrewer)
display.brewer.all()

library(ggplot2)
volumeassayed <- ggplot(data=Volume_Recovery,
aes(METHOD,EFFECTIVE_VOL, fill=METHOD))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank())+
  ggtitle("Effective Volume Assayed")+
  labs(y="Volume (mL)")+
  scale_fill_brewer(palette="RdBu",
                    name = "Method")
volumeassayed

ggsave("VolumeAssayed.png", plot=volumeassayed, device="png",
dpi=300, limitsize=TRUE)
```

## Percent Recovery
```{r, tidy=TRUE}
recoverystats <- matrix(ncol=7, nrow=6)
colnames(recoverystats) <- c("Minimum", "1st Quartile",
"Median", "Mean", "3rd Quartile", "Maximum", "Non-Detect")
rownames(recoverystats) <- c("BMFS", "BMFS - Vertrel", "Skimmed
Milk", "Skimmed Milk - Vertrel", "Ultrafiltration", "PEG")

recoverystats[1,1:7] <-
summary(Volume_Recovery$RECOVERY[Volume_Recovery$METHOD=="BMFS"]
)
recoverystats[2,1:6] <-
summary(Volume_Recovery$RECOVERY[Volume_Recovery$METHOD=="BMFS
Vertrel"])
recoverystats[3,1:7] <-
summary(Volume_Recovery$RECOVERY[Volume_Recovery$METHOD=="Skimme
d Milk"])
recoverystats[4,1:7] <-
summary(Volume_Recovery$RECOVERY[Volume_Recovery$METHOD=="Skimme
d Milk Vertrel"])
```

```
recoverystats[5,1:7] <-
summary(Volume_Recovery$RECOVERY[Volume_Recovery$METHOD=="Ultraf
iltration"])
recoverystats[6,1:6] <-
summary(Volume_Recovery$RECOVERY[Volume_Recovery$METHOD=="PEG"])
write.csv(recoverystats, "RecoveryStats.csv")

library(ggplot2)
percentrecovery <- ggplot(data=Volume_Recovery,
                          aes(METHOD, y = RECOVERY,
fill=METHOD))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12))+
  ggtitle("OC43 Recovery")+
  labs(y="Recovery")+
  scale_fill_brewer(palette="RdBu",
                    name = "Method")+
  scale_y_continuous(trans = "log10")
percentrecovery

ggsave("PercentRecovery.png", plot=percentrecovery,
device="png", dpi=300, limitsize=TRUE)
```

```
## Volume assayed and Recovery Figure
```{r, tidy = True}
library(ggplot2)
library("ggpubr")

vol_percent <- ggarrange(volumeassayed, percentrecovery,
                         ncol = 2,
                         nrow = 1,
                         common.legend = TRUE,
                         legend = "none",
                         labels = "AUTO")
vol_percent
ggsave(vol_percent, filename = "vol_percent.tiff",
       units = 'in', width = 9, height = 5, dpi = 600)
```

## Resuspension Volumes
```

```{r, tidy=TRUE}
resuspensionvol <- matrix(ncol=6, nrow=6)
colnames(resuspensionvol) <- c("Minimum", "1st Quartile",
"Median", "Mean", "3rd Quartile", "Maximum")
rownames(resuspensionvol) <- c("BMFS", "BMFS - Vertrel",
"Skimmed Milk", "Skimmed Milk - Vertrel", "Ultrafiltration",
"PEG")

resuspensionvol[1,1:6] <-

summary(Volume_Recovery$RESUSP_VOL[Volume_Recovery$METHOD=="BMFS
"])
resuspensionvol[2,1:6] <-

summary(Volume_Recovery$RESUSP_VOL[Volume_Recovery$METHOD=="BMFS
Vertrel"])
resuspensionvol[3,1:6] <-

summary(Volume_Recovery$RESUSP_VOL[Volume_Recovery$METHOD=="Skim
med Milk"])
resuspensionvol[4,1:6] <-

summary(Volume_Recovery$RESUSP_VOL[Volume_Recovery$METHOD=="Skim
med Milk Vertrel"])
resuspensionvol[5,1:6] <-

summary(Volume_Recovery$RESUSP_VOL[Volume_Recovery$METHOD=="Ultr
afiltration"])
resuspensionvol[6,1:6] <-

summary(Volume_Recovery$RESUSP_VOL[Volume_Recovery$METHOD=="PEG"
])

write.csv(resuspensionvol, "ResuspensionVolume.csv")

library(ggplot2)
ggplot(data=Volume_Recovery, aes(METHOD,RESUSP_VOL,
fill=METHOD))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=18),
        axis.title = element_text(size=14),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1),
        axis.title.x = element_blank())+
  ggtitle("Resuspension Volume")+
```

```
  labs(y="Volume (mL)")+
  scale_fill_brewer(palette="Greys",
                    name = "Method")

ggsave("ResuspensionVolume.png", plot=last_plot(), device="png",
dpi=300, limitsize=TRUE)
```

## N1 Standard Curves
```{r, tidy=TRUE}
library(readr)
library(RColorBrewer)
sars_stdcurve_cq <- read_csv("SARS Standard Curves.csv")

N1_reg_all <- lm(data=sars_stdcurve_cq, N1~log_cov)
summary(N1_reg_all)
lbl_all <- paste("R^2 == ",
round(summary(N1_reg_all)$r.squared,4))

N1_reg_sub <- lm(data=subset(sars_stdcurve_cq, log_cov > "2"),
                 N1~log_cov)
summary(N1_reg_sub)
lbl_sub <- paste("R^2 == ",
round(summary(N1_reg_sub)$r.squared,4))

int <- c(N1_reg_all[["coefficients"]][["(Intercept)"]],
N1_reg_sub[["coefficients"]][["(Intercept)"]])
slope <- c(N1_reg_all[["coefficients"]][["log_cov"]],
N1_reg_sub[["coefficients"]][["log_cov"]])
names <- c("all", "sub")
reglines <- data.frame(names, int, slope)
palette <- c("#053061", "#b2182b")

library(ggpmisc)
n1reg <- ggplot(data=sars_stdcurve_cq, aes(x=log_cov,y=N1))+
  theme_classic()+
  geom_point(data=subset(sars_stdcurve_cq, log_cov %in% c(5, 4,
3)),
             aes(x=log_cov, y=N1),
             alpha=0.3,
             color="#b2182b",
             shape = 19,
             size=2)+
  geom_point(data=subset(sars_stdcurve_cq, log_cov %in% c(2, 1,
0, -1)),
             aes(x=log_cov, y=N1),
             alpha=0.3,
             color="#053061",
```

```
            shape = 19,
            size=2)+
  labs(title = "N1 Standard Curve Regressions",
       y = "Cq",
       x = "Log(CoV)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=24),
        axis.title=element_text(size=16, face = "bold"),
        legend.text=element_text(size=14),
        axis.text=element_text(size=14),
        legend.position = "bottom")+
  scale_x_continuous(breaks = c(6, 5, 4, 3, 2, 1, 0, -1, -2))+
  ylim(20, 45)+
  annotate("text", x = 0.5, y=32,
           label="y = -2.55x + 38.49",
           color="#053061")+
  annotate("text", x = 0.5, y = 29,
           label = lbl_all,
           color = "#053061",
           parse = TRUE)+
  annotate("text", x=4, y=38,
           label="y=-3.06x + 40.54",
           color="#b2182b")+
  annotate("text", x = 4, y = 35,
           label = lbl_sub,
           color="#b2182b",
           parse = TRUE)+
  geom_abline(data = reglines, aes(intercept = int, slope =
slope, color = names),
              size = 0.5)+
  scale_color_manual(aesthetics = "color",
                     values = c(rev(brewer.pal(9,
"RdBu"))[1],rev(brewer.pal(9,"RdBu"))[9]),
                     labels = c("All Concentrations", "Subset of
Concentrations"),
                     name = "")+
  geom_smooth(data=subset(sars_stdcurve_cq, log_cov %in% c(5, 4,
3)),
              method = "lm",
              color = "#b2182b",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)+
  geom_smooth(method = "lm",
              color = "#053061",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)
```

```
n1reg
ggsave("N1reg.png", plot=n1reg, device="png", dpi=300,
limitsize=TRUE)

stds_lines <- read_csv("Standard Curve Lines.csv")

n1lines <- ggplot(data = sars_stdcurve_cq, aes(x=log_cov,
y=N1))+
  theme_classic()+
  geom_point(aes(x=log_cov, y=N1),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "N1 Standard Curves",
       y = "Cq",
       x = "Log(CoV)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  scale_x_continuous(breaks = c(5, 4, 3, 2, 1, 0, -1))+
  ylim(20, 45)+
  geom_abline(data=stds_lines,
              aes(slope=n1_slope, intercept=n1_int),
              alpha=0.4,
              color = "#053061")+
  geom_abline(data=stds_lines,
              aes(slope=n1_lin_slope, intercept=n1_lin_int),
              alpha=0.4,
              color="#b2182b")
n1lines

ggsave("N1lines.png", plot=n1lines, device="png", dpi=300,
limitsize=TRUE)
```


## N2 Standard Curves
```{r, tidy=TRUE}
N2_reg_all <- lm(data=sars_stdcurve_cq, N2~log_cov)
summary(N2_reg_all)
lbl_all <- paste("R^2 ==",
round(summary(N2_reg_all)$r.squared,4))

N2_reg_sub <- lm(data=subset(sars_stdcurve_cq, log_cov > "2"),
                 N2~log_cov)
```

```
summary(N2_reg_sub)
lbl_sub <- paste("R^2 ==",
round(summary(N2_reg_sub)$r.squared,4))

int <- c(N2_reg_all[["coefficients"]]["(Intercept)"],
N2_reg_sub[["coefficients"]]["(Intercept)"])
slope <- c(N2_reg_all[["coefficients"]]["log_cov"],
N2_reg_sub[["coefficients"]]["log_cov"])
names <- c("all", "sub")
reglines <- data.frame(names, int, slope)
palette <- c("#053061", "#b2182b")


library(ggpmisc)
n2reg <- ggplot(data=sars_stdcurve_cq, aes(x=log_cov,y=N2))+
  theme_classic()+
  geom_point(data=subset(sars_stdcurve_cq, log_cov %in% c(5, 4,
3)),
             aes(x=log_cov, y=N2),
             alpha=0.3,
             color="#b2182b",
             shape = 19,
             size=2)+
  geom_point(data=subset(sars_stdcurve_cq, log_cov %in% c(2, 1,
0, -1)),
             aes(x=log_cov, y=N2),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "N2 Standard Curve Variability",
       y = "Cq",
       x = "Log(CoV)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10),
        legend.position = "bottom")+
  scale_x_continuous(breaks = c(5, 4, 3, 2, 1, 0, -1))+
  ylim(20, 45)+
  geom_smooth(data=subset(sars_stdcurve_cq, log_cov %in% c(5, 4,
3)),
              method = "lm",
              color = "#b2182b",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)+
```

```
  geom_smooth(method = "lm",
              color = "#053061",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)+
  annotate("text", x = 0, y=30,
           label="y = -2.62x + 38.23",
           color="#053061")+
  annotate("text", x = 0, y = 27,
           label = lbl_all,
           color = "#053061",
           parse = TRUE)+
  annotate("text", x=4, y=36,
           label="y=-3.04x + 39.90",
           color="#b2182b")+
  annotate("text", x = 4, y = 33,
           label = lbl_sub,
           color = "#b2182b",
           parse = TRUE)+
  geom_abline(data = reglines, aes(intercept = int, slope =
slope, color = names),
              size = 0.5)+
  scale_color_manual(aesthetics = "color",
                     values = c(rev(brewer.pal(9,
"RdBu"))[1],rev(brewer.pal(9,"RdBu"))[9]),
                     labels = c("All Concentrations", "Subset of
Concentrations"),
                     name = "")

n2reg
ggsave("N2reg.png", plot=n2reg, device="png", dpi=300,
limitsize=TRUE)

n2lines <- ggplot(data = sars_stdcurve_cq, aes(x=log_cov,
y=N2))+
  theme_classic()+
  geom_point(aes(x=log_cov, y=N2),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "N2 Standard Curves",
       y = "Cq",
       x = "Log(CoV)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
```

```
        axis.text=element_text(size=10))+
  scale_x_continuous(breaks = c(5, 4, 3, 2, 1, 0, -1))+
  ylim(20, 45)+
  geom_abline(data=stds_lines,
              aes(slope=n2_slope, intercept=n2_int),
              alpha=0.4,
              color = "#053061")+
  geom_abline(data=stds_lines,
              aes(slope=n2_lin_slope, intercept=n2_lin_int),
              alpha=0.4,
              color="#b2182b")
n2lines

ggsave("N2lines.png", plot=n2lines, device="png", dpi=300,
limitsize=TRUE)
```

## N3 Standard Curves
```{r, tidy=TRUE}
N3_reg_all <- lm(data=sars_stdcurve_cq, N3~log_cov)
summary(N3_reg_all)
lbl_all <- paste("R^2 == ",
round(summary(N3_reg_all)$r.squared,4))

N3_reg_sub <- lm(data=subset(sars_stdcurve_cq, log_cov > "2"),
                 N3~log_cov)
summary(N3_reg_sub)
lbl_sub <- paste("R^2 == ",
round(summary(N3_reg_sub)$r.squared,4))

int <- c(N3_reg_all[["coefficients"]][["(Intercept)"]],
N3_reg_sub[["coefficients"]][["(Intercept)"]])
slope <- c(N3_reg_all[["coefficients"]][["log_cov"]],
N3_reg_sub[["coefficients"]][["log_cov"]])
names <- c("all", "sub")
reglines <- data.frame(names, int, slope)
palette <- c("#053061", "#b2182b")

library(ggpmisc)
n3reg <- ggplot(data=sars_stdcurve_cq, aes(x=log_cov,y=N3))+
  theme_classic()+
  geom_point(data=subset(sars_stdcurve_cq, log_cov %in% c(5, 4,
3)),
             aes(x=log_cov, y=N3),
             alpha=0.3,
             color="#b2182b",
             shape = 19,
             size=2)+
```

```
  geom_point(data=subset(sars_stdcurve_cq, log_cov %in% c(2, 1,
0, -1)),
             aes(x=log_cov, y=N3),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "N3 Standard Curve Variability",
       y = "Cq",
       x = "Log(CoV)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10),
        legend.position = "bottom")+
  scale_x_continuous(breaks = c(5, 4, 3, 2, 1, 0, -1))+
  ylim(20, 45)+
  geom_smooth(data=subset(sars_stdcurve_cq, log_cov %in% c(5, 4,
3)),
              method = "lm",
              color = "#b2182b",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)+
  geom_smooth(method = "lm",
              color = "#053061",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)+
  annotate("text", x = 0, y=32,
           label="y = -2.57x + 38.50",
           color="#053061")+
  annotate("text", x = 0, y = 29,
           label = lbl_all,
           color = "#053061",
           parse = TRUE)+
  annotate("text", x=4, y=38,
           label="y=-3.02x + 40.28",
           color="#b2182b")+
  annotate("text", x = 4, y = 35,
           label = lbl_sub,
           color = "#b2182b",
           parse = TRUE)+
  geom_abline(data = reglines, aes(intercept = int, slope =
slope, color = names),
              size = 0.5)+
  scale_color_manual(aesthetics = "color",
```

```
                      values = c(rev(brewer.pal(9,
"RdBu"))[1],rev(brewer.pal(9,"RdBu"))[9]),
                      labels = c("All Concentrations", "Subset of
Concentrations"),
                      name = "")
n3reg
ggsave("N3reg.png", plot=n3reg, device="png", dpi=300,
limitsize=TRUE)

n3lines <- ggplot(data = sars_stdcurve_cq, aes(x=log_cov,
y=N3))+
  theme_classic()+
  geom_point(aes(x=log_cov, y=N3),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "N3 Standard Curves",
       y = "Cq",
       x = "Log(CoV)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  scale_x_continuous(breaks = c(5, 4, 3, 2, 1, 0, -1))+
  ylim(20, 45)+
  geom_abline(data=stds_lines,
              aes(slope=n3_slope, intercept=n3_int),
              alpha=0.4,
              color = "#053061")+
  geom_abline(data=stds_lines,
              aes(slope=n3_lin_slope, intercept=n3_lin_int),
              alpha=0.4,
              color="#b2182b")
n3lines

ggsave("N3lines.png", plot=n3lines, device="png", dpi=300,
limitsize=TRUE)
```

##SARS Stds Figure
```{r, tidy = TRUE}
library(ggplot2)
library("ggpubr")

sars_lines <- ggarrange(n1lines, n2lines, n3lines, n1reg, n2reg,
n3reg,
```

```
                          ncol = 3,
                          nrow = 2,
                          common.legend = TRUE,
                          legend = "bottom",
                          labels = "AUTO")
sars_lines
ggsave(sars_lines, filename = "sars_lines.tiff",
       units = 'in', width = 13, height = 6, dpi = 600)

```

```
## OC43 Standard Curves
```{r, tidy = TRUE}
library(readr)
oc43_stdcurve_cq <- read_csv("OC43 Standard Curves.csv")
View(oc43_stdcurve_cq)

OC43_reg_all <- lm(data=oc43_stdcurve_cq, OC43~log_cov)
summary(OC43_reg_all)
lbl_all <- paste("R^2 == ",
round(summary(OC43_reg_all)$r.squared,4))

library(ggpmisc)
oc43reg <- ggplot(data=oc43_stdcurve_cq, aes(x=log_cov,y=OC43))+
  theme_classic()+
  geom_point(aes(x=log_cov, y=OC43),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "OC43 Standard Curve Regression",
       y = "Cq",
       x = "Log(TCID50 OC43)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  scale_x_continuous(breaks = c(4, 3, 2, 1, 0, -1, -2, -3, -4, -
5, -6))+
  ylim(15, 43)+
  geom_smooth(method = "lm",
              show.legend = TRUE,
              color = "#053061",
              linetype = 1,
              size = 0.5,
              fullrange = TRUE)+
  annotate("text", x = 2, y=32,
```
```

```
           label="y = -3.14x + 24.80",
           color="#053061")+
  annotate("text", x = 2, y = 29,
           label = lbl_all,
           color = "#053061",
           parse = TRUE)
oc43reg
ggsave("OC43reg.png",plot=oc43reg, device="png", dpi=300,
limitsize=TRUE)

oc43lines <- ggplot(data = oc43_stdcurve_cq, aes(x=log_cov,
y=OC43))+
  theme_classic()+
  geom_point(aes(x=log_cov, y=OC43),
             alpha=0.3,
             color="#053061",
             shape = 19,
             size=2)+
  labs(title = "OC43 Standard Curves",
       y = "Cq",
       x = "Log(TCID50 OC43)")+
  theme(plot.title=element_text(face="bold", hjust=0.5,
size=14),
        axis.title=element_text(size=12),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  scale_x_continuous(breaks = c(4, 3, 2, 1, 0, -1, -2, -3, -4, -
5, -6))+
  ylim(15, 43)+
  geom_abline(data=stds_lines,
              aes(slope=oc43_slope, intercept=oc43_int),
              alpha=0.4,
              color = "#053061")
oc43lines

ggsave("OC43lines.png",plot=oc43lines, device="png", dpi=300,
limitsize=TRUE)

library(ggplot2)
library("ggpubr")

oc43_figure <- ggarrange(oc43lines, oc43reg,
                         ncol = 2,
                         nrow = 1,
                         common.legend = TRUE,
                         legend = "right",
                         labels = "AUTO")
oc43_figure
```

```
ggsave(oc43_figure, filename = "oc43_figure.tiff",
       units = 'in', width = 10, height = 5, dpi = 600)
```

#SARS Balloon Plots
```{r, tidy=TRUE}
library(readr)
assay_balloon <- read_csv("assay_balloon.csv")
View(assay_balloon)
assay_quant <- quantile(assay_balloon$PERCENT, c(0, 0.125, 0.25,
0.375, 0.5, 0.625, 0.75, 0.875, 1.0))

library(RColorBrewer)
display.brewer.all(colorblindFriendly = TRUE)
display.brewer.pal(n = 9, name = "Blues")
brewer.pal(n = 9, name = "Blues")

library(gridExtra)
methods <- c("", "BMFS", "BMFS Vertrel", "PEG",
"Sludge","Skimmed Milk", "Skimmed Milk - Vertrel",
             "Ultrafiltration")
samplesize <- c("Sample Size", 45, 9, 4, 21, 45, 45, 22)
n_table <- t(data.frame(methods, samplesize))
tt <- ttheme_default(colhead = list(fg_params = list(parse =
TRUE)), base_size = 7)
n_tbl <- tableGrob(n_table, rows=NULL, theme=tt)

pres_pos <- read_csv("pres_pos.csv")

library(ggplot2)
library(dplyr)
library(gt)
library(ggpubr)

quantile(pres_pos$Percent, probs = seq(0,1,0.125))
library(RColorBrewer)
display.brewer.pal(n = 8, name = 'Reds')
display.brewer.pal(n = 8, name = "RdBu")
rev(brewer.pal(n = 8, name = 'RdBu'))
brewer.pal(n = 8, name = "Reds")

pres_pos_test <- read_csv("pres_pos_test.csv")

heat_pos <- gt(data = pres_pos_test, rowname_col = "Dilution")
%>%
  tab_header(title = md("**Presumptive Positive for SARS-CoV-
2**")) %>%
  tab_spanner(label = "BMFS",


```

```
              columns = vars(Percent_BMFS, Size_BMFS)) %>%
  tab_spanner(label = "BMFS-V",
              columns = vars(Percent_BMFS_V, Size_BMFS_V)) %>%
  tab_spanner(label = "Skimmed Milk",
              columns = vars(Percent_SM, Size_SM)) %>%
  tab_spanner(label = "Skimmed Milk - V",
              columns = vars(Percent_SM_V, Size_SM_V)) %>%
  tab_spanner(label = "Sludge",
              columns = vars(Percent_SL, Size_SL)) %>%
  tab_spanner(label = "PEG",
              columns = vars(Percent_PEG, Size_PEG)) %>%
  tab_spanner(label = "Ultrafiltration",
              columns = vars(Percent_UF, Size_UF)) %>%
  cols_label(Percent_BMFS = "%",
             Size_BMFS = "n",
             Percent_BMFS_V = "%",
             Size_BMFS_V = "n",
             Percent_SM = "%",
             Size_SM = "n",
             Percent_SM_V = "%",
             Size_SM_V = "n",
             Percent_SL = "%",
             Size_SL = "n",
             Percent_PEG = "%",
             Size_PEG = "n",
             Percent_UF = "%",
             Size_UF = "n") %>%
  tab_style(style = cell_borders(sides = "right",
                                 color = "lightgray"),
            locations = cells_body(columns = vars(Size_BMFS,
Size_BMFS_V, Size_SM, Size_SM_V,
                                                  Size_SL,
Size_PEG))) %>%
  tab_style(style = cell_fill(color = "#FFF5F0"),
            locations = cells_body(columns = vars(Percent_PEG)))
%>%
  tab_style(style = cell_fill(color = "#FEE0D2"),
            locations = cells_body(columns = vars(Percent_UF),
                                   rows = "Undiluted")) %>%
  tab_style(style = cell_fill(color = "#FEE0D2"),
            locations = cells_body(columns = vars(Percent_BMFS,
Percent_BMFS_V),
                                   rows = "Undiluted")) %>%
  tab_style(style = cell_fill(color = "#FCBBA1"),
            locations = cells_body(columns = vars(Percent_BMFS,
Percent_BMFS_V),
                                   rows = "1:10 Dilution")) %>%
  tab_style(style = cell_fill(color ="#FC9272"),
```

```
            locations = cells_body(columns = vars(Percent_SM,
Percent_SM_V),
                                   rows = "Undiluted")) %>%
  tab_style(style = cell_fill(color = "#FB6A4A"),
            locations = cells_body(columns = vars(Percent_SM,
Percent_SM_V),
                                   rows = "1:10 Dilution")) %>%
  tab_style(style = cell_fill(color = "#EF3B2C"),
            locations = cells_body(columns = vars(Percent_UF),
                                   rows = "1:10 Dilution")) %>%
  tab_style(style = cell_fill(color = "#CB181D"),
            locations = cells_body(columns = vars(Percent_SL),
                                   rows = "Undiluted")) %>%
  tab_style(style = cell_fill(color = "#99000D"),
            locations = cells_body(columns = vars(Percent_SL),
                                   rows = "1:10 Dilution"))
heat_pos
webshot::install_phantomjs()
gtsave(heat_pos, filename = "heat_pos.png")

quantile(pres_pos$Percent, probs = seq(0,1,0.125))
library(RColorBrewer)
display.brewer.pal(n = 9, name = 'Blues')
brewer.pal(n = 9, name = 'Blues')

library(ggpubr)
assayballoon <- ggballoonplot(data = assay_balloon,
                              aes(x = ASSAY, y = METHOD),
                              fill = "PERCENT",
                              size = "SIZE")+
  scale_fill_gradient(low = "#FFFFFF",
                      high = "#252525",
                      name = "Percent Positive")+
  #scale_fill_gradientn(colors = my_cols,
                       #name = "Percent Positive")+
  geom_text(data = assay_balloon, nudge_x = 0.42, aes(label =
PERCENT))+
  ggtitle("Percent Positivity for SARS-CoV-2 by Assay")+
  guides()+
  theme(legend.position="bottom",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text.x = element_text(size = 14, color = "black",
face = "bold"),
        axis.text.y = element_text(size = 14, color = "black",
face = "bold"))+
```

```
  labs(size = guide_legend("Sample Size"))
assayballoon
ggsave("AssayBalloonGrey.png", plot = assayballoon,
device="png", dpi=300, limitsize=TRUE)

plant_balloon <- read_csv("plant_balloon.csv")
View(plant_balloon)
plant_quant <- quantile(plant_balloon$PERCENT, c(0, 0.125, 0.25,
0.375, 0.5, 0.625, 0.75, 0.875, 1.0))

methods <- c("", "BMFS", "BMFS Vertrel", "PEG",
"Sludge","Skimmed Milk",
             "Skimmed Milk - Vertrel", "Ultrafiltration")
wp_samplesize <- c("Plant C", 15, 3, 4, 6, 15, 15, 8)
sp_samplesize <- c("Plant B", 15, 3, 0, 7, 15, 15, 7)
bw_samplesize <-  c("Plant A", 15, 3, 0, 4, 15, 15, 7)
n_table <- t(data.frame(methods, bw_samplesize, sp_samplesize,
wp_samplesize))
tt <- ttheme_default(colhead = list(fg_params = list(parse =
TRUE)), base_size = 10)
n_tbl <- tableGrob(n_table, rows=NULL, theme=tt)

plantballoon <- ggballoonplot(data = plant_balloon,
              aes( x = PLANT, y = METHOD),
              fill = "PERCENT",
              size = "SIZE")+
  scale_fill_gradient(low = "#F7FBFF",
                      high = "#08519C",
                      name = "Percent Positive")+
  #scale_fill_gradientn(colors = my_cols)+
  geom_text(data = plant_balloon, nudge_x = 0.4, aes(label =
PERCENT))+
  ggtitle("Percent Positivity for SARS-CoV-2 by Treatment
Plant")+
  guides()+
  theme(legend.position="bottom",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text.x = element_text(size = 14, color = "black",
face = "bold"),
        axis.text.y = element_text(size = 14, color = "black",
face = "bold"))+
  labs(size = guide_legend("Sample Size"),
       fill = guide_legend("Percent Positive"))
plantballoon
```

```
# Balloon Plot Figures
```{r, tidy = TRUE}
library(ggplot2)
library("ggpubr")
library("png")

grid.arrange(plantballoon, n_tbl, nrow = 2,  as.table = TRUE,
heights = c(30, 10),
             bottom = "Sample Size", widths = 45)
g_plant <- arrangeGrob(plantballoon, n_tbl, nrow = 2,  as.table
= TRUE,
                       heights = c(30, 10), bottom = "Sample
Size", widths = 45)
ggsave("PlantBalloonBlue.png",g_plant, device="png", dpi=300,
limitsize=TRUE)
```

# N1 Cq Values
```{r, tidy = TRUE}
library(readxl)
ct_values <- read_excel("OC43 and nCoV Ct values Methods
Comparison copy.xlsx",
                        col_types = c("text", "numeric",
"numeric", "numeric", "numeric", "numeric",
                                      "numeric", "numeric",
"numeric",  "numeric", "numeric", "numeric",
                                      "numeric", "numeric",
"numeric", "numeric", "numeric", "numeric",
                                      "numeric", "numeric",
"text", "numeric", "numeric", "numeric",
                                      "numeric", "numeric",
"numeric", "numeric", "numeric", "numeric",
                                      "numeric", "numeric",
"numeric", "numeric", "numeric", "numeric",
                                      "numeric"))

method <- as.vector(ct_values$Method)
week <- as.vector(ct_values$`Week of Year`)
wwtp <- as.vector(ct_values$Sample)
N1 <- as.vector(ct_values$N1)
N2 <- as.vector(ct_values$N2)
N3 <- as.vector(ct_values$N3)
pres_pos <- as.vector(ct_values$`Presumptive Positive`)
data <- data.frame(method, week, wwtp, N1, N2, N3, pres_pos)

mean <- mean(data$N1[data$N1>1 & data$N1<40])
line <- c("mean")
```
```

```
meanline <- data.frame(line, x = c(-Inf, Inf), mean)

library(ggplot2)

N1_cq <- ggplot(data = data, aes(x = week, y = N1))+
  geom_point(aes(x = week, y = N1, color = wwtp, shape = wwtp),
size = 3)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant")+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant")+
  ggtitle("N1")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  geom_hline(data = meanline,
             aes(yintercept = mean, linetype = "line"),
             size = 1)+
  scale_linetype_manual(name = c(""),
                        values = 1,
                        labels = "Mean")+
  ylim(25,45)
N1_cq

ggsave("N1cq.png", plot=N1_cq, device="png", dpi=300,
limitsize=TRUE)

```

#N2 Cq Values
```{r, tidy = TRUE}

mean <- mean(data$N2[data$N2>1 & data$N2<40])
line <- c("mean")
meanline <- data.frame(line, x = c(-Inf, Inf), mean)

library(ggplot2)

N2_cq <- ggplot(data = data, aes(x = week, y = N2))+
  geom_point(aes(x = week, y = N2, color = wwtp, shape = wwtp),
size = 3)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant")+
```

```
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant")+
  ggtitle("N2")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  geom_hline(data = meanline,
             aes(yintercept = mean, linetype = "line"),
             size = 1)+
  scale_linetype_manual(name = c(""),
                        values = 1,
                        labels = "Mean")+
  ylim(25,45)
N2_cq

ggsave("N2cq.png", plot=N2_cq, device="png", dpi=300,
limitsize=TRUE)
```


#N3 Cq Values
```{r, tidy = TRUE}

mean <- mean(data$N3[data$N3>1 & data$N3<40])
line <- c("mean")
meanline <- data.frame(line, x = c(-Inf, Inf), mean)

library(ggplot2)

N3_cq <- ggplot(data = data, aes(x = week, y = N3))+
  geom_point(aes(x = week, y = N3, color = wwtp, shape = wwtp),
size = 3)+
  scale_shape_manual(values = c(16, 17, 15),
                     name = "Wastewater Treatment Plant")+
  scale_color_manual(values = c("#b2182b", "#92c5de",
"#053061"),
                     name = "Wastewater Treatment Plant")+
  ggtitle("N3")+
  labs(y="Cq Value", x="Week of the Year")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
```

```
        legend.text=element_text(size=12),
        axis.text=element_text(size=10))+
  geom_hline(data = meanline,
             aes(yintercept = mean, linetype = "line"),
             size = 1)+
  scale_linetype_manual(name = c(""),
                        values = 1,
                        labels = "Mean")+
  ylim(25,45)
N3_cq

ggsave("N3cq.png", plot=N3_cq, device="png", dpi=300,
limitsize=TRUE)
```


#SARS Assays Figure
```{r, tidy = TRUE}
library(ggplot2)
library("ggpubr")

sars_figure <- ggarrange(N1_cq, N2_cq, N3_cq,
                         ncol = 3,
                         nrow = 1,
                         common.legend = TRUE,
                         legend = "bottom",
                         labels = "AUTO")
sars_figure
annotate_sars_cq <- annotate_figure(sars_figure,
                                    top = text_grob("SARS-CoV-2
Cq Values", face = "bold", size = 24))
annotate_sars_cq
ggsave(annotate_sars_cq, filename = "sars_cq.tiff",
       units = 'in', width = 13, height = 4, dpi = 600)
```


#SARS Cq Box plots by Assay
```{r, tidy = TRUE}

box_data <- data.frame(assay = NA, cq = NA, method = NA)
box_data[1:603,2] <- c(data$N1, data$N2, data$N3)
box_data[1:201,1] <- "N1"
box_data[202:402,1] <- "N2"
box_data[403:603,1] <- "N3"
box_data$method <- data$method
```

```
sars_cq <- ggplot(data = box_data, aes(x = assay, y = cq, fill =
assay))+
  theme_classic()+
  geom_boxplot(data = subset(box_data, cq > 0))+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=18),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12))+
  labs(y="Cq", x = "SARS-CoV-2 Assay")+
  #scale_fill_brewer(palette="RdBu")+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "Assay")
sars_cq
ggsave("SARS cq boxplot.tiff", plot=sars_cq, device="tiff",
dpi=300, limitsize=TRUE)
```


#SARS Cq Box plots by Method
```{r, tidy = TRUE}

method_box <- ggplot(data = box_data, aes(x = method, y = cq,
fill = method))+
  theme_classic()+
  geom_boxplot(data = subset(box_data, cq > 0))+
  scale_fill_brewer(palette="RdBu",
                    guide = FALSE)+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=18),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12))+
  labs(y="Cq", x = "Method")+
  scale_x_discrete(labels = c("BMFS", "BMFS - Vertrel", "PEG",
"Sludge", "Skimmed Milk",
                              "Skimmed Milk - Vertrel",
"Ultrafiltration"))
method_box
ggsave("SARS cq boxplot by method.tiff", plot = method_box,
device = "tiff", dpi = 300, limitsize = TRUE)
```

````
---
title: "SARS Data Updated"
author: "Sarah Philo"
date: "7/8/2021"
geometry: margin = 1.75cm
output: pdf_document
editor_options:
  chunk_output_type: console
---

```{r, tidy = TRUE}
knitr::opts_chunk$set(echo = TRUE)

setwd("~/OneDrive/Documents/UW/EOHML/CoV/Ongoing
Surveillance/Figures v2")
```


##Load and Manipulate Data
```{r, tidy = TRUE}
library(readr)

ww_data <- read_csv("SARS2 in WW.csv",
                    col_types = cols(exp_date = col_date(format
= "%Y-%m-%d"),
                                     sample_date =
col_date(format = "%Y-%m-%d")))

ww_data$effective_vol <- ww_data$method_vol*
  (ww_data$ext_vol/ww_data$final_vol)*
  (ww_data$pcr_vol/ww_data$elu_vol)

ww_data$stock_conc_construct <- 10^((ww_data$seed_cq -
ww_data$intercept)/
                                     ww_data$slope)

ww_data$stock <- ((ww_data$stock_conc*ww_data$ext_vol)/
                    ww_data$elu_vol)*(5/1000)

ww_data$oc43_conc <- (ww_data$stock_conc*ww_data$seed_vol)/
  ww_data$comp_vol

ww_data$est_oc43_assayed <- NA

ww_data$act_oc43_assayed <- 10^((ww_data$oc43 -
ww_data$intercept)/
                                  ww_data$slope)
````
````

```
ww_data$uniqueID <- c(1:length(ww_data$record_id))
uniqueID <- ww_data$uniqueID

for(i in uniqueID) {
  if(is.na(ww_data$std_positive_control[ww_data$uniqueID ==
i])){
    } else {
       if(ww_data$std_positive_control[ww_data$uniqueID == i] ==
1){
         ww_data$est_oc43_assayed[ww_data$uniqueID == i] <-
           (ww_data$oc43_conc[ww_data$uniqueID == i]*
              ww_data$effective_vol[ww_data$uniqueID == i])/1000
    ##something is happening here, don't worry about it tonight
  } else {
    ww_data$est_oc43_assayed[ww_data$uniqueID == i] <-
      (ww_data$stock_conc_construct[ww_data$uniqueID ==
i]/11.67)*
      (ww_data$seed_vol[ww_data$uniqueID == i]/
         ww_data$comp_vol[ww_data$uniqueID == i])*
      (ww_data$effective_vol[ww_data$uniqueID == i])
    }
  }
  }

ww_data$percent_recovery <-
  (ww_data$act_oc43_assayed/ww_data$est_oc43_assayed)*100

ww_data$method <- factor(ww_data$method,
                         levels = c(1, 2, 3, 4, 5, 6, 7, 8, 9,
10, 11, 12, 14, 15, 16, 17),
                         labels = c("Skimmed Milk",
                                    "Skimmed Milk - Vertrel",
                                    "Skimmed Milk - Direct",
                                    "Ultrafiltration",
                                    "Sludge",
                                    "BMFS",
                                    "BMFS - Vertrel",
                                    "PEG",
                                    "NucleoSpin",
                                    "PowerBiofilm",
                                    "HA Membrane Filtration",
                                    "CeresNano",
                                    "UF - Pellet",
                                    "Sludge - Grab",
                                    "NucleoMag",
                                    "CeresNano + ViralRNA"))
#Update this as more methods get used
```

```
ww_data$wwtp <- factor(ww_data$wwtp,
                       levels = c(1, 2, 3, 4, 5),
                       labels = c("West Point", "South Plant",
"Bright Water",
                                  "Lynden", "Alderwood"))

ww_data <- subset(ww_data, wwtp %in% c("West Point", "South
Plant", "Bright Water"))

oc43_data <-
  data.frame(
    ww_data$record_id, ww_data$wwtp, ww_data$method,
ww_data$exp_date, ww_data$sample_date,
    ww_data$comp_vol, ww_data$seed_vol, ww_data$method_vol,
ww_data$final_vol,
    ww_data$ext_vol, ww_data$elu_vol, ww_data$pcr_vol,
ww_data$slope, ww_data$intercept,
    ww_data$oc43_1, ww_data$oc43_2, ww_data$oc43,
    ww_data$dil_oc43_1, ww_data$dil_oc43_2, ww_data$dil_oc43
  )

write.csv(x = oc43_data,
          file = "oc43_data.csv")
```


##Volume Assayed
```{r, tidy = TRUE}
ww_data$method <- as.factor(ww_data$method)

library(ggplot2)
volumeassayed <- ggplot(data = ww_data,
       aes(x = method, y = effective_vol, fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank())+
  ggtitle("Effective Volume Assayed")+
  labs(y="Volume (mL)")+
  scale_fill_brewer(palette="RdBu",
                    name = "Method")
volumeassayed
```

```
ggsave("volumeassayed.png", plot = volumeassayed, device =
"png", dpi = 300, limitsize = TRUE)

percentrecovery <- ggplot(data = ww_data, aes(x = method,
                                              y =
percent_recovery,
                                              fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12))+
  ggtitle("OC43 Recovery")+
  labs(y="Recovery")+
  scale_fill_brewer(palette="RdBu",
                    name = "Method")
  #scale_y_continuous(trans = "log10")
percentrecovery
ggsave("percentrecovery.png", plot = percentrecovery, device =
"png", dpi = 300, limitsize = TRUE)

library("ggpubr")

vol_percent <- ggarrange(volumeassayed, percentrecovery,
                         nrow = 2,
                         ncol = 1,
                         common.legend = TRUE,
                         legend = "none",
                         labels = "AUTO")
vol_percent
ggsave(vol_percent, filename = "vol_percent.tiff",
       units = 'in', width = 8.5, height = 11, dpi = 600)
```


##N1 Graphs Over Time, SM and SM-D
```{r, tidy = TRUE}
ww_data_sm_smd_sub <- subset(ww_data, method %in% c("Skimmed
Milk", "Skimmed Milk - Direct"))
wp_subset <- subset(ww_data_sm_smd_sub, wwtp %in% c("West
Point"))
sp_subset <- subset(ww_data_sm_smd_sub, wwtp %in% c("South
Plant"))
```

```
bw_subset <- subset(ww_data_sm_smd_sub, wwtp %in% c("Bright
Water"))

library(ggplot2)

wp_n1 <- ggplot(data = wp_subset,
       aes(x = sample_date,
           y = n1,
           fill = wwtp,
           shape = wwtp))+
  geom_point(aes(x = sample_date, y = n1, color = wwtp), size =
2)+
  facet_wrap(~method)+
  scale_shape_manual(values = c(16))+
  scale_color_manual(values = c("#67001F"))+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "1 month")+
  ylim(32.5, 45)+
  theme(legend.position="bottom",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10),
        axis.text.x = element_text(angle = 60, hjust = 1))+
  ggtitle("West Point")+
  labs(y = "N1 Cq Value",
       x = "Date")
wp_n1

wp_subset$n1_gc <- 10^((wp_subset$n1 -
wp_subset$n1_int)/wp_subset$n1_slope)
wp_subset$n1_gcL <-
(wp_subset$n1_gc/wp_subset$effective_vol)*1000

library(readr)
biweekly_cases_seattle <- read_csv("biweekly_cases_seattle.csv",
                                   col_types = cols(Week_End =
col_date(format = "%Y-%m-%d"),
                                                    Week_Start =
col_date(format = "%Y-%m-%d")))
daily_cases_seattle <- read_csv("City_daily_cases.csv",
                                col_types = cols(Collection_Date
= col_date(format = "%m/%d/%Y")))

coeff <- 1

wp_n1_gc <- ggplot(data = wp_subset,
```

```
                  aes(x = sample_date,
                      y = n1_gcL))+
  geom_col(aes(x = sample_date,
               y = n1_gcL,
               fill = wwtp),
           position = "dodge",
           width = 2)+
  geom_col(data = biweekly_cases_seattle,
           aes(x = Week_End, y = Confirmed_Cases/coeff, color =
City),
           width = 2,
           size = 0.05,
           fill = "#CC6600")+
  theme(legend.position = "bottom",
        plot.title=element_text(face="bold", hjust=0.5, size=24,
color = "#4b2e83"),
        axis.title=element_text(size=18),
        legend.text=element_text(size=12),
        legend.title = element_text(size = 12),
        axis.text=element_text(size=18),
        axis.text.x = element_text(angle = 45, hjust = 1, size =
16),
        panel.background = element_rect(fill = "white"),
        axis.line.x.bottom = element_line(color = "black"),
        axis.line.y = element_line(color = "black"),
        panel.grid.major = element_line(color = "lightgray",
size = 0.15))+
  ggtitle("N1 Gene Copy - West Point")+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "1 month",
               limits = as.Date(c("2020-04-01", "2021-10-01")))+
  labs(x = "Sample Date",
       y = "Gene Copy / L")+
  scale_fill_manual(values = c("#4b2e83"),
                    name = "Detected SARS in WW")+
  scale_color_manual(values = "#CC6600",
                     name = "Confirmed Bi-Weekly Cases")+
  scale_y_continuous(sec.axis = sec_axis(trans = ~.*coeff,
                                         name = "Confirmed
Weekly Cases"),
                     limits = c(0, 16000))
wp_n1_gc

ggsave(wp_n1_gc, filename = "wp_n1_gc.tiff",
       units = 'in', width = 10, height = 6, dpi = 600)

sp_n1 <- ggplot(data = sp_subset,
       aes(x = sample_date,
```

```
           y = n1,
           fill = wwtp,
           shape = wwtp))+
  geom_point(aes(x = sample_date, y = n1, color = wwtp), size =
2)+
  facet_wrap(~method)+
  scale_shape_manual(values = c(17))+
  scale_color_manual(values = c("#053061"))+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "1 month")+
  ylim(32.5, 45)+
  theme(legend.position="bottom",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10),
        axis.text.x = element_text(angle = 60, hjust = 1))+
  ggtitle("South Plant")+
  labs(y = "N1 Cq Value",
       x = "Date")
sp_n1

sp_subset$n1_gc <- 10^((sp_subset$n1 -
sp_subset$n1_int)/sp_subset$n1_slope)
sp_subset$n1_gcL <-
(sp_subset$n1_gc/sp_subset$effective_vol)*1000

coeff <- 0.1

sp_n1_gc <- ggplot(data = sp_subset,
                   aes(x = sample_date,
                       y = n1_gcL))+
  geom_col(aes(x = sample_date,
               y = n1_gcL,
               fill = wwtp),
           position = "dodge",
           width = 2)+
  geom_col(data = biweekly_cases_seattle,
           aes(x = Week_End, y = Confirmed_Cases/coeff, color =
City),
           width = 2,
           size = 0.05,
           fill = "#CC6600")+
  theme(legend.position = "bottom",
        plot.title=element_text(face="bold", hjust=0.5, size=24,
color = "#4b2e83"),
        axis.title=element_text(size=18),
```

```
        legend.text=element_text(size=12),
        legend.title = element_text(size = 12),
        axis.text=element_text(size=18),
        axis.text.x = element_text(angle = 45, hjust = 1, size =
16),
        panel.background = element_rect(fill = "white"),
        axis.line.x.bottom = element_line(color = "black"),
        axis.line.y = element_line(color = "black"),
        panel.grid.major = element_line(color = "lightgray",
size = 0.15))+
  ggtitle("N1 Gene Copy - South Plant")+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "1 month",
               limits = as.Date(c("2020-03-15", "2021-10-01")))+
  labs(x = "Sample Date",
       y = "Gene Copy / L")+
  scale_fill_manual(values = c("#4b2e83"),
                    name = "Detected SARS in WW")+
  scale_color_manual(values = "#CC6600",
                     name = "Confirmed Bi-Weekly Cases")+
  scale_y_continuous(sec.axis = sec_axis(trans = ~.*coeff,
                                         name = "Confirmed
Weekly Cases"),
                     limits =  c(0, 190000))
sp_n1_gc

ggsave(sp_n1_gc, filename = "sp_n1_gc.tiff",
       units = 'in', width = 10, height = 6, dpi = 600)

bw_subset$n1_gc <- 10^((bw_subset$n1 -
bw_subset$n1_int)/bw_subset$n1_slope)
bw_subset$n1_gcL <-
(bw_subset$n1_gc/bw_subset$effective_vol)*1000

coeff <- 0.1

bw_n1 <- ggplot(data = bw_subset,
       aes(x = sample_date,
           y = n1,
           fill = wwtp,
           shape = wwtp))+
  geom_point(aes(x = sample_date, y = n1, color = wwtp), size =
2)+
  facet_wrap(~method)+
  scale_shape_manual(values = c(18))+
  scale_color_manual(values = c("#92c5de"))+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "1 month")+
```

```
  ylim(32.5, 45)+
  theme(legend.position="bottom",
        plot.title=element_text(face="bold", hjust=0.5,
size=18),
        axis.title=element_text(size=14),
        legend.text=element_text(size=12),
        axis.text=element_text(size=10),
        axis.text.x = element_text(angle = 60, hjust = 1))+
  ggtitle("Bright Water")+
  labs(y = "N1 Cq Value",
       x = "Date")
bw_n1

bw_n1_gc <- ggplot(data = bw_subset,
                   aes(x = sample_date,
                       y = n1_gcL))+
  geom_col(aes(x = sample_date,
               y = n1_gcL,
               fill = wwtp),
           position = "dodge",
           width = 2)+
  geom_col(data = biweekly_cases_seattle,
           aes(x = Week_End, y = Confirmed_Cases/coeff, color =
City),
           width = 2,
           size = 0.05,
           fill = "#CC6600")+
  theme(legend.position = "bottom",
        plot.title=element_text(face="bold", hjust=0.5, size=24,
color = "#4b2e83"),
        axis.title=element_text(size=18),
        legend.text=element_text(size=12),
        legend.title = element_text(size = 12),
        axis.text=element_text(size=18),
        axis.text.x = element_text(angle = 45, hjust = 1, size =
16),
        panel.background = element_rect(fill = "white"),
        axis.line.x.bottom = element_line(color = "black"),
        axis.line.y = element_line(color = "black"),
        panel.grid.major = element_line(color = "lightgray",
size = 0.15))+
  ggtitle("N1 Gene Copy - Brightwater")+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "1 month",
               limits = as.Date(c("2020-03-15", "2021-10-01")))+
  labs(x = "Sample Date",
       y = "Gene Copy / L")+
  scale_fill_manual(values = c("#4b2e83"),
```

```
                    name = "Detected SARS in WW")+
  scale_color_manual(values = "#CC6600",
                     name = "Confirmed Bi-Weekly Cases")+
  scale_y_continuous(sec.axis = sec_axis(trans = ~.*coeff,
                                         name = "Confirmed
Weekly Cases"),
                     limits =  c(0, 100000))
bw_n1_gc

ggsave(plot = bw_n1_gc, filename = "bw_n1_gc.tiff",
       units = 'in', width = 10, height = 6, dpi = 600)

library("ggpubr")

n1_total <- ggarrange(wp_n1, sp_n1, bw_n1,
                      ncol = 1,
                      nrow = 3,
                      common.legend = TRUE,
                      legend = "none",
                      labels = "AUTO")
n1_total
ggsave(n1_total, filename = "n1_cq_total.tiff",
       units = 'in', width = 5, height = 9, dpi = 600)

n1_gcL <- as.vector(c(wp_subset$n1_gcL, sp_subset$n1_gcL,
bw_subset$n1_gcL))
```

```
##Stats for SM-D
```{r, tidy = TRUE}
unique_dates <- unique(ww_data$sample_date)
ww_data$sm_boolean <- 0

#want all days with BOTH SM and SM-D, nothing more
for(i in unique_dates){
  method_date <- ww_data$method[ww_data$sample_date==i]
  if(length(method_date[method_date == "Skimmed Milk - Direct"])
>= 1){
    if(length(method_date[method_date == "Skimmed Milk"]) >= 1){
      ww_data$sm_boolean[ww_data$sample_date==i] = 1}
    }
  }

data_sm <- subset(ww_data, sm_boolean %in% c(1))
```
```

```
tab_sm <- table(data_sm$pres_pos[data_sm$method=="Skimmed
Milk"],
                data_sm$pres_pos[data_sm$method=="Skimmed Milk -
Direct"],
                dnn = c("SM", "SM-DIR"))
tab_sm

fisher.test(tab_sm)
mcnemar.test(tab_sm)

tab_sm_dil <-
table(data_sm$pres_pos_dil[data_sm$method=="Skimmed Milk"],

data_sm$pres_pos_dil[data_sm$method=="Skimmed Milk - Direct"],
                     dnn = c("SM", "SM-DIR"))
tab_sm_dil

fisher.test(tab_sm_dil)
mcnemar.test(tab_sm_dil)

tab_dir_vert <- table(data_sm$pres_pos[data_sm$method=="Skimmed
Milk - Vertrel"],
                      data_sm$pres_pos[data_sm$method=="Skimmed
Milk - Direct"],
                      dnn = c("SM - V", "SM-DIR"))
tab_dir_vert
mcnemar.test(tab_dir_vert)

tab_dir_vert_dil <-
table(data_sm$pres_pos_dil[data_sm$method=="Skimmed Milk -
Vertrel"],

data_sm$pres_pos_dil[data_sm$method=="Skimmed Milk - Direct"],
                           dnn = c("SM - V", "SM-DIR"))
tab_dir_vert_dil
mcnemar.test(tab_dir_vert_dil)

my_comparisons <- list(c("Skimmed Milk", "Skimmed Milk -
Vertrel"),
                       c("Skimmed Milk", "Skimmed Milk -
Direct"),
                       c("Skimmed Milk - Vertrel", "Skimmed Milk
- Direct"))

library(ggplot2)
smd_volumeassayed <- ggplot(data = subset(data_sm, method %in%
c("Skimmed Milk",
```

```
"Skimmed Milk - Vertrel",

"Skimmed Milk - Direct")),
       aes(x = method, y = effective_vol, fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank())+
  ggtitle("Effective Volume Assayed")+
  labs(y="Volume (mL)")+
  scale_fill_brewer(palette="Purples",
                    name = "Method")+
  geom_label(y = 8,
             x = 1,
             fill = "white",
             size = 3,
             label = paste('n =',

length(data_sm$effective_vol[data_sm$method == "Skimmed
Milk"])))+
  geom_label(y = 8,
             x = 2,
             fill = "white",
             size = 3,
             label = paste('n =',

length(data_sm$effective_vol[data_sm$method == "Skimmed Milk -
Vertrel"])))+

  geom_label(y = 8,
             x = 3,
             fill = "white",
             size = 3,
             label = paste('n =',

length(data_sm$effective_vol[data_sm$method == "Skimmed Milk -
Direct"])))+
  ylim(0,9)
smd_volumeassayed

ggsave("smd_volumeassayed.png", plot = smd_volumeassayed, device
= "png", dpi = 300, limitsize = TRUE)
```

```
smd_recovery <- ggplot(data = subset(data_sm, method %in%
c("Skimmed Milk",

"Skimmed Milk - Vertrel",

"Skimmed Milk - Direct")),
       aes(x = method, y = percent_recovery, fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12))+
  ggtitle("OC43 Recovery")+
  labs(y="Recovery")+
  #stat_compare_means(method = "t.test",
                     #paired = TRUE,
                     #label.y = c(60))+
  #scale_y_continuous(trans = "log10")+
  scale_fill_brewer(palette="Purples",
                    name = "Method")+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100))+
  annotate("text", x = 1, y = 58, size = 3,
           label = "Paired t-test")+
  stat_compare_means(comparisons = my_comparisons,
                     method = "t.test",
                     paired = TRUE,
                     size = 3)
smd_recovery
ggsave("smd_recovery.png", plot = smd_recovery, device = "png",
dpi = 300, limitsize = TRUE)

library("ggpubr")

smd_vol_percent <- ggarrange(smd_volumeassayed, smd_recovery,
                             nrow = 1,
                             ncol = 2,
                             common.legend = TRUE,
                             legend = "none",
                             labels = "auto")
smd_vol_percent
ggsave(smd_vol_percent, filename = "smd_vol_percent.tiff",
       units = 'in', width = 6.8, height = 3.7, dpi = 600)
```

```
wp_sampledate <- data_sm$sample_date[data_sm$wwtp=="West
Point"&data_sm$method=="Skimmed Milk - Direct"]
wp_compvol <- data_sm$comp_vol[data_sm$wwtp=="West
Point"&data_sm$method=="Skimmed Milk - Direct"]
sp_sampledate <- data_sm$sample_date[data_sm$wwtp=="South
Plant"&data_sm$method=="Skimmed Milk - Direct"]
sp_compvol <- data_sm$comp_vol[data_sm$wwtp=="South
Plant"&data_sm$method=="Skimmed Milk - Direct"]
bw_sampledate <- data_sm$sample_date[data_sm$wwtp=="Bright
Water"&data_sm$method=="Skimmed Milk - Direct"]
bw_compvol <- data_sm$comp_vol[data_sm$wwtp=="Bright
Water"&data_sm$method=="Skimmed Milk - Direct"]

smd_sampledate <- data.frame(wp_sampledate, wp_compvol,
                             sp_sampledate, sp_compvol,
                             bw_sampledate, bw_compvol)
library(dplyr)
smd_sampledate %>% rename("Plant A" = wp_sampledate,
                          "Plant B" = sp_sampledate,
                          "Plant C" = bw_sampledate)
write.csv(x = smd_sampledate,
          file = "smd_sampledate.csv")

method <- c("Skimmed Milk", "Skimmed Milk - Vertrel", "Skimmed
Milk - Direct")

max_recovery <-
  c(max(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk"], na.rm = TRUE),
    max(data_sm$percent_recovery[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm = TRUE),
    max(data_sm$percent_recovery[data_sm$method=="Skimmed Milk -
Direct"], na.rm = TRUE))
max_recovery <- round(max_recovery, digits = 2)

recovery_quantile_75 <-
  c(quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk"],
             probs = c(0.75), na.rm = TRUE),
    quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk - Vertrel"],
             probs = c(0.75), na.rm = TRUE),
    quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk - Direct"],
             probs = c(0.75), na.rm = TRUE))
recovery_quantile_75 <- round(recovery_quantile_75, digits = 2)

recovery_med <-
```

```
  c(quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk"],
             probs = c(0.5), na.rm = TRUE),
    quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk - Vertrel"],
             probs = c(0.5), na.rm = TRUE),
    quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk - Direct"],
             probs = c(0.5), na.rm = TRUE))
recovery_med <- round(recovery_med, digits = 2)

recovery_quantile_25 <-
  c(quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk"],
             probs = c(0.25), na.rm = TRUE),
    quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk - Vertrel"],
             probs = c(0.25), na.rm = TRUE),
    quantile(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk - Direct"],
             probs = c(0.25), na.rm = TRUE))
recovery_quantile_25 <- round(recovery_quantile_25, digits = 2)

recovery_min <-
  c(min(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk"], na.rm = TRUE),
    min(data_sm$percent_recovery[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm = TRUE),
    min(data_sm$percent_recovery[data_sm$method=="Skimmed Milk -
Direct"], na.rm = TRUE))
recovery_min <- round(recovery_min, digits = 2)

mean_recovery <-
  c(mean(data_sm$percent_recovery[data_sm$method=="Skimmed
Milk"], na.rm = TRUE),
    mean(data_sm$percent_recovery[data_sm$method=="Skimmed Milk
- Vertrel"], na.rm = TRUE),
    mean(data_sm$percent_recovery[data_sm$method=="Skimmed Milk
- Direct"], na.rm = TRUE))
mean_recovery <- round(mean_recovery, digits = 2)

recovery_stats <- data.frame(method,
                             max_recovery,
                             recovery_quantile_75,
                             recovery_med,
                             recovery_quantile_25,
                             recovery_min,
                             mean_recovery)
```

```
recovery_stats_test <- data.frame(t(recovery_stats[-1]))
colnames(recovery_stats_test) <- recovery_stats[,1]
rownames(recovery_stats_test) <- c("Max", "75th Quant.",
"Median",
                                   "25th Quant.", "Minimum",
"Mean")

write.csv(x = recovery_stats_test,
          file = "recovery_stats.csv")

max_vol_assayed <-
  c(max(data_sm$effective_vol[data_sm$method=="Skimmed Milk"],
na.rm = TRUE),
    max(data_sm$effective_vol[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm = TRUE),
    max(data_sm$effective_vol[data_sm$method=="Skimmed Milk -
Direct"], na.rm = TRUE))
max_vol_assayed <- round(max_vol_assayed, digits = 2)

vol_assayed_quantile_75 <-
  c(quantile(data_sm$effective_vol[data_sm$method=="Skimmed
Milk"],
             probs = c(0.75), na.rm = TRUE),
    quantile(data_sm$effective_vol[data_sm$method=="Skimmed Milk
- Vertrel"],
             probs = c(0.75), na.rm = TRUE),
    quantile(data_sm$effective_vol[data_sm$method=="Skimmed Milk
- Direct"],
             probs = c(0.75), na.rm = TRUE))
vol_assayed_quantile_75 <- round(vol_assayed_quantile_75, digits
= 2)

vol_assayed_med <-
  c(quantile(data_sm$effective_vol[data_sm$method=="Skimmed
Milk"],
             probs = c(0.5), na.rm = TRUE),
    quantile(data_sm$effective_vol[data_sm$method=="Skimmed Milk
- Vertrel"],
             probs = c(0.5), na.rm = TRUE),
    quantile(data_sm$effective_vol[data_sm$method=="Skimmed Milk
- Direct"],
             probs = c(0.5), na.rm = TRUE))
vol_assayed_med <- round(vol_assayed_med, digits = 2)

vol_assayed_quantile_25 <-
  c(quantile(data_sm$effective_vol[data_sm$method=="Skimmed
Milk"],
             probs = c(0.25), na.rm = TRUE),
```

```
    quantile(data_sm$effective_vol[data_sm$method=="Skimmed Milk
- Vertrel"],
             probs = c(0.25), na.rm = TRUE),
    quantile(data_sm$effective_vol[data_sm$method=="Skimmed Milk
- Direct"],
             probs = c(0.25), na.rm = TRUE))
vol_assayed_quantile_25 <- round(vol_assayed_quantile_25, digits
= 2)

vol_assayed_min <-
  c(min(data_sm$effective_vol[data_sm$method=="Skimmed Milk"],
na.rm = TRUE),
    min(data_sm$effective_vol[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm = TRUE),
    min(data_sm$effective_vol[data_sm$method=="Skimmed Milk -
Direct"], na.rm = TRUE))
vol_assayed_min <- round(vol_assayed_min, digits = 2)

mean_vol_assayed <-
  c(mean(data_sm$effective_vol[data_sm$method=="Skimmed Milk"],
na.rm = TRUE),
    mean(data_sm$effective_vol[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm = TRUE),
    mean(data_sm$effective_vol[data_sm$method=="Skimmed Milk -
Direct"], na.rm = TRUE))
mean_vol_assayed <- round(mean_vol_assayed, digits = 2)

vol_assayed_stats <- data.frame(method,
                                max_vol_assayed,
                                vol_assayed_quantile_75,
                                vol_assayed_med,
                                vol_assayed_quantile_25,
                                vol_assayed_min,
                                mean_vol_assayed)
vol_assayed_stats_test <- data.frame(t(vol_assayed_stats[-1]))
colnames(vol_assayed_stats_test) <- vol_assayed_stats[,1]
rownames(vol_assayed_stats_test) <- c("Max", "75th Quant.",
"Median",
                                      "25th Quant.", "Minimum",
"Mean")

write.csv(x = vol_assayed_stats_test,
          file = "vol_assayed_stats.csv")
```

#Control Charts Skimmed Milk Methods
```{r, tidy = TRUE}
```

```
average <- mean(data_sm$oc43[data_sm$method=="Skimmed Milk"],
na.rm=TRUE)
stdev <- sd(data_sm$oc43[data_sm$method=="Skimmed Milk"],
na.rm=TRUE)
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

library(ggplot2)
sm_shewart <- ggplot(data = subset(x = data_sm, method=="Skimmed
Milk"),
                     aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(12, 36)+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "4 weeks")+
  ggtitle("Skimmed Milk")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
```

```
        axis.text=element_text(size=9),
        axis.text.x = element_text(angle = 45, hjust=1))
sm_shewart

average <- mean(data_sm$oc43[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm=TRUE)
stdev <- sd(data_sm$oc43[data_sm$method=="Skimmed Milk -
Vertrel"], na.rm=TRUE)
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

smv_shewart <- ggplot(data = subset(x = data_sm,
method=="Skimmed Milk - Vertrel"),
                      aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(12, 36)+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "4 weeks")+
  ggtitle("Skimmed Milk - Vertrel")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
```

```
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9),
        axis.text.x = element_text(angle = 45, hjust=1))
smv_shewart

average <- mean(data_sm$oc43[data_sm$method=="Skimmed Milk -
Direct"], na.rm=TRUE)
stdev <- sd(data_sm$oc43[data_sm$method=="Skimmed Milk -
Direct"], na.rm=TRUE)
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

smd_shewart <- ggplot(data = subset(x = data_sm,
method=="Skimmed Milk - Direct"),
                      aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(12, 36)+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "4 weeks")+
```

```
  ggtitle("Skimmed Milk - Direct")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9),
        axis.text.x = element_text(angle = 45, hjust=1))
smd_shewart

library(ggplot2)
library("ggpubr")
control_charts <- ggarrange(sm_shewart, smv_shewart,
smd_shewart,
                            ncol = 1,
                            nrow = 3,
                            common.legend = TRUE,
                            legend = "right",
                            labels = "auto")
control_charts
annotate_control <- annotate_figure(control_charts,
                                    left = text_grob("OC43 Cq
Values",
                                                     rot = 90,
                                                     face =
"bold"))
annotate_control
ggsave(control_charts, filename = "control_charts.tiff",
       units = 'in', width = 3.3, height = 6.8, dpi = 600)

mean_oc43_cq <-
  c(mean(data_sm$oc43[data_sm$method=="Skimmed Milk"], na.rm =
TRUE),
    mean(data_sm$oc43[data_sm$method=="Skimmed Milk - Vertrel"],
na.rm = TRUE),
    mean(data_sm$oc43[data_sm$method=="Skimmed Milk - Direct"],
na.rm = TRUE))
mean_oc43_cq <- round(mean_oc43_cq, digits = 1)

sd_oc43_cq <-
  c(sd(data_sm$oc43[data_sm$method=="Skimmed Milk"], na.rm =
TRUE),
    sd(data_sm$oc43[data_sm$method=="Skimmed Milk - Vertrel"],
na.rm = TRUE),
    sd(data_sm$oc43[data_sm$method=="Skimmed Milk - Direct"],
na.rm = TRUE))
sd_oc43_cq <- round(sd_oc43_cq, digits = 1)
```

```
var_oc43_cq <-
  c(var(data_sm$oc43[data_sm$method=="Skimmed Milk"], na.rm =
TRUE),
    var(data_sm$oc43[data_sm$method=="Skimmed Milk - Vertrel"],
na.rm = TRUE),
    var(data_sm$oc43[data_sm$method=="Skimmed Milk - Direct"],
na.rm = TRUE))
var_oc43_cq <- round(var_oc43_cq, digits = 1)

max_oc43_cq <-
  c(max(data_sm$oc43[data_sm$method=="Skimmed Milk"], na.rm =
TRUE),
    max(data_sm$oc43[data_sm$method=="Skimmed Milk - Vertrel"],
na.rm = TRUE),
    max(data_sm$oc43[data_sm$method=="Skimmed Milk - Direct"],
na.rm = TRUE))
max_oc43_cq <- round(max_oc43_cq, digits = 1)

min_oc43_cq <-
  c(min(data_sm$oc43[data_sm$method=="Skimmed Milk"], na.rm =
TRUE),
    min(data_sm$oc43[data_sm$method=="Skimmed Milk - Vertrel"],
na.rm = TRUE),
    min(data_sm$oc43[data_sm$method=="Skimmed Milk - Direct"],
na.rm = TRUE))
min_oc43_cq <- round(min_oc43_cq, digits = 1)

miss_oc43_cq <-
  c(sum(is.na(data_sm$oc43[data_sm$method=="Skimmed Milk"])),
    sum(is.na(data_sm$oc43[data_sm$method=="Skimmed Milk -
Vertrel"])),
    sum(is.na(data_sm$oc43[data_sm$method=="Skimmed Milk -
Direct"])))
miss_oc43_cq <- round(miss_oc43_cq, digits = 0)

oc43_cq_stats <- data.frame(method,
                            mean_oc43_cq,
                            sd_oc43_cq,
                            var_oc43_cq,
                            max_oc43_cq,
                            min_oc43_cq,
                            miss_oc43_cq)
oc43_cq_test <- data.frame(t(oc43_cq_stats[-1]))
colnames(oc43_cq_test) <- oc43_cq_stats[,1]
rownames(oc43_cq_test) <- c("Mean", "Std. Dev.", "Variance",
                            "Max", "Min", "Non. Detect.")
```

```
write.csv(x = oc43_cq_test,
          file = "oc43_cq_stats.csv")
```


#Balloon Plots SARS-CoV-2 Skimmed Milk Methods
```{r, tidy = TRUE}

size <- as.vector(c(length(data_sm$method[data_sm$method ==
"Skimmed Milk"]),
                    length(data_sm$method[data_sm$method ==
"Skimmed Milk - Vertrel"]),
                    length(data_sm$method[data_sm$method ==
"Skimmed Milk - Direct"])))

sars_pos <- as.vector(c((sum(data_sm$pres_pos[data_sm$method ==
"Skimmed Milk"]) /
                  length(data_sm$pres_pos[data_sm$method ==
"Skimmed Milk"]))*100,
                (sum(data_sm$pres_pos[data_sm$method == "Skimmed
Milk - Vertrel"]) /
                  length(data_sm$pres_pos[data_sm$method ==
"Skimmed Milk - Vertrel"]))*100,
                (sum(data_sm$pres_pos[data_sm$method == "Skimmed
Milk - Direct"]) /
                  length(data_sm$pres_pos[data_sm$method ==
"Skimmed Milk - Direct"]))*100))

sars_pos_dil <-
  as.vector(c((sum(data_sm$pres_pos_dil[data_sm$method ==
"Skimmed Milk"]) /
                  length(data_sm$pres_pos_dil[data_sm$method ==
"Skimmed Milk"]))*100,
                (sum(data_sm$pres_pos_dil[data_sm$method ==
"Skimmed Milk - Vertrel"]) /
                  length(data_sm$pres_pos_dil[data_sm$method ==
"Skimmed Milk - Vertrel"]))*100,
                (sum(data_sm$pres_pos_dil[data_sm$method ==
"Skimmed Milk - Direct"]) /
                  length(data_sm$pres_pos_dil[data_sm$method ==
"Skimmed Milk - Direct"]))*100))

n1_pos <- as.vector(c((sum(data_sm$n1_dummy[data_sm$method ==
"Skimmed Milk"]) /
                  length(data_sm$n1_dummy[data_sm$method ==
"Skimmed Milk"]))*100,
                (sum(data_sm$n1_dummy[data_sm$method == "Skimmed
Milk - Vertrel"]) /
```

```
                  length(data_sm$n1_dummy[data_sm$method ==
"Skimmed Milk - Vertrel"]))*100,
               (sum(data_sm$n1_dummy[data_sm$method == "Skimmed
Milk - Direct"]) /
                  length(data_sm$n1_dummy[data_sm$method ==
"Skimmed Milk - Direct"]))*100))

n1_pos_dil <-
  as.vector(c((sum(data_sm$n1_dil_dummy[data_sm$method ==
"Skimmed Milk"]) /
                  length(data_sm$n1_dil_dummy[data_sm$method ==
"Skimmed Milk"]))*100,
               (sum(data_sm$n1_dil_dummy[data_sm$method ==
"Skimmed Milk - Vertrel"]) /
                  length(data_sm$n1_dil_dummy[data_sm$method ==
"Skimmed Milk - Vertrel"]))*100,
               (sum(data_sm$n1_dil_dummy[data_sm$method ==
"Skimmed Milk - Direct"]) /
                  length(data_sm$n1_dil_dummy[data_sm$method ==
"Skimmed Milk - Direct"]))*100))

n2_pos <- as.vector(c((sum(data_sm$n2_dummy[data_sm$method ==
"Skimmed Milk"]) /
                  length(data_sm$n2_dummy[data_sm$method ==
"Skimmed Milk"]))*100,
               (sum(data_sm$n2_dummy[data_sm$method == "Skimmed
Milk - Vertrel"]) /
                  length(data_sm$n2_dummy[data_sm$method ==
"Skimmed Milk - Vertrel"]))*100,
               (sum(data_sm$n2_dummy[data_sm$method == "Skimmed
Milk - Direct"]) /
                  length(data_sm$n2_dummy[data_sm$method ==
"Skimmed Milk - Direct"]))*100))

n2_pos_dil <-
  as.vector(c((sum(data_sm$n2_dil_dummy[data_sm$method ==
"Skimmed Milk"]) /
                  length(data_sm$n2_dil_dummy[data_sm$method ==
"Skimmed Milk"]))*100,
               (sum(data_sm$n2_dil_dummy[data_sm$method ==
"Skimmed Milk - Vertrel"]) /
                  length(data_sm$n2_dil_dummy[data_sm$method ==
"Skimmed Milk - Vertrel"]))*100,
               (sum(data_sm$n2_dil_dummy[data_sm$method ==
"Skimmed Milk - Direct"]) /
                  length(data_sm$n2_dil_dummy[data_sm$method ==
"Skimmed Milk - Direct"]))*100))
```

```
positive <- as.vector(c(sars_pos, n1_pos, n2_pos))

positive_dil <- as.vector(c(sars_pos_dil, n1_pos_dil,
n2_pos_dil))

assays <- as.factor(c("SARS", "SARS", "SARS",
                      "N1", "N1", "N1",
                      "N2", "N2", "N2"))

assay_balloon <- cbind.data.frame(method, assays, size,
positive)
#need to have the two variables that are serving as x and Y for
balloon plots next to each other
assay_balloon$method <- as.factor(assay_balloon$method)

write.csv(x = assay_balloon, file = "assay_balloon.csv")
library(readr)
assay_balloon_test <- read_csv("assay_balloon.csv",
                               col_types = cols(X1 =
col_skip()))

library(ggpubr)
assayballoonplot <- ggballoonplot(data = assay_balloon_test,
              aes(x = assays,
                  y = method),
                  size = "size",
                  fill = "positive")+
  geom_text(data = assay_balloon_test, nudge_x = 0.4,
            aes(label = round(x = positive,
                              digits = 1)))+
  scale_size_continuous(range = c(5, 10),
                        name = "Sample Size")+
  scale_fill_gradient(low = "#DDDDDD",
                      high = "#4B2E83",
                      name = "Percent Positive",
                      limits = c(0, 100))+
  ggtitle("Non-Diluted")+
  guides()+
  theme(legend.position = "right",
        plot.title = element_text(#face = "bold",
                                  hjust = 0.5,
                                  size = 12),
        axis.title = element_text(size = 12),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 12, face = "bold"),
        axis.text.x = element_text(size = 12, color = "black",
face = "bold",
                                   angle = 30),
```

```
        axis.text.y = element_text(size = 12, color = "black",
face = "bold"))
assayballoonplot
ggsave(filename = "assayballoonplot.png",
       plot = assayballoonplot,
       device = "png",
       dpi = 300,
       limitsize = TRUE)

assay_balloon_dil <- cbind.data.frame(method, assays, size,
positive_dil)
#need to have the two variables that are serving as x and Y for
balloon plots next to each other
assay_balloon_dil$method <- as.factor(assay_balloon_dil$method)

write.csv(x = assay_balloon_dil, file = "assay_balloon_dil.csv")
library(readr)
assay_balloon_dil_test <- read_csv("assay_balloon_dil.csv",
                                   col_types = cols(X1 =
col_skip()))

assayballoonplot_dil <- ggballoonplot(data =
assay_balloon_dil_test,
              aes(x = assays,
                  y = method),
                  size = "size",
                  fill = "positive_dil")+
  geom_text(data = assay_balloon_dil_test, nudge_x = 0.4,
            aes(label = round(x = positive_dil,
                              digits = 1)))+
  scale_size_continuous(range = c(5, 10),
                        name = "Sample Size")+
  scale_fill_gradient(low = "#DDDDDD",
                      high = "#4B2E83",
                      name = "Percent Positive",
                      limits = c(0,100))+
  ggtitle(bquote("Dilution" ~10^-1))+
  guides()+
  theme(legend.position = "right",
        plot.title = element_text(face = "bold",
                                  hjust = 0.5,
                                  size = 12),
        axis.title = element_text(size = 12),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 12, face = "bold"),
        axis.text.x = element_text(size = 12, color = "black",
face = "bold",
                                   angle = 30),
```

```
        axis.text.y = element_text(size = 12, color = "black",
face = "bold"))
assayballoonplot_dil
ggsave(filename = "assayballoonplot_dil.png",
       plot = assayballoonplot_dil,
       device = "png",
       dpi = 300,
       limitsize = TRUE)


library(ggplot2)
library("ggpubr")
balloon_plots <- ggarrange(assayballoonplot,
assayballoonplot_dil,
                           ncol = 2,
                           nrow = 1,
                           common.legend = TRUE,
                           legend = "bottom",
                           labels = "auto",
                           font.label = list(size = 12))
balloon_plots
#annotate_balloon <- annotate_figure(balloon_plots,
 #                                   top = text_grob("Percent
Positivity for SARS-CoV-2",
  #                                                    face =
"bold", size = 24))
#annotate_balloon
ggsave(balloon_plots, filename = "balloon_plots.tiff",
       units = 'in', width = 6.8, height = 3.7, dpi = 600)
```


##N1 and N2 Cq values
```{r, tidy = TRUE}
library(ggplot2)

smd_n1_cq <- ggplot(data = subset(data_sm, method %in%
c("Skimmed Milk",

"Skimmed Milk - Vertrel",

"Skimmed Milk - Direct")),
       aes(x = method, y = n1, fill = method))+
  theme_classic()+
  geom_boxplot()+
  theme(plot.title = element_text(hjust=0.5, size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
```

```
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = "none")+
  labs(y = "Cq Values")+
  ggtitle("N1")+
  scale_fill_brewer(palette = "Purples",
                    name = "Method")+
  scale_y_continuous(limits = c(30, 45))
smd_n1_cq
ggsave(plot = smd_n1_cq, filename = "smd_n1_cq.png", device =
"png", dpi = 300, limitsize = TRUE)

smd_n2_cq <- ggplot(data = subset(data_sm, method %in%
c("Skimmed Milk",

"Skimmed Milk - Vertrel",

"Skimmed Milk - Direct")),
       aes(x = method, y = n2, fill = method))+
  theme_classic()+
  geom_boxplot()+
  theme(plot.title = element_text(hjust=0.5, size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = "none")+
  labs(y = "Cq Values")+
  ggtitle("N2")+
  scale_fill_brewer(palette = "Purples",
                    name = "Method")+
  scale_y_continuous(limits = c(30, 45))
smd_n2_cq
ggsave(plot = smd_n2_cq, filename = "smd_n2_cq.png", device =
"png", dpi = 300,
       limitsize = TRUE)

smd_dil_n1_cq <- ggplot(data = subset(data_sm, method %in%
c("Skimmed Milk",

"Skimmed Milk - Vertrel",

"Skimmed Milk - Direct")),
       aes(x = method, y = dil_n1, fill = method))+
  theme_classic()+
  geom_boxplot()+
```

```
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = "none")+
  labs(y = "Cq Values")+
  ggtitle(bquote("N1" ~10^-1))+
  scale_fill_brewer(palette = "Purples",
                    name = "Method")+
  scale_y_continuous(limits = c(30, 45))
smd_dil_n1_cq
ggsave(plot = smd_dil_n1_cq, filename = "smd_dil_n1_cq.png",
device = "png", dpi = 300, limitsize = TRUE)

smd_dil_n2_cq <- ggplot(data = subset(data_sm, method %in%
c("Skimmed Milk",

"Skimmed Milk - Vertrel",

"Skimmed Milk - Direct")),
       aes(x = method, y = dil_n2, fill = method))+
  theme_classic()+
  geom_boxplot()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=16),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = "none")+
  labs(y = "Cq Values")+
  ggtitle(bquote("N2" ~10^-1))+
  scale_fill_brewer(palette = "Purples",
                    name = "Method")+
  scale_y_continuous(limits = c(30, 45))
smd_dil_n2_cq
ggsave(plot = smd_dil_n2_cq, filename = "smd_dil_n2_cq.png",
device = "png", dpi = 300, limitsize = TRUE)

library(ggplot2)
library("ggpubr")
sars_cq <- ggarrange(smd_n1_cq, smd_n2_cq, smd_dil_n1_cq,
smd_dil_n2_cq,
                     ncol = 2,
```

```
                    nrow = 2,
                    common.legend = FALSE,
                    labels = "auto",
                    font.label = list(size = 12))
sars_cq
ggsave(plot = sars_cq, filenam = "sars_cq.tiff", units = 'in',
width = 11, height = 8.5, dpi = 600)

sars_cq_data <- cbind.data.frame(data_sm$method, data_sm$wwtp,
data_sm$sample_date,
                                 data_sm$n1, data_sm$dil_n1,
                                 data_sm$n2, data_sm$dil_n2)

write.csv(x = sars_cq_data, file = "sars_cq_data.csv")
```

```
##Stats for SM-V
```{r, tidy = TRUE}
ww_data$smv_boolean <- 0

#want all days with BOTH SM and SM-V, nothing more
for(i in unique_dates){
  method_date <- ww_data$method[ww_data$sample_date==i]
  if(length(method_date[method_date == "Skimmed Milk -
Vertrel"]) > 1){
    if(length(method_date[method_date == "Skimmed Milk"]) > 1){
      ww_data$smv_boolean[ww_data$sample_date==i] = 1}
    }
  }

data_smv <- subset(ww_data, smv_boolean %in% c(1))

tab_smv <- table(data_smv$pres_pos[data_smv$method=="Skimmed
Milk"],
                 data_smv$pres_pos[data_smv$method=="Skimmed Milk
- Vertrel"],
                 dnn = c("SM", "SM-V"))
tab_smv

fisher.test(tab_smv)
mcnemar.test(tab_smv)

tab_smv_dil <-
table(data_smv$pres_pos_dil[data_smv$method=="Skimmed Milk"],
```
```

```
data_smv$pres_pos_dil[data_smv$method=="Skimmed Milk -
Vertrel"],
                      dnn = c("SM", "SM-V"))
tab_smv_dil
fisher.test(tab_smv_dil)
mcnemar.test(tab_smv_dil)
```


##Stats for CeresNano
```{r, tidy = TRUE}
ww_data$ceres_boolean <- 0
unique_dates <- unique(ww_data$sample_date)

#want all days with BOTH SM-D and Ceres, nothing more
#this also isn't including BW samples because the sample dates
are different but the exp dates are all the same
for(i in unique_dates){
  method_date <- ww_data$method[ww_data$sample_date==i]
  if(length(method_date[method_date == "CeresNano"]) >= 1){
    if(length(method_date[method_date == "Skimmed Milk -
Direct"]) >= 1){
      ww_data$ceres_boolean[ww_data$sample_date==i] = 1}
    }
  }

data_ceres <- subset(ww_data, ceres_boolean %in% c(1))

data_ceres$method <- as.factor(data_ceres$method)
data_ceres$pres_pos <- as.factor(data_ceres$pres_pos)
data_ceres$pres_pos_dil <- as.factor(data_ceres$pres_pos_dil)

tab_ceres <-
table(data_ceres$pres_pos[data_ceres$method=="CeresNano"],

data_ceres$pres_pos[data_ceres$method=="Skimmed Milk - Direct"],
                    dnn = c("CeresNano", "SM-D"))
tab_ceres

fisher.test(tab_ceres)
mcnemar.test(tab_ceres)

tab_ceres_dil <-
table(data_ceres$pres_pos_dil[data_ceres$method=="CeresNano"],

data_ceres$pres_pos_dil[data_ceres$method=="Skimmed Milk -
Direct"],
```

```
                        dnn = c("CeresNano", "SM-D"))
tab_ceres_dil
#this isn't working because there aren't any 0's for
CeresNano...
#not sure how I can force a zero into this table
fisher.test(tab_ceres_dil)
mcnemar.test(tab_ceres_dil)

```


##Stats for PowerBiofilm
```{r, tidy = TRUE}
ww_data$bio_boolean <- 0
write.csv(x = ww_data, file = "ww_data.csv")

#this isn't working anymore....
for(i in unique_dates){
  method_date <- ww_data$method[ww_data$sample_date==i]
  if(length(method_date[method_date == "PowerBiofilm"]) >= 1){
    if(length(method_date[method_date == "Skimmed Milk -
Direct"]) >= 1){
      ww_data$bio_boolean[ww_data$sample_date==i] = 1}
    }
}

data_bio <- subset(ww_data, bio_boolean %in% c(1))

data_bio$method <- as.factor(data_bio$method)
data_bio$pres_pos <- as.factor(data_bio$pres_pos)
data_bio$pres_pos_dil <- as.factor(data_bio$pres_pos_dil)

write.csv(x = data_bio, file = "data_bio.csv")
tab_bio <-
table(data_bio$pres_pos[data_bio$method=="PowerBiofilm"],
                  data_bio$pres_pos[data_bio$method=="Skimmed
Milk - Direct"],
                  dnn = c("PowerBiofilm", "ViralRNA"))
tab_bio

fisher.test(tab_bio)
mcnemar.test(tab_bio)

tab_bio_dil <-
table(data_bio$pres_pos_dil[data_bio$method=="Skimmed Milk -
Direct"],

data_bio$pres_pos_dil[data_bio$method=="PowerBiofilm"],
```

```
                    dnn = c("ViralRNA", "PowerBiofilm"))
tab_bio_dil
#Same thing as with CeresNano, PowerBiofilm detects all SARS so
there are no zeros...
fisher.test(tab_bio_dil)
mcnemar.test(tab_bio_dil)

#Comparing OC43 Recovery
shapiro.test(
  with(
    data_bio,
    percent_recovery[method == "Skimmed Milk - Direct"]
    - percent_recovery[method == "PowerBiofilm"]))

# Subset data
viral_rna <- subset(data_bio,  method == "Skimmed Milk -
Direct", percent_recovery,
                    drop = TRUE)
print(summary(viral_rna))
t.test(viral_rna)

powerbiofilm <- subset(data_bio,  method == "PowerBiofilm",
percent_recovery,
                       drop = TRUE)
print(summary(powerbiofilm))
t.test(powerbiofilm)

res <- wilcox.test(viral_rna, powerbiofilm, paired = TRUE)
res

#Graph Percent Recovery
ext_comp <- data.frame(ViralRNA = viral_rna, PowerBiofilm =
powerbiofilm)

library(ggbreak)

powerbiofilm_recovery <-
  ggpaired(data = subset(data_bio, method %in% c("Skimmed Milk -
Direct", "PowerBiofilm")),
           x = "method", y = "percent_recovery",
           fill = "method",
           palette = c("#9E9AC8", "#6A51A3"),
           line.color = "gray")+
  scale_x_discrete(labels = c("ViralRNA","PowerBiofilm")) +
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
```

```
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_text(hjust = .7, face = "bold"),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("")+
  labs(y = "OC43 % Recovery")+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 10))+
  annotate(geom = "text",
           x = 1.2,
           y = 35,
           label = "p < 0.001",
           size = 3.5)+
  annotate(geom = "text",
           x = 1.2,
           y = 30,
           label = "n = 18",
           size = 3.5)+
  scale_y_break(breaks = c(50, 98))
powerbiofilm_recovery

ggsave("powerbiofilm_recovery.png",
       plot = powerbiofilm_recovery, device = "png", dpi = 300,
limitsize = TRUE)

#Comparing Dilutions with Slope
data_bio$oc43_inhibit <- data_bio$oc43 - data_bio$dil_oc43
summary(data_bio$oc43_inhibit)

# Subset data
inhibition <-
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"), oc43_inhibit,
         drop = TRUE)
slope <-
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"), slope,
         drop = TRUE)
method <-
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"), method,
         drop = TRUE)
wwtp <-
```

```
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"), wwtp,
         drop = TRUE)

inhibit_comp <- data.frame(method, wwtp, inhibition, slope)

inhibit_comp$method <- factor(inhibit_comp$method,
                              levels = c("Skimmed Milk -
Direct", "PowerBiofilm"),
                              labels = c("ViralRNA",
"PowerBiofilm"))

qqnorm(inhibit_comp$inhibition[inhibit_comp$method ==
"PowerBiofilm"])
x <-  inhibit_comp$inhibition[inhibit_comp$method ==
"PowerBiofilm"]
x <- x[complete.cases(x)]
k <- 2

## Create a function to compute statistic to
## test for outliers in both tails.
tm = function(x,k){
  n = length(x)
  ## Compute the absolute residuals.
  r = abs(x - mean(x))
  ## Sort data according to size of residual.
  df = data.frame(x,r)
  dfs = df[order(df$r),]
  ## Create a subset of the data without the largest k values.
  klarge = c((n-k+1):n)
  subx = dfs$x[-klarge]
  ## Compute the sums of squares.
  ksub = (subx - mean(subx))**2
  all = (df$x - mean(df$x))**2
  ## Compute the test statistic.
  ek = sum(ksub)/sum(all)
}

## Call the function and compute value of test statistic for
data.
ekstat = tm(x,k)
ekstat

## Compute critical value based on simulation.
test = c(1:10000)
for (i in 1:10000){
xx = rnorm(length(x))
test[i] = tm(xx,k)}
```

```
quantile(test,0.05)

length(inhibit_comp$inhibition[inhibit_comp$inhibition<0])

inhibit_severe <- subset(inhibit_comp, inhibition > 0)
inhibit_normal <- subset(inhibit_comp, inhibition < 0)

length(inhibit_normal$inhibition[inhibit_normal$method=="ViralRN
A"])
length(inhibit_normal$inhibition[inhibit_normal$method=="PowerBi
ofilm"])

summary(inhibit_normal$inhibition[inhibit_normal$method=="ViralR
NA"])
summary(inhibit_normal$inhibition[inhibit_normal$method=="PowerB
iofilm"])

t.test(inhibit_normal$inhibition[inhibit_normal$method=="ViralRN
A"])
t.test(inhibit_normal$inhibition[inhibit_normal$method=="PowerBi
ofilm"])

summary(inhibit_normal$slope[inhibit_normal$method=="ViralRNA"])
summary(inhibit_normal$slope[inhibit_normal$method=="PowerBiofil
m"])

t.test(inhibit_normal$slope[inhibit_normal$method=="ViralRNA"])
t.test(inhibit_normal$slope[inhibit_normal$method=="PowerBiofilm
"])

#Test for normality
shapiro.test(
  with(
    inhibit_comp,
    inhibition[method == "PowerBiofilm"]
    - slope[method == "PowerBiofilm"]))

shapiro.test(
  with(
    inhibit_comp,
    inhibition[method == "ViralRNA"]
    - slope[method == "ViralRNA"]))

#Statistical Test
res <-
  wilcox.test(x =
inhibit_normal$inhibition[inhibit_normal$method ==
"PowerBiofilm"],
```

```
              y = inhibit_normal$slope[inhibit_normal$method ==
"PowerBiofilm"],
              paired = TRUE)
res

control <-
  wilcox.test(x =
inhibit_normal$inhibition[inhibit_normal$method == "ViralRNA"],
              y = inhibit_normal$slope[inhibit_normal$method ==
"ViralRNA"],
              paired = TRUE)
control

# A data frame with labels for each facet
test_labels <-
  data.frame(method = c("ViralRNA", "PowerBiofilm"),
             label = c("Wilcoxon signed-rank", "Wilcoxon signed-
rank"),
             pvalue = c("p < 0.0001", "p < 0.0001"),
             n = c("n = 18", "n = 15*"))

biofilm_pairs <-
  ggpaired(data = inhibit_normal,
           cond1 = "slope", cond2 = "inhibition",
           xlab = "PowerBiofilm",
           ylab = "",
           fill = c("#6A51A3", "#6A51A3", "#9E9AC8", "#9E9AC8"),
           line.color = "gray")+
  scale_x_discrete(labels = c("Slope","Difference"))+
  facet_wrap(~method)+
  geom_text(x = 1.2, y = -0.4, data = test_labels, aes(label =
pvalue))+
  geom_text(x = 1.2, y = -0.8, data = test_labels, aes(label =
n))+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"),
        strip.text.x = element_text(size = 8, face = "bold"))+
  scale_y_continuous(limits = c(-4, 0),
                     breaks = seq(-4,00, by = 1))+
```

```
  ggtitle("")
biofilm_pairs

biofilm_pairs_severe <-
  ggpaired(data = inhibit_severe,
           cond1 = "slope", cond2 = "inhibition",
           xlab = "PowerBiofilm",
           ylab = "",
           #fill = c("#6A51A3", "#6A51A3", "#9E9AC8",
"#9E9AC8"),
           line.color = "gray")+
  scale_x_discrete(labels = c("Slope","Difference"))+
  #facet_wrap(~method)+
  #geom_text(x = 1.2, y = -0.4, data = test_labels, aes(label =
pvalue))+
  #geom_text(x = 1.2, y = -0.8, data = test_labels, aes(label =
n))+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=12),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"),
        strip.text.x = element_text(size = 8, face = "bold"))+
  scale_y_continuous(limits = c(-4, 16),
                     breaks = seq(-4,16, by = 2))+
  ggtitle("PowerBiofilm")
biofilm_pairs_severe
```


#Stats for NucleoSpin
```{r, tidy = TRUE}
ww_data$ns_boolean <- 0
unique_dates <- unique(ww_data$sample_date)

#want all days with BOTH SM-D and NucleoSpin, nothing more
for(i in unique_dates){
  method_date <- ww_data$method[ww_data$sample_date==i]
  if(length(method_date[method_date == "NucleoSpin"]) >= 1){
    if(length(method_date[method_date == "Skimmed Milk -
Direct"]) >= 1){
      ww_data$ns_boolean[ww_data$sample_date==i] = 1}
```

```
    }
}

data_ns <- subset(ww_data, ns_boolean %in% c(1))

data_ns$method <- as.factor(data_ns$method)
data_ns$pres_pos <- as.factor(data_ns$pres_pos)
data_ns$pres_pos_dil <- as.factor(data_ns$pres_pos_dil)

tab_ns <- table(data_ns$pres_pos[data_ns$method=="NucleoSpin"],
                data_ns$pres_pos[data_ns$method=="Skimmed Milk -
Direct"],
                dnn = c("NucleoSpin", "ViralRNA"))
tab_ns

fisher.test(tab_ns)
mcnemar.test(tab_ns)

tab_ns_dil <-
table(data_ns$pres_pos_dil[data_ns$method=="NucleoSpin"],

data_ns$pres_pos_dil[data_ns$method=="Skimmed Milk - Direct"],
                    dnn = c("NucleoSpin", "ViralRNA"))
tab_ns_dil
#this isn't working because there aren't any 0's for
CeresNano...
#not sure how I can force a zero into this table
fisher.test(tab_ns_dil)
mcnemar.test(tab_ns_dil)

#Comparing OC43 Recovery

shapiro.test(
  with(
    data_ns,
    percent_recovery[method == "Skimmed Milk - Direct"] -
      percent_recovery[method == "NucleoSpin"]
  )
)

viral_rna <- subset(data_ns, method == "Skimmed Milk - Direct",
percent_recovery,
                    drop = TRUE)
print(summary(viral_rna))
t.test(viral_rna)

nucleospin <- subset(data_ns, method == "NucleoSpin",
percent_recovery,
```

```
                     drop = TRUE)
print(summary(nucleospin))
t.test(nucleospin)

res <- wilcox.test(viral_rna, nucleospin, paired = TRUE)
res

library(ggbreak)

nucleospin_recovery <-
  ggpaired(data = subset(data_ns, method %in% c("Skimmed Milk -
Direct", "NucleoSpin")),
           x = "method", y = "percent_recovery",
           fill = "method",
           palette = c("#9E9AC8", "#6A51A3"),
           line.color = "gray")+
  scale_x_discrete(labels = c("ViralRNA", "NucleoSpin"))+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_text(hjust = .7, face = "bold"),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("")+
  labs(y = "OC43 % Recovery")+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 10))+
  annotate(geom = "text",
           x = 1.2,
           y = 35,
           label = "p < 0.01",
           size = 3.5)+
  annotate(geom = "text",
           x = 1.2,
           y = 30,
           label = "n = 9",
           size = 3.5)+
  scale_y_break(breaks = c(50, 98))
nucleospin_recovery

ggsave("nucleospin_recovery.png",
```

```
       plot = nucleospin_recovery, device = "png", dpi = 300,
limitsize = TRUE)

#Comparing Dilutions with Slope
data_ns$oc43_inhibit <- data_ns$oc43 - data_ns$dil_oc43
summary(data_ns$oc43_inhibit)

# Subset data
inhibition <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"), oc43_inhibit,
         drop = TRUE)
slope <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"), slope,
         drop = TRUE)
method <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"), method,
         drop = TRUE)
wwtp <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"), wwtp,
         drop = TRUE)

inhibit_comp <- data.frame(method, wwtp, inhibition, slope)

inhibit_comp$method <- factor(inhibit_comp$method,
                              levels = c("Skimmed Milk -
Direct", "NucleoSpin"),
                              labels = c("ViralRNA",
"NucleoSpin"))
length(inhibit_comp$inhibition[inhibit_comp$inhibition>0])

qqnorm(inhibit_comp$inhibition[inhibit_comp$method ==
"NucleoSpin"])

#Test for normality
shapiro.test(
  with(
    inhibit_comp,
    inhibition[method == "NucleoSpin"]
    - slope[method == "NucleoSpin"]))

shapiro.test(
  with(
    inhibit_comp,
    inhibition[method == "ViralRNA"]
```

```
    - slope[method == "ViralRNA"]))

#Statistical Test
res <-
  wilcox.test(
    x = inhibit_comp$inhibition[inhibit_comp$method ==
"NucleoSpin"],
    y = inhibit_comp$slope[inhibit_comp$method == "NucleoSpin"],
    paired = TRUE)
res

control <-
  wilcox.test(
    x = inhibit_comp$inhibition[inhibit_comp$method ==
"ViralRNA"],
    y = inhibit_comp$slope[inhibit_comp$method == "ViralRNA"],
    paired = TRUE)
control

# A data frame with labels for each facet
test_labels <-
  data.frame(method = c("ViralRNA", "NucleoSpin"),
             label = c("Wilcoxon signed-rank", "Wilcoxon signed-
rank"),
             pvalue = c("p = 0.02", "p = 0.04"),
             n = c("n = 9"))

nucleospin_pairs <-
  ggpaired(data = inhibit_comp,
           cond1 = "slope", cond2 = "inhibition",
           ylab = "",
           fill = c("#6A51A3", "#6A51A3", "#9E9AC8", "#9E9AC8"),
           line.color = "gray")+
  scale_x_discrete(labels = c("Slope","Difference"))+
  facet_wrap(~method)+
  geom_text(x = 1.2, y = -0.4, data = test_labels, aes(label =
pvalue))+
  geom_text(x = 1.2, y = -0.8, data = test_labels, aes(label =
n))+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
```

```
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"),
        strip.text.x = element_text(size = 8, face = "bold"))+
  scale_y_continuous(limits = c(-4, 0),
                     breaks = seq(-4,00, by = 1))+
  ggtitle("")
nucleospin_pairs
```


#Stats for NucleoMag
```{r, tidy = TRUE}
ww_data$nm_boolean <- 0
unique_dates <- unique(ww_data$sample_date)

#want all days with BOTH SM-D and NucleoSpin, nothing more
for(i in unique_dates){
  method_date <- ww_data$method[ww_data$sample_date==i]
  if(length(method_date[method_date == "NucleoMag"]) >= 1){
    if(length(method_date[method_date == "Skimmed Milk -
Direct"]) >= 1){
      ww_data$nm_boolean[ww_data$sample_date==i] = 1}
    }
}

data_nm <- subset(ww_data, nm_boolean %in% c(1))

data_nm$method <- as.factor(data_nm$method)
data_nm$pres_pos <- as.factor(data_nm$pres_pos)
data_nm$pres_pos_dil <- as.factor(data_nm$pres_pos_dil)

tab_nm <- table(data_nm$pres_pos[data_nm$method=="NucleoMag"],
                data_nm$pres_pos[data_nm$method=="Skimmed Milk -
Direct"],
                dnn = c("NucleoMag", "ViralRNA"))
tab_nm

fisher.test(tab_nm)
mcnemar.test(tab_nm)

tab_nm_dil <-
table(data_nm$pres_pos_dil[data_nm$method=="NucleoMag"],

data_nm$pres_pos_dil[data_nm$method=="Skimmed Milk - Direct"],
                     dnn = c("NucleoMag", "ViralRNA"))
tab_nm_dil
#this isn't working because there aren't any 0's for
CeresNano...
```

```
#not sure how I can force a zero into this table
fisher.test(tab_nm_dil)
mcnemar.test(tab_nm_dil)

#Comparing OC43 Recovery

shapiro.test(
  with(
    data_nm,
    percent_recovery[method == "Skimmed Milk - Direct"] -
      percent_recovery[method == "NucleoMag"]
  )
)

viral_rna <- subset(data_nm, method == "Skimmed Milk - Direct",
percent_recovery,
                    drop = TRUE)
summary(viral_rna)
t.test(viral_rna)

nucleomag <- subset(data_nm, method == "NucleoMag",
percent_recovery,
                     drop = TRUE)
summary(nucleomag)
t.test(nucleomag)

res <- wilcox.test(viral_rna, nucleomag, paired = TRUE)
res

library(ggbreak)

nucleomag_recovery <-
  ggpaired(data = subset(data_nm, method %in% c("Skimmed Milk -
Direct", "NucleoMag")),
           x = "method", y = "percent_recovery",
           fill = "method",
           palette = c("#9E9AC8", "#6A51A3"),
           line.color = "gray")+
  scale_x_discrete(labels = c("ViralRNA", "NucleoMag")) +
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_text(hjust = .7, face = "bold"),
        legend.text = element_text(size = 8),
```

```
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("")+
  labs(y = "OC43 % Recovery")+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 10))+
  annotate(geom = "text",
           x = 1.2,
           y = 35,
           label = "p < 0.001",
           size = 3.5)+
  annotate(geom = "text",
           x = 1.2,
           y = 30,
           label = "n = 18",
           size = 3.5)+
  scale_y_break(breaks = c(50, 98))
nucleomag_recovery

ggsave("nucleomag_recovery.png",
       plot = nucleomag_recovery, device = "png", dpi = 300,
limitsize = TRUE)

#Comparing Dilutions with Slope
data_nm$oc43_inhibit <- data_nm$oc43 - data_nm$dil_oc43
summary(data_nm$oc43_inhibit)

# Subset data
inhibition <-
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"), oc43_inhibit,
         drop = TRUE)
slope <-
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"), slope,
         drop = TRUE)
method <-
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"), method,
         drop = TRUE)
wwtp <-
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"), wwtp,
         drop = TRUE)

inhibit_comp <- data.frame(method, wwtp, inhibition, slope)
```

```
inhibit_comp$method <- factor(inhibit_comp$method,
                              levels = c("Skimmed Milk -
Direct", "NucleoMag"),
                              labels = c("ViralRNA",
"NucleoMag"))

qqnorm(inhibit_comp$inhibition[inhibit_comp$method ==
"NucleoMag"])
x <-  inhibit_comp$inhibition[inhibit_comp$method ==
"NucleoMag"]
x <- x[complete.cases(x)]
k <-  3

## Create a function to compute statistic to
## test for outliers in both tails.
tm = function(x,k){

n = length(x)

## Compute the absolute residuals.
r = abs(x - mean(x))

## Sort data according to size of residual.
df = data.frame(x,r)
dfs = df[order(df$r),]

## Create a subset of the data without the largest k values.
klarge = c((n-k+1):n)
subx = dfs$x[-klarge]

## Compute the sums of squares.
ksub = (subx - mean(subx))**2
all = (df$x - mean(df$x))**2

## Compute the test statistic.
ek = sum(ksub)/sum(all)
}

## Call the function and compute value of test statistic for
data.
ekstat = tm(x,k)
ekstat

## Compute critical value based on simulation.
test = c(1:10000)
for (i in 1:10000){
xx = rnorm(length(x))
```

```
test[i] = tm(xx,k)}
quantile(test,0.05)

length(inhibit_comp$inhibition[inhibit_comp$inhibition<0])
length(inhibit_comp$inhibition[inhibit_comp$inhibition>0])

inhibit_severe <- subset(inhibit_comp, inhibition > 0)
inhibit_normal <- subset(inhibit_comp, inhibition < 0)

length(inhibit_normal$inhibition[inhibit_normal$method=="ViralRN
A"])
length(inhibit_normal$inhibition[inhibit_normal$method=="NucleoM
ag"])

summary(inhibit_normal$inhibition[inhibit_normal$method=="ViralR
NA"])
summary(inhibit_normal$inhibition[inhibit_normal$method=="Nucleo
Mag"])

t.test(inhibit_normal$inhibition[inhibit_normal$method=="ViralRN
A"])
t.test(inhibit_normal$inhibition[inhibit_normal$method=="NucleoM
ag"])

summary(inhibit_normal$slope[inhibit_normal$method=="ViralRNA"])
summary(inhibit_normal$slope[inhibit_normal$method=="NucleoMag"]
)

t.test(inhibit_normal$slope[inhibit_normal$method=="ViralRNA"])
t.test(inhibit_normal$slope[inhibit_normal$method=="NucleoMag"])

#Statistical Test
res <-
  wilcox.test(
    x = inhibit_normal$inhibition[inhibit_normal$method ==
"NucleoMag"],
    y = inhibit_normal$slope[inhibit_normal$method ==
"NucleoMag"],
    paired = TRUE)
res

control <-
  wilcox.test(
    x = inhibit_normal$inhibition[inhibit_normal$method ==
"ViralRNA"],
    y = inhibit_normal$slope[inhibit_normal$method ==
"ViralRNA"],
    paired = TRUE)
```

```
control

# A data frame with labels for each facet
test_labels <-
  data.frame(method = c("ViralRNA", "NucleoMag"),
             label = c("Wilcoxon signed-rank", "Wilcoxon signed-
rank"),
             pvalue = c("p = 0.06", "p = 0.76"),
             n = c("n = 17", "n = 14*"))

nucleomag_pairs <-
  ggpaired(data = inhibit_normal,
           cond1 = "slope", cond2 = "inhibition",
           ylab = "",
           fill = c("#6A51A3", "#6A51A3", "#9E9AC8", "#9E9AC8"),
           line.color = "gray")+
  scale_x_discrete(labels = c("Slope","Difference"))+
  facet_wrap(~method)+
  geom_text(x = 1.2, y = -0.4, data = test_labels, aes(label =
pvalue))+
  geom_text(x = 1.2, y = -0.8, data = test_labels, aes(label =
n))+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"),
        strip.text.x = element_text(size = 8, face = "bold"))+
  scale_y_continuous(limits = c(-4, 0),
                     breaks = seq(-4,00, by = 1))+
  ggtitle("")
nucleomag_pairs

nucleomag_pairs_severe <-
  ggpaired(data = inhibit_severe,
           cond1 = "slope", cond2 = "inhibition",
           ylab = "",
           fill = c("#9E9AC8", "#9E9AC8"),
           line.color = "gray")+
  scale_x_discrete(labels = c("Slope","Difference"))+
  #facet_wrap(~method)+
```

```
  #geom_text(x = 1.2, y = -0.4, data = test_labels, aes(label =
pvalue))+
  #geom_text(x = 1.2, y = -0.8, data = test_labels, aes(label =
n))+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=12),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"),
        strip.text.x = element_text(size = 8, face = "bold"))+
  scale_y_continuous(limits = c(-4, 16),
                     breaks = seq(-4,16, by = 2))+
  ggtitle("NucleoMag")
nucleomag_pairs_severe
```

#Cq Difference Figure
```{r, tidy = TRUE}
library(ggpubr)
inhibition_fig <-
  ggarrange(
    biofilm_pairs, nucleospin_pairs, nucleomag_pairs,
    nrow = 3, ncol = 1,
    legend = "none",
    labels = "auto",
    font.label = list(size = 12)
  )
inhibition_fig
ggsave(plot = inhibition_fig, filename = "inhibition_fig.tiff",
       units = 'in', width = 5.4, height = 9, dpi = 600)

inhibition_fig_severe <-
  ggarrange(
    print(biofilm_pairs_severe), print(nucleomag_pairs_severe),
    nrow = 1, ncol = 2,
    legend = "none",
    labels = "auto",
    font.label = list(size = 12)
  )
inhibition_fig_severe
ggsave(plot = inhibition_fig_severe, filename =
"inhibition_fig_severe.tiff",
```

```
       units = 'in', width = 7, height = 2.9, dpi = 600)
```

#Recovery Comparisons Figure
```{r, tidy = TRUE}
library(ggpubr)

#need to add print to the ggarrange plots or I lose the axis
breaks
ext_kit_recovery <- ggarrange(print(powerbiofilm_recovery),
                              print(nucleospin_recovery),
                              print(nucleomag_recovery),
                              nrow = 1, ncol = 3,
                              legend = "none",
                              labels = "auto",
                              font.label = list(size = 12))
ext_kit_recovery

ggsave(ext_kit_recovery, filename = "ext_kit_recovery.tiff",
       units = 'in', width = 7, height = 2.9, dpi = 600)
```

#Extraction Kit OC43 Recovery
```{r, tidy = TRUE}

ext_data <- subset(ww_data, method %in% c("Skimmed Milk -
Direct",
                                          "NucleoSpin",
                                          "PowerBiofilm",
                                          "NucleoMag"))

ext_data$method <- factor(ext_data$method,
                          levels = c("Skimmed Milk - Direct",
                                     "NucleoSpin",
                                     "PowerBiofilm",
                                     "NucleoMag"),
                          labels = c("ViralRNA",
                                     "NucleoSpin",
                                     "PowerBiofilm",
                                     "NucleoMag"))

ext_data$method <- as.factor(ext_data$method)

ext_data <- subset(ext_data, exp_date > "2021-07-11" | exp_date
< "2021-04-28")
ext_data <- subset(ext_data, exp_date > "2021-02-10")
```

```
library(ggplot2)

my_comparisons <- list(c("ViralRNA", "NucleoSpin"),
                       c("ViralRNA", "PowerBiofilm"),
                       c("ViralRNA", "NucleoMag"),
                       c("NucleoSpin", "PowerBiofilm"),
                       c("NucleoSpin", "NucleoMag"),
                       c("PowerBiofilm", "NucleoMag"))

library(RColorBrewer)
brewer.pal(n = 9, "Purples")

ext_recovery <-
  ggplot(data = ext_data, aes(x = method, y = percent_recovery,
fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=14),
        axis.title = element_text(size=14),
        axis.text = element_text(size=14),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12),
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("OC43 Recovery")+
  labs(y = "Recovery")+
  scale_fill_manual(values = c("#A9A9A9", "#9E9AC8", "#6A51A3",
"#3F007D"),
                    name = "Method")+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 20))
ext_recovery

ggsave("ext_recovery.png", plot = ext_recovery, device = "png",
dpi = 300, limitsize = TRUE)

ext_vol_assayed <-
  ggplot(data = ext_data, aes(x = method, y = effective_vol,
color = method, fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=12),
        axis.title = element_text(size=12),
        axis.text = element_text(size=12),
        axis.text.x = element_text(angle = 45, hjust=1),
```

```
        axis.title.x = element_blank(),
        legend.text = element_text(size = 10),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("Effective Volume Assayed")+
  labs(y = "Volume (mL)")+
  scale_fill_manual(values = c("#A9A9A9", "#9E9AC8", "#6A51A3",
"#3F007D"),
                    name = "Method")+
  scale_color_manual(values = c("#A9A9A9", "#9E9AC8", "#6A51A3",
"#3F007D"),
                    name = "Method")+
  scale_y_continuous(limits = c(0, 7),
                     breaks = seq(0, 7, by = 1))
ext_vol_assayed

ggsave(plot = ext_vol_assayed, filename =
"ext_vol_assayed.tiff",
       units = 'in', width = 3.33, height = 2.5, dpi = 600)

ext_vol_percent <- ggarrange(ext_vol_assayed, ext_recovery,
                             nrow = 1,
                             ncol = 2,
                             common.legend = TRUE,
                             legend = "none",
                             labels = "auto")
ext_vol_percent
ggsave(ext_vol_percent, filename = "ext_vol_percent.tiff",
       units = 'in', width = 6.8, height = 3.7, dpi = 600)
```

#Extraction Comparison Control Charts
```{r, tidy = TRUE}

#ViralRNA
average <- mean(ext_data$oc43[ext_data$method=="ViralRNA"],
na.rm=TRUE)
stdev <- sd(ext_data$oc43[ext_data$method=="ViralRNA"],
na.rm=TRUE)
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)
```

```
viralrna_shewart <-
  ggplot(data = subset(x = ext_data, method=="ViralRNA"),

         aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(5, 40)+
  scale_x_date(date_labels = "%d %b '%y",
               date_breaks = "1 month")+
  ggtitle("ViralRNA")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9),
        axis.text.x = element_text(angle = 45, hjust=1))
viralrna_shewart

#Power Biofilm
average <- mean(ext_data$oc43[ext_data$method=="PowerBiofilm"],
na.rm=TRUE)
stdev <- sd(ext_data$oc43[ext_data$method=="PowerBiofilm"],
na.rm=TRUE)
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev
```

```
lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

powerbiofilm_shewart <-
  ggplot(data = subset(x = ext_data, method=="PowerBiofilm"),
         aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(5, 40)+
  scale_x_date(date_labels = "%d %b '%y",
               date_breaks = "1 week")+
  ggtitle("PowerBiofilm")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9),
        axis.text.x = element_text(angle = 45, hjust=1))
powerbiofilm_shewart

#NucleoSpin
average <- mean(ext_data$oc43[ext_data$method=="NucleoSpin"],
na.rm=TRUE)
stdev <- sd(ext_data$oc43[ext_data$method=="NucleoSpin"],
na.rm=TRUE)
```

```
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

nucleospin_shewart <-
  ggplot(data = subset(x = ext_data, method=="NucleoSpin"),
         aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(5, 40)+
  scale_x_date(date_labels = "%d %b '%y",
               date_breaks = "1 week")+
  ggtitle("NucleoSpin")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9),
        axis.text.x = element_text(angle = 45, hjust=1))
nucleospin_shewart

#NucleoMag
```

```
average <- mean(ext_data$oc43[ext_data$method=="NucleoMag"],
na.rm=TRUE)
stdev <- sd(ext_data$oc43[ext_data$method=="NucleoMag"],
na.rm=TRUE)
lcl <- average-3*stdev
ucl <- average+3*stdev
lwl <- average-stdev
uwl <- average+stdev

lines <- c("1", "1", "2", "2", "3", "3", "4", "4", "5", "5")
y <- c(ucl, ucl, uwl, uwl, average, average, lwl, lwl, lcl, lcl)
shewartlines <- data.frame(lines, x = c(-Inf, Inf), y)
shewartlines$lines <- as.factor(shewartlines$lines)

nucleomag_shewart <-
  ggplot(data = subset(x = ext_data, method=="NucleoMag"),
         aes(x = sample_date, y = oc43)) +
  geom_hline(data = shewartlines,
             aes(yintercept = y, linetype = lines),
             size = 1)+
  scale_linetype_manual(values = c("dotted", "twodash", "solid",
"twodash",
                                   "dotted"),
                        labels = c("UCL", "UWL", "Average",
"LWL", "LCL"),
                        name = "Lines")+
  geom_point(aes(shape = wwtp, fill = wwtp), size=3)+
  scale_shape_manual(values = c(21, 25, 22),
                     name = "WWTP",
                     labels = c("Plant A", "Plant B", "Plant
C"))+
  scale_fill_manual(values = c("#b2182b", "#92c5de", "#053061"),
                    name = "WWTP",
                    labels = c("Plant A", "Plant B", "Plant
C"))+
  #annotate("rect",xmin=-Inf, xmax=Inf, ymin=40, ymax=45,
alpha=0.4, fill="red")+
  ylim(5, 40)+
  scale_x_date(date_labels = "%d %b '%y",
               date_breaks = "1 week")+
  ggtitle("NucleoMag")+
  labs(y="OC43 Cq Value", x="Sample Date")+
  theme(legend.position="right",
        plot.title=element_text(face="bold", hjust=0.5,
size=12),
        axis.title=element_text(size=11),
        legend.text=element_text(size=9),
        axis.text=element_text(size=9),
```

```
        axis.text.x = element_text(angle = 45, hjust=1))
nucleomag_shewart

library(ggplot2)
library("ggpubr")

ext_control <-
  ggarrange(viralrna_shewart, powerbiofilm_shewart,
nucleomag_shewart, nucleospin_shewart,
            ncol = 2,
            nrow = 2,
            common.legend = TRUE,
            legend = "right",
            labels = "AUTO")
ext_control

ggsave(plot = ext_control, filename = "ext_control.tiff", units
= 'in',
       width = 10, height = 6, dpi = 600)
```

#Descriptive data for Slopes, Inhibition
```{r, tidy = TRUE}

#Differences between Undiluted and Diluted OC43

ext_data$oc43_inhibit <- ext_data$oc43 - ext_data$dil_oc43
ext_data$oc43_dif <- ext_data$slope - ext_data$oc43_inhibit

summary(ext_data$slope)
t.test(ext_data$slope)

#ViralRNA
summary(ext_data$oc43_inhibit[ext_data$method=="ViralRNA"])
t.test(ext_data$oc43_inhibit[ext_data$method=="ViralRNA"])
length(ext_data$oc43_inhibit[ext_data$method=="ViralRNA"])

#PowerBiofilm
summary(data_bio$oc43_inhibit[data_bio$method=="PowerBiofilm"])
t.test(data_bio$oc43_inhibit[data_bio$method=="PowerBiofilm"])
length(data_bio$oc43_inhibit[data_bio$method=="PowerBiofilm"])

summary(data_bio$oc43_inhibit[data_bio$method=="Skimmed Milk -
Direct"])
t.test(data_bio$oc43_inhibit[data_bio$method=="Skimmed Milk -
Direct"])
length(data_bio$oc43_inhibit[data_bio$method=="Skimmed Milk -
Direct"])
```

```
median(data_bio$slope)
t.test(data_bio$slope)

#NucleoSpin
summary(data_ns$oc43_inhibit[data_ns$method=="NucleoSpin"])
t.test(data_ns$oc43_inhibit[data_ns$method=="NucleoSpin"])
length(data_ns$oc43_inhibit[data_ns$method=="NucleoSpin"])

summary(data_ns$oc43_inhibit[data_ns$method=="Skimmed Milk -
Direct"])
t.test(data_ns$oc43_inhibit[data_ns$method=="Skimmed Milk -
Direct"])
length(data_ns$oc43_inhibit[data_ns$method=="Skimmed Milk -
Direct"])

median(data_ns$slope)
t.test(data_ns$slope)

#NucleoMag
summary(data_nm$oc43_inhibit[data_nm$method=="NucleoMag"])
t.test(data_nm$oc43_inhibit[data_nm$method=="NucleoMag"])
length(data_nm$oc43_inhibit[data_nm$method=="NucleoMag"])

summary(data_nm$oc43_inhibit[data_nm$method=="Skimmed Milk -
Direct"])
t.test(data_nm$oc43_inhibit[data_nm$method=="Skimmed Milk -
Direct"])
length(data_nm$oc43_inhibit[data_nm$method=="Skimmed Milk -
Direct"])

median(data_nm$slope)
t.test(data_nm$slope)
```

#SARS Balloon Plots Extraction Comparison
```{r, tidy = TRUE}

size <- as.vector(c(length(ext_data$method[ext_data$method ==
"ViralRNA"]),
                    length(ext_data$method[ext_data$method ==
"PowerBiofilm"]),
                    length(ext_data$method[ext_data$method ==
"NucleoMag"]),
                    length(ext_data$method[ext_data$method ==
"NucleoSpin"])))
```

```
sars_pos <- as.vector(c((sum(ext_data$pres_pos[ext_data$method
== "ViralRNA"]) /
                  length(ext_data$pres_pos[ext_data$method ==
"ViralRNA"]))*100,
                (sum(ext_data$pres_pos[ext_data$method ==
"PowerBiofilm"]) /
                  length(ext_data$pres_pos[ext_data$method ==
"PowerBiofilm"]))*100,
                (sum(ext_data$pres_pos[ext_data$method ==
"NucleoMag"]) /
                  length(ext_data$pres_pos[ext_data$method ==
"NucleoMag"]))*100,
                (sum(ext_data$pres_pos[ext_data$method ==
"NucleoSpin"]) /
                  length(ext_data$pres_pos[ext_data$method ==
"NucleoSpin"]))*100))

sars_pos_dil <-
  as.vector(c((sum(ext_data$pres_pos_dil[ext_data$method ==
"ViralRNA"]) /
                  length(ext_data$pres_pos_dil[ext_data$method ==
"ViralRNA"]))*100,
                (sum(ext_data$pres_pos_dil[ext_data$method ==
"PowerBiofilm"]) /
                  length(ext_data$pres_pos_dil[ext_data$method ==
"PowerBiofilm"]))*100,
                (sum(ext_data$pres_pos_dil[ext_data$method ==
"NucleoMag"]) /
                  length(ext_data$pres_pos_dil[ext_data$method ==
"NucleoMag"]))*100,
                (sum(ext_data$pres_pos_dil[ext_data$method ==
"NucleoSpin"]) /
                  length(ext_data$pres_pos_dil[ext_data$method ==
"NucleoSpin"]))*100))

n1_pos <- as.vector(c((sum(ext_data$n1_dummy[ext_data$method ==
"ViralRNA"]) /
                  length(ext_data$n1_dummy[ext_data$method ==
"ViralRNA"]))*100,
                (sum(ext_data$n1_dummy[ext_data$method ==
"PowerBiofilm"]) /
                  length(ext_data$n1_dummy[ext_data$method ==
"PowerBiofilm"]))*100,
                (sum(ext_data$n1_dummy[ext_data$method ==
"NucleoMag"]) /
                  length(ext_data$n1_dummy[ext_data$method ==
"NucleoMag"]))*100,
```

```
              (sum(ext_data$n1_dummy[ext_data$method ==
"NucleoSpin"]) /
                 length(ext_data$n1_dummy[ext_data$method ==
"NucleoSpin"]))*100))

n1_pos_dil <-
  as.vector(c((sum(ext_data$n1_dil_dummy[ext_data$method ==
"ViralRNA"]) /
                 length(ext_data$n1_dil_dummy[ext_data$method ==
"ViralRNA"]))*100,
              (sum(ext_data$n1_dil_dummy[ext_data$method ==
"PowerBiofilm"]) /
                 length(ext_data$n1_dil_dummy[ext_data$method ==
"PowerBiofilm"]))*100,
              (sum(ext_data$n1_dil_dummy[ext_data$method ==
"NucleoMag"]) /
                 length(ext_data$n1_dil_dummy[ext_data$method ==
"NucleoMag"]))*100,
              (sum(ext_data$n1_dil_dummy[ext_data$method ==
"NucleoSpin"]) /
                 length(ext_data$n1_dil_dummy[ext_data$method ==
"NucleoSpin"]))*100))

n2_pos <- as.vector(c((sum(ext_data$n2_dummy[ext_data$method ==
"ViralRNA"]) /
                 length(ext_data$n2_dummy[ext_data$method ==
"ViralRNA"]))*100,
              (sum(ext_data$n2_dummy[ext_data$method ==
"PowerBiofilm"]) /
                 length(ext_data$n2_dummy[ext_data$method ==
"PowerBiofilm"]))*100,
              (sum(ext_data$n2_dummy[ext_data$method ==
"NucleoMag"]) /
                 length(ext_data$n2_dummy[ext_data$method ==
"NucleoMag"]))*100,
              (sum(ext_data$n2_dummy[ext_data$method ==
"NucleoSpin"]) /
                 length(ext_data$n2_dummy[ext_data$method ==
"NucleoSpin"]))*100))

n2_pos_dil <-
  as.vector(c((sum(ext_data$n2_dil_dummy[ext_data$method ==
"ViralRNA"]) /
                 length(ext_data$n2_dil_dummy[ext_data$method ==
"ViralRNA"]))*100,
              (sum(ext_data$n2_dil_dummy[ext_data$method ==
"PowerBiofilm"]) /
```

```
                length(ext_data$n2_dil_dummy[ext_data$method ==
"PowerBiofilm"]))*100,
              (sum(ext_data$n2_dil_dummy[ext_data$method ==
"NucleoMag"]) /
                length(ext_data$n2_dil_dummy[ext_data$method ==
"NucleoMag"]))*100,
              (sum(ext_data$n2_dil_dummy[ext_data$method ==
"NucleoSpin"]) /
                length(ext_data$n2_dil_dummy[ext_data$method ==
"NucleoSpin"]))*100))

positive <- as.vector(c(sars_pos, n1_pos, n2_pos))

positive_dil <- as.vector(c(sars_pos_dil, n1_pos_dil,
n2_pos_dil))

assays <- as.factor(c("SARS", "SARS", "SARS", "SARS",
                      "N1", "N1", "N1", "N1",
                      "N2", "N2", "N2", "N2"))

extraction <- as.factor(c("ViralRNA", "PowerBiofilm",
"NucleoMag", "NucleoSpin"))

extraction_balloon <- cbind.data.frame(extraction, assays, size,
positive)
#need to have the two variables that are serving as x and Y for
balloon plots next to each other

library(ggpubr)
extballoonplot <-
  ggballoonplot(data = extraction_balloon,
                aes(x = assays,
                    y = method),
                size = "size",
                fill = "positive")+
  geom_text(data = extraction_balloon, nudge_x = 0.4,
            aes(label = round(x = positive,
                              digits = 1)),
            size = 3)+
  scale_size_continuous(range = c(5, 10),
                        name = "Sample Size")+
  scale_fill_gradient2(low = "#EBEBEB",
                       mid = "#D9CEEE",
                       high = "#4B2E83",
                       midpoint = 55,
                       name = "Percent Positive",
                       limits = c(0, 100))+
  ggtitle("Non-Diluted")+
```

```
  guides()+
  theme(legend.position = "right",
        plot.title = element_text(#face = "bold",
                                  hjust = 0.5,
                                  size = 10),
        axis.title = element_text(size = 10),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 10, face = "bold"),
        axis.text.x = element_text(size = 10, color = "black",
face = "bold",
                                   angle = 30),
        axis.text.y = element_text(size = 10, color = "black",
face = "bold"))
extballoonplot

extraction_balloon_dil <- cbind.data.frame(extraction, assays,
size, positive_dil)

extballoonplot_dil <-
  ggballoonplot(data = extraction_balloon_dil,
                aes(x = assays,
                    y = method),
                size = "size",
                fill = "positive_dil")+
  geom_text(data = extraction_balloon_dil, nudge_x = 0.4,
            aes(label = round(x = positive_dil,
                              digits = 1)),
            size = 3)+
  scale_size_continuous(range = c(5, 10),
                        name = "Sample Size")+
  scale_fill_gradient2(low = "#EBEBEB",
                       mid = "#D9CEEE",
                       high = "#4B2E83",
                       midpoint = 55,
                       name = "Percent Positive",
                       limits = c(0, 100))+
  ggtitle("Dilution" ~10^-1)+
  guides()+
  theme(legend.position = "right",
        plot.title = element_text(#face = "bold",
                                  hjust = 0.5,
                                  size = 10),
        axis.title = element_text(size = 10),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 10, face = "bold"),
        axis.text.x = element_text(size = 10, color = "black",
face = "bold",
                                   angle = 30),
```

```
        axis.text.y = element_text(size = 10, color = "black",
face = "bold"))
extballoonplot_dil

library("ggpubr")
ext_balloon_plots <- ggarrange(extballoonplot,
extballoonplot_dil,
                               ncol = 2,
                               nrow = 1,
                               common.legend = TRUE,
                               legend = "bottom",
                               labels = "auto",
                               font.label = list(size = 12))
ext_balloon_plots
#annotate_balloon <- annotate_figure(balloon_plots,
 #                                  top = text_grob("Percent
Positivity for SARS-CoV-2",
  #                                                  face =
"bold", size = 24))
#annotate_balloon
ggsave(ext_balloon_plots, filename = "ext_balloon_plots.tiff",
       units = 'in', width = 7, height = 3.7, dpi = 600)
```

#Comparing Ratios of SARS diluted vs undiluted for PowerBiofilm Comparisons

```{r, tidy = TRUE}

data_bio$uniqueID <- c(1:length(data_bio$record_id))
uniqueID <- data_bio$uniqueID

data_bio$n1_actual <- NA
for(i in uniqueID){
  if(data_bio$n1_dil_dummy[data_bio$uniqueID == i] == 1){
    if(data_bio$n1_dummy[data_bio$uniqueID == i] == 0){
      data_bio$n1_actual[data_bio$uniqueID == i] <- 45
    } else {
      data_bio$n1_actual[data_bio$uniqueID == i] <-
data_bio$n1[data_bio$uniqueID == i]
    }
  }
}

data_bio$n1_expected <- NA
for(i in uniqueID){
  if(data_bio$n1_dil_dummy[data_bio$uniqueID == i] == 1){
    data_bio$n1_expected[data_bio$uniqueID == i] <-
```

```
      data_bio$dil_n1[data_bio$uniqueID == i] +
data_bio$n1_slope[data_bio$uniqueID == i]
  }
}

data_bio$n1_ideal <- NA
for(i in uniqueID){
  if(data_bio$n1_dil_dummy[data_bio$uniqueID == i] == 1){
    data_bio$n1_ideal[data_bio$uniqueID == i] <-
      data_bio$dil_n1[data_bio$uniqueID == i] - 3.3
  }
}

data_bio$n1_inhibit <-
  data_bio$n1_actual - data_bio$n1_expected

data_bio$inhibit_severe <- NA
for(i in uniqueID){
  if(!is.na(data_bio$n1_inhibit[data_bio$uniqueID == i])){
    if(data_bio$n1_inhibit[data_bio$uniqueID == i] > 1){
      data_bio$inhibit_severe[data_bio$uniqueID == i] <- 1
    } else {
      data_bio$inhibit_severe[data_bio$uniqueID == i] <- 0
    }
  }
}

tab_bio_inbibit <-
table(data_bio$inhibit_severe[data_bio$method ==
"PowerBiofilm"],

data_bio$inhibit_severe[data_bio$method == "Skimmed Milk -
Direct"],
                          dnn = c("PowerBiofilm", "ViralRNA"))
tab_bio_inbibit

# Subset data
n1_inhibition <-
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"),
         n1_inhibit,
         drop = TRUE)
method <-
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"),
         method,
         drop = TRUE)
wwtp <-
```

```
  subset(data_bio, method %in% c("PowerBiofilm", "Skimmed Milk -
Direct"),
         wwtp,
         drop = TRUE)

n1_inhibit_comp <- data.frame(method, wwtp, n1_inhibition)

n1_inhibit_comp$method <-
  factor(n1_inhibit_comp$method,
         levels = c("Skimmed Milk - Direct", "PowerBiofilm"),
         labels = c("ViralRNA", "PowerBiofilm"))

bio_inhibit <- subset(n1_inhibit_comp, method %in%
c("PowerBiofilm"))
viralrna_inhibit <- subset(n1_inhibit_comp, method %in%
c("ViralRNA"))

severe <-
  length(
    bio_inhibit$n1_inhibition
    [bio_inhibit$n1_inhibition > 1 &
!is.na(bio_inhibit$n1_inhibition)])

normal <-
  length(
    bio_inhibit$n1_inhibition
    [bio_inhibit$n1_inhibition < 1 &
!is.na(bio_inhibit$n1_inhibition)])

inhibit_ratio <- severe/normal
inhibit_ratio

severe_viralrna <-
  length(
    viralrna_inhibit$n1_inhibition
    [viralrna_inhibit$n1_inhibition > 1 &
!is.na(viralrna_inhibit$n1_inhibition)])

normal_viralrna <-
  length(
    viralrna_inhibit$n1_inhibition
    [viralrna_inhibit$n1_inhibition < 1 &
!is.na(viralrna_inhibit$n1_inhibition)])

inhibit_ratio_viralrna <- severe_viralrna/normal_viralrna
inhibit_ratio_viralrna
```

```
#Comparing Ratios of SARS diluted vs undiluted for NucleoSpin
Comparisons
```{r, tidy = TRUE}
data_ns$uniqueID <- c(1:length(data_ns$record_id))
uniqueID <- data_ns$uniqueID

data_ns$n1_actual <- NA
for(i in uniqueID){
  if(data_ns$n1_dil_dummy[data_ns$uniqueID == i] == 1){
    if(data_ns$n1_dummy[data_ns$uniqueID == i] == 0){
      data_ns$n1_actual[data_ns$uniqueID == i] <- 45
    } else {
      data_ns$n1_actual[data_ns$uniqueID == i] <-
data_ns$n1[data_ns$uniqueID == i]
    }
  }
}

data_ns$n1_expected <- NA
for(i in uniqueID){
  if(data_ns$n1_dil_dummy[data_ns$uniqueID == i] == 1){
    data_ns$n1_expected[data_ns$uniqueID == i] <-
      data_ns$dil_n1[data_ns$uniqueID == i] +
data_ns$n1_slope[data_ns$uniqueID == i]
  }
}

data_ns$n1_ideal <- NA
for(i in uniqueID){
  if(data_ns$n1_dil_dummy[data_ns$uniqueID == i] == 1){
    data_ns$n1_ideal[data_ns$uniqueID == i] <-
      data_ns$dil_n1[data_ns$uniqueID == i] - 3.3
  }
}

data_ns$n1_inhibit <-
  data_ns$n1_actual - data_ns$n1_expected

# Subset data
n1_inhibition <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"),
         n1_inhibit,
         drop = TRUE)
method <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"),
         method,
```
```

```
         drop = TRUE)
wwtp <-
  subset(data_ns, method %in% c("NucleoSpin", "Skimmed Milk -
Direct"),
         wwtp,
         drop = TRUE)

n1_inhibit_comp <- data.frame(method, wwtp, n1_inhibition)

n1_inhibit_comp$method <-
  factor(n1_inhibit_comp$method,
         levels = c("Skimmed Milk - Direct", "NucleoSpin"),
         labels = c("ViralRNA", "NucleoSpin"))

ns_inhibit <- subset(n1_inhibit_comp, method %in%
c("NucleoSpin"))
viralrna_inhibit <- subset(n1_inhibit_comp, method %in%
c("ViralRNA"))

severe <-
  length(
    ns_inhibit$n1_inhibition
    [ns_inhibit$n1_inhibition > 1 &
!is.na(ns_inhibit$n1_inhibition)])

normal <-
  length(
    ns_inhibit$n1_inhibition
    [ns_inhibit$n1_inhibition < 1 &
!is.na(ns_inhibit$n1_inhibition)])

inhibit_ratio <- severe/normal
inhibit_ratio

severe_viralrna <-
  length(
    viralrna_inhibit$n1_inhibition
    [viralrna_inhibit$n1_inhibition > 1 &
!is.na(viralrna_inhibit$n1_inhibition)])

normal_viralrna <-
  length(
    viralrna_inhibit$n1_inhibition
    [viralrna_inhibit$n1_inhibition < 1 &
!is.na(viralrna_inhibit$n1_inhibition)])

inhibit_ratio_viralrna <- severe_viralrna/normal_viralrna
inhibit_ratio_viralrna
```

```

#Comparing Ratios of SARS diluted vs undiluted for NucleoMag
Comparisons
```{r, tidy = TRUE}

data_nm$uniqueID <- c(1:length(data_nm$record_id))
uniqueID <- data_nm$uniqueID

data_nm$n1_actual <- NA
for(i in uniqueID){
  if(data_nm$n1_dil_dummy[data_nm$uniqueID == i] == 1){
    if(data_nm$n1_dummy[data_nm$uniqueID == i] == 0){
      data_nm$n1_actual[data_nm$uniqueID == i] <- 45
    } else {
      data_nm$n1_actual[data_nm$uniqueID == i] <-
data_nm$n1[data_nm$uniqueID == i]
    }
  }
}

data_nm$n1_expected <- NA
for(i in uniqueID){
  if(data_nm$n1_dil_dummy[data_nm$uniqueID == i] == 1){
    data_nm$n1_expected[data_nm$uniqueID == i] <-
      data_nm$dil_n1[data_nm$uniqueID == i] +
data_nm$n1_slope[data_nm$uniqueID == i]
  }
}

data_nm$n1_ideal <- NA
for(i in uniqueID){
  if(data_nm$n1_dil_dummy[data_nm$uniqueID == i] == 1){
    data_nm$n1_ideal[data_nm$uniqueID == i] <-
      data_nm$dil_n1[data_nm$uniqueID == i] - 3.3
  }
}

data_nm$n1_inhibit <-
  data_nm$n1_actual - data_nm$n1_expected

# Subset data
n1_inhibition <-
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"),
         n1_inhibit,
         drop = TRUE)
method <-
```
```

```
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"),
         method,
         drop = TRUE)
wwtp <-
  subset(data_nm, method %in% c("NucleoMag", "Skimmed Milk -
Direct"),
         wwtp,
         drop = TRUE)

n1_inhibit_comp <- data.frame(method, wwtp, n1_inhibition)

n1_inhibit_comp$method <-
  factor(n1_inhibit_comp$method,
         levels = c("Skimmed Milk - Direct", "NucleoMag"),
         labels = c("ViralRNA", "NucleoMag"))

ns_inhibit <- subset(n1_inhibit_comp, method %in%
c("NucleoMag"))
viralrna_inhibit <- subset(n1_inhibit_comp, method %in%
c("ViralRNA"))

severe <-
  length(
    ns_inhibit$n1_inhibition
    [ns_inhibit$n1_inhibition > 1 &
!is.na(ns_inhibit$n1_inhibition)])

normal <-
  length(
    ns_inhibit$n1_inhibition
    [ns_inhibit$n1_inhibition < 1 &
!is.na(ns_inhibit$n1_inhibition)])

inhibit_ratio <- severe/normal
inhibit_ratio

severe_viralrna <-
  length(
    viralrna_inhibit$n1_inhibition
    [viralrna_inhibit$n1_inhibition > 1 &
!is.na(viralrna_inhibit$n1_inhibition)])

normal_viralrna <-
  length(
    viralrna_inhibit$n1_inhibition
    [viralrna_inhibit$n1_inhibition < 1 &
!is.na(viralrna_inhibit$n1_inhibition)])
```

```
inhibit_ratio_viralrna <- severe_viralrna/normal_viralrna
inhibit_ratio_viralrna
```

#ViralRNA inhibition
```{r, tidy = TRUE}
ext_data$uniqueID <- c(1:length(ext_data$record_id))
uniqueID <- ext_data$uniqueID

ext_data$n1_actual <- NA
for(i in uniqueID){
  if(ext_data$n1_dil_dummy[ext_data$uniqueID == i] == 1){
    if(ext_data$n1_dummy[ext_data$uniqueID == i] == 0){
      ext_data$n1_actual[ext_data$uniqueID == i] <- 45
    } else {
      ext_data$n1_actual[ext_data$uniqueID == i] <-
ext_data$n1[ext_data$uniqueID == i]
    }
  }
}

ext_data$n1_expected <- NA
for(i in uniqueID){
  if(ext_data$n1_dil_dummy[ext_data$uniqueID == i] == 1){
    ext_data$n1_expected[ext_data$uniqueID == i] <-
      ext_data$dil_n1[ext_data$uniqueID == i] +
ext_data$n1_slope[ext_data$uniqueID == i]
  }
}

ext_data$n1_ideal <- NA
for(i in uniqueID){
  if(ext_data$n1_dil_dummy[ext_data$uniqueID == i] == 1){
    ext_data$n1_ideal[ext_data$uniqueID == i] <-
      ext_data$dil_n1[ext_data$uniqueID == i] - 3.3
  }
}

ext_data$n1_inhibit <-
  ext_data$n1_actual - ext_data$n1_expected

# Subset data
n1_inhibition <-
  subset(ext_data, method %in% c("ViralRNA"),
         n1_inhibit,
         drop = TRUE)
method <-
```

```
  subset(ext_data, method %in% c("ViralRNA"),
         method,
         drop = TRUE)
wwtp <-
  subset(ext_data, method %in% c("ViralRNA"),
         wwtp,
         drop = TRUE)

n1_inhibit_comp <- data.frame(method, wwtp, n1_inhibition)

n1_inhibit_comp$method <-
  factor(n1_inhibit_comp$method,
         levels = c("ViralRNA"),
         labels = c("ViralRNA"))

severe_viralrna <-
  length(
    n1_inhibit_comp$n1_inhibition
    [n1_inhibit_comp$n1_inhibition > 1 &
!is.na(n1_inhibit_comp$n1_inhibition)])

normal_viralrna <-
  length(
    n1_inhibit_comp$n1_inhibition
    [n1_inhibit_comp$n1_inhibition < 1 &
!is.na(n1_inhibit_comp$n1_inhibition)])

inhibit_ratio_viralrna <- severe_viralrna/normal_viralrna
inhibit_ratio_viralrna
```



```{r, tidy = TRUE}

#
https://www.itl.nist.gov/div898/handbook/prc/section4/prc474.htm

## Set the proportions of interest.
p = c(17/18, 1/7, 16/18)
N = length(p)
value = critical.range = c()

## Compute critical values.
for (i in 1:(N-1))
   { for (j in (i+1):N)
    {
     value = c(value,(abs(p[i]-p[j])))
```

```
---
title: "Methods Comp 2 Figures"
author: "Sarah Philo"
date: "6/17/2021"
geometry: margin = 1.75cm
output: pdf_document
editor_options:
  chunk_output_type: console
---

```{r, tidy = TRUE}
knitr::opts_chunk$set(echo = TRUE)

setwd("~/OneDrive/Documents/UW/EOHML/CoV/Methods Comp 2")
```


#Load Data into R and Manipulate to what I need
```{r, tidy = TRUE}
library(readr)

ceresnano <- read_csv("20210709 CeresNano Report.csv",
                      col_types = cols(exp_date =
col_date(format = "%m/%d/%y"),
                                       sample_date =
col_date(format = "%m/%d/%y")))

ceresnano$effective_vol <- ceresnano$method_vol *
  (ceresnano$ext_vol/ceresnano$final_vol) *
  (ceresnano$pcr_vol/ceresnano$elu_vol)

ceresnano$stock_conc_construct <- 10^((ceresnano$seed_cq -
ceresnano$intercept)/
                                        ceresnano$slope)

ceresnano$stock <- ((ceresnano$stock_conc*ceresnano$ext_vol)/
                      ceresnano$elu_vol)*(5/1000)

ceresnano$oc43_conc <-
(ceresnano$stock_conc*ceresnano$seed_vol)/
  ceresnano$comp_vol

ceresnano$est_oc43_assayed <- NA

ceresnano$act_oc43_assayed <- 10^((ceresnano$oc43 -
ceresnano$intercept)/
                                    ceresnano$slope)
```
```

```
ceresnano$uniqueID <- c(1:length(ceresnano$record_id))
uniqueID <- ceresnano$uniqueID

for(i in uniqueID) {
  if(is.na(ceresnano$std_positive_control[ceresnano$uniqueID ==
i])){
    } else {
       if(ceresnano$std_positive_control[ceresnano$uniqueID ==
i] == 1){
    ceresnano$est_oc43_assayed[ceresnano$uniqueID == i] <-
      (ceresnano$oc43_conc[ceresnano$uniqueID == i]*
         ceresnano$effective_vol[ceresnano$uniqueID == i])/1000
    ##something is happening here, don't worry about it tonight
  } else {
    ceresnano$est_oc43_assayed[ceresnano$uniqueID == i] <-
      (ceresnano$stock_conc_construct[ceresnano$uniqueID ==
i]/11.67)*
      (ceresnano$seed_vol[ceresnano$uniqueID == i]/
         ceresnano$comp_vol[ceresnano$uniqueID == i])*
      (ceresnano$effective_vol[ceresnano$uniqueID == i])
    }
  }
  }

ceresnano$percent_recovery <-
(ceresnano$act_oc43_assayed/ceresnano$est_oc43_assayed)*100

ceresnano$method <- factor(ceresnano$method,
                           levels = c(3, 12, 13),
                           labels = c("Viral RNA Mini",
                                      "CeresNano",
                                      "HA Membrane Filtration"))

ceresnano$wwtp <- factor(ceresnano$wwtp,
                         levels = c(1, 2, 3),
                         labels = c("West Point", "South Plant",
"Bright Water"))
```


#Plots
```{r, tidy = TRUE}
library(ggplot2)
library(RColorBrewer)
display.brewer.pal(n = 9, name = 'Purples')
brewer.pal(n = 9, name = 'Purples')

percentrecovery <- ggplot(data = ceresnano,
```

```
      aes(x = method, y = percent_recovery, fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=42),
        axis.title = element_text(size=24, face = "bold"),
        axis.text = element_text(size=24, face = "bold"),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = "none")+
  ggtitle("OC43 Recovery")+
  labs(y="% Recovery")+
  scale_fill_manual(values = c("#4B2E83", "#B7A57A", "#85754D"),
                    name = "Method")
#+scale_y_continuous(trans = "log10")
percentrecovery

ggsave("percent_recovery.png", plot = percentrecovery,
device="png", dpi=300, limitsize=TRUE)
```


##Balloon Plots
```{r, tidy = TRUE}

size <- as.vector(c(length(ceresnano$method[ceresnano$method ==
"CeresNano"]),
                    length(ceresnano$method[ceresnano$method ==
"Viral RNA Mini"]),
                    length(ceresnano$method[ceresnano$method ==
"HA Membrane Filtration"])))

method <- unique(ceresnano$method)

sars_pos <- as.vector(c((sum(ceresnano$pres_pos[ceresnano$method
== "CeresNano"]) /
                  length(ceresnano$pres_pos[ceresnano$method ==
"CeresNano"]))*100,
                (sum(ceresnano$pres_pos[ceresnano$method == "Viral
RNA Mini"]) /
                  length(ceresnano$pres_pos[ceresnano$method ==
"Viral RNA Mini"]))*100,
                (sum(ceresnano$pres_pos[ceresnano$method == "HA
Membrane Filtration"]) /
                  length(ceresnano$pres_pos[ceresnano$method ==
"HA Membrane Filtration"]))*100))
```

```
n1_pos <- as.vector(c((sum(ceresnano$n1_dummy[ceresnano$method
== "CeresNano"]) /
                  length(ceresnano$n1_dummy[ceresnano$method ==
"CeresNano"]))*100,
              (sum(ceresnano$n1_dummy[ceresnano$method == "Viral
RNA Mini"]) /
                  length(ceresnano$n1_dummy[ceresnano$method ==
"Viral RNA Mini"]))*100,
              (sum(ceresnano$n1_dummy[ceresnano$method == "HA
Membrane Filtration"]) /
                  length(ceresnano$n1_dummy[ceresnano$method ==
"HA Membrane Filtration"]))*100))

n2_pos <- as.vector(c((sum(ceresnano$n2_dummy[ceresnano$method
== "CeresNano"]) /
                  length(ceresnano$n2_dummy[ceresnano$method ==
"CeresNano"]))*100,
              (sum(ceresnano$n2_dummy[ceresnano$method == "Viral
RNA Mini"]) /
                  length(ceresnano$n2_dummy[ceresnano$method ==
"Viral RNA Mini"]))*100,
              (sum(ceresnano$n2_dummy[ceresnano$method == "HA
Membrane Filtration"]) /
                  length(ceresnano$n2_dummy[ceresnano$method ==
"HA Membrane Filtration"]))*100))

positive <- as.vector(c(sars_pos, n1_pos, n2_pos))

assays <- as.factor(c("SARS", "SARS", "SARS",
                      "N1", "N1", "N1",
                      "N2", "N2", "N2"))

assay_balloon <- cbind.data.frame(method, assays, size,
positive)
#need to have the two variables that are serving as x and Y for
balloon plots next to each other
assay_balloon$method <- as.factor(assay_balloon$method)

write.csv(x = assay_balloon, file = "assay_balloon.csv")
library(readr)
assay_balloon_test <- read_csv("assay_balloon.csv",
                               col_types = cols(X1 =
col_skip()))

library(ggpubr)
assayballoonplot <- ggballoonplot(data = assay_balloon_test,
              aes(x = assays,
                  y = method),
```

```
                  size = "size",
                  fill = "positive")+
  geom_text(data = assay_balloon_test, nudge_x = 0.2,
            aes(label = round(x = positive,
                              digits = 1)))+
  scale_size_continuous(range = c(5, 10),
                        name = "Sample Size")+
  scale_fill_gradient(low = "#DDDDDD",
                      high = "#4B2E83",
                      name = "Percent Positive")+
  ggtitle("Percent Positivity for SARS-CoV-2 by Method")+
  guides()+
  theme(legend.position = "right",
        plot.title = element_text(face = "bold",
                                  hjust = 0.5,
                                  size = 24),
        axis.title = element_text(size = 14),
        legend.text = element_text(size = 12),
        legend.title = element_text(size = 16, face = "bold"),
        axis.text.x = element_text(size = 14, color = "black",
face = "bold",
                                   angle = 0, hjust = 0.5),
        axis.text.y = element_text(size = 14, color = "black",
face = "bold"))
assayballoonplot
ggsave(filename = "assayballoonplot.png",
       plot = assayballoonplot,
       device = "png",
       dpi = 300,
       limitsize = TRUE)

```


```{r, tidy = TRUE}
library(readr)

viral_rna <- read_csv("Viral RNA CeresNano vs SMD.csv",
                      col_types = cols(exp_date =
col_date(format = "%Y-%m-%d"),
                                       sample_date =
col_date(format = "%Y-%m-%d")))

viral_rna$effective_vol <- viral_rna$method_vol *
  (viral_rna$ext_vol/viral_rna$final_vol) *
  (viral_rna$pcr_vol/viral_rna$elu_vol)
```

```
viral_rna$stock_conc_construct <- 10^((viral_rna$seed_cq -
viral_rna$intercept)/
                                        viral_rna$slope)

viral_rna$stock <- ((viral_rna$stock_conc*viral_rna$ext_vol)/
                      viral_rna$elu_vol)*(5/1000)

viral_rna$oc43_conc <-
(viral_rna$stock_conc*viral_rna$seed_vol)/
  viral_rna$comp_vol

viral_rna$est_oc43_assayed <- NA

viral_rna$act_oc43_assayed <- 10^((viral_rna$oc43 -
viral_rna$intercept)/
                                    viral_rna$slope)

viral_rna$uniqueID <- c(1:length(viral_rna$record_id))
uniqueID <- viral_rna$uniqueID

for(i in uniqueID) {
  if(is.na(viral_rna$std_positive_control[viral_rna$uniqueID ==
i])){
    } else {
       if(viral_rna$std_positive_control[viral_rna$uniqueID ==
i] == 1){
    viral_rna$est_oc43_assayed[viral_rna$uniqueID == i] <-
      (viral_rna$oc43_conc[viral_rna$uniqueID == i]*
         viral_rna$effective_vol[viral_rna$uniqueID == i])/1000
    ##something is happening here, don't worry about it tonight
  } else {
    viral_rna$est_oc43_assayed[viral_rna$uniqueID == i] <-
      (viral_rna$stock_conc_construct[viral_rna$uniqueID ==
i]/11.67)*
      (viral_rna$seed_vol[viral_rna$uniqueID == i]/
         viral_rna$comp_vol[viral_rna$uniqueID == i])*
      (viral_rna$effective_vol[viral_rna$uniqueID == i])
    }
  }
  }

viral_rna$percent_recovery <-
(viral_rna$act_oc43_assayed/viral_rna$est_oc43_assayed)*100

viral_rna$method <- factor(viral_rna$method,
                           levels = c(3, 11, 12, 16, 17),
                           labels = c("SMD + ViralRNA",
                                      "HA Membrane Filtration",
```

```
                                        "CeresNano + NucleoMag",
                                        "SMD + NucleoMag",
                                        "CeresNano + ViralRNA"))
viral_rna <- viral_rna %>%
  filter(method != "SMD + NucleoMag")

viral_rna$wwtp <- factor(viral_rna$wwtp,
                         levels = c(1, 2, 3),
                         labels = c("West Point",
                                    "South Plant",
                                    "Brightwater"))

summary(viral_rna$effective_vol[viral_rna$method == "SMD +
ViralRNA"])
summary(viral_rna$percent_recovery[viral_rna$method == "SMD +
ViralRNA"])
summary(viral_rna$effective_vol[viral_rna$method == "HA Membrane
Filtration"])
summary(viral_rna$percent_recovery[viral_rna$method == "HA
Membrane Filtration"])
summary(viral_rna$effective_vol[viral_rna$method == "CeresNano +
NucleoMag"])
summary(viral_rna$percent_recovery[viral_rna$method ==
"CeresNano + NucleoMag"])
summary(viral_rna$effective_vol[viral_rna$method == "CeresNano +
ViralRNA"])
summary(viral_rna$percent_recovery[viral_rna$method ==
"CeresNano + ViralRNA"])

viral_rna$hold_time <- viral_rna$exp_date -
viral_rna$sample_date

smd_date <- data.frame(viral_rna$method[viral_rna$method=="SMD +
ViralRNA"],

viral_rna$sample_date[viral_rna$method=="SMD + ViralRNA"])
colnames(smd_date) <- c("SMD", "date")
hamf_date <- data.frame((viral_rna$method[viral_rna$method=="HA
Membrane Filtration"]),

viral_rna$sample_date[viral_rna$method=="HA Membrane
Filtration"])
colnames(hamf_date) <- c("HA_MF", "date")
ceresnano_nm_date <-
data.frame(viral_rna$method[viral_rna$method=="CeresNano +
NucleoMag"],
```

```
viral_rna$sample_date[viral_rna$method=="CeresNano +
NucleoMag"])
colnames(ceresnano_nm_date) <- c("CN_NM", "date")
ceresnano_viral_date <-
data.frame(viral_rna$method[viral_rna$method=="CeresNano +
ViralRNA"],

viral_rna$sample_date[viral_rna$method=="CeresNano + ViralRNA"])
colnames(ceresnano_viral_date) <- c("CN_Viral", "date")

library(dplyr)
sampling_dates <- smd_date %>%
  full_join(hamf_date, by = "date") %>%
  full_join(ceresnano_nm_date, by = "date") %>%
  full_join(ceresnano_viral_date, by = "date") %>%
  distinct()
sampling_dates <- sampling_dates[,c(2, 1, 3, 4, 5, 6)]

write.csv(sampling_dates, file = "method_comp2_dates.csv")

```


#Viral RNA Extraction
```{r, tidy = TRUE}
library(ggplot2)

viral_rna_recovery <-
  ggplot(data = viral_rna,
       aes(x = method, y = percent_recovery, fill = method))+
  geom_boxplot()+
  theme_classic()+
  theme(axis.text.x = element_text(angle = 90, hjust=1, vjust =
0.5,
                                   face = "plain", size = 12),
        axis.text.y = element_text(face = "plain", size = 12),
        axis.title.x = element_blank(),
        axis.title.y = element_text(size = 12, face = "bold",
vjust = 0.5),
        legend.text = element_text(size = 12),
        legend.position = "none",
        axis.line = element_line(color = "#8E8E8E"),
        axis.ticks = element_line(color = "#8E8E8E",),
        panel.grid.major.y = element_line(color = "#8E8E8E"),
        panel.grid.minor.y = element_line(color = "light gray")
        )+
  #ggtitle("OC43 Recovery")+
```

```
  labs(y="Method Recovery")+
  scale_fill_manual(values = c("#85754D", "#8E8E8E", "#4b2e83",
"#B7A57A"))
#+scale_y_continuous(trans = "log10")
viral_rna_recovery

ggsave(plot = viral_rna_recovery, filename =
"viral_rna_recovery.tiff",
       units = 'in', width = 8.5, height = 6, dpi = 600)

viral_rna_volumeassayed <-
  ggplot(data = viral_rna,
       aes(x = method, y = effective_vol, color = method))+
  geom_boxplot()+
  theme_classic()+
  theme(axis.text.x = element_text(angle = 45, hjust=1,
                                   face = "plain", size = 12),
        axis.text.y = element_text(face = "plain", size = 12),
        axis.title.x = element_blank(),
        axis.title.y = element_text(size = 12, face = "bold",
vjust = 0.5),
        legend.text = element_text(size = 12),
        legend.position = "none",
        axis.line = element_line(color = "#8E8E8E"),
        axis.ticks = element_line(color = "#8E8E8E",),
        panel.grid.major.y = element_line(color = "#8E8E8E"),
        panel.grid.minor.y = element_line(color = "light gray")
        )+
  labs(y="Effective Vol. Assayed (mL)")+
  scale_color_manual(values = c("#85754D", "#8E8E8E", "#4b2e83",
"#B7A57A"))+
  scale_y_continuous(limits = c(0, 7))+
  scale_x_discrete(labels = c("CeresNano + NucleoMag" =
"Nanotrap + NucleoMag",
                              "CeresNano + ViralRNA" = "Nanotrap
+ ViralRNA",
                              "HA Membrane Filtration" =
"Membrane Filtration"))
viral_rna_volumeassayed

ggsave(plot = viral_rna_volumeassayed, filename =
"method_comp2_vol_assayed.png",
       units = 'in', width = 4, height = 4, dpi = 300)

viral_rna_vol_percent <-
  ggarrange(viral_rna_volumeassayed, viral_rna_recovery,
            ncol = 2,
            labels = "AUTO",
```

```
            font.label = list(size = 20))
viral_rna_vol_percent

ggsave(viral_rna_vol_percent, filename =
"viral_rna_vol_percent.tiff",
       units = 'in', width = 8.5, height = 5, dpi = 600)
```

#Paired Recovery Figures
```{r, tidy = TRUE}

viral_rna$hamf_boolean <- 0
unique_dates <- unique(viral_rna$sample_date)

#want all days with BOTH SM-D and HA Membrane Filtration,
nothing more
for(i in unique_dates){
  method_date <- viral_rna$method[viral_rna$sample_date==i]
  if(length(method_date[method_date == "HA Membrane
Filtration"]) >= 1){
    if(length(method_date[method_date == "SMD + ViralRNA"]) >=
1){
      viral_rna$hamf_boolean[viral_rna$sample_date==i] = 1}
    }
}

ha_mf <- subset(viral_rna, hamf_boolean %in% c(1))

ha_mf$method <- as.factor(ha_mf$method)
ha_mf$pres_pos <- as.factor(ha_mf$pres_pos)
ha_mf$pres_pos_dil <- as.factor(ha_mf$pres_pos_dil)

wilcox.test(ha_mf$percent_recovery[ha_mf$method=="SMD +
ViralRNA"],
            ha_mf$percent_recovery[ha_mf$method=="HA Membrane
Filtration"],
            paired = TRUE)

library(ggbreak)
mf_recovery <-
  ggpaired(data = subset(ha_mf, method %in% c("SMD + ViralRNA",
"HA Membrane Filtration")),
           x = "method", y = "percent_recovery",
           fill = "method",
           palette = c("#9E9AC8", "#6A51A3"),
           line.color = "gray")+
  theme_classic()+
```

```
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_text(hjust = .7, face = "bold"),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("")+
  labs(y = "OC43 % Recovery")+
  stat_compare_means(method = "wilcox.test", paired = TRUE,
                     comparisons = list(c("SMD + ViralRNA", "HA
Membrane Filtration")),
                     label.y = 12)+
  scale_y_continuous(labels = scales::percent_format(scale = 1,
accuracy = 1L),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 5),
                     expand = expansion(mult = c(0.05, 0.1)))+
  annotate(geom = "text",
           x = 1.5,
           y = 10,
           label = "n = 15",
           size = 3.5)+
  scale_y_break(breaks = c(15, 98))+
  scale_x_discrete(
    labels =
      c("HA Membrane Filtration" = "Membrane Filtration")
  )
mf_recovery

ggsave("mf_recovery.png",
       plot = mf_recovery, device = "png", dpi = 300, limitsize
= TRUE)

viral_rna$cnnm_boolean <- 0

#want all days with BOTH SM-D and HA Membrane Filtration,
nothing more
for(i in unique_dates){
  method_date <- viral_rna$method[viral_rna$sample_date==i]
  if(length(method_date[method_date == "CeresNano + NucleoMag"])
>= 1){
    if(length(method_date[method_date == "SMD + ViralRNA"]) >=
1){
```

```
      viral_rna$cnnm_boolean[viral_rna$sample_date==i] = 1}
    }
}

cn_nm <- subset(viral_rna, cnnm_boolean %in% c(1))

cn_nm$method <- as.factor(cn_nm$method)
cn_nm$pres_pos <- as.factor(cn_nm$pres_pos)
cn_nm$pres_pos_dil <- as.factor(cn_nm$pres_pos_dil)

wilcox.test(cn_nm$percent_recovery[cn_nm$method=="SMD +
ViralRNA"],
            cn_nm$percent_recovery[cn_nm$method=="CeresNano +
NucleoMag"],
            paired = TRUE)

library(ggbreak)
cnnm_recovery <-
  ggpaired(data = subset(cn_nm, method %in% c("SMD + ViralRNA",
"CeresNano + NucleoMag")),
           x = "method", y = "percent_recovery",
           fill = "method",
           palette = c("#9E9AC8", "#6A51A3"),
           line.color = "gray")+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_text(hjust = .7, face = "bold"),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("")+
  labs(y = "OC43 % Recovery")+
  stat_compare_means(method = "wilcox.test", paired = TRUE,
                     comparisons = list(c("SMD + ViralRNA",
"CeresNano + NucleoMag")),
                     label.y = 95)+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 20),
                     expand = expansion(mult = c(0.05, 0.1)))+
  scale_x_discrete(labels = c("CeresNano + NucleoMag" =
"Nanotrap + NucleoMag"))+
```

```
  annotate(geom = "text",
           x = 1.5,
           y = 85,
           label = "n = 18",
           size = 3.5)
  #scale_y_break(breaks = c(80, 98))
cnnm_recovery

ggsave("cnnms_recovery.png",
       plot = cnnm_recovery, device = "png", dpi = 300,
limitsize = TRUE)

viral_rna$cnviral_boolean <- 0

for(i in unique_dates){
  method_date <- viral_rna$method[viral_rna$sample_date==i]
  if(length(method_date[method_date == "CeresNano + ViralRNA"])
>= 1){
    if(length(method_date[method_date == "SMD + ViralRNA"]) >=
1){
      viral_rna$cnviral_boolean[viral_rna$sample_date==i] = 1}
    }
}

cn_viral <- subset(viral_rna, cnviral_boolean %in% c(1))

cn_viral$method <- as.factor(cn_viral$method)
cn_viral$pres_pos <- as.factor(cn_viral$pres_pos)
cn_viral$pres_pos_dil <- as.factor(cn_viral$pres_pos_dil)

wilcox.test(cn_viral$percent_recovery[cn_viral$method=="SMD +
ViralRNA"],

cn_viral$percent_recovery[cn_viral$method=="CeresNano +
ViralRNA"],
            paired = TRUE)

library(ggbreak)
cnviral_recovery <-
  ggpaired(data = subset(cn_viral, method %in% c("SMD +
ViralRNA", "CeresNano + ViralRNA")),
           x = "method", y = "percent_recovery",
           fill = "method",
           palette = c("#9E9AC8", "#6A51A3"),
           line.color = "gray")+
  theme_classic()+
  theme(plot.title = element_text(face="bold", hjust=0.5,
size=10),
```

```
        axis.title = element_text(size=10),
        axis.text = element_text(size=10),
        axis.text.x = element_text(angle = 45, hjust=1),
        axis.title.x = element_blank(),
        axis.title.y = element_text(hjust = .7, face = "bold"),
        legend.text = element_text(size = 8),
        legend.title = element_text(size = 8),
        legend.position = "None",
        panel.grid.major.y = element_line(color = "#F5F5F5"))+
  ggtitle("")+
  labs(y = "OC43 % Recovery")+
  stat_compare_means(method = "wilcox.test", paired = TRUE,
                     comparisons = list(c("SMD + ViralRNA",
"CeresNano + ViralRNA")),
                     label.y = 20)+
  scale_y_continuous(labels = scales::percent_format(scale = 1),
                     limits = c(0, 100),
                     breaks = seq(0,100, by = 10),
                     expand = expansion(mult = c(0.05, 0.1)))+
  scale_x_discrete(labels = c("CeresNano + ViralRNA" = "Nanotrap
+ ViralRNA"))+
  annotate(geom = "text",
           x = 1.5,
           y = 15,
           label = "n = 15",
           size = 3.5)+
  scale_y_break(breaks = c(25, 98))
cnviral_recovery

ggsave("cnviral_recovery.png",
       plot = cnviral_recovery, device = "png", dpi = 300,
limitsize = TRUE)

method_comp_recovery <-
  ggarrange(print(mf_recovery),
            print(cnnm_recovery),
            print(cnviral_recovery),
            nrow = 1, ncol = 3,
            legend = "none",
            labels = "AUTO",
            font.label = list(size = 12))
method_comp_recovery

ggsave(method_comp_recovery, filename =
"method_comp_recovery.tiff",
       units = 'in', width = 7, height = 2.9, dpi = 600)

```
```
```

```
#SARS Balloon Plots for All CN & MN Comparisons
```{r, tidy = TRUE}
size <- as.vector(c(length(viral_rna$method[viral_rna$method ==
"CeresNano + NucleoMag"]),
                    length(viral_rna$method[viral_rna$method ==
"CeresNano + ViralRNA"]),
                    length(viral_rna$method[viral_rna$method ==
"HA Membrane Filtration"]),
                    length(viral_rna$method[viral_rna$method ==
"SMD + ViralRNA"])))

method <- unique(viral_rna$method)

sars_pos <- as.vector(c((sum(viral_rna$pres_pos[viral_rna$method
== "CeresNano + NucleoMag"]) /

length(viral_rna$pres_pos[viral_rna$method == "CeresNano +
NucleoMag"]))*100,
                        (sum(viral_rna$pres_pos[viral_rna$method
== "CeresNano + ViralRNA"]) /

length(viral_rna$pres_pos[viral_rna$method == "CeresNano +
ViralRNA"]))*100,
                        (sum(viral_rna$pres_pos[viral_rna$method
== "HA Membrane Filtration"]) /

length(viral_rna$pres_pos[viral_rna$method == "HA Membrane
Filtration"]))*100,
                        (sum(viral_rna$pres_pos[viral_rna$method
== "SMD + ViralRNA"]) /

length(viral_rna$pres_pos[viral_rna$method == "SMD +
ViralRNA"]))*100))

n1_pos <- as.vector(c((sum(viral_rna$n1_dummy[viral_rna$method
== "CeresNano + NucleoMag"]) /

length(viral_rna$n1_dummy[viral_rna$method == "CeresNano +
NucleoMag"]))*100,
                        (sum(viral_rna$n1_dummy[viral_rna$method
== "CeresNano + ViralRNA"]) /

length(viral_rna$n1_dummy[viral_rna$method == "CeresNano +
ViralRNA"]))*100,
                        (sum(viral_rna$n1_dummy[viral_rna$method
== "HA Membrane Filtration"]) /
```
```

```
length(viral_rna$n1_dummy[viral_rna$method == "HA Membrane
Filtration"]))*100,
                          (sum(viral_rna$n1_dummy[viral_rna$method
== "SMD + ViralRNA"]) /

length(viral_rna$n1_dummy[viral_rna$method == "SMD +
ViralRNA"]))*100))

n2_pos <- as.vector(c((sum(viral_rna$n2_dummy[viral_rna$method
== "CeresNano + NucleoMag"]) /

length(viral_rna$n2_dummy[viral_rna$method == "CeresNano +
NucleoMag"]))*100,
                          (sum(viral_rna$n2_dummy[viral_rna$method
== "CeresNano + ViralRNA"]) /

length(viral_rna$n2_dummy[viral_rna$method == "CeresNano +
ViralRNA"]))*100,
                          (sum(viral_rna$n2_dummy[viral_rna$method
== "HA Membrane Filtration"]) /

length(viral_rna$n2_dummy[viral_rna$method == "HA Membrane
Filtration"]))*100,
                          (sum(viral_rna$n2_dummy[viral_rna$method
== "SMD + ViralRNA"]) /

length(viral_rna$n2_dummy[viral_rna$method == "SMD +
ViralRNA"]))*100))

positive <- as.vector(c(sars_pos, n1_pos, n2_pos))

assays <- as.factor(c("SARS", "SARS", "SARS", "SARS",
                      "N1", "N1", "N1", "N1",
                      "N2", "N2", "N2", "N2"))

assay_balloon <- cbind.data.frame(method, assays, size,
positive)
#need to have the two variables that are serving as x and Y for
balloon plots next to each other
assay_balloon$method <- as.factor(assay_balloon$method)

library(ggpubr)
methods_comp2_balloonplot <-
  ggballoonplot(data = assay_balloon,
                aes(x = assays,
                    y = method),
                    size = "size",
```

```
                  fill = "positive")+
  geom_text(data = assay_balloon, nudge_x = 0.35,
            aes(label = round(x = positive,
                              digits = 1)),
            size = 5)+
  scale_size_continuous(range = c(4, 12),
                        name = "Sample Size")+
  scale_fill_gradient2(low = "#EBEBEB",
                       mid = "#D9CEEE",
                       high = "#4B2E83",
                       midpoint = 50,
                       name = "Percent Positive")+
  ggtitle("Non-Diluted")+
  guides()+
  theme(legend.position = "bottom",
        plot.title = element_text(hjust=0.5, size=20),
        axis.title = element_text(size=16, face = "bold"),
        axis.text = element_text(size=16,
                                 family = "Arial",
                                 face = "bold"),
        axis.text.x = element_text(angle = 40, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12, color = "black"),
        legend.title = element_text(face = "bold", color =
"black"),
        axis.line = element_line(color = "#8E8E8E"),
        axis.ticks = element_line(color = "#8E8E8E",),
        panel.grid.major = element_line(color = "light gray"))+
  ylab("")+
  scale_x_discrete(
    labels = c("CeresNano + NucleoMag" = "Nanotrap + NucleoMag",
               "CeresNano + ViralRNA" = "Nanotrap + ViralRNA",
               "HA Membrane Filtration" = "Membrane Filtration")
  )
methods_comp2_balloonplot

ggsave(filename = "methods_comp2_balloonplot.png",
       plot = methods_comp2_balloonplot,
       device = "png",
       dpi = 300,
       limitsize = TRUE)


sars_pos_dil <-
as.vector(c((sum(viral_rna$pres_pos_dil[viral_rna$method ==
"CeresNano + NucleoMag"]) /
```

```
length(viral_rna$pres_pos_dil[viral_rna$method == "CeresNano +
NucleoMag"]))*100,

(sum(viral_rna$pres_pos_dil[viral_rna$method == "CeresNano +
ViralRNA"]) /

length(viral_rna$pres_pos_dil[viral_rna$method == "CeresNano +
ViralRNA"]))*100,

(sum(viral_rna$pres_pos_dil[viral_rna$method == "HA Membrane
Filtration"]) /

length(viral_rna$pres_pos_dil[viral_rna$method == "HA Membrane
Filtration"]))*100,

(sum(viral_rna$pres_pos_dil[viral_rna$method == "SMD +
ViralRNA"]) /

length(viral_rna$pres_pos_dil[viral_rna$method == "SMD +
ViralRNA"]))*100))

n1_pos_dil <-
as.vector(c((sum(viral_rna$n1_dil_dummy[viral_rna$method ==
"CeresNano + NucleoMag"]) /

length(viral_rna$n1_dil_dummy[viral_rna$method == "CeresNano +
NucleoMag"]))*100,

(sum(viral_rna$n1_dil_dummy[viral_rna$method == "CeresNano +
ViralRNA"]) /

length(viral_rna$n1_dil_dummy[viral_rna$method == "CeresNano +
ViralRNA"]))*100,

(sum(viral_rna$n1_dil_dummy[viral_rna$method == "HA Membrane
Filtration"]) /

length(viral_rna$n1_dil_dummy[viral_rna$method == "HA Membrane
Filtration"]))*100,

(sum(viral_rna$n1_dil_dummy[viral_rna$method == "SMD +
ViralRNA"]) /

length(viral_rna$n1_dil_dummy[viral_rna$method == "SMD +
ViralRNA"]))*100))
```

```
n2_pos_dil <-
as.vector(c((sum(viral_rna$n2_dil_dummy[viral_rna$method ==
"CeresNano + NucleoMag"]) /

length(viral_rna$n2_dil_dummy[viral_rna$method == "CeresNano +
NucleoMag"]))*100,

(sum(viral_rna$n2_dil_dummy[viral_rna$method == "CeresNano +
ViralRNA"]) /

length(viral_rna$n2_dil_dummy[viral_rna$method == "CeresNano +
ViralRNA"]))*100,

(sum(viral_rna$n2_dil_dummy[viral_rna$method == "HA Membrane
Filtration"]) /

length(viral_rna$n2_dil_dummy[viral_rna$method == "HA Membrane
Filtration"]))*100,

(sum(viral_rna$n2_dil_dummy[viral_rna$method == "SMD +
ViralRNA"]) /

length(viral_rna$n2_dil_dummy[viral_rna$method == "SMD +
ViralRNA"]))*100))

positive_dil <- as.vector(c(sars_pos_dil, n1_pos_dil,
n2_pos_dil))

assay_balloon_dil <- cbind.data.frame(method, assays, size,
positive_dil)
#need to have the two variables that are serving as x and Y for
balloon plots next to each other
assay_balloon_dil$method <- as.factor(assay_balloon_dil$method)

library(ggpubr)
methods_comp2_balloonplot_dil <- ggballoonplot(data =
assay_balloon_dil,
              aes(x = assays,
                  y = method),
                  size = "size",
                  fill = "positive_dil")+
  geom_text(data = assay_balloon_dil, nudge_x = 0.35,
            aes(label = round(x = positive_dil,
                              digits = 1)),
            size = 5)+
  scale_size_continuous(range = c(4,12),
                        name = "Sample Size")+
  scale_fill_gradient2(low = "#EBEBEB",
```

```
                        mid = "#D9CEEE",
                        high = "#4B2E83",
                        midpoint = 50,
                        name = "Percent Positive")+
  ggtitle(bquote("Dilution" ~10^-1))+
  guides()+
  theme(legend.position = "bottom",
        plot.title = element_text(hjust=0.5, size=20),
        axis.title = element_text(size=16, face = "bold"),
        axis.text = element_text(size=16,
                                 face = "bold",
                                 family = "Arial"),
        axis.text.x = element_text(angle = 40, hjust=1),
        axis.title.x = element_blank(),
        legend.text = element_text(size = 12, color = "black"),
        legend.title = element_text(face = "bold", color =
"black"),
        axis.line = element_line(color = "#8E8E8E"),
        axis.ticks = element_line(color = "#8E8E8E",),
        panel.grid.major = element_line(color = "light gray"))+
  ylab("")+
  scale_x_discrete(
    labels = c("CeresNano + NucleoMag" = "Nanotrap + NucleoMag",
               "CeresNano + ViralRNA" = "Nanotrap + ViralRNA",
               "HA Membrane Filtration" = "Membrane Filtration")
  )
methods_comp2_balloonplot_dil

ggsave(filename = "methods_comp2_balloonplot_dil.png",
       plot = methods_comp2_balloonplot_dil,
       device = "png",
       dpi = 300,
       limitsize = TRUE)

library("ggpubr")
balloon_plots <- ggarrange(methods_comp2_balloonplot,
methods_comp2_balloonplot_dil,
                           ncol = 2,
                           nrow = 1,
                           common.legend = TRUE,
                           legend = "bottom",
                           labels = "AUTO",
                           font.label = list(size = 20))
balloon_plots
ggsave(balloon_plots, filename = "balloon_plots.tiff",
       units = 'in', width = 10, height = 5, dpi = 600)

annotate_balloon <-
```

```
  annotate_figure(balloon_plots,
                  top = text_grob("Percent Positivity for SARS-
CoV-2 by Method",
                                  face = "bold", size = 24))
annotate_balloon
ggsave(annotate_balloon, filename =
"balloon_plots_annotate.tiff",
       units = 'in', width = 10, height = 5, dpi = 600)
```

#SARS2 detection stats
```{r, tidy = TRUE}
tab_mf <-
  table(
    ha_mf$pres_pos[ha_mf$method == "SMD + ViralRNA"],
    ha_mf$pres_pos[ha_mf$method == "HA Membrane Filtration"],
    dnn = c("SMD", "HA-MF")
  )
tab_mf
mcnemar_mf <- mcnemar.test(tab_mf)
mcnemar_mf

tab_mf_dil <-
  table(
    ha_mf$pres_pos_dil[ha_mf$method == "SMD + ViralRNA"],
    ha_mf$pres_pos_dil[ha_mf$method == "HA Membrane
Filtration"],
    dnn = c("SMD", "HA-MF")
  )
tab_mf_dil
mcnemar_mf_dil <- mcnemar.test(tab_mf_dil)
mcnemar_mf_dil

tab_cn <-
  table(
    cn_nm$pres_pos[cn_nm$method == "SMD + ViralRNA"],
    cn_nm$pres_pos[cn_nm$method == "CeresNano + NucleoMag"],
    dnn = c("SMD", "CN+NM")
  )
tab_cn
mcnemar_cnnm <- mcnemar.test(tab_cn)
mcnemar_cnnm

tab_cn_dil <-
  table(
    cn_nm$pres_pos_dil[cn_nm$method == "SMD + ViralRNA"],
    cn_nm$pres_pos_dil[cn_nm$method == "CeresNano + NucleoMag"],
    dnn = c("SMD", "CN+NM")


```

```
  )
tab_cn_dil
mcnemar_cnnm_dil <- mcnemar.test(tab_cn_dil)
mcnemar_cnnm_dil

tab_cnviral <-
  table(
    cn_viral$pres_pos[cn_viral$method == "SMD + ViralRNA"],
    cn_viral$pres_pos[cn_viral$method == "CeresNano +
ViralRNA"],
    dnn = c("SMD", "CN+Viral")
  )
tab_cnviral
mcnemar_cnviral <- mcnemar.test(tab_cnviral)
mcnemar_cnviral

tab_cnviral_dil <-
  table(
    cn_viral$pres_pos_dil[cn_viral$method == "SMD + ViralRNA"],
    cn_viral$pres_pos_dil[cn_viral$method == "CeresNano +
ViralRNA"],
    dnn = c("SMD", "CN+Viral")
  )
tab_cnviral_dil
mcnemar_cnviral_dil <- mcnemar.test(tab_cnviral_dil)
mcnemar_cnviral_dil
```

# 3 Wastewater Surveillance for Antibacterial Resistance

## 3.1 Wastewater Surveillance for Antibacterial-Resistance Genes During the late 2020 SARS-CoV-2 Peak in Two Different Populations

### 3.1.1 Abstract

The SARS-CoV-2 pandemic challenged the capacity of medical systems around the world. Medical professionals often prescribed antibiotics in the absence of other medical options for COVID-19 and to stave off secondary bacterial infections. The US CDC reported a rise in resistant infections after the COVID-19 pandemic started. How and if the pandemic contributed to antibacterial resistance abundance in the environment is not well understood. Wastewater treatment plants are hotspots of antibacterial resistance and good locations for environmental surveillance because water from an entire population is collected there. This study aimed to understand how rising COVID-19 cases from October 2020 – February 2021 in Portugal and King County contributed to antibacterial-resistance genes detected in wastewater. Primary influent wastewater was collected from two treatment plants in King County and four treatment plants in Portugal, and hospital effluent was collected from three hospitals in Portugal. Genomic extracts were tested with qPCR and dPCR for antibacterial-resistance genes conferring resistance against antibiotics under threat as identified by the US Centers for Disease Control and Prevention. Random-effects models were fit for log-transformed gene-abundances to assess trends in gene abundance. Pearson's correlation coefficients were calculated for effluent and influent samples collected in the same metro area in Portugal, and Wilcoxon signed rank tests were conducted on paired

dPCR and qPCR samples collected in King County. Both *mecA* and $bla_{CTX\text{-}M\ Groups2,\ 8,\ 25}$ significantly increased in Portugal during the sampling period. There was no statistical evidence of correlation between samples collected in the same metro area. Results in this study further highlight the importance of surveilling for antibacterial resistance genes in environments impacted by humans.

### 3.1.2 Introduction

The first cases of COVID-19 in Portugal were reported to the World Health Organization (WHO) on March 2, 2020 (WHO, 2020). The Portuguese government declared a state of emergency on March 19, 2020, which began the first lockdown (RTP, 2020). The number of cases escalated rapidly after this, with over 900 daily cases reported on April 1, 2020 (WHO, 2020). Daily reported cases remained at about this level over the summer until they started to rise again in September and October. Within one year of the first cases being reported in the country, there were over seven-hundred thousand cases and sixteen-thousand deaths reported in Portugal (WHO, 2020). SARS-CoV-2 was detected in the United States earlier than in Portugal, with the first confirmed COVID-19 case identified in Snohomish County, Washington from a sample collected on January 18, 2020 (NPR, 2020). There were over eighty thousand confirmed cases and one thousand deaths in King County, Washington within one year of the first confirmed case (King County Public Health, 2020).

A clinical diagnostic test was quickly developed and rapidly distributed around the world (Corman *et al.*, 2020, Sheridan, 2020), but the demand for tests rapidly outpaced the capacity for conducting those tests. This created a need for disease surveillance separate from clinical testing such as wastewater surveillance. Wastewater surveillance has been successfully used to supplement clinical surveillance for other

infectious diseases, most notably for poliovirus elimination and eradication (The World Health Organization, 2003, Hovi *et al.*, 2012). Some of the earliest reports in March 2020 detected SARS-CoV-2, the causative virus for COVID-19, in wastewater all over the world (Ahmed *et al.*, 2020, Medema *et al.*, 2020, Peccia *et al.*, 2020). A Portuguese effort to conduct SARS-CoV-2 wastewater surveillance began on April 27, 2020 and continued until December 2, 2020 (Monteiro *et al.*, 2022). In King County, wastewater was collected to test for SARS-CoV-2 throughout 2020 and 2021 (Philo *et al.*, 2021, Philo *et al.*, 2022).

Although existing anti-viral medications and antibiotics such as remdesivir and azithromycin, respectively, were being investigated for their clinical efficacy against SARS-CoV-2 infection (Chibber *et al.*, 2020), no medication had been clinically proven to be effective. Despite this lack of efficacy data, antibiotics were widely prescribed to COVID-19 patients. A cross sectional survey carried out from April 7 – 28, 2020 indicated that 60% of doctors in Portugal routinely prescribed antibiotics to patients admitted to the hospital, and β-lactams with a macrolide is the most common combination of medication (Beović *et al.*, 2020). In the same cross-sectional survey, all doctors surveyed in North America routinely prescribed antibiotics to patients (Beović *et al.*, 2020). Other data suggest as many as three quarters of hospitalized patients were prescribed antibiotics (Langford *et al.*, 2021, Russell *et al.*, 2021, Centers for Disease Control and Prevention *et al.*, 2022), even though a meta-analysis suggests that bacterial co-infection was low (Langford *et al.*, 2021). Additionally, patients who were mechanically ventilated were more likely to be prescribed antibiotics (Langford *et al.*, 2021). However, widespread use of antibiotics doesn't come without risk. Use of antibiotics leads to elevated resistance and increased risk of resistant infections

(Centers for Disease Control and Prevention, 2019, Hutchings *et al.*, 2019). Finally, surface disinfectants were widely used during the COVID-19 pandemic. Over half of the surface disinfectants approved to disinfect SARS-CoV-2 contained a quaternary ammonium compound (QAC) (Hora *et al.*, 2020). QACs have been shown to upregulate efflux pumps associated with multidrug resistance in bacteria (Merchel Piovesan Pereira & Tagkopoulos, 2019). The wide use of antibiotics and surface disinfectants suggest that bacteria would have experienced increased resistance.

While it is not fully understood how the COVID-19 pandemic directly affected antimicrobial resistance (AMR), it is apparent that AMR became more widespread. Almost every organism threatened by AMR and tracked by the US Centers for Disease Control and Prevention (CDC) showed an increase in resistance compared to 2019 (Centers for Disease Control and Prevention *et al.*, 2022). Bacteria such as carbapenem-resistant Enterobacterales (CRE) were exhibiting decreased resistance before the pandemic, but hospital-onset infections increased 35% after the pandemic (Centers for Disease Control and Prevention *et al.*, 2022). In an Italian ICU, incidence of CRE increased from 6.7% in 2019 to 50% in March and April 2020 (Tiri *et al.*, 2020). If antibacterial resistance genes (ARGs) associated with clinical resistance were also more abundant in wastewater is not well understood.

Given the apparent increase in AMR and the widespread adoption of wastewater surveillance for SARS-CoV-2, there have been numerous calls to include AMR targets in these wastewater surveillance efforts (Pruden *et al.*, 2021, Jin *et al.*, 2022). The CDC has recently named AMR as a future target for the National Wastewater Surveillance System (NWSS) (Centers for Disease Control and Prevention, 2022). Additionally, a framework for designing a wastewater surveillance program was recently published that

lays out different objectives and the methods needed to achieve those objectives (Liguori *et al.*, 2022). Despite these goals, very little research has been done to assess the impacts of COVID-19 on AMR in wastewater. A study carried out in Las Vegas, Nevada was able to detect increases in ARGs in wastewater collected at the wastewater treatment plant (WWTP) corresponding to the antibiotics used to treat COVID-19 infections in late 2020 and early 2021 (Harrington *et al.*, 2022). As of December 2022, the paper by Harrington, *et al.,* is the only published study assessing how the COVID-19 pandemic affected AMR in wastewater, however they used sequencing to identify ARGs. Furthermore, there are very few, if any, studies looking at this problem in multiple communities, both in and outside of the United States. Liguori *et al.* (2022) have suggested that qPCR of clinically relevant ARGs can be used on influent and effluent wastewater to quantify rates of removal. Munir *et al.* (2011) use qPCR in this way to compare ARG removal from different WWTPs in Michigan. Using qPCR for ARG surveillance is a low-cost option that can help identify the scope of the problem before implementing high-cost methods such as sequencing.

The current research aimed to understand how the COVID-19 pandemic in late 2020 contributed to detectable ARGs in wastewater by assaying hospital effluent wastewater and primary influent wastewater at WWTPs collected in Portugal and primary influent collected at WWTPs in King County. ARGs were detected using qPCR in all samples, and a subset of samples in King County were also assayed with dPCR to compare differences in detection methods. The goal of this research is to assess trends in ARGs over time and to understand if metro areas in Portugal trended together.

**Table 3.1.1**: Sampling dates for wastewater collected in Portugal.

| | North 1 | | North 2 | | South 1 | South 2 | | South 3 |
|---|---|---|---|---|---|---|---|---|
| **Week** | *WWTP* | *Hospital* | *WWTP* | *Hospital* | *WWTP* | *WWTP* | *Hospital* | *WWTP* |
| **34** | 8/18/20 | 8/18/20 | 8/18/20 | 8/18/20 | 8/18/20 | 8/18/20 | 8/18/20 | 8/18/20 |
| **36** | 9/1/20 | 9/1/20 | 9/1/20 | 9/1/20 | 9/1/20 | 9/1/20 | 9/1/20 | 9/1/20 |
| **38** | 9/15/20 | - | 9/15/20 | 9/15/20 | 9/15/20 | 9/15/20 | 9/15/20 | 9/15/20 |
| **40** | - | - | - | - | 9/29/20 | 9/29/20 | 9/29/20 | 9/29/20 |
| **41** | 10/12/20 | 10/12/20 | 10/12/20 | 10/12/20 | - | - | - | - |
| **42** | - | - | - | - | 10/15/20 | 10/14/20 | 10/13/20 | 10/13/20 |
| **43** | 10/26/20 | 10/26/20 | 10/26/20 | - | - | - | - | 10/26/20 |
| **44** | - | - | - | 10/28/20 | 10/28/20 | 10/28/20 | 10/27/20 | - |
| **45** | 11/9/20 | 11/9/20 | 11/9/20 | - | - | - | - | 11/9/20 |
| **46** | - | - | - | 11/11/20 | 11/11/20 | 11/11/20 | 11/10/20 | - |
| **47** | 11/23/20 | 11/23/20 | 11/23/20 | - | - | - | - | - |
| **48** | 11/25/20 | - | - | 11/25/20 | 11/25/20 | 11/25/20 | - | 11/25/20 |

## 3.1.3 Experimental Methods

### 3.1.3.1 Portugal

#### *3.1.3.1.1 Wastewater Sampling*

Wastewater concentrate originally collected to conduct SARS-CoV-2 wastewater surveillance from August 18, 2020, through November 25, 2020 was used in this study (Monteiro et al., 2022) (Table 3.1.1). Samples were collected from five WWTPs and from three hospitals where patients were referred for COVID-19 care. Two WWTPs are in the North and three are in the South. The three hospitals are associated with the water catchment areas of some of the WWTPs. Because the goal was to assess changes in ARGs contemporaneous with changes in the SARS-Cov-2 pandemic in Portugal, samples were selected around the first major pandemic peak in November 2020. To establish a “background” ARG prevalence, samples collected before COVID-19 cases started rising were processed. Starting August 18, 2020, the first weekly sample collected every-other week was chosen for ARG analysis (Table 3.1.1). The last samples processed were collected on November 25, 2020. Due to logistical constraints,

some locations did not have samples collected every week. A total of 59 samples collected throughout the country were assessed for ARGs.

Twenty-four-hour composite samples were collected using autosamplers (ISCO, Inc., Lincoln, NE, USA) at all sampling locations except the hospitals in the north, where grab samples were collected. All samples were transported in a refrigerated bag at 5°C with temperature control (+/- 3°C) to the Laboratório Análises do Instituto Superior Técnico (LAIST) at the Instituto Superior Técnico (IST) in Lisbon, Portugal within eight hours of collection. Samples were processed within 24 hours upon arrival to the LAIST.

#### *3.1.3.1.2 Concentration and DNA Extraction*

Wastewater was concentrated according to Monteiro *et al.* (2022) using Inuvai R180 hollow fiber filters with a molecular weight cut off ≤ 18.8 kDa (Inuvai, Fresenius Medical Care, Bad Homburg, Germany). The step-by-step protocol is included in the appendices in Section 3.3.1. Briefly, the filters were first primed using a 0.9% v/V NaCl solution. After priming, the wastewater sample was filtered at a flow of 200 – 250 mL/min. Next, the sample was eluted using 1x PBS (pH = 7.4) supplemented with 0.01% of sodium polyphosphate, 0.01% Tween 80, and 0.001% antifoam. The total elution volume was 300mL.

The elution was further concentrated using polyethylene glycol (PEG) precipitation. PEG 8000 was added for a 20% v/V final concentration (60g per 300mL of eluate) with NaCl (7.17g per 300mL). The sample bottles were agitated until the PEG dissolved and were then placed on a rocking table at 4°C overnight. Samples were next centrifuged at 10,000xG for 30 minutes and resuspended in 1x PBS (pH = 7.4) for a final volume of five milliliters. Samples were stored at -80°C until extraction using the QIAamp Fast DNA Stool Mini Kit (QIAGEN, Hilden, Germany) following the protocol for

**Table 3.1.2**: Sampling dates for wastewater collected in King County.

| | West Point | South Plant |
|---|---|---|
| *Week* | *WWTP* | *WWTP* |
| 36 | 9/1/20 | 9/1/20 |
| 38 | 9/15/20 | 9/15/20 |
| 40 | 9/29/20 | 9/29/20 |
| 42 | 10/13/20 | 10/13/20 |
| 44 | 10/27/20 | 10/27/20 |
| 46 | 11/10/20 | 11/10/20 |
| 48 | 11/24/20 | 11/24/20 |
| 50 | 12/8/20 | 12/8/20 |
| 52 | 12/22/20 | 12/22/20 |
| 1 | 1/5/21 | 1/5/21 |
| 3 | 1/19/21 | 1/19/21 |
| 5 | 2/2/21 | 2/2/21 |
| 7 | 2/16/21 | 2/16/21 |

human DNA analysis with the following modifications. The input material was 220 µL of wastewater concentrate and the DNA was eluted into 100uL of the included elution buffer (buffer ATE). Extracts were kept at -20°C until analysis.

### 3.1.3.2 King County

#### *3.1.3.2.1 Wastewater Sampling*

Wastewater concentrate previously collected in King County to test for SARS-CoV-2 was used in this study (Philo *et al.*, 2021, Philo *et al.*, 2022). Samples were collected from two WWTPs serving the Seattle-area in King County, WA. Samples were selected around increased COVID-19 cases from November 2020 to January 2021. Samples collected every other week from this date were tested for ARGs (Table 3.1.2). The earliest samples tested were collected on September 1, 2020, until February 16, 2021. In total, 26 samples were tested, with 13 from each WWTP. Primary influent wastewater was grab sampled. All samples were transported on ice to the Environmental and Occupational Health Microbiology Lab (EOHML) at the University of

Washington Seattle campus the same day of sample collection. Samples were stored at 4°C and processed within 48 hours upon arrival to the EOHML.

#### *3.1.3.2.2 Concentration and DNA Extraction*

Wastewater was concentrated following the skimmed milk flocculation protocol according to Philo *et al.* (2021). A step-by-step protocol is included in the appendices in section 3.3.2. Briefly, a 5% skimmed milk solution was added to 250mL of wastewater (1% v/V final). The pH of the wastewater was dropped to 3-4 using 5M HCl and shaken at about 200RPM for two hours. After shaking, the wastewater was centrifuged at 3500G and 4°C for 30 minutes and pellets were resuspended in 3.0mL of sterile 1x PBS (pH = 7.4). The resuspended wastewater concentrate was stored at -80°C until extraction.

Genetic material was extracted using the ViralRNA Mini Kit (QIAGEN, Germantown, MD, USA). Each sample was extracted in duplicate with a doubled input volume for a total extraction volume of 560uL using the necessary volume adjustments described in the manufacturer's protocol. Each column was eluted in 60uL of Buffer AVE for a total elution volume of 120uL. Duplicate extracts were combined and re-aliquoted before being stored at -20°C for later molecular analysis.

### 3.1.3.3 qPCR and dPCR

Specific gene targets were selected because they confer resistance against some of the last line antibiotics and/or have been identified as threats to public health by the United States Centers for Disease Control and Prevention (Centers for Disease Control and Prevention, 2019). Samples were tested for genes conferring resistance to extended-spectrum beta-lactams and cephalosporins (*bla*$_{CMY}$, *bla*$_{SHV}$, and *bla*$_{CTX\text{-}M}$) (Roschanski *et al.*, 2014), colistin (*mcr1*) (Donà *et al.*, 2017), vancomycin (*vanA*) (He *et*

**Table 3.1.3**: Primer and probe sequences for ARG targets.

| Target | Name | Sequence (5` - 3`) | Portugal Probe | King County Probe | Amplicon Length | Reference |
|---|---|---|---|---|---|---|
| *bla*$_{CMY}$ | CMY_fwd<br>CMY_rev<br>CMY_probe | GGCAAACAGTGGCAGGGTAT<br>AATGCGGCTTTATCCCTAACG<br>CCTACCGCTGCAGATCCCCGATG | ROX | FAM | 131bp | Roschanski *et al.* (2014) |
| *bla*$_{SHV}$ | SHV_fwd<br>SHV_rev<br>SHV_probe | TCCCATGATGAGCACCTTTAAA<br>TCCTGCTGGCGATAGTGGAT<br>TGCCGGTGACGAACAGCTGGAG | Cy5 | FAM | 134bp | Roschanski *et al.* (2014) |
| *bla*$_{CTX-M}$ | CTX-A_fwd<br>CTX-A_rev<br>CTX-A_p | CGGGCRATGGCGCARAC<br>TGCRCCGGTSGTATTGCC<br>CCARCGGGCGCAGYTGGTGAC | Yak. Yellow | FAM | 135bp | Roschanski *et al.* (2014) |
| *bla*$_{CTX-M}$ | CTX-B_fwd<br>CTX-B_rev<br>CTX-B_p | ACCGAGCCSACGCTCAA<br>CCGCTGCCGGTTTTATC<br>CCCGCGYGATACCACCACGC | Yak. Yellow | FAM | 220bp | Roschanski *et al.* (2014) |
| *mcr*-1 | mcr-1-FW<br>mcr-1-RV | ACGCCATCTGCAACACCAA<br>GCCAACGAGCATACCGACA | - | - | 87bp | Donà *et al.* (2017) |
| *mecA* | mecA-F<br>mecA-R<br>mecA-p | AAAACTAGGTGTTGGTGAAGATATACC<br>GAAAGGATCTGTACTGGGTTAATCAG<br>TTCACCTTGTCCGTAACCTGAATCAGCT | FAM | FAM | 177bp | Sabet *et al.* (2007) |
| *vanA* | vanA-F<br>vanA-R<br>vanA-P | GCCGGAAAAAGGCTCTGAA<br>TTTTTTGCCGTTTCCTGTATCC<br>CGCAGTTATAACCGTTCCCGCAGACC | FAM | FAM | 120bp | He *et al.* (2020) |

*al.*, 2020), and methicillin (*mecA*) (Sabet *et al.*, 2007). Samples were assessed for ARGs using pre-existing qPCR assays (Table 3.1.3). All qPCR reactions in Portugal were carried out on either the Applied Biosystems QuantStudio ™ 5 System or the 7500 Fast Real-Time PCR System (Applied Biosystems Corp., Waltham, MA, USA). DNA was amplified using the Luna® Universal qPCR Master Mix for SYBR assays and the Luna ® Universal Probe qPCR Master Mix for TaqMan probe assays (New England

**Table 3.1.4**: Primer and probe concentrations and reaction conditions for qPCR in Portugal and King County.

| Assays | Concentrations | Reaction Conditions |
|---|---|---|
| *bla*$_{CMY}$, *bla*$_{SHV}$, *bla*$_{CTX-M}$ | Forward: 0.4uM<br>Reverse: 0.4uM<br>Probe: 0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 50°C for 15 seconds<br>• 70°C for 25 seconds |
| *mcr1* | Forward: 0.2uM<br>Reverse:0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 63°C for 10 seconds<br>• 72°C for 10 seconds<br>Melt curve |
| *mecA* | Forward: 0.4uM<br>Reverse: 0.4uM<br>Probe: 0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 30 seconds<br>• 55°C for 45 seconds |
| *vanA* | Forward: 0.5uM<br>Reverse: 0.5uM<br>Probe: 0.3uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 60°C for 60 seconds |

Biolabs, Ipswich, MA, USA) in a total reaction volume of 15µL with 1.0uL of DNA extract, 7.5uL of the respective master mix, and variable amounts of water, primers, and probes according to their respective publications. Reaction conditions and primer and probe concentrations are in Table 3.1.4. All qPCR reactions in Seattle were carried out using the iTaq Universal Probes Supermix or the iTaq Universal SYBR Green Supermix and the BioRad CFX qPCR systems (Bio-Rad Laboratories, Hercules, CA, USA). The total reaction volume was 15uL with 7.5uL of the respective master mix, 2.0uL of extract, and variable amounts of water, primers, and probes. Reaction conditions and primer and probe concentrations are in Table 3.1.4.

All samples were run in duplicate with duplicate 10-fold dilutions. A negative control of nuclease free water was included with each run. Samples were quantified using standard curves generated using 10-fold dilutions of positive controls. A subset of samples collected in King County were additionally assayed using dPCR on the Absolute Q™ Digital PCR system (Applied Biosystems Corp., Waltham, MA, USA) to compare efficiencies of qPCR and dPCR. The existing qPCR assays were adapted to dPCR for a total reaction volume of 9uL with 1.80uL of Absolute Q™ DNA dPCR Mix (5x), 2.0uL of genetic material, and variable amounts of water, primers, and probes.

**Table 3.1.5**: Primer and probe concentrations and reaction conditions for dPCR in King County.

| Assays | Concentrations | Reaction Conditions |
|---|---|---|
| *bla*$_{CMY}$, *bla*$_{SHV}$, *bla*$_{CTX-M}$ | Forward: 0.9uM<br>Reverse: 0.9uM<br>Probe: 0.2uM | 95°C for 10 minutes<br>45 cycles:<br>• 95°C for 5 seconds<br>• 50°C for 15 seconds<br>• 70°C for 15 seconds |
| *mecA* | Forward: 0.9uM<br>Reverse: 0.9uM<br>Probe: 0.2uM | 95°C for 10 minutes<br>45 cycles:<br>• 95°C for 5 seconds<br>• 55°C for 15 seconds |
| *vanA* | Forward: 0.9uM<br>Reverse: 0.9uM<br>Probe: 0.25uM | 95°C for 10 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 60°C for 60 seconds |

Reaction conditions and primer and probe concentrations for dPCR carried out in Seattle are in Table 3.1.5.

#### *3.1.3.3.1 Positive Controls*

To obtain positive controls for qPCR reactions in Portugal, wastewater concentrate was screened for the ARGs using the respective assays. Potential positive reactions were then run using electrophoresis on a 2.5% agarose gel. Gels were stained with ethidium bromide, and bands containing the correct number of basepairs were cut out. Lengths are listed in Table 3.1.3. Gels were extracted using the GeneJET Gel Extraction Kit (Thermo Scientific, Waltham, MA, USA) following the manufacturer's protocol for DNA segments less than 500 basepairs and that will be used for sequencing. Extracted DNA segments were then re-run on their respective qPCR assays to confirm the correct sequence was isolated. Positive controls in Seattle were gBlocks obtained from IDT (Integrated DNA Technologies, Inc., Coralville, IA, USA).

To be used as positive controls in standard curves, the number of gene copies was analyzed using digital PCR (dPCR) or a Qubit ™ 3 Fluorometer (Invitrogen Corp., Waltham, MA, USA). In Portugal, *blaCTX-M*, *mecA*, and *vanA* positive controls were quantified analyzed with dPCR using the QuantStudio™ 3D Digital PCR system

**Table 3.1.6**: Primer and probe concentrations and reaction conditions for dPCR in Portugal

| Assays | Concentrations | Reaction Conditions |
|---|---|---|
| $bla_{SHV}$ | Forward: 0.4uM<br>Reverse: 0.4uM<br>Probe: 0.25uM | 95°C for 1 minute<br>45 cycles:<br>• 95°C for 15 seconds<br>• 50°C for 15 seconds<br>• 70°C for 20 seconds |
| *mecA* | Forward: 0.4uM<br>Reverse: 0.4uM<br>Probe: 0.2uM | 95°C for 3 minutes<br>45 cycles:<br>• 95°C for 30 seconds<br>• 55°C for 120 seconds |
| *vanA* | Forward: 0.5uM<br>Reverse: 0.5uM<br>Probe: 0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 30 seconds<br>• 60°C for 120 seconds |

(Applied Biosystems Corp., Waltham, MA, USA) at LAIST in Lisbon. In Lisbon, dPCR was carried out with 1uL of genomic material and reaction conditions and primer and probe concentrations are in Table 3.1.6. The remaining genes were shipped to the Environmental and Occupational Health Microbiology Lab (EOHML) in Seattle for quantification with the Qubit™ 3 Fluorometer (*mcr1*) or the QuantStudio™ Absolute Q™ Digital PCR system (*blaSHV* and *blaCMY*) using 2uL of genomic material and FAM labeled probes. The gBlocks were similarly quantified using dPCR on the QuantStudio™ Absolute Q™ Digital PCR system or the Qubit™ 3 Fluorometer (*mcr1*). Reaction conditions and primer and probe concentrations are in Table 3.1.5. DNA concentrations obtained from the Qubit™ were turned into copy numbers using Equation 1 (Tyra, 2017).

Equation 1: $C_{Qubit} \times M \times \left(1 \times 10^{-15}\, {}^{mol}/_{fmol}\right) \times (6.022 \times 10^{23}) = {}^{copies}/_{\mu L}$

$C_{Qubit} = concentration\ in\ \frac{ng}{uL}\ from\ Qubit$

$M = molecular\ weight$

### 3.1.3.4 Data Management and Statistical Analyses

Daily COVID-19 case data in Portugal were downloaded on October 17, 2022 from the World Health Organization COVID-19 dashboard (WHO, 2020). Daily COVID-19 case data for Seattle were downloaded on October 10, 2022 from the King County Public Health COVID-19 data dashboard (King County Public Health, 2020). All qPCR data were analyzed using either Microsoft Excel (Microsoft Corp., Redmond, WA, USA), the Standard Curve app in the ThermoFisher Connect Platform (ThermoFisher Scientific, Waltham, MA, USA), or Bio-Rad's CFX Maestro for Mac App (Bio-Rad Laboratories, Hercules, CA, USA). Samples with non-exponential amplification were considered false-positive. Gene copies were calculated using the average of the two

replicate reactions and the associated standard curves. To calculate gene copies/L in the original sample, qPCR and dPCR values were adjusted using the effective volume assayed of each sample, or the initial volume of wastewater processed by qPCR after concentration and extraction (Philo *et al.*, 2021). King County qPCR and dPCR ARG data were further adjusted using the daily influent flow and the estimated population served by each wastewater treatment plant using Equation 2. $C_{ARG}$ is the concentration of ARGs in gc/L from the PCR reaction, $Q_{Inf}$ is the daily influent flow at the WWTP, and population is the estimated number of people served by the WWTP (2022). This adjustment was chosen because flow does not directly scale to population served in these WWTPs. Both COVID-19 case data and dPCR and qPCR results and were maintained in Microsoft Excel. Statistical comparisons, data manipulation, and data visualization was carried out using RStudio (2019 RStudio: Integrated Development for R. RStudio, Inc., Boston, MA, USA, URL http://www.rstudio.com/).

Equation 2: $\left[ARG_{adj.}\right] = \frac{(C_{ARG} \times Q_{Inf.})}{population}$

To assess trends in ARG abundance across the study period, random effects models were fit for the Portugal and King County datasets, each regressing log-transformed gene-copies on day with a random effect for each sampling site. This assumes gene-copies within a site are more similar than gene-copies between sites. To determine if metro areas with hospital effluent and WWTP influent trended together in Portugal, Pearson's correlation coefficients were calculated for samples collected in the same week. To test for differences in ARG abundance between qPCR and dPCR, Wilcoxon Signed Rank tests were carried out on paired samples using the built-in R function (wilcox.test(paired = TRUE)).

### 3.1.4 Results and Discussion

#### 3.1.4.1 COVID-19 Cases

The average number of daily incident COVID-19 cases for Portugal from the beginning of the pandemic until August 18, 2020, is 243 (95% CI: 215 – 272). During the study period from August 18 to November 25, 2020, the average number of daily incident COVID-19 cases is 1,940 (95% CI: 1.562 – 2,318). Cases continued to rise until November 11, 2020, when they peaked at 6,297 incident cases. Cases started to drop until the end of year holidays, when they rose rapidly again and peaked at 15,251 incident cases on January 28, 2021. The average number of daily incident COVID-19 cases in Seattle from the beginning of the pandemic until the beginning of this study is 21 (95% CI: 18 – 24). During the study period, the average number of daily incident cases is 88 (95% CI: 78 – 99) peaking at 295 cases on November 30, 2020. As in Portugal, cases began to drop in December and started rising around the end of year holidays. This rise peaked at 285 incident cases on January 4, 2021.

#### 3.1.4.2 Antibacterial-Resistance Gene Abundance

It has been suggested that WWTPs serve as a reservoir for AMR (Guo *et al.*, 2017), due to the presence of both antibiotic residues in the wastewater and numerous bacteria of human and animal origin that harbor ARGs (Rizzo *et al.*, 2013). ARGs and AMR bacteria are routinely detected at WWTPs. However, when ARGs are detected, the genetic context around them and if they are intra- or extracellular is not always known. While intracellular ARGs are more common in wastewater than extracellular ARGs (Zarei-Baygi & Smith, 2021), extracellular ARGs are more likely to move between species of bacteria via horizontal gene transfer. Surveilling ARGs in wastewater can help give a better idea of the underlying prevalence in the population.

This study looks at a wide variety of ARGs belonging to different resistance families. The beta-lactamase genes $bla_{SHV}$ and $bla_{CTX\text{-}M}$ tested for in this study are among the most identified extended-spectrum beta-lactamase (ESBL) genes (Poirel *et al.*, 2012, Roschanski *et al.*, 2014) and confer resistance against later generation cephalosporins. These genes are particularly concerning in Enterobacterales spp. (Roschanski *et al.*, 2014, Centers for Disease Control and Prevention, 2019). The most abundant gene target on average during the sampling period in both Portugal and King County is $bla_{SHV}$, with $1.85x10^8$ gene copies/L and $4.26x10^{10}$ gene copies/person/day, respectively (Table 3.1.7). The abundance of $bla_{SHV}$ in other studies varies widely. Quach-Cu *et al.* (2018) found high abundance of $bla_{TEM}$ and $bla_{SHV}$ in the same qPCR assay, with about $10^7$ copies of these genes per L of wastewater. However, Harnisz & Korzeniewska (2018) found that $bla_{SHV}$ was among the least common genes detected in bacteria isolates from wastewater using a different PCR assay.

The least abundant gene targets in Portugal are $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ with $4.76x10^3$ gene copies/L (Table 3.1.7). However, $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ is highly abundant in King

**Table 3.1.7**: Average ARG abundance in Portugal (gc/L) and King County (gc/person/day) across the study period.

| Metro | WW Type | $bla_{CMY}$ | $bla_{shv}$ | $bla_{CTX\text{-}M\text{-}1,\ 9}$ | $bla_{CTX\text{-}M\text{-}2,\ 8,\ 25}$ | *mcr1* | *mecA* | *vanA* | *n* |
|---|---|---|---|---|---|---|---|---|---|
| North 1 | WWTP | 3.38E+05 | 1.19E+07 | 1.65E+07 | 7.72E+03 | 7.80E+03 | 1.11E+03 | 7.91E+02 | 8 |
| North 1 | Hospital | 1.81E+04 | 1.28E+09 | 6.13E+06 | 2.99E+02 | 9.70E+02 | 2.60E+04 | 3.17E+04 | 6 |
| North 2 | WWTP | 1.01E+05 | 3.57E+07 | 1.57E+07 | 5.12E+03 | 2.52E+04 | 2.64E+03 | 1.16E+03 | 7 |
| North 2 | Hospital | 8.52E+04 | 1.33E+08 | 5.75E+07 | 4.04E+02 | 2.95E+03 | 4.65E+04 | 7.02E+04 | 7 |
| South 1 | WWTP | 8.24E+04 | 3.54E+07 | 1.21E+07 | 6.05E+03 | 2.60E+04 | 2.56E+03 | 1.30E+03 | 8 |
| South 2 | WWTP | 8.71E+04 | 9.19E+07 | 1.25E+07 | 5.88E+03 | 6.20E+04 | 7.34E+02 | 4.47E+03 | 8 |
| South 2 | Hospital | 1.61E+04 | 3.70E+07 | 6.89E+06 | 5.20E+03 | 5.74E+03 | 5.81E+02 | 1.61E+05 | 7 |
| South 3 | WWTP | 5.81E+04 | 7.90E+07 | 9.37E+06 | 5.27E+03 | 3.41E+04 | 1.86E+03 | 5.05E+02 | 8 |
| **Portugal Total** | | **1.03E+05** | **1.85E+08** | **1.70E+07** | **4.76E+03** | **2.27E+04** | **1.04E+04** | **3.66E+04** | **59** |
| South Plant | | 1.00E+09 | 4.87E+10 | 8.49E+07 | 7.98E+08 | 2.89E+04 | 4.92E+04 | 2.26E+05 | 13 |
| West Point | | 1.14E+09 | 3.65E+10 | 2.25E+09 | 3.52E+10 | 5.38E+04 | 6.10E+06 | 3.37E+06 | 13 |
| **King County Total** | | **1.07E+09** | **4.26E+10** | **1.17E+09** | **1.73E+10** | **3.35E+04** | **2.68E+06** | **1.87E+06** | **26** |

County, with $1.73x10^{10}$ gc/person/day. Calero-Cáceres *et al.* (2014) detected about three log gene copies per mL of wastewater of $bla_{CTX\text{-}M\ Group\ 1}$, while Harnisz & Korzeniewska (2018) only detected $bla_{CTX\text{-}M}$ in a handful of isolates, suggesting low abundance compared to the currently presented data. However, because $bla_{CTX\text{-}M}$ genes are frequently associated with mobile genetic elements, detection at any abundance suggests the genes can move easily (Cantón *et al.*, 2012). The last beta-lactamase gene tested for in this study, $bla_{CMY}$, is of particular importance because it is plasmid associated and can therefore move easily between bacteria (Roschanski *et al.*, 2014). On average, $bla_{CMY}$ was more abundant in King County than Portugal (Table 3.1.7). However, both King County and Portugal had high concentrations on average, with $1.07x10^{9}$ gene copies/person/day and $1.03x10^{5}$ gene copies/L, respectively. Other evidence of $bla_{CMY}$ quantification in wastewater is limited, with many studies reporting detection of $bla_{CMY}$ but not quantification (Kwak *et al.*, 2015, Zhang *et al.*, 2019). Previous work in Washington State identified $bla_{CMY}$ in 70.9% of wastewater samples (Zhang *et al.*, 2019), which is lower than in the current study in which $bla_{CMY}$ is detected in all samples in King County. Quantification in wastewater relative to the 16s rRNA gene has been carried out, but this is limited by the efficiency of 16s amplification. A study of hospital effluent in Finland found that $bla_{CMY}$ is $10^{-5}$ to $10^{-4}$ times less abundant than 16s rRNA (Majlander *et al.*, 2021).

Of the remaining genes tested for in this study, all have shown prior evidence of detection in wastewater with varied concentrations. Using qPCR, *mecA* abundance in untreated wastewater ranged from $3.05x10^{2}$ to $2.68x10^{3}$ gene copies/mL. The average *vanA* abundance in the same wastewater is $8.28x10^{5}$ gene copies/mL (Zieliński *et al.*, 2020). In that publication, *mecA* and *vanA* are at higher levels than what was detected

in Portugal and King County, on average (Table 3.1.7). The least abundant gene target in King County is *mcr1*, with $3.35 \times 10^4$ gene copies/person/day on average (Table 3.1.7).

### 3.1.4.3 Trends Over Time

The main goal of this study was to determine if environmental AMR changed during the late 2020 surge in COVID-19 cases before medications had been developed. To answer this question, random effects models were fit for Portugal and King County data each regressing log-transformed gene-copies on day with a random effect for each site to allow for different baseline ARG abundance by site. In Portugal, the only two genes that significantly increased over the course of the sampling period are *mecA* and $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ (Table 3.1.8, Appendices Figure 3.3.1). Across the study period in Portugal, the average *mecA* concentration is 1.05 times higher than the previous day (95% CI: 1.041 – 1.062, $p < 0.001$). Additionally, *mecA*, concentrations tend to be higher in hospital effluent compared to WWTP influent, particularly in the north regions. This suggests that the hospital signal is still detectable at the WWTPs but has been diluted throughout the system. However, in the south 2 region, *mecA* is higher in the WWTP influent than the hospital effluent. This region is a large metro area and WWTP influent contains wastewater from other hospitals and clinics that could be increasing the signal. In King County, *mecA* also increased during the study period. Average daily concentrations are 1.01 times higher than the previous day (1.004 – 1.013, $p < 0.001$). One other qPCR target, $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$, increased over the study period. Average daily concentration is 1.01 times higher than the previous day (95% CI: 1.002 – 1.021, $p = 0.02$). However, these results are likely heavily influenced by relatively higher concentrations in several of the samples at the end of the study. Additionally, changes in $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ in Portugal and *mecA* in King County are small. It is likely that

qPCR is not precise enough and these results should be assessed critically. There is no statistical evidence that any of the other genes in Portugal changed over time, which aligns with the data visually (Table 3.1.8, Appendices Figure 3.3.1).

There is strong clinical evidence to back up the rise in *mecA* detected in Portugal and King County over the course of the study period. First, *S. aureus* is the most isolated organism in intensive care units (ICUs) and the incidence of MRSA ventilator associated pneumonia (VAP) rises to over 30% after five days of mechanical ventilation (Lam & Wunderink, 2006). Because COVID-19 patients are often ventilated for over a week (Hazard *et al.*, 2020), they are also at increased risk for MRSA infections. A review by Bassetti *et al.* (2022) found that MRSA is the most common secondary

**Table 3.1.8**: Results from random effects models assessing changes in log-transformed gene abundance over time in Portugal and King County. Both *mecA* and $bla_{CTX\text{-}M\text{-}2,\ 8,\ 25}$ significantly increase in the Portuguese wastewater during the sampling period. All beta-lactamase genes significantly decrease in the King County wastewater, but these are not adjusted for increased flow during the winter months due to higher precipitation.

| | **Portugal** | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 95% CI | | | 95% CI | | |
| Targets | Est. | Lower | Upper | Exp. Est. | Lower | Upper | p-value |
| $bla_{CMY}$ | 0.0093 | 1.20E-05 | 0.019 | 1.01 | 1.000 | 1.019 | 0.06 |
| $bla_{SHV}$ | -0.0033 | -0.020 | 0.013 | 1.00 | 0.980 | 1.013 | 0.7 |
| $bla_{CTX\text{-}M\ Groups\ 1,\ 9}$ | 0.0046 | -0.0023 | 0.012 | 1.00 | 0.998 | 1.012 | 0.2 |
| $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ | 0.011 | 0.0022 | 0.021 | 1.01 | 1.002 | 1.021 | 0.02 |
| *mecA* | 0.050 | 0.040 | 0.060 | 1.05 | 1.041 | 1.062 | < 0.001 |
| *vanA* | 0.018 | -0.0019 | 0.038 | 1.02 | 0.998 | 1.039 | 0.08 |
| *mcr1* | 0.0024 | -0.0058 | 0.011 | 1.00 | 0.994 | 1.011 | 0.57 |
| | **King County, WA** | | | | | | |
| | | 95% CI | | | 95% CI | | |
| | Est. | Lower | Upper | Exp. Est. | Lower | Upper | p-value |
| $bla_{CMY}$ | -0.0082 | -0.015 | -0.0017 | 0.99 | 0.985 | 0.999 | 0.02 |
| $bla_{SHV}$ | -0.0088 | -0.017 | -0.00021 | 0.99 | 0.983 | 1.000 | 0.06 |
| $bla_{CTX\text{-}M\ Groups\ 1,\ 9}$ | -0.0035 | -0.011 | 0.0037 | 0.99 | 0.989 | 1.004 | 0.34 |
| $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ | -0.0090 | -0.022 | 0.0037 | 0.99 | 0.978 | 1.004 | 0.17 |
| *mecA* | 0.0088 | 0.0044 | 0.013 | 1.01 | 1.004 | 1.013 | <0.001 |
| *vanA* | 0.0070 | -0.0012 | 0.016 | 1.01 | 0.999 | 1.016 | 0.11 |
| *mcr1* | -0.0020 | -0.012 | 0.0077 | 1.00 | 0.988 | 1.008 | 0.69 |

infection in ventilated COVID-19 patients. Additionally, the US CDC showed that hospital onset MRSA cases increased 13% in 2020 compared to 2019 (2022). It is likely that rising MRSA infections associated with COVID-19 patients led to the increase in *mecA* wastewater abundance detected in this study. The current study does not attempt to isolate MRSA from the wastewater or sequence the genes to determine their origin, but that poses an interesting area for future work. Additionally, future work should identify MRSA infections and antibiotics used in the hospitals sampled in this study to better describe this relationship. Finally, future studies should compare detected ARGs with the numbers of hospitalized and ventilated patients to better understand the relationship between the clinics and environmental ARGs.

In King County, after adjustment for flow and estimated population, the only other gene target tested for that changed was $bla_{CMY}$. (Table 3.1.8, Appendices Figure 3.3.2). Over the course of the study, average daily *blaCMY* abundance was 0.99 times smaller than the previous day (95% CI: 0.985 – 0.999, p = 0.02). As with *mecA*, this change is small. Data on AMR infections, VAP in the hospitals, and antibiotics used in the water catchment area would further support these data showing changes in gene targets.

The US CDC identified increased AMR during the first year of the pandemic in all the organisms they track for which data was available in 2022. This includes bacteria with some of the ARGs tested for in this study, like ESBL-Enterobacterales, methicillin-resistant *S. aureus* (MRSA), and vancomycin-resistant Enterococcus (VRE) (Centers for Disease Control and Prevention *et al.*, 2022). A recently published guidance document for monitoring AMR in wastewater suggests that qPCR can be used to quantify ARG removal in a system because it is more high throughput than other molecular techniques and is not specific to ARGs in certain bacteria (Liguori *et al.*, 2022). While

this study was not able to detect changes for most of the ARG targets in wastewater, this study demonstrates that qPCR can be used for general surveillance and to assess changes in gene abundance over time in certain settings.

The current study is also important because there is little other work assessing changes in environmental AMR during the pandemic. As of December 2022, there is one other manuscript trying to understand this question (Harrington *et al.*, 2022). They collected primary influent wastewater weekly from a WWTP in Nevada from November 2020 through the end of January 2021 and found significant increases in beta-lactam and fluoroquinolone ARGs in December 2020 associated with rising COVID-19 cases (Harrington *et al.*, 2022). Harrington *et al.* (2022) were also able to show significant decreases in ARGs from December to January when COVID-19 cases again dropped. It is possible the current study did not assay samples frequently enough to detect changes.

#### 3.1.4.4 Correlations

Pearson's correlation coefficients were calculated for log transformed ARG abundances in hospital effluent and WWTP influent samples collected in the same week

**Table 3.1.9**: Pearson's Correlation Coefficients for ARG abundance within water catchment areas in Portugal. 95% confidence intervals are not provided for comparisons with three or fewer pairs.

| | North 1 | | | | North 2 | | | | South 2 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 95% CI | | | | 95% CI | | | | 95% CI | | |
| Targets | Est. (n) | Lower | Upper | *p* | Est. (n) | Lower | Upper | *p* | Est. (n) | Lower | Upper | *p* |
| *mecA* | 0.83 (4) | -0.65 | 1.00 | 0.17 | 0.65 (4) | -0.83 | 0.99 | 0.35 | 0.23 (3) | - | - | 0.85 |
| *vanA* | 0.91 (4) | -0.42 | 1.00 | 0.09 | 0.30 (4) | -0.93 | 0.98 | 0.70 | 0.16 (7) | -0.68 | 0.81 | 0.74 |
| *mcr1* | -0.59 (6) | -0.95 | 0.43 | 0.22 | -0.91 (3) | - | - | 0.27 | -0.56 (5) | -0.97 | 0.64 | 0.33 |
| $bla_{CMY}$ | -0.79 (6) | -0.98 | 0.05 | 0.06 | -0.72 (4) | -0.99 | 0.78 | 0.28 | 0.42 (7) | -0.49 | 0.89 | 0.35 |
| $bla_{SHV}$ | 0.95 (6) | 0.59 | 0.99 | <0.01 | 0.93 (4) | -0.30 | 1.00 | 0.07 | 0.80 (7) | 0.13 | 0.95 | 0.03 |
| $bla_{CTX\text{-}M\ Groups\ 1,\ 9}$ | 0.56 (6) | -0.46 | 0.94 | 0.24 | -0.24 (4) | -0.98 | 0.94 | 0.76 | 0.69 (7) | -0.13 | 0.95 | 0.08 |
| $bla_{CTX\text{-}M\ Groups2,\ 8,\ 25}$ | 0.76 (5) | -0.38 | 0.98 | 0.14 | -0.76 (4) | -0.99 | 0.74 | 0.24 | 0.43 (7) | -0.49 | 0.89 | 0.34 |

**Table 3.1.10**: Average gc/person/day for each gene target in subset of King County samples obtained using qPCR and dPCR with Wilcoxon signed-rank test p-values.

| | qPCR | | | dPCR | | | |
|---|---|---|---|---|---|---|---|
| | | 95% CI | | | 95% CI | | |
| *Targets* | Average | Lower | Upper | Average | Lower | Upper | *p* |
| *mecA* | 3.21E+06 | 7.44E+05 | 5.68E+06 | 6.58E+06 | 4.64E+06 | 8.52E+06 | 0.0012207 |
| *vanA* | 1.33E+06 | -1.21E+05 | 2.78E+06 | 2.70E+07 | 1.16E+07 | 4.24E+07 | 0.00048828 |
| $bla_{CMY}$ | 1.29E+09 | 3.10E+08 | 2.28E+09 | 3.28E+10 | 1.25E+10 | 5.31E+10 | 0.00012207 |
| $bla_{SHV}$ | 4.38E+10 | 2.40E+10 | 6.36E+10 | 1.16E+10 | 5.22E+09 | 1.80E+10 | 0.00012207 |
| $bla_{CTX\text{-}M\ Groups\ 1,9}$ | 1.12E+09 | 2.34E+08 | 2.00E+09 | 2.57E+10 | 1.51E+10 | 3.63E+10 | 0.00012207 |
| $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ | 2.26E+10 | 8.23E+08 | 4.44E+10 | 2.54E+09 | 1.25E+09 | 3.84E+09 | 0.26757813 |

from the same water catchment areas in Portugal (Table 3.1.9). There were three total areas to compare, two in the north and one in the south. There was little statistical evidence that ARGs within metro areas trended together, except $bla_{SHV}$, which showed strong correlations in concentrations in the North 1 and South 2 regions (Table 3.1.9). Correlation coefficients were 0.95 (95% CI: 0.59 – 0.99, $p < 0.001$) and 0.80 (95% CI: 0.13 – 0.95, $p = 0.03$), in North 1 and South 2, respectively. It is important to note that not all samples within a metro area were collected on the same day and they are not directly related to each other. This likely reduces our ability to associate ARG concentrations within water catchment areas. Future work should take into consideration residence time of the wastewater conveyance system to aim to collect samples that are more closely associated with each other.

### 3.1.4.5 dPCR Method Comparisons

To understand if there are differences in results between molecular detection platform, adjusted gene abundances obtained using dPCR and qPCR (gene copies/person/day) were statistically compared using Wilcoxon signed rank tests. Wilcoxon signed rank tests shows that there is a statistically significant change in gene copies/L between qPCR and dPCR for all ARGs assayed except $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ (Table 3.1.10). Mean gene copies/person/day were higher for dPCR for all targets except $bla_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$. These data show the chosen detection platform may have a

substantial effect on final abundance calculations. There have long been concerns that inhibitors present in genomic extracts inhibit amplification in qPCR and RT-qPCR (Demeke & Jenkins, 2010), particularly for extracts of large volume environmental water samples (Gibson *et al.*, 2012). Digital-droplet reverse-transcriptase PCR (RT-ddPCR) is less inhibited by wastewater inhibitors than RT-qPCR at medium and low levels of inhibitors (Rački *et al.*, 2014). Additionally, SARS-CoV-2 was more likely to be detected in wastewater concentrate processed by RT-dPCR than RT-qPCR (Ahmed *et al.*, 2022). Finally, dPCR results do not depend on a standard curve to quantify the samples, reducing between-reaction variability (Quan *et al.*, 2018). Data presented in this paper support the utility of dPCR over qPCR for sample quantification.

### 3.1.5 Conclusions and Limitations

This study is the first study to describe AMR in wastewater during the pandemic in two different countries and is only one of two to try to quantify changes in environmental AMR due to the SARS-CoV-2 pandemic. The data presented here were able to show increases in two gene targets in Portugal, indicating that qPCR can be used to supplement hospital data for AMR in certain settings. However, these data are limited in several ways. First, the samples were initially collected, concentrated, and extracted for surveillance of an RNA virus, SARS-CoV-2. Methods optimized for assaying viral RNA may not be as efficient for bacterial DNA. We could be systematically underestimating DNA abundances in these samples, particularly for King County where samples were extracted with a ViralRNA kit. Second, samples were selected to cover a longer time rather than samples as frequently as possible. This could have obscured true changes in ARG abundance over time. Finally, this study does not attempt to describe which organisms are harboring the detected ARGs and the

genetic context around the ARGs. This leaves numerous questions about the dangers these ARGs pose to public health unanswered. Future work should aim to collect samples more frequently and to better understand AMR infections at the time of sample collection. There is also a large unknown about specific infection control practices and standards of care in the hospitals in Portugal. Data on the number and types of antibiotics used, as well as how patients are typically treated will help shed light on the utility of wastewater surveillance for ARGs. Additionally, molecular methods should be used that are less subject to inhibitors. Concentrating wastewater using skimmed milk flocculation results in significant inhibition of the PCR assays, and the protocol was optimized for isolating SARS-CoV-2 RNA, not bacterial DNA. This could result in an underestimation of ARG abundance in King County. Finally, sequencing should be carried out to describe the genetic context of detected ARGs in wastewater.

### 3.1.6 Acknowledgments

Statistical advice for this project was provided by students and faculty through the Department of Biostatistics Consulting Program at the University of Washington, Seattle. Sample collection was facilitated by the West Point Process Lab, the South Treatment Plant collection team, and collaborators in Portugal. Research reported in this publication was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under award number T32ES015459. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This project was also funded by the US Fulbright Student Program. This work could also not have been carried out without the support of all members of the Environmental and Occupational Health Microbiology lab

in Seattle, WA and the Laboratório Análises do Instituto Superior Técnico in Lisbon, Portugal.

## 3.2 Antibacterial Resistance Genes in Rural and Urban Wastewater Treatment Plants

### 3.2.1 Abstract

Antimicrobial resistance is a growing public health concern. With the widespread use of antibiotics in medical facilities and agriculture, numerous bacterial infections are no longer treatable or require later-generation broad-spectrum antibiotics that have more side effects. Much of the existing infrastructure for antimicrobial resistance surveillance is in medical facilities and focuses on treating infections. However, the recent expansion of wastewater surveillance for the SARS-CoV-2 pandemic has resulted in numerous calls to incorporate additional targets, including antimicrobial resistant bacteria and associated genes. The goal of the current project is to compare seven antibacterial resistance genes between urban and rural or peri urban treatment plants in Portugal and King County. While there are many studies looking at resistance in samples collected from single wastewater treatment plants, not many include samples from multiple countries and treatment plants serving both densely populated and rural populations. There was no statistical evidence for differences in resistance genes between the treatment plants in Portugal, while every gene except for *mcr1* was found to be statistically different between the treatment plants in King County. Data collected in this project can help understand how different communities contribute to environmental antibacterial resistance to better implement control programs.

### 3.2.2 Introduction

Antimicrobial resistance (AMR) has been detected for almost as long as antibiotics have been in use. Within a few years of the discovery of penicillin and clinical use, resistant bacteria had emerged (Saga & Tamaguchi, 2009, Hutchings *et al.*, 2019).

Surveillance is crucial to understand the scope of the problem. Historically, AMR surveillance has focused on clinical settings. The first international AMR surveillance program was the European Antimicrobial Resistance Surveillance Network which only included data from clinical labs for invasive bacterial isolates (Aarestrup & Woolhouse, 2020). The World Health Organization (WHO) later launched the Global Antimicrobial Resistance Surveillance System in 2015 to standardize surveillance in clinical settings (Tornimbene *et al.*, 2018), but this program also does not include environmental surveillance. More recently, the Global Tricycle Surveillance project established protocols for environmental sampling, but this project focuses on extended-spectrum beta-lactamase (ESBL) *E. coli* and does not search for other antibacterial-resistance genes (ARGs) (World Health Organization, 2021).

Wastewater surveillance was widely adopted during the SARS-CoV-2 pandemic to supplement clinical surveillance. There have been calls to capitalize on the expanded wastewater surveillance capacity by incorporating new targets such as ARGs into existing wastewater surveillance programs (Pruden *et al.*, 2021, Centers for Disease Control and Prevention, 2022). The environment is both a reservoir of ARGs and a facilitator of ARG movement between bacteria (Thakur & Gray, 2019). ARGs are considered emerging contaminants because they can persist in microbial populations and the environment and may not be removed by wastewater treatment (Pruden *et al.*, 2006, Quach-Cu *et al.*, 2018, Pärnänen *et al.*, 2019). Molecular methods have the potential to vastly improve upon culture methods for AMR surveillance. Molecular methods such as metagenomic sequencing and highly parallel qPCR arrays have shown that ARG abundance in wastewater is highly correlated with socioeconomic and environmental factors and mirrors resistance detected in clinics (Hendriksen *et al.*,

2019, Pärnänen *et al.*, 2019). Despite the promise of these methods, there are no standardized protocols or reporting mechanisms for sampling and assaying wastewater for ARGs (Pruden *et al.*, 2021). While frameworks for environmental surveillance (ES) of AMR have been suggested (Huijbers *et al.*, 2019, Liguori *et al.*, 2022), national and international ES programs do not standardize protocols between sampling labs (Centers for Disease Control and Prevention, 2022, Larsson *et al.*, 2022).

This study aims to quantify differences in ARG targets between differently sized wastewater treatment plants (WWTPs) in two geographically distinct international communities. Primary influent wastewater was sampled from four different WWTPs using methods previously shown to be successful for ARG testing. ARGs were detected using qPCR and differences between large urban and smaller rural/peri urban were statistically assessed. Results establish that ARG abundance between geographically close but distinct communities can be different. Studying AMR only at a state or national level may mask these small geographic differences.

## 3.2.3 Methods

### 3.2.3.1 Portugal Wastewater Sampling and Concentration

Wastewater was collected from two WWTPs in Portugal serving geographically distinct communities every other week (Table 3.2.1). One WWTP serves an urban community of over 200,000 people and contains two hospitals in the water catchment area. It is additionally near several large transportation centers and is heavily populated. It can process up to 54,500 cubic meters of wastewater a day. The other treatment plant serves a rural community of just under 30,000 people. It can process just over 4,000 cubic meters of wastewater a day and is in a historical agriculture center for the country. Twenty-four-hour composite samples were collected using autosamplers. All samples

were transported in a refrigerated bag at 5°C with temperature control (+/- 3°C) to the Laboratório Análises do Instituto Superior Técnico (LAIST).

Wastewater was concentrated using hollow-fiber ultrafiltration according to (Monteiro *et al.*, 2022) using Inuvai R180 filters with a molecular weight cut off less than or equal to 18.8 kDa (Inuvai, Fresenius Medical Care, Bad Homburg, Germany). Filters were first primed with 0.9% v/V NaCl solution, and then wastewater was filtered at a flow of 200 – 250 mL/min. The sample was eluted from the filters using 1x PBS supplemented with 0.01% sodium polyphosphate, 0.01% Tween 80, and 0.001% antifoam for a total elution volume of 300mL. The elution underwent secondary concentration with polyethylene glycol (PEG) precipitation. PEG 8000 (25% v/V) and NaCl (7.17g per 300mL) was added to the eluate and the sample bottles agitated until it completely dissolved. The samples were then rocked overnight at 4°C and centrifuged the next day at 10,000G for 30 minutes. Sample pellets were resuspended in 5.0mL of 1xPBS and stored at -80°C until extraction. A complete concentration protocol is included in the appendices in Section 3.3.1.

Genetic material was extracted from the concentrate using the QIAamp Fast DNA Stool Mini Kit (QIAGEN, Hilden, Germany) following the manufacturer's protocol for human DNA analysis and the following modifications. The input material was 220uL

**Table 3.2.1**: Sampling dates for wastewater collected in Portugal and King County.

| **Portugal** | | **King County** | |
|---|---|---|---|
| *Urban* | *Rural* | *Urban* | *Rural* |
| 03/07/22 | 03/07/22 | 10/25/22 | 10/24/22 |
| 03/24/22 | 03/24/22 | 11/01/22 | 10/31/22 |
| 04/04/22 | 04/04/22 | 11/08/22 | 11/07/22 |
| 04/20/22 | 04/20/22 | 11/15/22 | 11/14/22 |
| 05/02/22 | 05/02/22 | 11/22/22 | 11/21/22 |
| 05/09/22 | 05/10/22 | | |

of concentrate, and the final elution volume was 100uL. All extracts were stored at -20°C until analysis.

#### 3.2.3.2 King County Wastewater Sampling and Concentration

Primary influent wastewater was collected weekly by grab sample from two WWTPs in King County (Table 3.2.1). As with Portugal, these WWTPs were selected because one serves a larger urban center and the other serves a larger peri-urban/rural community. The urban treatment plant serves roughly 780 thousand people and processes on average 95 million gallons of sewage a day. There are two major medical centers in the water catchment area with both pediatric and adult level 1 trauma centers and referral hospitals. The peri-urban treatment plant serves about 268 thousand people and 18 million gallons a day, on average (2022). There were five samples collected from each WWTP for a total of 10 samples. Samples were transported on ice to the Environmental and Occupational Health Microbiology Lab (EOHML) the same day of collection. Samples were held at 4°C and processed within 48 hours of arrival to the EOHML.

Samples were concentrated following the protocol in Philo *et al.* (2022). A full step by step protocol is included in section 2.6.2. Briefly, a 5% skimmed milk solution was added to 0.1L of wastewater (1% v/V final) and the pH dropped to 3 – 4 using 5M HCl. The solution was then shaken at 200RPM for two hours and divided into 50mL conicals, centrifuged at 3500G and 4°C for 30 minutes. The supernatant was poured off to produce two pellets per 100mL of wastewater.

Buffer AVL from the QIAamp Viral RNA Mini Kit (QIAGEN, Germantown, MD, USA) was prepared with carrier RNA according to the manufacturers protocol. Each pellet was resuspended in 560uL of buffer AVL and transferred to a PowerBead Pro

**Table 3.2.2**: Primer and probe sequences for ARG targets.

| Target | Name | Sequence (5` - 3`) | Portugal Probe | Seattle Probe | Reference |
|---|---|---|---|---|---|
| *bla*$_{CMY}$ | CMY_fwd<br>CMY_rev<br>CMY_probe | GGCAAACAGTGGCAGGGTAT<br>AATGCGGCTTTATCCCTAACG<br>CCTACCGCTGCAGATCCCCGATG | ROX | FAM | Roschanski *et al.* (2014) |
| *bla*$_{SHV}$ | SHV_fwd<br>SHV_rev<br>SHV_probe | TCCCATGATGAGCACCTTTAAA<br>TCCTGCTGGCGATAGTGGAT<br>TGCCGGTGACGAACAGCTGGAG | Cy5 | FAM | Roschanski *et al.* (2014) |
| *bla*$_{CTX-M}$ | CTX-A_fwd<br>CTX-A_rev<br>CTX-A_p | CGGGCRATGGCGCARAC<br>TGCRCCGGTSGTATTGCC<br>CCARCGGGCGCAGYTGGTGAC | Yak. Yellow | FAM | Roschanski *et al.* (2014) |
| *bla*$_{CTX-M}$ | CTX-B_fwd<br>CTX-B_rev<br>CTX-B_p | ACCGAGCCSACGCTCAA<br>CCGCTGCCGGTTTTATC<br>CCCGCGYGATACCACCACGC | Yak. Yellow | FAM | Roschanski *et al.* (2014) |
| *mcr*-1 | mcr-1-FW<br>mcr-1-RV | ACGCCATCTGCAACACCAA<br>GCCAACGAGCATACCGACA | - | - | Donà *et al.* (2017) |
| *mecA* | mecA-F<br>mecA-R<br>mecA-p | AAAACTAGGTGTTGGTGAAGATATACC<br>GAAAGGATCTGTACTGGGTTAATCAG<br>TTCACCTTGTCCGTAACCTGAATCAGCT | FAM | FAM | Sabet *et al.* (2007) |
| *vanA* | vanA-F<br>vanA-R<br>vanA-P | GCCGGAAAAAGGCTCTGAA<br>TTTTTTGCCGTTTCCTGTATCC<br>CGCAGTTATAACCGTTCCCGCAGACC | FAM | FAM | He *et al.* (2020) |

tube (Cat. No. 19301, QIAGEN, Germantown, MD, USA). The bead tubes were vortexed horizontally for 10 minutes and then centrifuged at 15,000G for one minute to pellet the solids. The supernatant was transferred to a clean 1.5mL tube without disturbing the solids or beads. Ethanol was added to the 1.5mL tube (560uL, 100% EtOH) and mixed by pulse vortexing for 15 seconds. The mixture was then applied to columns and centrifuged at 15,000G for three minutes and repeated until the entire sample passed through the columns. One column is used per pellet for two columns per 100mL wastewater sample.

The remainder of the manufacturer's spin protocol was followed until the elution step. 40uL of Buffer AVE was added to each column and incubated for one minute. After elution, the same 40uL was applied back to the column for a second elution. The elution from replicate columns was mixed and re-aliquoted for a total volume of 80uL. Samples could either be immediately assayed or stored at -20°C for long term storage.

### 3.2.3.3 qPCR

Samples in Lisbon, Portugal and King County, WA were tested for genes conferring resistance to extended-spectrum beta-lactams and cephalosporins (*bla*$_{CMY}$, *bla*$_{SHV}$, and *bla*$_{CTX-M}$) (Roschanski *et al.*, 2014), colistin (*mcr1*) (Donà *et al.*, 2017), vancomycin (*vanA*) (He *et al.*, 2020), and methicillin (*mecA*) (Sabet *et al.*, 2007) using pre-existing qPCR assays (Section 3.1, Table 3.2.2). In Portugal, qPCR reactions were carried out on the Applied Biosystems QuantStudio ™ 5 System (Applied Biosystems Corp., Waltham, MA, USA) with Luna ® Universal qPCR Master Mix for SYBR assays or Luna ® Universal Probe qPCR Master Mix for TaqMan probe assays (New England Biolabs, Ipswich, MA, USA). The total reaction volume was 15uL with 1.0uL of extract, 7.5uL of the respective master mix, and variable amounts of water, primers, and probes according to their respective assays (Table 3.2.3). All qPCR reactions in King County used the BioRad CFX qPCR systems and iTaq Universal Probes Supermix or iTaq Universal SYBR Green Supermix (Bio-Rad Laboratories, Hercules, CA, USA). The total reaction volume was 15uL with 7.5uL of mastermix, 2.0uL of extract, and variable amounts of primers, probes, and water (Section 3.1, Table 3.2.3).

**Table 3.2.3**: Primer and probe concentrations and reaction conditions for qPCR in Portugal and King County.

| Assays | Concentrations | Reaction Conditions |
|---|---|---|
| *bla*$_{CMY}$, *bla*$_{SHV}$, *bla*$_{CTX-M}$ | Forward: 0.4uM<br>Reverse: 0.4uM<br>Probe: 0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 50°C for 15 seconds<br>• 70°C for 25 seconds |
| *mcr1* | Forward: 0.2uM<br>Reverse:0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 63°C for 10 seconds<br>• 72°C for 10 seconds<br>Melt curve |
| *mecA* | Forward: 0.4uM<br>Reverse: 0.4uM<br>Probe: 0.2uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 30 seconds<br>• 55°C for 45 seconds |
| *vanA* | Forward: 0.5uM<br>Reverse: 0.5uM<br>Probe: 0.3uM | 95°C for 5 minutes<br>45 cycles:<br>• 95°C for 15 seconds<br>• 60°C for 60 seconds |

All samples were run in duplicate with duplicate 10-fold dilutions. Negative controls (nuclease free water) were included in each reaction. Samples were quantified using standard curves generated with 10-fold dilutions of positive controls. Positive controls used in Portugal were previously described (Section 3.1). Briefly, samples were screened using the qPCR assays with positive reactions cleaned up using gel electrophoresis (2.5% agarose) and the GeneJet Gel Extraction Kit (Thermo Scientific, Waltham, MA, USA). Positive controls in Seattle were gBlocks obtained from IDT (Integrated DNA Technologies, Inc., Coralville, IA, USA).

#### 3.2.3.4 Data Management and Statistical Analyses

All qPCR data were analyzed with the ThermoFisher Connect Platform Standard Curve app (ThermoFisher Scientific, Waltham, MA, USA) or the Bio-Rad CFX Maestro for Mac App (Bio-Rad Laboratories, Hercules, CA, USA). Data were collated and maintained in Microsoft Excel (Microsoft Corp., Redmond, WA, USA). Gene copies were calculated using the average of the two replicate reactions and the associated standard curves. To calculate gene copies/L in the original sample, qPCR values were adjusted using the effective volume assayed of each sample. The effective volume assayed is the initial volume of wastewater processed by qPCR after concentration and extraction (Philo *et al.*, 2021). Statistical comparisons, data manipulation, and data visualization was carried out using RStudio (2019 RStudio: Integrated Development for R. RStudio, Inc., Boston, MA, USA, URL http://www.rstudio.com/). Wilcoxon Rank Sum tests were carried out using the built-in R function (wilcox.test()) to assess the difference in ARG abundance between urban and rural communities.

#### 3.2.3.4.1 *Flow and Population Adjustment*

Daily influent flow rate and the estimated population served (2022) was obtained for the King County area treatment plants to adjust ARG concentrations. Flowrate was not available for the Portugal WWTPs. ARG abundances for each target were adjusted using Equation 1. $C_{ARG}$ is the concentration of ARGs in gc/L from the qPCR reaction, $Q_{Inf}$ is the daily influent flow at the WWTP, and population is the estimated number of people served by the WWTP.

**Equation 1**: $$\left[ARG_{adj.}\right] = \frac{(C_{ARG} \times Q_{Inf.})}{population}$$

## 3.2.4 Results and Discussion

### 3.2.4.1 Antibacterial-Resistance Gene Abundances

The most abundant gene on average in Portugal is *bla*$_{SHV}$ (1.x83$^{7}$ gc/L) and in King County is *bla*$_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ (7.31x10$^{9}$ gc/person/day) (Table 3.2.4). This aligns with previous work (Section 3.1), in which *bla*$_{SHV}$ was the most abundant gene in samples collected in Portugal (Table 3.1.7). However, *blaSHV* abundances are lower in King County samples than in Section 3.1. Additionally, *mcr1* is the least abundant gene detected on average in King County. This gene was not detected in any of the samples collected at the urban WWTP (Table 3.2.4) and was only detected in one sample at the

**Table 3.2.4**: Average abundances of ARGs in Portugal (gc/L) and King County (gc/person/day) across the study period.

| Location | WWTP | *bla*$_{CMY}$ | *bla*$_{shv}$ | *bla*$_{CTX\text{-}M\ Groups\ -1,\ 9}$ | *bla*$_{CTX\text{-}M\ Groups\ 2,\ 8,\ 25}$ | *mecA* | *vanA* | *mcr1* |
|---|---|---|---|---|---|---|---|---|
| King County | Rural | 4.12E+06 | 1.17E+08 | 1.91E+08 | 6.07E+08 | 5.34E+05 | 5.17E+06 | 1.73E+04 |
| King County | Urban | 4.12E+07 | 6.88E+09 | 1.45E+09 | 1.40E+10 | 8.12E+06 | 1.58E+08 | NA |
| **King County Mean** | | **2.27E+07** | **1.83E+07** | **8.21E+08** | **7.31E+09** | **4.33E+06** | **8.17E+07** | **1.73E+04** |
| Portugal | Urban | 5.42E+04 | 2.52E+07 | 1.41E+07 | 1.06E+06 | 7.29E+02 | 2.54E+01 | 3.66E+03 |
| Portugal | Rural | 5.68E+04 | 1.14E+07 | 1.55E+07 | 1.56E+03 | 7.99E+02 | 9.88E+01 | 9.87E+03 |
| **Portugal Mean** | | **5.55E+04** | **1.83E+07** | **1.48E+07** | **5.81E+05** | **7.64E+02** | **5.88E+01** | **7.05E+03** |

**Table 3.2.5**: Wilcoxon Rank-Sum p-values comparing ARG abundance between WWTPs for each region.

| Region | *bla*$_{CMY}$ | *bla*$_{SHV}$ | *bla*$_{CTX\text{-}M\ Groups\ 1,9}$ | *bla*$_{CTX\text{-}M\ Groups\ 2,8,25}$ | *mecA* | *vanA* | *mcr1* |
|---|---|---|---|---|---|---|---|
| *King County* | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | 0.008 | NA |
| *Portugal* | 0.59 | 0.31 | 1 | 0.33 | 0.84 | 0.54 | 0.33 |

rural treatment plant. The data presented here together with data in Section 3.1 suggest there is relative temporal stability in ARGs in the two communities despite sample collection in different seasons and years. The samples collected in the current study were collected in spring and fall 2022, and samples collected in the previous study were collected in winter 2020/21 (Table 3.1.1, 3.1.2). However, sampling locations between the two studies are not the same, which could explain some of the differences. The Global Sewage Surveillance Project has also found temporal stability in antimicrobial resistance by collecting samples from WWTPs all over the world for several years (Munk *et al.*, 2022). The temporal stability in relative ARG abundances in King County and Portugal coincides with the Munk *et al.* (2022) study that resistomes are stable over time.

#### 3.2.4.2 Urban and Rural Differences

Wilcoxon rank-sum tests were carried out comparing ARG abundance in gene copies/L of wastewater between peri-urban and urban settings in King County and Portugal. Every gene target assayed except *mcr1* is significantly different between peri-urban and urban wastewater samples in King County (Table 3.2.5, Figure 3.2.1). On the other hand, there were no significant differences detected between wastewater samples collected in urban and rural settings in Portugal (Table 3.2.5, Figure 3.2.1). It is likely that much of the differences in ARGs can be described by population size. Zhu *et al.* (2022) collected surface water samples from a river along a gradient of urbanization and found that population size was a key parameter for the ARGs detected in those

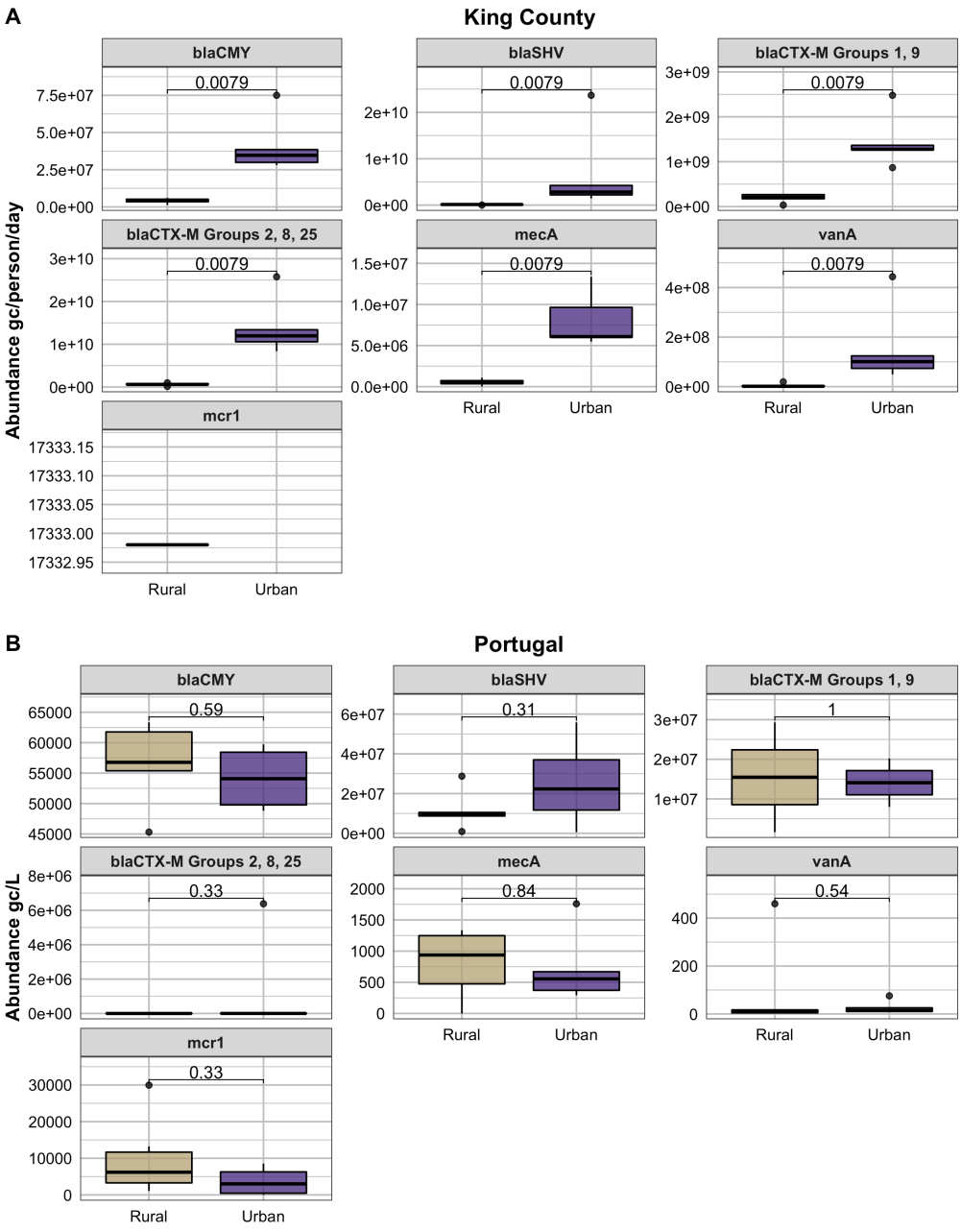



**Figure 3.2.1**: ARG abundance (gc/L) between urban and rural wastewater treatment plants for each gene target and region sampled with Wilcoxon Rank-Sum p-values. ARG abundance is significantly different for each gene target in **A)** King County except *mcr1*. There are no statistically significant differences in ARG distribution between the two **B)** Portuguese treatment plants sampled.

samples. Because the urban WWTP in King County serves about three times as many people as the peri urban WWTP, it is likely that the differences in ARGs are due to population size. However, even though the urban WWTP in Portugal serves roughly six times as many people as the peri urban WWTP, there are no differences in ARGs. Both

Portuguese treatment plants are smaller than the rural treatment plant in King County, suggesting that ARG differences are not as apparent between smaller communities. The data presented here suggest treatment plants serving smaller populations are likely to be more similar regardless of whether they are serving a rural or urban community.

This could be also due to the relative high use of antimicrobials in European pig and poultry farming despite a ban on antimicrobials for growth promotion (Woolhouse *et al.*, 2015). It is well established that livestock agriculture facilities contain large amounts of antimicrobial-resistant bacteria. ARGs are commonly detected at high levels in manure, wastewater, and animals at livestock farms (Tao *et al.*, 2014, Wang *et al.*, 2016, Mobasseri *et al.*, 2019). Because the community around the rural treatment plant in Portugal contains livestock agriculture, there could be higher levels of ARGs relative to the urban WWTP. Additionally, antimicrobials are the second most commonly used pesticide in plant agriculture (Miller *et al.*, 2022). They are commonly applied to plants to prevent infections. This could also be increasing the ARGs in the rural WWTP relative to the urban WWTP in Portugal.

There are very few studies specifically looking at differences in ARGs between urban and rural wastewater treatment plants. There are many studies comparing different cities and countries (Hendriksen *et al.*, 2019, Huijbers *et al.*, 2020, Munk *et al.*, 2022). The metagenomic data coming out of the Global Sewage Surveillance Project suggests that European and North American resistomes are more similar than those in other parts of the world (Munk *et al.*, 2022), but they are potentially losing finer differences when observing data at a country scale. Metagenomics also only offers relative abundances, making direct comparison with qPCR results difficult. Additionally, there are numerous studies show higher ARG abundance in surface waters impacted

by cities compared to rural settings (Ahmed *et al.*, 2021, McInnes Ross *et al.*, 2021, Na *et al.*, 2021, Zhu *et al.*, 2022). However, samples collected in settings impacted by livestock agriculture facilities had more AMR bacteria than in urban settings (Osbiston *et al.*, 2021), indicating that environmental factors are also important.

### 3.2.5 Conclusions

This study is one of the first to try to understand differences in ARG abundance between differently sized WWTPs in two countries. There were significant differences in ARGs between the two WWTPs in King County in all gene targets for which statistical comparisons were possible, while there were no significant differences between the WWTPs for any of the ARG targets in Portugal. However, this paper contributes to a growing body of work supporting the use of wastewater for AMR surveillance. A limitation of this study is the use of qPCR for detection because there are many factors that can reduce the efficiency of qPCR reactions. However, qPCR is more robust than many sequencing methods for absolute quantification. Future work should aim to use methods less susceptible to inhibitors such as dPCR. Additionally, this study is not able to make direct comparisons between King County and Portugal because different wastewater concentration and extraction protocols were used. Future studies should aim to use the same methods at all sites for more comparable data.

### 3.2.6 Acknowledgments

Sample collection was facilitated by the West Point Process Lab, the Brightwater Operations staff, and collaborators in Portugal. Research reported in this publication was supported by the National Institute of Environmental Health Sciences of the National Institutes of Health under award number T32ES015459. The content is solely the responsibility of the authors and does not necessarily represent the official views of

the National Institutes of Health. The work in this project was further supported by the Fulbright US Student Program, Fulbright Portugal, and the Luso – American Development Foundation Open/Study Research award. This work could also not have been carried out without the support of all members of the Environmental and Occupational Health Microbiology lab in Seattle, WA and the Laboratório Análises do Instituto Superior Técnico in Lisbon, Portugal.

# 3.3 Antimicrobial Resistance Appendices

## 3.3.1 AMR and COVID-19

### 3.3.1.1 Hollow-Fiber Ultrafiltration

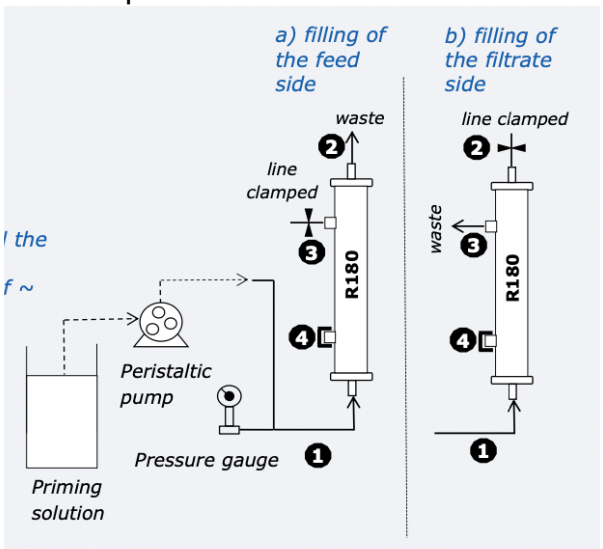
Methods per Monteiro *et al.* (2022) and the Inuvai R180 sampling protocol as provided by LAIST

**Preparation**

1. Collect influent wastewater in 1L polypropylene bottles. Store at 4-8°C for up to 4 days. If storing for longer, freeze at -80°C.
2. Autoclave or ensure available:
    a. 60mL luer-lock syringe
    b. Tubing
    c. 200mL graduated beakers
    d. PEG 8000
    e. 500mL PP bottles
    f. 50mL conicals
    g. Serological pipettes
3. Prepare the following reagents:
    a. 0.9% NaCl solution (~600mL per sample)
    b. Elution Buffer (minimum 300mL per sample) 1x PBS (pH = 7.4) with:
        i. 0.01% 10x NaPP solution
        ii. 0.001% Antifoam + 0.01% Tween 80 solution (initial solution 1% antifoam/10% Tween 80)

**Priming the filter**

1. Set up the filter:

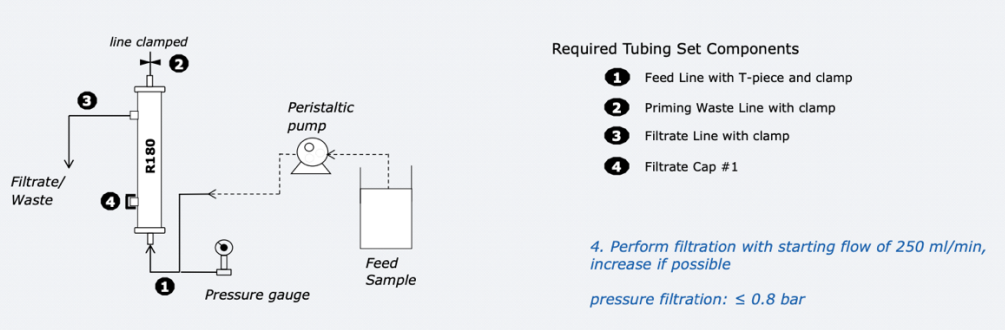



    a. Bottom input tube connected to the peristaltic pump (1)

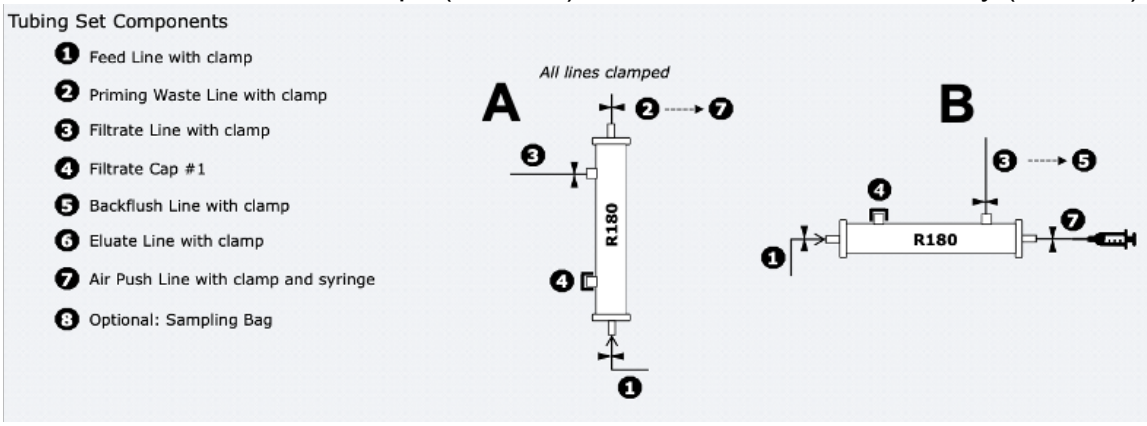
   b. Bottom-side output is capped (4)
   c. Top output is attached to small tubing with a luer-lock adaptor and clamp (2)
   d. Top-side output is attached to long tubing with a clamp (3)
2. Put input tubing for 1 into 0.9% NaCl solution with peristaltic pump
3. Prime the inside of the filter by clamping output 3 with output 2 open, pump just enough until the solution starts to come out of the top
4. Prime the outside of the filter by clamping output 2 and opening output 3, pump until the solution starts to leave the filter from output 3 (Figure 1b)
5. Clamp all inputs and outputs until filtration
6. Repeat priming process on all necessary filters

**Filtering the sample**

1. Set up the filters according to the below image:



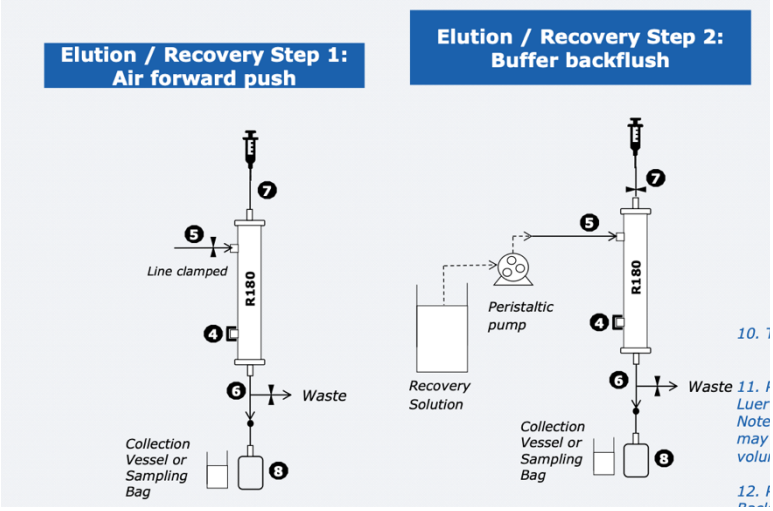

2. Measure the pH and chlorine content of the WW using pH and chlorine strips and record
   a. If chlorine is detected, do not use sample as chlorine disinfectant will alter the results
3. Close line 2, open line 3 and start filtering until the entire sample is through
   a. Ensure flow is around 200 – 250 mL/min and pressure stays between 0.8 – 1kg
4. After filtration, close all clamps (A below) and turn the filter horizontally (B below)



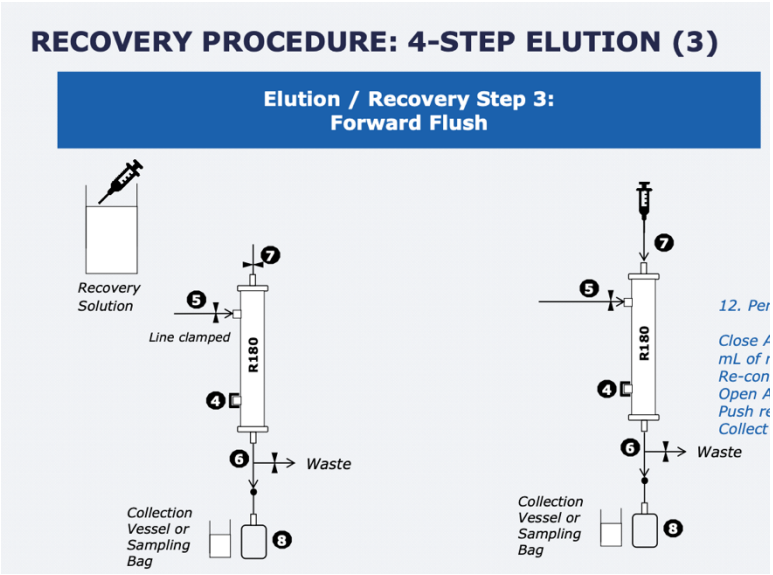

5. Connect a 60mL Luer-lock syringe to line 7

## Elution

1. Set up the filter for elution per the image below:




2. Air Push
    a. Open line 7 and keep line 5 closed
    b. Push 60mL of air through the syringe into a graduated beaker
    c. Measure the volume of wastewater received
3. Backflush
    a. Close line 7, open line 5
    b. Pump elution buffer through the filter until 200mL is collected from line 6
    c. Close all lines
4. Forward Flush




    a. Disconnect the syringe and fill it with elution buffer so that the final eluate volume equals 300mL
    b. Reattach the syringe to line 7
    c. Open lines 7 and 8
    d. Push elution buffer through the syringe and measure volume recovered

e. If 300mL has not been collected:
    i. Conduct a final air push through line 7
    ii. Pull syringe to volume needed to meet 300mL
f. Close all clamps
g. Remove tubing and put caps on the filter
h. Place filter in biohazard waste

**Secondary Concentration**

1. Weigh out PEG 8000 for a 20% v/V final concentration
    a. 60g/300mL of eluate
2. Parafilm sample bottles
3. Agitate eluate until PEG 8000 is completely dissolved
4. Shake at 4°C and 150 RPM overnight
5. After shaking, transfer sample to centrifuge bottles (approximately 6-7 for 300mL)
    a. Gently swirl sample between pours to ensure even distribution between each conical
6. Centrifuge at 10,000 xG for 30 minutes, 4°C
    a. Discard supernatant
7. Add 5mL of 1x PBS (pH = 7.4) to the pellets conicals
    a. Transfer all pellets and PBS to a single conical
8. Vortex at maximum speed for 10 minutes
9. Samples can either be directly extracted or stored at -80°C for future extraction

#### 3.3.1.2 Skimmed Milk Flocculation

**Preparation**

1. Collect influent wastewater in 1-L polypropylene bottles. Store at 4-8°C for up to 4 days.
2. Autoclave/ensure available:
    - 3 sets of tubing for peristaltic pump (2 input and 1 waste)
        i. **Make sure tubing is long enough**
    - 2 10-L carboys (1 sample and 1 waste)
    - 1-L polypropylene bottles
3. Prepare the following reagents:
    - 200mL 1X phosphate-buffered saline (PBS)
    - 5% skimmed milk (5mL/500mL of wastewater)
    - 5 M HCl
    - 5 M NaOH
4. Ensure other supplies available:
    - 50mL conicals
    - 15-mL conicals
    - 25-mL serological pipets
    - 10-mL serological pipets
    - Vertrel
    - pH strips
    - Chlorine strips

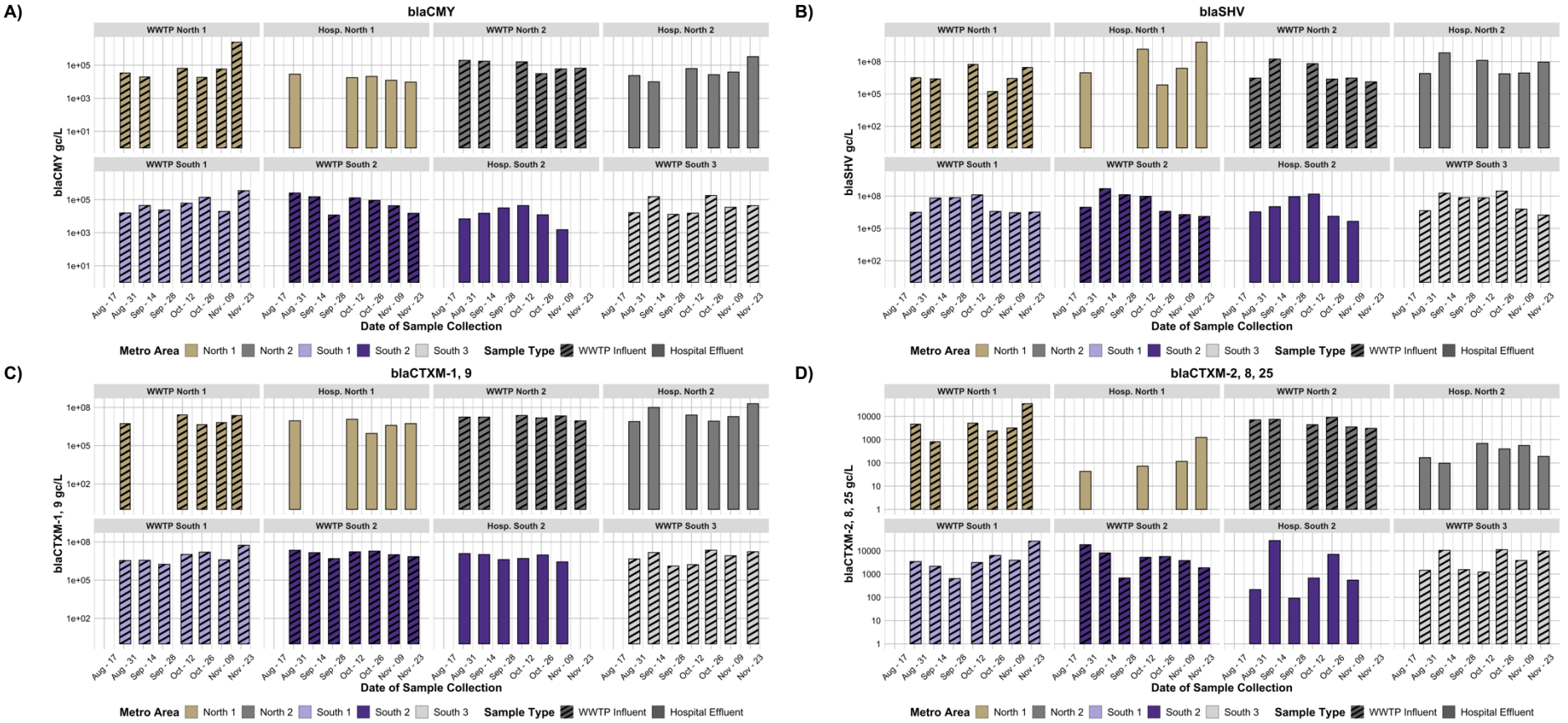
**Skimmed Milk Flocculation**

1. Combine WW from a single WWTP in a 10-L carboy using a graduated cylinder or peristaltic pump
2. Measure pH and chlorine content of WW using pH and chlorine strips
3. Place water sample on cart and mix by gently moving back and forth
4. Pump wastewater using a peristaltic pump into graduated cylinders:
    - Connect tubing to the 10-L PP sample carboy cap outlet, thread thru the peristaltic pump, and hold tubing outlet over graduated cylinder
    - Place the 10-L PP wastewater carboy in a large autoclave tray as secondary containment
    - Place the peristaltic pump on top of an upside-down autoclave tray to raise the height

- Place graduated cylinder in a small autoclave tray as secondary containment

5. Place water sample on cart, and mix by gently moving back and forth
1. Skimmed milk flocculation: pump 0.25L of wastewater sample using a peristaltic pump into a 1-L graduated cylinder and transfer 1.0L or 0.5L bottle.
2. Add 2.5 mL of 5% skimmed milk solution to the 0.25-L sample
3. Adjust pH to 3.0-4.0 using 5 M HCl and pH strips.
4. Parafilm sample bottle.
5. Shake sample on a shaker for 2 hours at room temperature (20-25°C) at 200 RPM.
6. After shaking, transfer sample to centrifuge conicals (approximately 2-3 for 100ml, 5-6 for 250ml and 20-22 for 1L).
    - Gently swirl sample thoroughly to evenly distribute flocs
7. Spin sample at 3500 x G, 4°C for 30 minutes.
8. Discard supernatant
9. Add 3.0mL sterile PBS, pH 7.4 to pellet of first culture analysis conical, and pipet up and down to combine the pellets for each sample
10. Vortex 10 minutes at maximum speed on vortex adapter to completely resuspend pellet.
11. Record volume
12. Samples can either be directly extracted or stored at -80°C for future extraction

### 3.3.1.3 Supplemental Figures



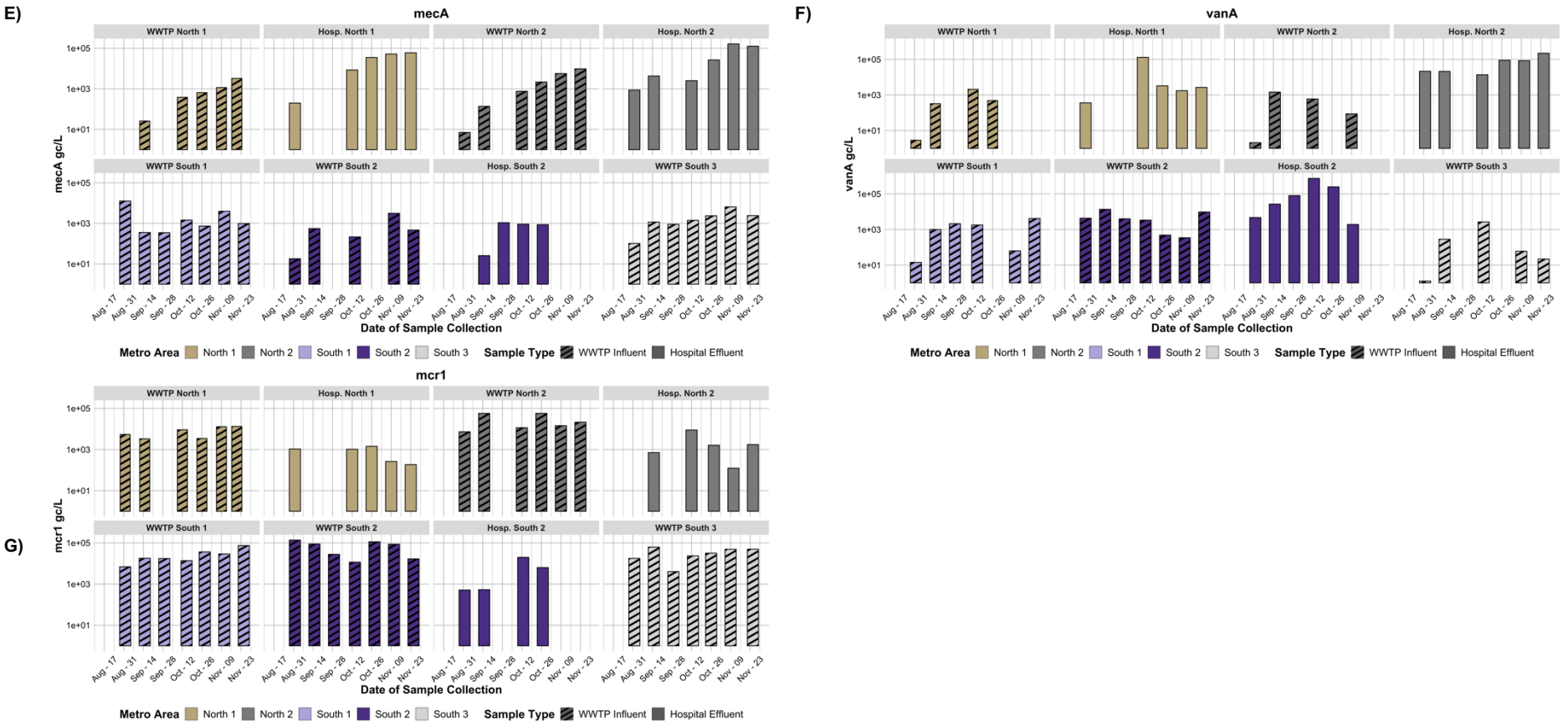
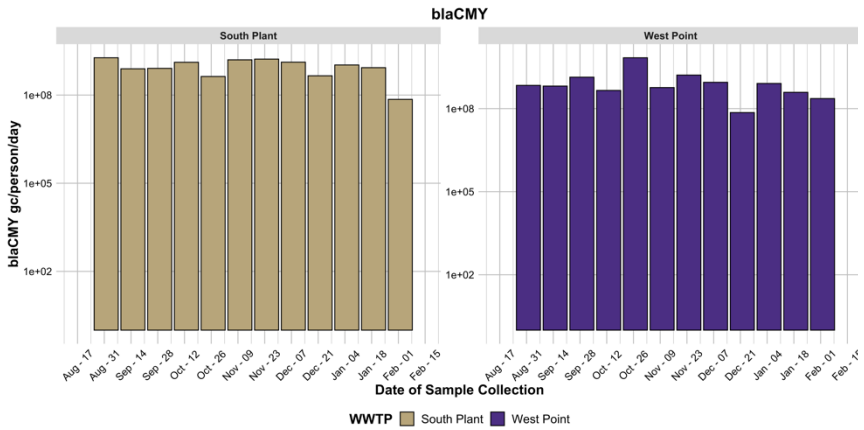
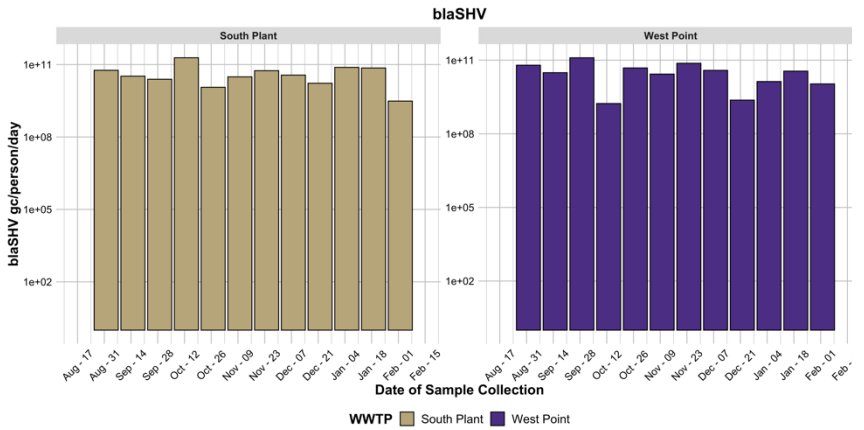



**Figure 3.3.1**: ARG Abundance (gc/L) in Portugal adjusted for effective volume assayed. Random effects models were run on log transformed gene copies to assess changes in abundance over time. Both **D)** *bla*$_{CTXM\text{-}2,8,25}$ and **E)** *mecA* significantly increased over time ($p = 0.02$ and $p < 0.001$, respectively). All other gene targets displayed no significant trend over time in Portugal.

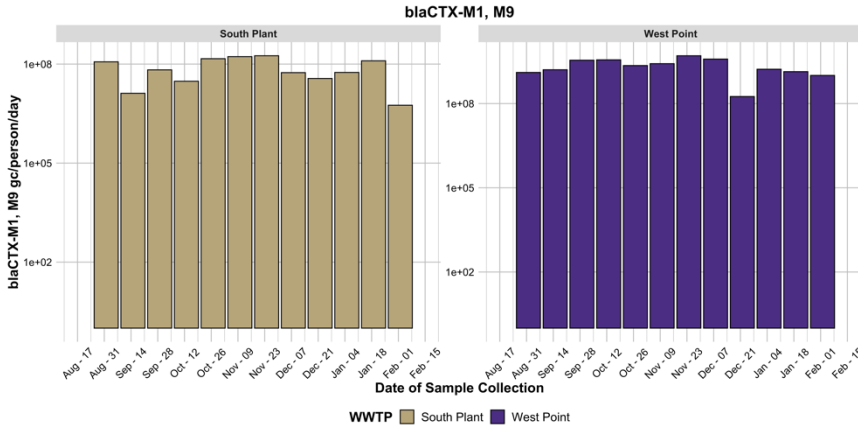
A)


blaCMY
South Plant
West Point
blaCMY gc/person/day
1e+08
1e+05
1e+02
Aug - 17
Aug - 31
Sep - 14
Sep - 28
Oct - 12
Oct - 26
Nov - 09
Nov - 23
Dec - 07
Dec - 21
Jan - 04
Jan - 18
Feb - 01
Feb - 15
Date of Sample Collection
WWTP
South Plant
West Point

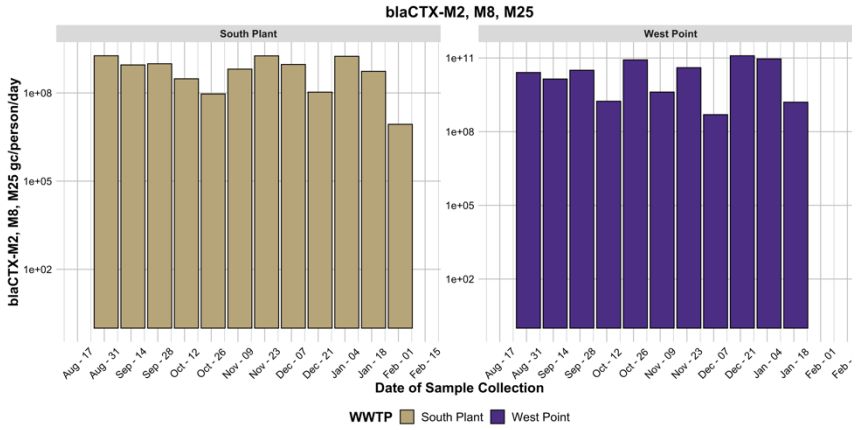
B)


blaSHV
South Plant
West Point
blaSHV gc/person/day
1e+11
1e+08
1e+05
1e+02
Aug - 17
Aug - 31
Sep - 14
Sep - 28
Oct - 12
Oct - 26
Nov - 09
Nov - 23
Dec - 07
Dec - 21
Jan - 04
Jan - 18
Feb - 01
Feb - 15
Date of Sample Collection
WWTP
South Plant
West Point

C)


blaCTX-M1, M9
South Plant
West Point
blaCTX-M1, M9 gc/person/day
1e+08
1e+05
1e+02
Aug - 17
Aug - 31
Sep - 14
Sep - 28
Oct - 12
Oct - 26
Nov - 09
Nov - 23
Dec - 07
Dec - 21
Jan - 04
Jan - 18
Feb - 01
Feb - 15
Date of Sample Collection
WWTP
South Plant
West Point

D)


blaCTX-M2, M8, M25
South Plant
West Point
blaCTX-M2, M8, M25 gc/person/day
1e+11
1e+08
1e+05
1e+02
Aug - 17
Aug - 31
Sep - 14
Sep - 28
Oct - 12
Oct - 26
Nov - 09
Nov - 23
Dec - 07
Dec - 21
Jan - 04
Jan - 18
Feb - 01
Feb - 15
Date of Sample Collection
WWTP
South Plant
West Point

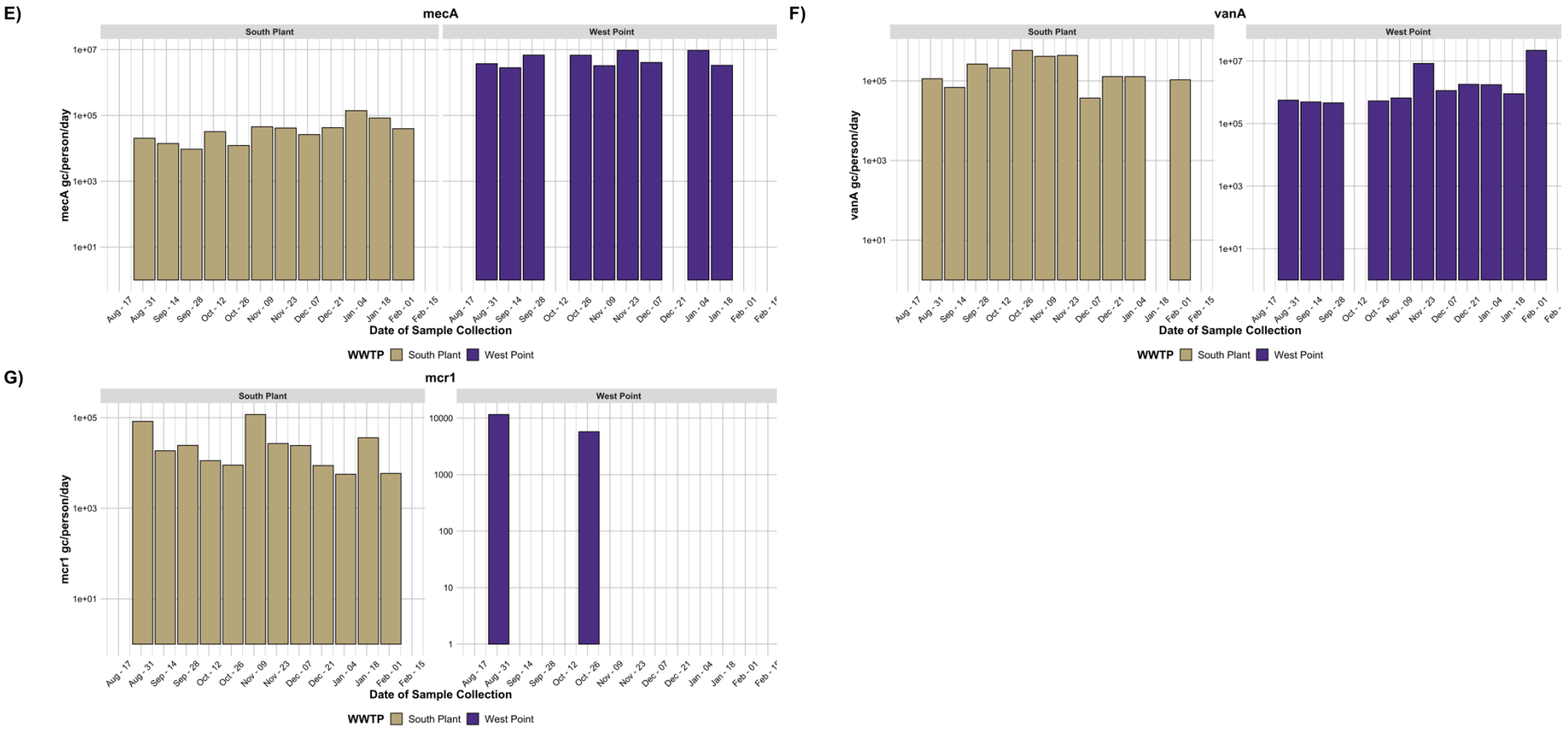



Figure 3.3.2: ARG Abundance in Seattle adjusted for effective volume assayed. Random effects models were run on log transformed gene copies to assess changes in abundance over time. All the beta-lactamase genes **(A, B, C, D)** significantly decreased over the course of the study. However, these genes are not adjusted for daily flow. Higher precipitation in the Seattle-area during winter months could lead to a dilution of the signal.

### 3.3.2 AMR R Code

````
---
title: "Lisbon AMR Data"
author: "Sarah Philo"
date: "2/28/2022"
geometry: margin = 1.75cm
output: pdf_document
editor_options:
---

#set WD
```{r, tidy = TRUE}
knitr::opts_chunk$set(echo = TRUE)
library(readr)
library(dplyr)
library(tidyr)
library(ggplot2)
library(ggpattern)
library(tidyverse)
```

# Load data and Manipulate
```{r, tidy = TRUE}
library(readr)
lisbon <-
  read_csv("Lisbon AMR Sample Results.csv",
           col_types = cols(sample_date = col_date(format =
"%m/%d/%y")))

#if you're having issues, check the date loaded properly, R has
LOTS of issues with this when starting over...

lisbon$ext_vol <- 110
lisbon$elu_vol <- 100
lisbon$pcr_vol <- 1
lisbon$volume_final <- lisbon$volume_final*1000

#Effective Volume in mL
lisbon$effective_vol <-
  lisbon$volume_initial*
  (lisbon$ext_vol/lisbon$volume_final)*
  (lisbon$pcr_vol/lisbon$elu_vol)*1000

summary(lisbon$effective_vol)

lisbon$metro <- NA
lisbon$metro[lisbon$location == "ETAR Serzedelo" |
````
````

```
               lisbon$location == "ETAR Hospital (Saída)"] <- 1
lisbon$metro[lisbon$location == "ETAR Gaia Litoral" |
               lisbon$location == "Rede Drenagem H. S. Silva"]
<- 2
lisbon$metro[lisbon$location == "ETAR Guia"] <- 3
lisbon$metro[lisbon$location == "ETAR Alcântara" |
lisbon$location == "ETAR Hospital C.C."] <- 4
lisbon$metro[lisbon$location == "ETAR Beirolas"] <- 5

lisbon$ww_type <- 1
lisbon$ww_type[lisbon$location == "ETAR Hospital C.C." |
                 lisbon$location == "Rede Drenagem H. S. Silva"
|
                 lisbon$location == "ETAR Hospital (Saída)"] <-
2

# Serzedelo = north 1, gaia = north 2, lisbon = south 2, cascais
= south 1, loures = south 3
lisbon$metro <- factor(lisbon$metro,
                       levels = c(1, 2, 3, 4, 5),
                       labels = c("North 1", "North 2",
                                  "South 1", "South 2", "South
3"))

lisbon$location <- factor(lisbon$location,
                          levels = c("ETAR Serzedelo", "ETAR
Hospital (Saída)",
                                     "ETAR Gaia Litoral", "Rede
Drenagem H. S. Silva",
                                     "ETAR Guia",
                                     "ETAR Alcântara", "ETAR
Hospital C.C.",
                                     "ETAR Beirolas"))

site_names <- c(
  `ETAR Alcântara` = "WWTP South 2",
  `ETAR Beirolas` = "WWTP South 3",
  `ETAR Hospital C.C.` = "Hosp. South 2",
  `ETAR Guia` = "WWTP South 1",
  `Rede Drenagem H. S. Silva` = "Hosp. North 2",
  `ETAR Gaia Litoral` = "WWTP North 2",
  `ETAR Hospital (Saída)` = "Hosp. North 1",
  `ETAR Serzedelo` = "WWTP North 1"
)

lisbon$ww_type <- factor(lisbon$ww_type,
                         levels = c(1, 2),
```

```
                          labels = c("WWTP Influent", "Hospital
Effluent"))

lisbon$week <- strftime(lisbon$sample_date, format = "%V")

hosp_n1 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Hospital (Saída)"],
             lisbon$week[lisbon$location=="ETAR Hospital
(Saída)"])
colnames(hosp_n1) <- c("hosp_n1", "week")

wwtp_n1 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Serzedelo"],
             lisbon$week[lisbon$location=="ETAR Serzedelo"])
colnames(wwtp_n1) <- c("wwtp_n1", "week")

hosp_n2 <- data.frame(lisbon$sample_date[lisbon$location=="Rede
Drenagem H. S. Silva"],
             lisbon$week[lisbon$location=="Rede Drenagem H. S.
Silva"])
colnames(hosp_n2) <- c("hosp_n2", "week")

wwtp_n2 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Gaia Litoral"],
             lisbon$week[lisbon$location=="ETAR Gaia Litoral"])
colnames(wwtp_n2) <- c("wwtp_n2", "week")

wwtp_s1 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Guia"],
             lisbon$week[lisbon$location=="ETAR Guia"])
colnames(wwtp_s1) <- c("wwtp_s1", "week")

wwtp_s2 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Alcântara"],
             lisbon$week[lisbon$location=="ETAR Alcântara"])
colnames(wwtp_s2) <- c("wwtp_s2", "week")

wwtp_s3 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Beirolas"],
             lisbon$week[lisbon$location=="ETAR Beirolas"])
colnames(wwtp_s3) <- c("wwtp_s3", "week")

hosp_s2 <- data.frame(lisbon$sample_date[lisbon$location=="ETAR
Hospital C.C."],
             lisbon$week[lisbon$location=="ETAR Hospital C.C."])
colnames(hosp_s2) <- c("hosp_s2", "week")

sampling_dates <- wwtp_n1 %>%
```

```
  full_join(hosp_n1, by = "week") %>%
  full_join(hosp_n2, by = "week") %>%
  full_join(wwtp_n2, by = "week") %>%
  full_join(wwtp_s1, by = "week") %>%
  full_join(wwtp_s2, by = "week") %>%
  full_join(hosp_s2, by = "week") %>%
  full_join(wwtp_s3, by = "week")

write.csv(sampling_dates, file = "port_sample_dates.csv")
```

#mecA gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)
library(ggbreak)

lisbon$mecA_gcL <-
  (lisbon$mecA_gc/lisbon$effective_vol)*1000

mecA_lisbon <-
  ggplot(data = lisbon, aes(x = sample_date, y = mecA_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "mecA gc/L", x = "Date of Sample Collection",
         title = "mecA")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
```

```
                                    vjust = .5,
                                    color = "black"),
          axis.text.y = element_text(size = 12,
                                    color = "black"),
          axis.title = element_text(size = 16,
                                   color = "black",
                                   face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                   face = "bold",
                                   hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))
mecA_lisbon


ggsave(plot = mecA_lisbon,
       filename = "mecA_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

lisbon$uniqueID <- c(1:length(lisbon$record_id))
uniqueID <- lisbon$uniqueID
mecA <- length(uniqueID)
location <- lisbon$location
sampleDate <- lisbon$sample_date

for(i in uniqueID){
  if(!is.na(lisbon$mecA_gcL[lisbon$uniqueID == i])){
    mecA[i] <- lisbon$mecA_gcL[lisbon$uniqueID == i]
  } else{
    mecA[i] <- 0
  }
}
```

```
mecA <- data.frame(mecA, location, sampleDate)

#https://www.rdocumentation.org/packages/trend/versions/1.1.4/to
pics/mk.test
#mk.test(x = mecA$mecA)
#mk.test(x = log(mecA$mecA)[mecA$mecA>0])
#mk.test(x = mecA$mecA[mecA$location == "ETAR Hospital
(Saída)"])
```


#vanA gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

lisbon$vanA_gcL <-
  (lisbon$vanA_gc/lisbon$effective_vol)*1000

vanA_lisbon <-
  ggplot(data = lisbon,
         aes(x = sample_date, y = vanA_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "vanA gc/L", x = "Date of Sample Collection",
         title = "vanA")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
```

```
                                    vjust = .5,
                                    color = "black"),
          axis.text.y = element_text(size = 12,
                                    color = "black"),
          axis.title = element_text(size = 16,
                                   color = "black",
                                   face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                   face = "bold",
                                   hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                   list(
                                     pattern = c("stripe",
"none"),
                                     pattern_spacing = 0.01)))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
vanA_lisbon

ggsave(plot = vanA_lisbon,
       filename = "vanA_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

#blaTEM gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

lisbon$blaTEM_gcL <-
  (lisbon$bla_tem_gc/lisbon$effective_vol)*1000
```

```
blaTEM_lisbon <-
  ggplot(data = lisbon,
         aes(x = sample_date, y = blaTEM_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "blaTEM gc/L", x = "Date of Sample Collection",
         title = "blaTEM")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
```

```
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
blaTEM_lisbon

ggsave(plot = blaTEM_lisbon,
       filename = "blaTEM_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

```
#blaSHV gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

lisbon$blaSHV_gcL <-
  (lisbon$bla_shv_gc/lisbon$effective_vol)*1000

blaSHV_lisbon <-
  ggplot(data = lisbon,
         aes(x = sample_date, y = blaSHV_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "blaSHV gc/L", x = "Date of Sample Collection",
         title = "blaSHV")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
```
```

```
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
blaSHV_lisbon

ggsave(plot = blaSHV_lisbon,
       filename = "blaSHV_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

#blaCMY gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

lisbon$blaCMY_gcL <-
  (lisbon$bla_cmy_gc/lisbon$effective_vol)*1000

blaCMY_lisbon <-
  ggplot(data = lisbon,
```

```
         aes(x = sample_date, y = blaCMY_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "blaCMY gc/L", x = "Date of Sample Collection",
         title = "blaCMY")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
```

```
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
blaCMY_lisbon

ggsave(plot = blaCMY_lisbon,
       filename = "blaCMY_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

```

#blactxa gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

lisbon$blactxa_gcL <-
  (lisbon$bla_ctxa_gc/lisbon$effective_vol)*1000

blaCTXA_lisbon <-
  ggplot(data = lisbon,
         aes(x = sample_date, y = blactxa_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "blaCTXM-1, 9 gc/L", x = "Date of Sample
Collection",
         title = "blaCTXM-1, 9")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
```

```
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
blaCTXA_lisbon

ggsave(plot = blaCTXA_lisbon,
       filename = "blaCTXA_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

```
#blactxb gc/L
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

lisbon$blactxb_gcL <-
  (lisbon$bla_ctxb_gc/lisbon$effective_vol)*1000

blaCTXB_lisbon <-
  ggplot(data = lisbon,
```
```

```
         aes(x = sample_date, y = blactxb_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "blaCTXM-2, 8, 25 gc/L", x = "Date of Sample
Collection",
         title = "blaCTXM-2, 8, 25")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
```

```
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
blaCTXB_lisbon

ggsave(plot = blaCTXB_lisbon,
       filename = "blaCTXB_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

lisbon$blactxb_log_gcL <-
  log(lisbon$blactxb_gcL)

log_blaCTXB_lisbon <-
  ggplot(data = lisbon,
         aes(x = sample_date, y = blactxb_log_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
                 pattern = ww_type),
             pattern_fill = "black",
             pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "blaCTXM-B gc/L", x = "Date of Sample Collection",
         title = "Gene Copies of blaCTXM-B in Portugal
Wastewater",
         subtitle = "Late 2020")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
```

```
                                    color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
    guides(pattern = guide_legend(override.aes =
                                  list(
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))
log_blaCTXB_lisbon

ggsave(plot = log_blaCTXB_lisbon,
       filename = "log_blaCTXB_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)


```

#mcr1 gc/L
```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)

lisbon$mcr1_gcL <-
  (lisbon$mcr1_gc/lisbon$effective_vol)*1000

mcr1_lisbon <-
  ggplot(data = lisbon, aes(x = sample_date, y = mcr1_gcL))+
    geom_col_pattern(position = position_dodge2(preserve =
"single"),
             color = "black",
             aes(fill = metro,
                 width = 8,
```

```
                pattern = ww_type),
              pattern_fill = "black",
              pattern_color = "black")+
    facet_wrap(~location, nrow = 2,
               #scales = "free",
               labeller = as_labeller(site_names)
               )+
    scale_fill_manual(values = c("#B7A57A", "#797979",
"#ABA2D8", "#4b2e83", "light gray"),
                      name = "Metro Area")+
    scale_pattern_discrete(choices = c("stripe", "none"),
                           name = "Sample Type")+
    labs(y = "mcr1 gc/L", x = "Date of Sample Collection",
         title = "mcr1")+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_y_continuous(trans = 'log10')+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2020-12-01")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))
mcr1_lisbon
```

```
ggsave(plot = mcr1_lisbon,
       filename = "mcr1_lisbon.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

#ARG Figure
```{r, tidy = TRUE}
library(tidyr)
library(dplyr)

lisbon_longer <- lisbon %>%
  select(sample_date, location, metro, ww_type,
         )



```

#COVID-19 Cases and Deaths
```{r, tidy = TRUE}
port_covid <- read_csv("20221017 Portugal COVID Data.csv",
                       col_types = cols(Date_reported =
col_date(format = "%m/%d/%y")))
covid_portugal <-
  ggplot(data = port_covid, aes(x = Date_reported, y =
New_cases))+
    geom_col(aes(fill = Country))+
    scale_fill_manual(values = c("#4B2E83"))+
    theme(legend.position = "none",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.major.x = element_blank(),
          panel.grid.minor = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
```

```
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face = "bold"),
          axis.line = element_line(color = "black")
    )+
    scale_x_date(
      date_labels = "%b '%y",
      date_breaks = "1 month",
      limits = as.Date(c("2020-03-01", "2022-10-31"))
    )+
    labs(x = "Date Reported",
         y = "Incident Cases",
         title = "Incident COVID-19 Cases, Portugal",
         subtitle = "Entire Pandemic"

    )+
    scale_y_continuous(
      breaks = c(10000, 20000, 30000, 40000, 50000, 60000)
    )
covid_portugal
ggsave(plot = covid_portugal,
       filename = "covid_portugal.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

covid_portugal_early <-
    ggplot(data = port_covid, aes(x = Date_reported, y =
New_cases))+
    geom_col(aes(fill = Country))+
    scale_fill_manual(values = c("#4B2E83"))+
    theme(legend.position = "none",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.major.x = element_blank(),
          panel.grid.minor = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
```

```
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face = "bold"),
          axis.line = element_line(color = "black")
    )+
    scale_x_date(
      date_labels = "%b '%y",
      date_breaks = "1 month",
      limits = as.Date(c("2020-03-01", "2020-06-30"))
    )+
    labs(x = "Date Reported",
         y = "Incident Cases",
         title = "Incident COVID-19 Cases, Portugal",
         subtitle = "Early 2020"

    )+
    scale_y_continuous(
      breaks = c(10000, 20000, 30000, 40000, 50000, 60000)
    )
covid_portugal_early
ggsave(plot = covid_portugal_early,
       filename = "covid_portugal_early.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

max(port_covid$New_cases[port_covid$Date_reported<"2020-06-30"])
print(port_covid$Date_reported[port_covid$New_cases == 936 &
port_covid$Date_reported<"2020-06-30"])

covid_portugal_firstYear <-
    ggplot(data = port_covid, aes(x = Date_reported, y =
New_cases))+
    geom_col(aes(fill = Country))+
    scale_fill_manual(values = c("#4B2E83"))+
    theme(legend.position = "none",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.major.x = element_blank(),
          panel.grid.minor = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 16,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 16,
                                     color = "black"),
```

```
          axis.title = element_text(size = 24,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face = "bold"),
          axis.line = element_line(color = "black")
    )+
    scale_x_date(
      date_labels = "%b '%y",
      date_breaks = "1 month",
      limits = as.Date(c("2020-03-01", "2021-03-10"))
    )+
    labs(x = "Date Reported",
         y = "Incident Cases",
         title = "Incident COVID-19 Cases, Portugal",
         subtitle = "First 12 Months"

    )+
    scale_y_continuous(
      breaks = c(5000, 10000, 15000, 20000)
    )
covid_portugal_firstYear

ggsave(plot = covid_portugal_firstYear,
       filename = "covid_portugal_firstYear.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

print(port_covid$Cumulative_cases[port_covid$Date_reported ==
"2021-03-02"])
print(port_covid$Cumulative_deaths[port_covid$Date_reported ==
"2021-03-02"])
t.test(port_covid$New_cases[port_covid$Date_reported < "2020-08-
18"])
t.test(port_covid$New_cases[port_covid$Date_reported >= "2020-
08-19"
                            & port_covid$Date_reported < "2020-
11-25"])

max(port_covid$New_cases[port_covid$Date_reported > "2020-11-05"
```

```
                          & port_covid$Date_reported < "2020-12-
25"])
print(port_covid$Date_reported[port_covid$New_cases==6297])

max(port_covid$New_cases[port_covid$Date_reported > "2020-12-05"
                          & port_covid$Date_reported < "2021-03-
01"])
print(port_covid$Date_reported[port_covid$New_cases==15251])
```

#ARG Descriptive Stats
```{r, tidy = TRUE}

library(dplyr)
library(tidyverse)

total_mean <- lisbon %>%
  mutate(number = length(location)) %>%
  mutate(avg_mecA = mean(mecA_gcL, na.rm = TRUE)) %>%
  mutate(avg_tem = mean(blaTEM_gcL, na.rm = TRUE)) %>%
  mutate(avg_cmy = mean(blaCMY_gcL, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blactxb_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxa = mean(blactxa_gcL, na.rm = TRUE)) %>%
  mutate(avg_vanA = mean(vanA_gcL, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_gcL, na.rm = TRUE)) %>%
  select(
    location, avg_mecA, avg_tem, avg_ctxb, avg_ctxa,
    avg_cmy, avg_shv,
    avg_mcr1,
    avg_vanA, number
  ) %>%
  distinct(avg_mecA, .keep_all = TRUE)
total_mean$location <- "total"

mean <- lisbon %>%
  group_by(location) %>%
  mutate(number = length(location)) %>%
  mutate(avg_mecA = mean(mecA_gcL, na.rm = TRUE)) %>%
  mutate(avg_tem = mean(blaTEM_gcL, na.rm = TRUE)) %>%
  mutate(avg_cmy = mean(blaCMY_gcL, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blactxb_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxa = mean(blactxa_gcL, na.rm = TRUE)) %>%
  mutate(avg_vanA = mean(vanA_gcL, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_gcL, na.rm = TRUE)) %>%
  select(
```

```
    location, avg_mecA, avg_tem, avg_cmy, avg_shv, avg_ctxa,
avg_ctxb,
    avg_mcr1,
    avg_vanA, number
  ) %>%
  distinct(location, .keep_all = TRUE)

mean <- bind_rows(mean, total_mean)
write.csv(mean, file = "portugal_mean_values.csv")
```

## Reproducible Example
```{r, tidy = TRUE}
library(reprex)
library(datapasta)
bios_cons <- select(lisbon, sample_date, location, metro,
ww_type, mecA_gcL)
#dpasta(bios_cons)
#put the dataframe in front of the tribble output with all the
data to create reprex dataset
bios_cons <- tibble::tribble(
     ~sample_date,                  ~location,
~metro,            ~ww_type,        ~mecA_gcL,
     "2020-08-18",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent", 18.4882727272727,
     "2020-08-18",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 50.3941590909091,
     "2020-08-18",                 "ETAR Guia",
"Cascais",     "WWTP Influent", 29.4717045454545,
     "2020-08-18", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 393.417163636364,
     "2020-08-18",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent",               NA,
     "2020-08-18",     "ETAR Hospital (Saída)",
"Serzedelo", "Hospital Effluent",        349.04545,
     "2020-08-18",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 34.6176636363636,
     "2020-08-18",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent",         28.16385,
     "2020-09-01",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent", 18.0361409090909,
     "2020-09-01",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 102.054413636364,
     "2020-09-01",                 "ETAR Guia",
"Cascais",     "WWTP Influent",       12595.9129,
     "2020-09-01",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent",               NA,
```

```
     "2020-09-01",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 7.03949545454546,
     "2020-09-01", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 868.839036363636,
     "2020-09-01",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent",               NA,
     "2020-09-01",     "ETAR Hospital (Saída)",
"Serzedelo", "Hospital Effluent", 198.318059090909,
     "2020-09-15",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent", 25.8260272727273,
     "2020-09-15",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent", 25.6137636363636,
     "2020-09-15",                "ETAR Guia",
"Cascais",     "WWTP Influent", 355.540781818182,
     "2020-09-15",             "ETAR Beirolas",
"Loures",     "WWTP Influent",       1157.78595,
     "2020-09-15",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 137.049254545455,
     "2020-09-15",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent", 552.042131818182,
     "2020-09-15", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 4286.46825909091,
     "2020-09-29",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 895.701509090909,
     "2020-09-29",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent",               NA,
     "2020-09-29",                "ETAR Guia",
"Cascais",     "WWTP Influent", 346.531663636364,
     "2020-09-29",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent", 1079.87502727273,
     "2020-10-12",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent", 381.192368181818,
     "2020-10-12", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 2528.13493181818,
     "2020-10-12",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 756.684604545455,
     "2020-10-12",     "ETAR Hospital (Saída)",
"Serzedelo", "Hospital Effluent", 8508.83912727273,
     "2020-10-13",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 1411.28763181818,
     "2020-10-13",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent", 905.608263636364,
     "2020-10-14",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent", 215.147622727273,
     "2020-10-15",                "ETAR Guia",
"Cascais",     "WWTP Influent", 1447.97823181818,
     "2020-10-26",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 2358.00907272727,
```

```
     "2020-10-26",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent", 648.688081818182,
     "2020-10-26",     "ETAR Hospital (Saída)",
"Serzedelo", "Hospital Effluent",       35137.9222,
     "2020-10-26",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 2168.44668181818,
     "2020-10-28",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent",               NA,
     "2020-10-28",                "ETAR Guia",
"Cascais",     "WWTP Influent", 733.925290909091,
     "2020-10-28", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 26586.3100090909,
     "2020-10-27",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent", 865.128113636364,
     "2020-11-09",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 6499.15391818182,
     "2020-11-09",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent", 1151.60628636364,
     "2020-11-09",     "ETAR Hospital (Saída)",
"Serzedelo", "Hospital Effluent", 52069.5147272727,
     "2020-11-09",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 5693.81570909091,
     "2020-11-10",        "ETAR Hospital C.C.",
"Lisbon", "Hospital Effluent",               NA,
     "2020-11-11",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent", 3133.91083636364,
     "2020-11-11",                "ETAR Guia",
"Cascais",     "WWTP Influent", 3953.83291818182,
     "2020-11-11", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 164070.588318182,
     "2020-11-23",         "ETAR Gaia Litoral", "Vila Nova de
Gaia",     "WWTP Influent", 9699.95826818182,
     "2020-11-23",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent", 3271.29851363636,
     "2020-11-23",     "ETAR Hospital (Saída)",
"Serzedelo", "Hospital Effluent", 59860.8027727273,
     "2020-11-25",             "ETAR Beirolas",
"Loures",     "WWTP Influent", 2405.35844545455,
     "2020-11-25",            "ETAR Alcântara",
"Lisbon",     "WWTP Influent", 467.254363636364,
     "2020-11-25",                "ETAR Guia",
"Cascais",     "WWTP Influent", 983.799640909091,
     "2020-11-25",            "ETAR Serzedelo",
"Serzedelo",     "WWTP Influent", 1197.39103181818,
     "2020-11-25", "Rede Drenagem H. S. Silva", "Vila Nova de
Gaia", "Hospital Effluent", 126528.423727273
     )
```

```
#Stats on ARGs
```{r, tidy = TRUE}
library(lmerTest)

lisbon$day <- difftime(lisbon$sample_date,
min(lisbon$sample_date), units = "days")

mecAmod <-
  lmer(data = lisbon, log(mecA_gcL) ~ day + (1|location))
summary(mecAmod)
confint(mecAmod)
exp(confint(mecAmod))

temMod <-
  lmer(data = lisbon, log(blaTEM_gcL) ~ day + (1|location))
summary(temMod)
confint(temMod)
exp(confint(temMod))

cmyMod <-
  lmer(data = lisbon, log(blaCMY_gcL) ~ day + (1|location))
summary(cmyMod)
confint(cmyMod)
exp(confint(cmyMod))

shvMod <-
  lmer(data = lisbon, log(blaSHV_gcL) ~ day + (1|location))
summary(shvMod)
confint(shvMod)
exp(confint(shvMod))

ctxaMod <-
  lmer(data = lisbon, log(blactxa_gcL) ~ day + (1|location))
summary(ctxaMod)
confint(ctxaMod)
exp(confint(ctxaMod))

ctxbMod <-
  lmer(data = lisbon, log(blactxb_gcL) ~ day + (1|location))
summary(ctxbMod)
confint(ctxbMod)
exp(confint(ctxbMod))

vanaMod <-
  lmer(data = lisbon, log(vanA_gcL) ~ day + (1|location))
summary(vanaMod)
confint(vanaMod)
```
```

```
exp(confint(vanaMod))

mcr1Mod <-
  lmer(data = lisbon, log(mcr1_gcL) ~ day + (1|location))
summary(mcr1Mod)
confint(mcr1Mod)
exp(confint(mcr1Mod))

library(dplyr)
library(lubridate)

lisbon <- lisbon %>%
  mutate(sample_date = lubridate::parse_date_time(sample_date,
"ymd")) %>%
  arrange(sample_date)

lisbon$week <- strftime(lisbon$sample_date, format = "%V")

serzedelo_wwtp <- lisbon %>%
  filter(lisbon$location == "ETAR Serzedelo") %>%
  select(week, location, mecA_gcL, vanA_gcL,
         mcr1_gcL,
         blaTEM_gcL, blaCMY_gcL, blaSHV_gcL, blactxa_gcL,
blactxb_gcL) %>%
  arrange(week)

serzedelo_hosp <- lisbon %>%
  filter(lisbon$location == "ETAR Hospital (Saída)") %>%
  select(week, location, mecA_gcL, vanA_gcL,
         mcr1_gcL,
         blaTEM_gcL, blaCMY_gcL, blaSHV_gcL, blactxa_gcL,
blactxb_gcL) %>%
  arrange(week)

serzedelo <- left_join(serzedelo_hosp, serzedelo_wwtp,
                       by = c("week"))

serzedelo_mecA <- cor.test(log(serzedelo$mecA_gcL.x),
log(serzedelo$mecA_gcL.y))
serzedelo_vanA <- cor.test(log(serzedelo$vanA_gcL.x),
log(serzedelo$vanA_gcL.y))
serzedelo_mcr1 <- cor.test(log(serzedelo$mcr1_gcL.x),
log(serzedelo$mcr1_gcL.y))
serzedelo_tem <- cor.test(log(serzedelo$blaTEM_gcL.x),
log(serzedelo$blaTEM_gcL.y))
serzedelo_cmy <- cor.test(log(serzedelo$blaCMY_gcL.x),
log(serzedelo$blaCMY_gcL.y))
```

```
serzedelo_shv <- cor.test(log(serzedelo$blaSHV_gcL.x),
log(serzedelo$blaSHV_gcL.y))
serzedelo_ctxa <- cor.test(log(serzedelo$blactxa_gcL.x),
log(serzedelo$blactxa_gcL.y))
serzedelo_ctxb <- cor.test(log(serzedelo$blactxb_gcL.x),
log(serzedelo$blactxb_gcL.y))

gaia_wwtp <- lisbon %>%
  filter(lisbon$location == "ETAR Gaia Litoral") %>%
  select(week, location, mecA_gcL, vanA_gcL,
         mcr1_gcL,
         blaTEM_gcL, blaCMY_gcL, blaSHV_gcL, blactxa_gcL,
blactxb_gcL) %>%
  arrange(week)

gaia_hosp <- lisbon %>%
  filter(lisbon$location == "Rede Drenagem H. S. Silva") %>%
  select(week, location, mecA_gcL, vanA_gcL,
         mcr1_gcL,
         blaTEM_gcL, blaCMY_gcL, blaSHV_gcL, blactxa_gcL,
blactxb_gcL) %>%
  arrange(week)

gaia <- left_join(gaia_hosp, gaia_wwtp, by = c("week"))

gaia_mecA <- cor.test(log(gaia$mecA_gcL.x),
log(gaia$mecA_gcL.y))
gaia_vanA <- cor.test(log(gaia$vanA_gcL.x),
log(gaia$vanA_gcL.y))
gaia_mcr1 <- cor.test(log(gaia$mcr1_gcL.x),
log(gaia$mcr1_gcL.y))
gaia_tem <- cor.test(log(gaia$blaTEM_gcL.x),
log(gaia$blaTEM_gcL.y))
gaia_cmy <- cor.test(log(gaia$blaCMY_gcL.x),
log(gaia$blaCMY_gcL.y))
gaia_shv <- cor.test(log(gaia$blaSHV_gcL.x),
log(gaia$blaSHV_gcL.y))
gaia_ctxa <- cor.test(log(gaia$blactxa_gcL.x),
log(gaia$blactxa_gcL.y))
gaia_ctxb <- cor.test(log(gaia$blactxb_gcL.x),
log(gaia$blactxb_gcL.y))

lisbon_wwtp <- lisbon %>%
  filter(lisbon$location == "ETAR Alcântara") %>%
  select(week, location, mecA_gcL, vanA_gcL,
         mcr1_gcL,
         blaTEM_gcL, blaCMY_gcL, blaSHV_gcL, blactxa_gcL,
blactxb_gcL) %>%
```

```
  arrange(week)

lisbon_hosp <- lisbon %>%
  filter(lisbon$location == "ETAR Hospital C.C.") %>%
  select(week, location, mecA_gcL, vanA_gcL,
         mcr1_gcL,
         blaTEM_gcL, blaCMY_gcL, blaSHV_gcL, blactxa_gcL,
blactxb_gcL) %>%
  arrange(week)

lisbon_cor <- left_join(lisbon_hosp, lisbon_wwtp, by =
c("week"))

lisbon_cor_mecA <- cor.test(log(lisbon_cor$mecA_gcL.x),
log(lisbon_cor$mecA_gcL.y))
lisbon_cor_vanA <- cor.test(log(lisbon_cor$vanA_gcL.x),
log(lisbon_cor$vanA_gcL.y))
lisbon_cor_mcr1 <- cor.test(log(lisbon_cor$mcr1_gcL.x),
log(lisbon_cor$mcr1_gcL.y))
lisbon_cor_tem <- cor.test(log(lisbon_cor$blaTEM_gcL.x),
log(lisbon_cor$blaTEM_gcL.y))
lisbon_cor_cmy <- cor.test(log(lisbon_cor$blaCMY_gcL.x),
log(lisbon_cor$blaCMY_gcL.y))
lisbon_cor_shv <- cor.test(log(lisbon_cor$blaSHV_gcL.x),
log(lisbon_cor$blaSHV_gcL.y))
lisbon_cor_ctxa <- cor.test(log(lisbon_cor$blactxa_gcL.x),
log(lisbon_cor$blactxa_gcL.y))
lisbon_cor_ctxb <- cor.test(log(lisbon_cor$blactxb_gcL.x),
log(lisbon_cor$blactxb_gcL.y))

targets <- c("mecA", "vanA",
             "mcr1",
             "blaCMY", "blaSHV", "blaCTXM-1, 9", "blaCTXM-2, 8,
25")

serzedelo_est <- round(c(serzedelo_mecA$estimate,
serzedelo_vanA$estimate,
                  serzedelo_mcr1$estimate,
                  serzedelo_cmy$estimate,
serzedelo_shv$estimate, serzedelo_ctxa$estimate,
serzedelo_ctxb$estimate),
                  digits = 2)
serzedelo_p <- round(c(serzedelo_mecA$p.value,
serzedelo_vanA$p.value,
                serzedelo_mcr1$p.value,
                serzedelo_cmy$p.value, serzedelo_shv$p.value,
serzedelo_ctxa$p.value, serzedelo_ctxb$p.value),
                digits = 2)
```

```
serzedelo_lower <- round(c(serzedelo_mecA$conf.int[1],
serzedelo_vanA$conf.int[1],
                     serzedelo_mcr1$conf.int[1],
                     serzedelo_cmy$conf.int[1],
serzedelo_shv$conf.int[1],
                     serzedelo_ctxa$conf.int[1],
serzedelo_ctxb$conf.int[1]),
                     digits = 2)
serzedelo_upper <- round(c(serzedelo_mecA$conf.int[2],
serzedelo_vanA$conf.int[2],
                     serzedelo_mcr1$conf.int[2],
                     serzedelo_cmy$conf.int[2],
serzedelo_shv$conf.int[2],
                     serzedelo_ctxa$conf.int[2],
serzedelo_ctxb$conf.int[2]),
                     digits = 2)
serzedelo <- data.frame(cbind(targets, serzedelo_est,
serzedelo_lower, serzedelo_upper, serzedelo_p))


gaia_est <- round(c(gaia_mecA$estimate, gaia_vanA$estimate,
                    gaia_mcr1$estimate,
                    gaia_cmy$estimate, gaia_shv$estimate,
gaia_ctxa$estimate, gaia_ctxb$estimate),
                   digits = 2)
gaia_p <- round(c(gaia_mecA$p.value, gaia_vanA$p.value,
                  gaia_mcr1$p.value,
                  gaia_cmy$p.value, gaia_shv$p.value,
gaia_ctxa$p.value, gaia_ctxb$p.value),
                 digits = 2)
gaia_lower <- round(c(gaia_mecA$conf.int[1],
gaia_vanA$conf.int[1],
                     NA,
                     gaia_cmy$conf.int[1], gaia_shv$conf.int[1],
                     gaia_ctxa$conf.int[1],
gaia_ctxb$conf.int[1]),
                    digits = 2)
gaia_upper <- round(c(gaia_mecA$conf.int[2],
gaia_vanA$conf.int[2],
                     NA,
                     gaia_cmy$conf.int[2], gaia_shv$conf.int[2],
                     gaia_ctxa$conf.int[2],
gaia_ctxb$conf.int[2]),
                    digits = 2)
gaia <- data.frame(cbind(targets, gaia_est, gaia_lower,
gaia_upper, gaia_p))
```

```
lisbon_cor_est <- round(c(lisbon_cor_mecA$estimate,
lisbon_cor_vanA$estimate,
                  lisbon_cor_mcr1$estimate,
                  lisbon_cor_cmy$estimate,
lisbon_cor_shv$estimate, lisbon_cor_ctxa$estimate,
                  lisbon_cor_ctxb$estimate),
                 digits = 2)
lisbon_cor_p <- round(c(lisbon_cor_mecA$p.value,
lisbon_cor_vanA$p.value,
                lisbon_cor_mcr1$p.value,
                lisbon_cor_cmy$p.value, lisbon_cor_shv$p.value,
lisbon_cor_ctxa$p.value, lisbon_cor_ctxb$p.value),
                digits = 2)
lisbon_cor_lower <- round(c(NA, lisbon_cor_vanA$conf.int[1],
                    lisbon_cor_mcr1$conf.int[1],
                    lisbon_cor_cmy$conf.int[1],
lisbon_cor_shv$conf.int[1],
                    lisbon_cor_ctxa$conf.int[1],
lisbon_cor_ctxb$conf.int[1]),
                   digits = 2)
lisbon_cor_upper <- round(c(NA, lisbon_cor_vanA$conf.int[2],
                    lisbon_cor_mcr1$conf.int[2],
                    lisbon_cor_cmy$conf.int[2],
lisbon_cor_shv$conf.int[2],
                    lisbon_cor_ctxa$conf.int[2],
lisbon_cor_ctxb$conf.int[2]),
                   digits = 2)
lisbon_cor_tab <- data.frame(cbind(targets, lisbon_cor_est,
lisbon_cor_lower, lisbon_cor_upper, lisbon_cor_p))

cor_tab <- list(serzedelo, gaia, lisbon_cor_tab) %>%
  reduce(full_join, by = 'targets')
write.csv(cor_tab, file = "correlation_data.csv")
```

```
---
title: "Seattle AMR Figures"
author: "Sarah Philo"
date: "10/10/2022"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)

setwd("~/OneDrive/Documents/UW/EOHML/AMR/Seattle/Figures")
```
```

```
## Load and Prep Data
```{r}
library(readr)
seattle<-
  read_csv("seattle amr results.csv",
           col_types = cols(bla_ctxb = col_double(),
                            bla_ctxb_gc = col_double(),
                            exp_date = col_date(format =
"%m/%d/%y"),
                            sample_date = col_date(format =
"%m/%d/%y")))

seattle$method_vol <- seattle$method_vol*1000

#Effective Volume in mL
seattle$effective_vol <- seattle$method_vol*
  (seattle$ext_vol/seattle$final_vol)*
  (seattle$pcr_vol/seattle$elu_vol)
seattle$effective_vol <- seattle$effective_vol*0.001

seattle$hold_time <- seattle$exp_date - seattle$sample_date

seattle$week <- strftime(seattle$sample_date, format = "%V")

west_point <- data.frame(seattle$sample_date[seattle$wwtp=="West
Point"],
                         seattle$week[seattle$wwtp=="West
Point"])
colnames(west_point) <- c("west_point", "week")

south_plant <-
data.frame(seattle$sample_date[seattle$wwtp=="South Plant"],
                          seattle$week[seattle$wwtp=="South
Plant"])
colnames(south_plant) <- c("south_plant", "week")

sampling_dates <- west_point %>%
  full_join(south_plant, by = "week")
write.csv(sampling_dates, file = "seattle_sample_dates.csv")

wwtp_flow <-
read_csv("~/OneDrive/Documents/UW/EOHML/AMR/Seattle/Figures/wwtp
_flow.csv",
                       col_types = cols(sample_date =
col_date(format = "%m/%d/%y")))

seattle <- seattle %>%
  left_join(wwtp_flow, by = "sample_date")
```
```

```
seattle$wp_lpd <- (seattle$wp_flow*1000000*3.785)/787000
seattle$sp_lpd <- (seattle$sp_flow*1000000*3.785)/839000

uniqueID <- 1:26
seattle$uniqueID <- uniqueID
seattle$blaCMY_adj <- NA
seattle$blaSHV_adj <- NA
seattle$blaCTXA_adj <- NA
seattle$blaCTXB_adj <- NA
seattle$mecA_adj <- NA
seattle$vanA_adj <- NA
seattle$mcr1_adj <- NA
```

## mecA Graphs
```{r, tidy = TRUE}

library(ggplot2)
library(ggpattern)

seattle$mecA_gcL <-
  (seattle$mecA_gc/seattle$effective_vol)*1000

for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$mecA_adj[seattle$uniqueID == i] <-
      seattle$mecA_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$mecA_adj[seattle$uniqueID == i] <-
      seattle$mecA_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
  }
}

mecA_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = mecA_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
               #scales = "free"
               )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "mecA gc/person/day", x = "Date of Sample
Collection",
         title = "mecA"
```

```
          #subtitle = "Late 2020 to Early 2021"
          )+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  scale_y_continuous(trans = 'log10')+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))
mecA_seattle

ggsave(plot = mecA_seattle,
       filename = "mecA_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```


## blaCMY
```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)
```

```
seattle$bla_cmy_gcL <-
  (seattle$bla_cmy_gc/seattle$effective_vol)*1000

for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$blaCMY_adj[seattle$uniqueID == i] <-
      seattle$bla_cmy_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$blaCMY_adj[seattle$uniqueID == i] <-
      seattle$bla_cmy_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
  }
}

bla_cmy_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = blaCMY_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
               scales = "free"
               )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "blaCMY gc/person/day", x = "Date of Sample
Collection",
         title = "blaCMY"
         #subtitle = "Late 2020 to Early 2021"
         )+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
```

```
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))+
  scale_y_continuous(trans = 'log10')
bla_cmy_seattle

ggsave(plot = bla_cmy_seattle,
       filename = "bla_cmy_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

## blaSHV
```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)

seattle$bla_shv_gcL <-
  (seattle$bla_shv_gc/seattle$effective_vol)*1000

for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$blaSHV_adj[seattle$uniqueID == i] <-
      seattle$bla_shv_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$blaSHV_adj[seattle$uniqueID == i] <-
      seattle$bla_shv_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
  }
}

bla_shv_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = blaSHV_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
```

```
                scales = "free"
                )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "blaSHV gc/person/day", x = "Date of Sample
Collection",
         title = "blaSHV"
         #subtitle = "Late 2020 to Early 2021"
         )+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))+
  scale_y_continuous(trans = 'log10')
bla_shv_seattle

ggsave(plot = bla_shv_seattle,
       filename = "bla_shv_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

## blaCTXB

```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)

seattle$bla_ctxb_gcL <-
  (seattle$bla_ctxb_gc/seattle$effective_vol)*1000

for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$blaCTXB_adj[seattle$uniqueID == i] <-
      seattle$bla_ctxb_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$blaCTXB_adj[seattle$uniqueID == i] <-
      seattle$bla_ctxb_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
  }
}

bla_ctxb_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = blaCTXB_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
               scales = "free"
               )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "blaCTX-M2, M8, M25 gc/person/day", x = "Date of
Sample Collection",
         title = "blaCTX-M2, M8, M25"
         #subtitle = "Late 2020 to Early 2021"
         )+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
```

```
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))+
  scale_y_continuous(trans = 'log10')
bla_ctxb_seattle

ggsave(plot = bla_ctxb_seattle,
       filename = "bla_ctxb_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

## blaCTXA

```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)

seattle$bla_ctxa_gcL <-
  (seattle$bla_ctxa_gc/seattle$effective_vol)*1000

for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$blaCTXA_adj[seattle$uniqueID == i] <-
      seattle$bla_ctxa_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$blaCTXA_adj[seattle$uniqueID == i] <-
      seattle$bla_ctxa_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
```

```
  }
}

bla_ctxa_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = blaCTXA_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
               scales = "free"
               )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "blaCTX-M1, M9 gc/person/day", x = "Date of Sample
Collection",
         title = "blaCTX-M1, M9"
         #subtitle = "Late 2020 to Early 2021"
         )+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  guides(pattern = guide_legend(override.aes =
                                list(
```

```
                                    pattern = c("stripe", "none"),
                                    pattern_spacing = 0.01)))+
  scale_y_continuous(trans = 'log10')
bla_ctxa_seattle

ggsave(plot = bla_ctxa_seattle,
       filename = "bla_ctxa_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```



## mcr1
```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)

seattle$mcr1_gcL <-
  (seattle$mcr1_gc/seattle$effective_vol)*1000

for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$mcr1_adj[seattle$uniqueID == i] <-
      seattle$mcr1_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$mcr1_adj[seattle$uniqueID == i] <-
      seattle$mcr1_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
  }
}

mcr1_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = mcr1_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
               scales = "free"
               )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "mcr1 gc/person/day", x = "Date of Sample
Collection",
         title = "mcr1"
         #subtitle = "Late 2020 to Early 2021"
         )+
    theme(legend.position = "bottom",
```

```
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
                                       hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))+
  scale_y_continuous(trans = 'log10')
mcr1_seattle

ggsave(plot = mcr1_seattle,
       filename = "mcr1_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```


## vanA
```{r, tidy = TRUE}
library(ggplot2)
library(ggpattern)

seattle$vanA_gcL <-
  (seattle$vanA_gc/seattle$effective_vol)*1000
```

```
for(i in uniqueID){
  if(seattle$wwtp[seattle$uniqueID == i] == "West Point"){
    seattle$vanA_adj[seattle$uniqueID == i] <-
      seattle$vanA_gcL[seattle$uniqueID ==
i]*seattle$wp_lpd[seattle$uniqueID == i]
  } else {
    seattle$vanA_adj[seattle$uniqueID == i] <-
      seattle$vanA_gcL[seattle$uniqueID ==
i]*seattle$sp_lpd[seattle$uniqueID == i]
  }
}

vanA_seattle <-
  ggplot(data = seattle,
         aes(x = sample_date, y = vanA_adj))+
    geom_col(aes(fill = wwtp),
             color = "black")+
    facet_wrap(~wwtp, nrow = 1,
               scales = "free"
               )+
    scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                      name = "WWTP")+
    labs(y = "vanA gc/person/day", x = "Date of Sample
Collection",
         title = "vanA"
         #subtitle = "Late 2020 to Early 2021"
         )+
    theme(legend.position = "bottom",
          panel.background = element_blank(),
          panel.grid = element_line(color = "gray"),
          panel.grid.minor.y = element_blank(),
          axis.text.x = element_text(angle = 45,
                                     size = 12,
                                     vjust = .5,
                                     color = "black"),
          axis.text.y = element_text(size = 12,
                                     color = "black"),
          axis.title = element_text(size = 16,
                                    color = "black",
                                    face = "bold"),
          axis.ticks = element_blank(),
          legend.title = element_text(size = 16, face = "bold"),
          legend.text = element_text(size = 14),
          plot.title = element_text(size = 18,
                                    face = "bold",
                                    hjust = 0.5),
          plot.subtitle = element_text(face = "italic",
                                       size = 14,
```

```
                                        hjust = 0.5),
          strip.text.x = element_text(size = 12, face =
"bold"))+
    scale_x_date(
      date_labels = "%b - %d",
      date_breaks = "2 weeks",
      limits = as.Date(c("2020-08-15", "2021-02-14")))+
  guides(pattern = guide_legend(override.aes =
                                list(
                                  pattern = c("stripe", "none"),
                                  pattern_spacing = 0.01)))+
  scale_y_continuous(trans = 'log10')
vanA_seattle

ggsave(plot = vanA_seattle,
       filename = "vanA_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)
```

## Seattle Case Data
```{r, tidy = TRUE}
covid19_seattle <- read_csv("covid19_seattle.csv",
                            col_types = cols(date =
col_date(format = "%m/%d/%Y")))

library(ggplot2)

covid_seattle <-
  ggplot(data = covid19_seattle, aes(x = date, y = case_count))+
  geom_col(fill = "#4b2e83")+
  scale_x_date(date_labels = "%b '%y",
               date_breaks = "2 weeks",
               limits = as.Date(c("2020-02-15", "2021-02-14")))+
  theme(legend.position = "bottom",
        plot.title=element_text(face="bold", hjust=0.5, size=24,
color = "#4b2e83"),
        axis.title=element_text(size=24),
        legend.text=element_text(size=12),
        legend.title = element_text(size = 12),
        axis.text=element_text(size=16),
        axis.text.x = element_text(angle = 45, hjust = 1, size =
16),
        panel.background = element_rect(fill = "white"),
        axis.line.x.bottom = element_line(color = "black"),
        axis.line.y = element_line(color = "black"),
        panel.grid.major = element_line(color = "lightgray",
size = 0.15),
          plot.subtitle = element_text(face = "italic",
```

```
                                        size = 14,
                                        hjust = 0.5))+
  labs(title = "Incident Cases Seattle",
       x = "Date",
       y = "Case Count",
       subtitle = "First Twelve Months")
covid_seattle

ggsave(plot = covid_seattle,
       filename = "covid_seattle.tiff",
       units = 'in', width = 14, height = 7, dpi = 600)

t.test(covid19_seattle$case_count[covid19_seattle$date<"2020-09-
01"])
t.test(covid19_seattle$case_count[covid19_seattle$date>="2020-
09-01" & covid19_seattle$date<="2021-02-16"])
max(covid19_seattle$case_count[covid19_seattle$date>="2020-09-
01" & covid19_seattle$date<="2021-02-16"])
print(covid19_seattle$date[covid19_seattle$case_count==295])
max(covid19_seattle$case_count[covid19_seattle$date>="2020-12-
01" & covid19_seattle$date<="2021-02-16"])
print(covid19_seattle$date[covid19_seattle$case_count==285])
```

## Seattle Stats
```{r, tidy = TRUE}
library(lmerTest)

seattle$day <- difftime(seattle$sample_date,
min(seattle$sample_date), units = "days")

mecAmod_sea <-
  lmer(data = seattle, log(mecA_adj) ~ day + (1|wwtp))
summary(mecAmod_sea)
confint(mecAmod_sea)
exp(confint(mecAmod_sea))

#temMod_sea <-
  #lmer(data = seattle, log(blaTEM_gcL) ~ day + (1|wwtp))
#summary(temMod_sea)
#confint(temMod_sea)

cmyMod_sea <-
  lmer(data = seattle, log(blaCMY_adj) ~ day + (1|wwtp))
summary(cmyMod_sea)
confint(cmyMod_sea)
exp(confint(cmyMod_sea))
```

```
shvMod_sea <-
  lmer(data = seattle, log(blaSHV_adj) ~ day + (1|wwtp))
summary(shvMod_sea)
confint(shvMod_sea)
exp(confint(shvMod_sea))

ctxaMod_sea <-
  lmer(data = seattle, log(blaCTXA_adj) ~ day + (1|wwtp))
summary(ctxaMod_sea)
confint(ctxaMod_sea)
exp(confint(ctxaMod_sea))

ctxbMod_sea <-
  lmer(data = seattle, log(blaCTXB_adj) ~ day + (1|wwtp))
summary(ctxbMod_sea)
confint(ctxbMod_sea)
exp(confint(ctxbMod_sea))

vanaMod_sea <-
  lmer(data = seattle, log(vanA_adj) ~ day + (1|wwtp))
summary(vanaMod_sea)
confint(vanaMod_sea)
exp(confint(vanaMod_sea))

mcr1_sea <-
  lmer(data = seattle, log(mcr1_adj) ~ day + (1|wwtp))
summary(mcr1_sea)
confint(mcr1_sea)
exp(confint(mcr1_sea))

library(dplyr)
library(lubridate)

```

## Average Values
```{r tidy = TRUE}

library(dplyr)
library(tidyverse)

total_mean <- seattle %>%
  mutate(number = length(wwtp)) %>%
  mutate(avg_mecA = mean(mecA_adj, na.rm = TRUE)) %>%
  #mutate(avg_tem = mean(blaTEM_gcL, na.rm = TRUE)) %>%
  mutate(avg_cmy = mean(blaCMY_adj, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_adj, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blaCTXB_adj, na.rm = TRUE)) %>%
```

```
  mutate(avg_ctxa = mean(blaCTXA_adj, na.rm = TRUE)) %>%
  mutate(avg_vanA = mean(vanA_adj, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_adj, na.rm = TRUE)) %>%
  select(
    wwtp, avg_mecA,
    #avg_tem,
    avg_ctxb, avg_ctxa,
    avg_cmy, avg_shv,
    avg_mcr1,
    avg_vanA, number
  ) %>%
  distinct(avg_mecA, .keep_all = TRUE)
total_mean$wwtp <- "total"

mean <- seattle %>%
  group_by(wwtp) %>%
  mutate(number = length(wwtp)) %>%
  mutate(avg_mecA = mean(mecA_adj, na.rm = TRUE)) %>%
  #mutate(avg_tem = mean(bla_tem_gcL, na.rm = TRUE)) %>%
  mutate(avg_cmy = mean(blaCMY_adj, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_adj, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blaCTXB_adj, na.rm = TRUE)) %>%
  mutate(avg_ctxa = mean(blaCTXA_adj, na.rm = TRUE)) %>%
  mutate(avg_vanA = mean(vanA_adj, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_adj, na.rm = TRUE)) %>%
  select(
    wwtp, avg_mecA,
    #avg_tem,
    avg_cmy, avg_shv, avg_ctxa, avg_ctxb,
    avg_mcr1,
    avg_vanA, number
  ) %>%
  distinct(wwtp, .keep_all = TRUE)

mean <- bind_rows(mean, total_mean)
write.csv(mean, file = "seattle_mean_values.csv")

```

## dPCR data
```{r, tidy = TRUE}
seattle_dpcr <-

read_csv("~/OneDrive/Documents/UW/EOHML/AMR/Seattle/Figures/seat
tle dpcr amr results.csv",
           col_types = cols(exp_date = col_date(format =
"%m/%d/%y"),
```

```
                              sample_date = col_date(format =
"%m/%d/%y")))

seattle_dpcr$bla_cmy_gc <- seattle_dpcr$bla_cmy_gcuL*2
seattle_dpcr$bla_shv_gc <- seattle_dpcr$bla_shv_gcuL*2
seattle_dpcr$bla_ctxa_gc <- seattle_dpcr$bla_ctxa_gcuL*2
seattle_dpcr$bla_ctxb_gc <- seattle_dpcr$bla_ctxb_gcuL*2
seattle_dpcr$mecA_gc <- seattle_dpcr$mecA_gcuL*2
seattle_dpcr$vanA_gc <- seattle_dpcr$vanA_gc*2

library(dplyr)

#get dPCR values adjusted by effective volume assayed
effective_vol <- seattle %>%
  select(sample_date, wwtp, effective_vol)
seattle_dpcr <- left_join(seattle_dpcr, effective_vol, by =
c("sample_date", "wwtp"))
seattle_dpcr$mecA_gcL <-
(seattle_dpcr$mecA_gc/seattle_dpcr$effective_vol)*1000
seattle_dpcr$vanA_gcL <-
(seattle_dpcr$vanA_gc/seattle_dpcr$effective_vol)*1000
seattle_dpcr$bla_cmy_gcL <-
(seattle_dpcr$bla_cmy_gc/seattle_dpcr$effective_vol)*1000
seattle_dpcr$bla_shv_gcL <-
(seattle_dpcr$bla_shv_gc/seattle_dpcr$effective_vol)*1000
seattle_dpcr$bla_ctxa_gcL <-
(seattle_dpcr$bla_ctxa_gc/seattle_dpcr$effective_vol)*1000
seattle_dpcr$bla_ctxb_gcL <-
(seattle_dpcr$bla_ctxb_gc/seattle_dpcr$effective_vol)*1000

seattle_dpcr <- seattle_dpcr %>%
  left_join(wwtp_flow, by = "sample_date")
seattle_dpcr$wp_lpd <-
(seattle_dpcr$wp_flow*1000000*3.785)/787000
seattle_dpcr$sp_lpd <-
(seattle_dpcr$sp_flow*1000000*3.785)/839000

uniqueID <- 1:length(seattle_dpcr$sample_date)
seattle_dpcr$uniqueID <- uniqueID

for(i in uniqueID){
  if(seattle_dpcr$wwtp[seattle_dpcr$uniqueID == i] == "West
Point"){
    seattle_dpcr$mecA_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$mecA_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$wp_lpd[seattle_dpcr$uniqueID == i]
  } else {
    seattle_dpcr$mecA_adj[seattle_dpcr$uniqueID == i] <-
```

```
      seattle_dpcr$mecA_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$sp_lpd[seattle_dpcr$uniqueID == i]
  }
}

for(i in uniqueID){
  if(seattle_dpcr$wwtp[seattle_dpcr$uniqueID == i] == "West
Point"){
    seattle_dpcr$blaCMY_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_cmy_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$wp_lpd[seattle_dpcr$uniqueID == i]
  } else {
    seattle_dpcr$blaCMY_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_cmy_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$sp_lpd[seattle_dpcr$uniqueID == i]
  }
}

for(i in uniqueID){
  if(seattle_dpcr$wwtp[seattle_dpcr$uniqueID == i] == "West
Point"){
    seattle_dpcr$blaSHV_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_shv_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$wp_lpd[seattle_dpcr$uniqueID == i]
  } else {
    seattle_dpcr$blaSHV_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_shv_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$sp_lpd[seattle_dpcr$uniqueID == i]
  }
}

for(i in uniqueID){
  if(seattle_dpcr$wwtp[seattle_dpcr$uniqueID == i] == "West
Point"){
    seattle_dpcr$blaCTXB_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_ctxb_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$wp_lpd[seattle_dpcr$uniqueID == i]
  } else {
    seattle_dpcr$blaCTXB_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_ctxb_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$sp_lpd[seattle_dpcr$uniqueID == i]
  }
}

for(i in uniqueID){
  if(seattle_dpcr$wwtp[seattle_dpcr$uniqueID == i] == "West
Point"){
    seattle_dpcr$blaCTXA_adj[seattle_dpcr$uniqueID == i] <-
```

```
      seattle_dpcr$bla_ctxa_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$wp_lpd[seattle_dpcr$uniqueID == i]
  } else {
    seattle_dpcr$blaCTXA_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$bla_ctxa_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$sp_lpd[seattle_dpcr$uniqueID == i]
  }
}

for(i in uniqueID){
  if(seattle_dpcr$wwtp[seattle_dpcr$uniqueID == i] == "West
Point"){
    seattle_dpcr$vanA_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$vanA_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$wp_lpd[seattle_dpcr$uniqueID == i]
  } else {
    seattle_dpcr$vanA_adj[seattle_dpcr$uniqueID == i] <-
      seattle_dpcr$vanA_gcL[seattle_dpcr$uniqueID ==
i]*seattle_dpcr$sp_lpd[seattle_dpcr$uniqueID == i]
  }
}

seattle_sub <- seattle %>%
  select(sample_date, wwtp, effective_vol, mecA_adj, blaCMY_adj,
blaSHV_adj, blaCTXA_adj, blaCTXB_adj, vanA_adj)

dpcr_comp <- left_join(seattle_dpcr, seattle_sub, by =
c("sample_date", "wwtp"))

#Stats
mecAdpcr <- wilcox.test(dpcr_comp$mecA_adj.x,
dpcr_comp$mecA_adj.y, paired = TRUE)
vanAdpcr <- wilcox.test(dpcr_comp$vanA_adj.x,
dpcr_comp$vanA_adj.y, paired = TRUE)
blaCMYdpcr <- wilcox.test(dpcr_comp$blaCMY_adj.x,
dpcr_comp$blaCMY_adj.y, paired = TRUE)
blaSHVdpcr <- wilcox.test(dpcr_comp$blaSHV_adj.x,
dpcr_comp$blaSHV_adj.y, paired = TRUE)
blaCTXAdpcr <- wilcox.test(dpcr_comp$blaCTXA_adj.x,
dpcr_comp$blaCTXA_adj.y, paired = TRUE)
blaCTXBdpcr <- wilcox.test(dpcr_comp$blaCTXB_adj.x,
dpcr_comp$blaCTXB_adj.y, paired = TRUE)

#extract values for table
dpcr_pvalues <- c(mecAdpcr$p.value, vanAdpcr$p.value,
                  blaCMYdpcr$p.value, blaSHVdpcr$p.value,
                  blaCTXAdpcr$p.value, blaCTXBdpcr$p.value)
```

```
targets <- c("mecA", "vanA", "blaCMY", "blaSHV", "blaCTXM-A",
"blaCTXM-B")
qpcr_mean <- c(mean(dpcr_comp$mecA_adj.y, na.rm = TRUE),
               mean(dpcr_comp$vanA_adj.y, na.rm = TRUE),
               mean(dpcr_comp$blaCMY_adj.y, na.rm = TRUE),
               mean(dpcr_comp$blaSHV_adj.y, na.rm = TRUE),
               mean(dpcr_comp$blaCTXA_adj.y, na.rm = TRUE),
               mean(dpcr_comp$blaCTXB_adj.y, na.rm = TRUE))

qpcr_ci_lower <- c(t.test(dpcr_comp$mecA_adj.y)$conf.int[1],
                   t.test(dpcr_comp$vanA_adj.y)$conf.int[1],
                   t.test(dpcr_comp$blaCMY_adj.y)$conf.int[1],
                   t.test(dpcr_comp$blaSHV_adj.y)$conf.int[1],
                   t.test(dpcr_comp$blaCTXA_adj.y)$conf.int[1],
                   t.test(dpcr_comp$blaCTXB_adj.y)$conf.int[1])

qpcr_ci_upper <- c(t.test(dpcr_comp$mecA_adj.y)$conf.int[2],
                   t.test(dpcr_comp$vanA_adj.y)$conf.int[2],
                   t.test(dpcr_comp$blaCMY_adj.y)$conf.int[2],
                   t.test(dpcr_comp$blaSHV_adj.y)$conf.int[2],
                   t.test(dpcr_comp$blaCTXA_adj.y)$conf.int[2],
                   t.test(dpcr_comp$blaCTXB_adj.y)$conf.int[2])

dpcr_mean <- c(mean(dpcr_comp$mecA_adj.x, na.rm = TRUE),
               mean(dpcr_comp$vanA_adj.x, na.rm = TRUE),
               mean(dpcr_comp$blaCMY_adj.x, na.rm = TRUE),
               mean(dpcr_comp$blaSHV_adj.x, na.rm = TRUE),
               mean(dpcr_comp$blaCTXA_adj.x, na.rm = TRUE),
               mean(dpcr_comp$blaCTXB_adj.x, na.rm = TRUE))

dpcr_ci_lower <- c(t.test(dpcr_comp$mecA_adj.x)$conf.int[1],
                   t.test(dpcr_comp$vanA_adj.x)$conf.int[1],
                   t.test(dpcr_comp$blaCMY_adj.x)$conf.int[1],
                   t.test(dpcr_comp$blaSHV_adj.x)$conf.int[1],
                   t.test(dpcr_comp$blaCTXA_adj.x)$conf.int[1],
                   t.test(dpcr_comp$blaCTXB_adj.x)$conf.int[1])

dpcr_ci_upper <- c(t.test(dpcr_comp$mecA_adj.x)$conf.int[2],
                   t.test(dpcr_comp$vanA_adj.x)$conf.int[2],
                   t.test(dpcr_comp$blaCMY_adj.x)$conf.int[2],
                   t.test(dpcr_comp$blaSHV_adj.x)$conf.int[2],
                   t.test(dpcr_comp$blaCTXA_adj.x)$conf.int[2],
                   t.test(dpcr_comp$blaCTXB_adj.x)$conf.int[2])

pcr_comp_values <- cbind(targets,
                         qpcr_mean, qpcr_ci_lower,
qpcr_ci_upper,
```

```
                          dpcr_mean, dpcr_ci_lower,
dpcr_ci_upper,
                          dpcr_pvalues)
pcr_comp_values <- data.frame(pcr_comp_values)
write.csv(pcr_comp_values, file = "pcr_comparison_.csv")
```


```
---
title: "Prospective Sampling AMR"
author: "Sarah Philo"
date: "12/14/2022"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo=FALSE, message=FALSE, warnings=FALSE,
fig.align='center')

#Load packages
library(tidyverse) #for ggplot2, dplyr, etc
library(kableExtra) #for making nice-looking tables
library(knitr)
library(sandwich) #standard errors
library(readr)
library(qwraps2)
library(gt)
library(dagR)
library(rigr)
library(ggplot2)
library(ggpubr)
library(dplyr)
options(qwraps2_markup = "markdown")
options(tinytex.verbose = TRUE)
```


##Load and Prep Data

```{r Load Data}
#Seattle Data
pros_seattle <-
  read_csv("seattle_amr_prospective.csv",
           col_types = cols(exp_date = col_date(format =
"%m/%d/%y"),
                            sample_date = col_date(format =
"%m/%d/%y")))
pros_seattle$location <- pros_seattle$wwtp
```
```

```
#Effective volumr in mL
pros_seattle$method_vol <- pros_seattle$method_vol*1000
pros_seattle$effective_vol <- pros_seattle$method_vol*
  (pros_seattle$ext_vol/pros_seattle$final_vol)*
  (pros_seattle$pcr_vol/pros_seattle$elu_vol)
pros_seattle$effective_vol <- pros_seattle$effective_vol*0.001
summary(pros_seattle$effective_vol)

#Set Sampling Location Criteria
pros_seattle$country <- "USA"
pros_seattle$urban <- ifelse(pros_seattle$wwtp=="West Point", 1,
0)

#Calculate gc/L
pros_seattle$blaCMY_gcL <-
  (pros_seattle$bla_cmy_gc/pros_seattle$effective_vol)*1000
pros_seattle$blaSHV_gcL <-
  (pros_seattle$bla_shv_gc/pros_seattle$effective_vol)*1000
pros_seattle$blaCTXA_gcL <-
  (pros_seattle$bla_ctxa_gc/pros_seattle$effective_vol)*1000
pros_seattle$blaCTXB_gcL <-
  (pros_seattle$bla_ctxb_gc/pros_seattle$effective_vol)*1000
pros_seattle$mecA_gcL <-
  (pros_seattle$mecA_gc/pros_seattle$effective_vol)*1000
pros_seattle$vanA_gcL <-
  (pros_seattle$vanA_gc/pros_seattle$effective_vol)*1000
pros_seattle$mcr1_gcL <-
  (pros_seattle$mcr1_gc/pros_seattle$effective_vol)*1000

#Only pull the variables I need
pros_seattle <- pros_seattle %>%
  select(location, sample_date, country, urban, effective_vol,
         blaCMY_gcL, blaSHV_gcL, blaCTXA_gcL, blaCTXB_gcL,
mecA_gcL, vanA_gcL, mcr1_gcL)

seattle_flow <- read_csv("seattle_flow.csv",
                         col_types = cols(sample_date =
col_date(format = "%m/%d/%Y")))

#Adjust Seattle Data
pros_seattle <- merge(pros_seattle, seattle_flow)

pros_seattle$flow_wp <- pros_seattle$flow_wp*1000000
pros_seattle$flow_bw <- pros_seattle$flow_bw*1000000
pros_seattle$wp_lpd <- (pros_seattle$flow_wp*3.785)/787000
pros_seattle$bw_lpd <- (pros_seattle$flow_bw*3.785)/268000
pros_seattle$wp_min <- min(pros_seattle$wp_lpd)
```

```
pros_seattle$wp_norm <- pros_seattle$wp_lpd/pros_seattle$wp_min
pros_seattle$bw_min <- min(pros_seattle$bw_lpd)
pros_seattle$bw_norm <- pros_seattle$bw_lpd/pros_seattle$bw_min

dates <- pros_seattle$sample_date
pros_seattle$blaCMY_adj <- NA
pros_seattle$blaSHV_adj <- NA
pros_seattle$blaCTXA_adj <- NA
pros_seattle$blaCTXB_adj <- NA
pros_seattle$mecA_adj <- NA
pros_seattle$vanA_adj <- NA
pros_seattle$mcr1_adj <- NA

for(i in dates){
 if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$blaCMY_adj[pros_seattle$sample_date == i] <-
     pros_seattle$blaCMY_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$blaCMY_adj[pros_seattle$sample_date == i] <-
     pros_seattle$blaCMY_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
  if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$blaSHV_adj[pros_seattle$sample_date == i] <-
     pros_seattle$blaSHV_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$blaSHV_adj[pros_seattle$sample_date == i] <-
     pros_seattle$blaSHV_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
  if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$blaCTXA_adj[pros_seattle$sample_date == i] <-
     pros_seattle$blaCTXA_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$blaCTXA_adj[pros_seattle$sample_date == i] <-
     pros_seattle$blaCTXA_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
  if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$blaCTXB_adj[pros_seattle$sample_date == i] <-
```

```
    pros_seattle$blaCTXB_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$blaCTXB_adj[pros_seattle$sample_date == i] <-
    pros_seattle$blaCTXB_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
  if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$mecA_adj[pros_seattle$sample_date == i] <-
    pros_seattle$mecA_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$mecA_adj[pros_seattle$sample_date == i] <-
    pros_seattle$mecA_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
  if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$vanA_adj[pros_seattle$sample_date == i] <-
    pros_seattle$vanA_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$vanA_adj[pros_seattle$sample_date == i] <-
    pros_seattle$vanA_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
  if(pros_seattle$location[pros_seattle$sample_date == i] ==
"West Point"){
   pros_seattle$mcr1_adj[pros_seattle$sample_date == i] <-
    pros_seattle$mcr1_gcL[pros_seattle$sample_date ==
i]*pros_seattle$wp_lpd[pros_seattle$sample_date == i]
 } else {
   pros_seattle$mcr1_adj[pros_seattle$sample_date == i] <-
    pros_seattle$mcr1_gcL[pros_seattle$sample_date ==
i]*pros_seattle$bw_lpd[pros_seattle$sample_date == i]
 }
}

pros_seattle <- pros_seattle %>%
  select(location, sample_date, country, urban, effective_vol,
         blaCMY_adj, blaSHV_adj, blaCTXA_adj, blaCTXB_adj,
mecA_adj, vanA_adj, mcr1_adj)
colnames(pros_seattle) <- c("location", "sample_date",
"country", "urban", "effective_vol",
                            "blaCMY_gcL", "blaSHV_gcL",
"blaCTXA_gcL", "blaCTXB_gcL", "mecA_gcL", "vanA_gcL",
"mcr1_gcL")
```

```
#Portugal Data
pros_port <-
  read_csv("Prospective Samples Lisbon.csv",
           col_types = cols(sample_date = col_date(format =
"%m/%d/%y")))
pros_port$ext_vol <- 110
pros_port$elu_vol <- 100
pros_port$pcr_vol <- 1
pros_port$volume_final <- pros_port$volume_final*1000

#Effective Volume in mL
pros_port$effective_vol <-
  pros_port$volume_initial*
  (pros_port$ext_vol/pros_port$volume_final)*
  (pros_port$pcr_vol/pros_port$elu_vol)*1000
summary(pros_port$effective_vol)

#Set Sampling Location Criteria
pros_port$country <- "Portugal"
pros_port$urban <- ifelse(pros_port$location=="Beirolas", 1, 0)

#Calculate gc/L for each target
pros_port$blaCMY_gcL <-
  (pros_port$bla_cmy_gc/pros_port$effective_vol)*1000
pros_port$blaSHV_gcL <-
  (pros_port$bla_shv_gc/pros_port$effective_vol)*1000
pros_port$blaCTXA_gcL <-
  (pros_port$bla_ctxa_gc/pros_port$effective_vol)*1000
pros_port$blaCTXB_gcL <-
  (pros_port$bla_ctxb_gc/pros_port$effective_vol)*1000
pros_port$mecA_gcL <-
  (pros_port$mecA_gc/pros_port$effective_vol)*1000
pros_port$vanA_gcL <-
  (pros_port$vanA_gc/pros_port$effective_vol)*1000
pros_port$mcr1_gcL <-
  (pros_port$mcr1_gc/pros_port$effective_vol)*1000

pros_port <- pros_port %>%
  select(location, sample_date, country, urban, effective_vol,
         blaCMY_gcL, blaSHV_gcL, blaCTXA_gcL, blaCTXB_gcL,
mecA_gcL, vanA_gcL, mcr1_gcL)

prospective <- rbind(pros_seattle, pros_port)

prospective$urban <- factor(prospective$urban,
                            levels = c(0, 1),
```

```
                              labels = c("Rural", "Urban"))
prospective$country <- factor(prospective$country,
                              levels = c("USA", "Portugal"))

#Get Sample Dates Table

```


```{r boxplots}
cmy_boxplots <-
  ggplot(data = prospective, aes(x = urban, y = blaCMY_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "blaCMY")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
cmy_boxplots
ggsave(plot = cmy_boxplots, dpi = 300, filename =
"cmy_boxplots.tiff", height = 4.5, width = 7)

shv_boxplots <-
  ggplot(data = prospective, aes(x = urban, y = blaSHV_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
```

```
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "blaSHV")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
shv_boxplots
ggsave(plot = shv_boxplots, dpi = 300, filename =
"shv_boxplots.tiff", height = 4.5, width = 7)

ctxa_boxplots <-
  ggplot(data = prospective, aes(x = urban, y = blaCTXA_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "blaCTX-M-1,9")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
ctxa_boxplots
ggsave(plot = ctxa_boxplots, dpi = 300, filename =
"ctxa_boxplots.tiff", height = 4.5, width = 7)

ctxb_boxplots <-
```

```
  ggplot(data = prospective, aes(x = urban, y = blaCTXB_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "blaCTX-M-2,8,25")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
ctxb_boxplots
ggsave(plot = ctxb_boxplots, dpi = 300, filename =
"ctxb_boxplots.tiff", height = 4.5, width = 7)

meca_boxplots <-
  ggplot(data = prospective, aes(x = urban, y = mecA_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "mecA")+
```

```
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
meca_boxplots
ggsave(plot = meca_boxplots, dpi = 300, filename =
"meca_boxplots.tiff", height = 4.5, width = 7)

vana_boxplots <-
  ggplot(data = prospective, aes(x = urban, y = vanA_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "vanA")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
vana_boxplots
ggsave(plot = vana_boxplots, dpi = 300, filename =
"vana_boxplots.tiff", height = 4.5, width = 7)

mcr1_boxplots <-
  ggplot(data = prospective, aes(x = urban, y = mcr1_gcL))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"))+
```

```
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~country, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "mcr1")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.1)))
mcr1_boxplots
ggsave(plot = mcr1_boxplots, dpi = 300, filename =
"mcr1_boxplots.tiff", height = 4.5, width = 7)
```
```{r figure}
boxplots <-
  ggarrange(
    cmy_boxplots, shv_boxplots, ctxa_boxplots, ctxb_boxplots,
meca_boxplots, vana_boxplots, mcr1_boxplots,
    ncol = 2, nrow = 4,
    labels = "AUTO", legend = "none"
  )
boxplots
boxplots <- annotate_figure(boxplots, top = text_grob("ARG
Abundances (gc/L)", face = "bold", size = 20))
boxplots
ggsave(plot = boxplots, filename = "boxplots.tiff", height = 11,
width = 8.5)
```

#Boxplots v2
```{r long plot}
pros_long <- prospective %>%
  pivot_longer(cols = c(blaCMY_gcL, blaSHV_gcL, blaCTXA_gcL,
blaCTXB_gcL, mecA_gcL, vanA_gcL, mcr1_gcL),
               names_to = "target",
               values_to = "arg_abundance")

pros_long$target <-
  factor(pros_long$target,
         levels = c("blaCMY_gcL", "blaSHV_gcL", "blaCTXA_gcL",
"blaCTXB_gcL", "mecA_gcL", "vanA_gcL", "mcr1_gcL"),
         labels = c("blaCMY", "blaSHV", "blaCTX-M Groups 1, 9",
"blaCTX-M Groups 2, 8, 25", "mecA", "vanA", "mcr1"))

prospective$urban <- factor(prospective$urban,
                            levels = c(0, 1),
```

```
                          labels = c("Rural", "Urban"))

long_plot_usa <-
  ggplot(data = subset(pros_long, country == "USA"),
         aes(x = urban, y = arg_abundance))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"),
        plot.title = element_text(size = 14, face = "bold",
hjust = 0.5))+
  geom_boxplot(aes(fill = urban, color = urban),
               alpha = 0.8)+
  facet_wrap(~target, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "Abundance gc/person/day", title = "King
County")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.2)))
long_plot_usa

long_plot_port <-
  ggplot(data = subset(pros_long, country == "Portugal"),
         aes(x = urban, y = arg_abundance))+
  theme_bw()+
  theme(legend.position = "none",
        panel.background = element_blank(),
        panel.grid = element_line(color = "gray"),
        axis.text.x = element_text(size = 10,  vjust = .5, color
= "black"),
        axis.text.y = element_text(size = 10, color = "black"),
        axis.title = element_text(size = 12, color = "black",
face = "bold"),
        axis.ticks = element_blank(),
        strip.text.x = element_text(size = 10, face = "bold"),
        title = element_text(size = 14, face = "bold"))+
  geom_boxplot(aes(fill = urban, color = urban),
```

```
              alpha = 0.8)+
  facet_wrap(~target, scales = "free_y")+
  scale_fill_manual(values = c("#B7A57A", "#4b2e83"),
                    name = "Region")+
  scale_color_manual(values = c("black", "black"),
                     name = "Region")+
  labs(x = "", y = "Abundance gc/L", title = "Portugal")+
  stat_compare_means(method = "wilcox.test", comparisons =
list(c("Rural", "Urban")))+
  scale_y_continuous(expand = expansion(mult = c(0.05, 0.2)))
long_plot_port

long_plot <-
  ggarrange(
    long_plot_usa, long_plot_port,
    nrow = 2,
    labels = "AUTO", legend = "none"
  )
long_plot
ggsave(plot = long_plot, filename = "long_boxplots.tiff", height
= 11, width = 8.5)



```



#Wilcox tests
```{r statistical comparisons}
sea_cmy <-
wilcox.test(pros_seattle$blaCMY_gcL[pros_seattle$urban == 0],
pros_seattle$blaCMY_gcL[pros_seattle$urban == 1])
sea_shv <-
wilcox.test(pros_seattle$blaSHV_gcL[pros_seattle$urban == 0],
pros_seattle$blaSHV_gcL[pros_seattle$urban == 1])
sea_ctxa <-
wilcox.test(pros_seattle$blaCTXA_gcL[pros_seattle$urban == 0],
pros_seattle$blaCTXA_gcL[pros_seattle$urban == 1])
sea_ctxb <-
wilcox.test(pros_seattle$blaCTXB_gcL[pros_seattle$urban == 0],
pros_seattle$blaCTXB_gcL[pros_seattle$urban == 1])
sea_meca <- wilcox.test(pros_seattle$mecA_gcL[pros_seattle$urban
== 0], pros_seattle$mecA_gcL[pros_seattle$urban == 1])
sea_vana <- wilcox.test(pros_seattle$vanA_gcL[pros_seattle$urban
== 0], pros_seattle$vanA_gcL[pros_seattle$urban == 1])

sea_pvalues <-
```

```
  c(sea_cmy$p.value, sea_shv$p.value, sea_ctxa$p.value,
sea_ctxb$p.value,
    sea_meca$p.value, sea_vana$p.value, "")

port_cmy <- wilcox.test(pros_port$blaCMY_gcL[pros_port$urban ==
0], pros_port$blaCMY_gcL[pros_port$urban == 1])
port_shv <- wilcox.test(pros_port$blaSHV_gcL[pros_port$urban ==
0], pros_port$blaSHV_gcL[pros_port$urban == 1])
port_ctxa <- wilcox.test(pros_port$blaCTXA_gcL[pros_port$urban
== 0], pros_port$blaCTXA_gcL[pros_port$urban == 1])
port_ctxb <- wilcox.test(pros_port$blaCTXB_gcL[pros_port$urban
== 0], pros_port$blaCTXB_gcL[pros_port$urban == 1])
port_meca <- wilcox.test(pros_port$mecA_gcL[pros_port$urban ==
0], pros_port$mecA_gcL[pros_port$urban == 1])
port_vana <- wilcox.test(pros_port$vanA_gcL[pros_port$urban ==
0], pros_port$vanA_gcL[pros_port$urban == 1])
port_mcr1 <- wilcox.test(pros_port$mcr1_gcL[pros_port$urban ==
0], pros_port$mcr1_gcL[pros_port$urban == 1])

port_pvalues <-
  c(port_cmy$p.value, port_shv$p.value, port_ctxa$p.value,
port_ctxb$p.value,
    port_meca$p.value, port_vana$p.value, port_mcr1$p.value)

targets <- c("blaCMY", "blaSHV", "blaCTX-M-1,9", "blaCTXM-
2,8,25", "mecA", "vanA", "mcr1")

pvalues <- data.frame(rbind(sea_pvalues, port_pvalues))
colnames(pvalues) <- targets
write.csv(pvalues, file = "wilcox_p_values.csv")
```

#Descriptive Statistics
```{r descriptive stats}
mean <- prospective %>%
  group_by(country, urban) %>%
  mutate(avg_cmy = mean(blaCMY_gcL, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxa = mean(blaCTXA_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blaCTXB_gcL, na.rm = TRUE)) %>%
  mutate(avg_meca = mean(mecA_gcL, na.rm = TRUE)) %>%
  mutate(avg_vana = mean(vanA_gcL, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_gcL, na.rm = TRUE)) %>%
  distinct(country, urban, .keep_all = TRUE) %>%
  select(country, urban, avg_cmy, avg_shv, avg_ctxa, avg_ctxb,
avg_meca, avg_vana, avg_mcr1)

total_mean <- prospective %>%
```

```
  mutate(avg_cmy = mean(blaCMY_gcL, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxa = mean(blaCTXA_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blaCTXB_gcL, na.rm = TRUE)) %>%
  mutate(avg_meca = mean(mecA_gcL, na.rm = TRUE)) %>%
  mutate(avg_vana = mean(vanA_gcL, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_gcL, na.rm = TRUE)) %>%
  distinct(avg_cmy, .keep_all = TRUE) %>%
  select(avg_cmy, avg_shv, avg_ctxa, avg_ctxb, avg_meca,
avg_vana, avg_mcr1)

country_mean <- prospective %>%
  group_by(country) %>%
  mutate(avg_cmy = mean(blaCMY_gcL, na.rm = TRUE)) %>%
  mutate(avg_shv = mean(blaSHV_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxa = mean(blaCTXA_gcL, na.rm = TRUE)) %>%
  mutate(avg_ctxb = mean(blaCTXB_gcL, na.rm = TRUE)) %>%
  mutate(avg_meca = mean(mecA_gcL, na.rm = TRUE)) %>%
  mutate(avg_vana = mean(vanA_gcL, na.rm = TRUE)) %>%
  mutate(avg_mcr1 = mean(mcr1_gcL, na.rm = TRUE)) %>%
  distinct(country,.keep_all = TRUE) %>%
  select(country, urban, avg_cmy, avg_shv, avg_ctxa, avg_ctxb,
avg_meca, avg_vana, avg_mcr1)
write.csv(country_mean, file = "country_mean.csv")

mean_table <- bind_rows(mean, total_mean)

write.csv(mean_table, file = "mean_values.csv")
```


## Code appendix
```{r code appendix, ref.label=knitr::all_labels(), echo=TRUE,
eval=FALSE}
```

# 4 Conclusions

The main goals of this study were two-fold: to compare and optimize wastewater concentration methods for SARS-CoV-2, and to apply those methods to wastewater surveillance for antimicrobial resistance genes. We were able to show that many different methods will work to conduct SARS-CoV-2 wastewater surveillance. The specific protocol choice depends on different goals including precision of the measurement and recovery of the control organism. An initial methods comparison of the bag mediated filtration system, skimmed milk flocculation, ultrafiltration, and polyethylene glycol precipitation identified skimmed milk flocculation as a low-cost and effective SARS-CoV-2 method. Skimmed milk flocculation was later improved by resuspending the pellet in the extraction kit lysis buffer. It was then shown that the specific choice of extraction kit drastically affects recovery and SARS-CoV-2 detection. Switching between kits can have substantial effects on results. Finally, this study showed that there is still room for improvement by testing a novel technology: isolating virus from wastewater using magnetic beads. The CeresNano Nanotrap ® particles significantly improved on both OC43 recovery and SARS-CoV-2 detection.

Despite the extensive amount of work, there were some research questions that were not answered. First, all the work carried out used endogenous SARS-CoV-2. Research should be done using inactivated and SARS-CoV-2 to compare its behavior with OC43 and to quantify how much of the virus of interest is recovered with the chosen methods and where target loss occurs within each method. There are additionally unanswered questions about the types of inhibitors in the RNA extract and how best to remove inhibitors such as lactose from the skimmed milk flocculation. There were additional concentration methods that were not tested, such as extracting

wastewater solids and using different weights and concentrations of polyethylene glycol. We were also unable to test different sampling schemes such as grab or composite sampling, optimizing the time of day to collect samples, and sampling at different locations in the conveyance system (manholes vs lift stations vs treatment plants). Finally, quantifying the variability in SARS-CoV-2 wastewater data is crucial to help public health officials interpret the data. While a model was constructed, it needs to be refined and updated with more recent data that includes newly emerged variants.

Wastewater collected before, during, and after the late 2020 SARS-CoV-2 peak in Portugal and the Seattle-area was tested for seven different antimicrobial resistance genes (ARGs) to understand if widespread use of antimicrobials during the pandemic resulted in increased environmental resistance. This study showed that both *mecA* and $bla_{CTXM\text{-}2,8,25}$ significantly increased in Portugal from August – November 2020. The *mecA* increase could be due to co-infection with methicillin-resistant *S. aureus* infections commonly associated with ventilated patients. In Seattle, all the beta-lactamase genes tested statistically decreased. This could potentially be due to dilution of the signal in the watershed due to rain or shifting resistomes due to the pandemic. When comparing ARGs between urban and rural treatment plants, all genes except *mcr1* were significantly higher in the Seattle urban treatment plant, but there were no differences between the Portuguese treatment plants. Understanding which communities are hotspots for ARGs is important to control resistance in the world.

The antimicrobial resistance work is limited in a number of different ways. We were unable to conduct sequencing to understand the genetic context of the ARGs. This is crucial to understand if the detected ARGs are a threat to human and animal health. Additionally, more data must be collected about the occurrence of resistant infections in

hospitals and ventilated COVID-19 patients to more directly link changing environmental resistance with clinical settings. Wastewater gene abundance data should also be adjusted for influent flow rates at treatment plants to better control for signal dilution.

This project shows that wastewater surveillance is a useful tool in public health, but that it is not the answer to all public health surveillance challenges. In the right setting, it can prove to be incredibly useful. The specific use case must be considered when designing a wastewater surveillance program. Wastewater surveillance to determine if a pathogen is in a congregate living facility such as a university dorm or long-term care facility likely does not require quantification. Skimmed milk flocculation, while less efficient for quantifying SARS-CoV-2, would be sufficient for a use case in which presence/absence is the goal. However, if the goal is to sequence the sample, skimmed milk flocculation likely is not efficient enough. Concentration with the Nanotrap ® beads would be more suited to this situation. Guidance on choosing and optimizing methods for specific use cases is needed to help move the field of wastewater surveillance forward.

Data from this research suggest that the chosen extraction protocol strongly affects the results. OC43 recovery and SARS-CoV-2 detection was significantly different for each extraction kit. Many different concentration protocols will effectively isolate various pathogens, but none of that matters if the extraction kit doesn't clean up wastewater inhibitors effectively. In circumstances where researchers and practitioners cannot optimize each step, the extraction protocol should be the main focus. Throughout this project we lay out steps to take and the data needed to validate methods for different targets. The project contributes useful information to the growing field of wastewater surveillance.

# 5 References


(2022) Facts about the King County regional wastewater system. Vol. 2022 p.^pp.

Aarestrup F, M. & Woolhouse M, E. J. (2020) Using sewage for surveillance of antimicrobial resistance. *Science* **367**: 630-632.

Adalja A, Watson M, Toner E, Cicero A & Inglesby T (2018) The Characteristics of Pandemic Pathogens. p.^pp. Center for Health Security, Johns Hopkins Bloomberg School of Public Health.

Ahmed W, Harwood VJ, Gyawali P, Sidhu JPS & Toze S (2015) Comparison of Concentration Methods for Quantitative Detection of Sewage-Associated Viral Markers in Environmental Waters. *Applied and Environmental Microbiology* **81**: 2042-2049.

Ahmed W, Bivins A, Korajkic A, Metcalfe S, Smith WJM & Simpson SL (2023) Comparative analysis of Adsorption-Extraction (AE) and Nanotrap® Magnetic Virus Particles (NMVP) workflows for the recovery of endogenous enveloped and non-enveloped viruses in wastewater. *Science of The Total Environment* **859**: 160072.

Ahmed W, Bivins A, Simpson SL, Smith WJM, Metcalfe S, McMinn B, Symonds EM & Korajkic A (2021) Comparative analysis of rapid concentration methods for the recovery of SARS-CoV-2 and quantification of human enteric viruses and a sewage-associated marker gene in untreated wastewater. *Science of The Total Environment* **799**: 149386.

Ahmed W, Gyawali P, Hamilton KA, Joshi S, Aster D, Donner E, Simpson SL & Symonds EM (2021) Antibiotic Resistance and Sewage-Associated Marker Genes in Untreated Sewage and a River Characterized During Baseflow and Stormflow. *Frontiers in Microbiology* **12**.

Ahmed W, Smith WJM, Metcalfe S*, et al.* (2022) Comparison of RT-qPCR and RT-dPCR Platforms for the Trace Detection of SARS-CoV-2 RNA in Wastewater. *ACS ES&T Water* **2**: 1871-1880.

Ahmed W, Bivins A, Bertsch PM*, et al.* (2020) Surveillance of SARS-CoV-2 RNA in wastewater: Methods optimisation and quality control are crucial for generating reliable public health information. *Curr Opin Environ Sci Health*.

Ahmed W, Bertsch PM, Bivins A*, et al.* (2020) Comparison of virus concentration methods for the RT-qPCR-based recovery of murine hepatitis virus, a surrogate for SARS-CoV-2 from untreated wastewater. *Science of the Total Environment* **739**.

Ahmed W, Tscharke B, Bertsch PM*, et al.* (2021) SARS-CoV-2 RNA monitoring in wastewater as a potential early warning system for COVID-19 transmission in the community: A temporal case study. *Science of the Total Environment* **761**.

Ahmed W, Simpson SL, Bertsch PM*, et al.* (2022) Minimizing errors in RT-PCR detection and quantification of SARS-CoV-2 RNA for wastewater surveillance. *Science of The Total Environment* **805**: 149877.

Aleem A, Akbar Samad AB & Slenker AK (2022) Emerging Variants of SARS-CoV-2 And Novel Therapeutics Against Coronavirus (COVID-19). *StatPearls,*p.^pp. StatPearls Publishing



Anderson BD, Ma M-J, Wang G-L*, et al.* (2018) Prospective surveillance for influenza A virus in Chinese swine farms. *Emerging Microbes & Infections* **7**: 1-10.

Asghar H, Diop OM, Weldegebriel G*, et al.* (2014) Environmental Surveillance for Polioviruses in the Global Polio Eradication Initiative. *The Journal of Infectious Diseases* **210**: S294-S303.

Bailey ES, Borkenhagen LK, Choi JY, Greer AE, Culhane MR & Gray GC (2020) A feasibility study of conducting surveillance for swine pathogens in slurry from North Carolina swine farms. *Scientific Reports* **10**: 10059.

Barkovskii AL, Babb CM, Hurley D & Shin E (2015) Origins and environmental mobility of antibiotic resistance genes, virulence factors and bacteria in a tidal creek's watershed. *J Appl Microbiol* **118**: 764-776.

Bassetti M, Magnasco L, Vena A, Portunato F & Giacobbe DR (2022) Methicillin-resistant Staphylococcus aureus lung infection in coronavirus disease 2019: how common? *Current Opinion in Infectious Diseases* **35**.

Bengtsson-Palme J, Milakovic M, Švecová H, Ganjto M, Jonsson V, Grabic R & Udikovic-Kolic N (2019) Industrial wastewater treatment plant enriches antibiotic resistance genes and alters the structure of microbial communities. *Water Research* **162**: 437-445.

Beović B, Doušak M, Ferreira-Coimbra J, Nadrah K, Rubulotta F, Belliato M, Berger-Estilita J, Ayoade F, Rello J & Erdem H (2020) Antibiotic use in patients with COVID-19: a 'snapshot' Infectious Diseases International Research Initiative (ID-IRI) survey. *Journal of Antimicrobial Chemotherapy* **75**: 3386-3390.

Bertrand I, Challant J, Jeulin H, Hartard C, Mathieu L, Lopez S, Schvoerer E, Courtois S & Gantzer C (2021) Epidemiological surveillance of SARS-CoV-2 by genome quantification in wastewater applied to a city in the northeast of France: Comparison of ultrafiltration- and protein precipitation-based methods. *Int J Hyg Environ Health* **233**: 113692.

Betancourt WQ, Schmitz BW, Innes GK*, et al.* (2021) COVID-19 containment on a college campus via wastewater-based epidemiology, targeted clinical testing and an intervention. *Sci Total Environ* **779**: 146408.

Bibby K, Bivins A, Wu Z & North D (2021) Making Waves: Plausible Lead Time for Wastewater Based Epidemiology as an Early Warning System for COVID-19. *Water Research* **202**: 117438.

Bivins A, North D, Wu Z, Shaffer M, Ahmed W & Bibby K (2021) Within- and between-Day Variability of SARS-CoV-2 RNA in Municipal Wastewater during Periods of Varying COVID-19 Prevalence and Positivity. *ACS ES&T Water* **1**: 2097-2108.

Bivins A, North D, Ahmad A*, et al.* (2020) Wastewater-Based Epidemiology: Global Collaborative to Maximize Contributions in the Fight Against COVID-19. *Environmental Science & Technology* **54**: 7754-7757.

Bogler A, Packman A, Furman A*, et al.* (2020) Rethinking wastewater risks and monitoring in light of the COVID-19 pandemic. *Nature Sustainability* **3**: 981-990.

Borkenhagen LK, Mallinson KA, Tsao RW*, et al.* (2018) Surveillance for respiratory and diarrheal pathogens at the human-pig interface in Sarawak, Malaysia. *PLOS ONE* **13**: e0201295.

Burd EM (2010) Validation of Laboratory-Developed Molecular Assays for Infectious Diseases. *Clinical Microbiology Reviews* **23**: 550-576.

Calero-Cáceres W, Melgarejo A, Colomer-Lluch M, Stoll C, Lucena F, Jofre J & Muniesa M (2014) Sludge as a potential important source of antibiotic resistance genes in both the bacterial and bacteriophage fractions. *Environ Sci Technol* **48**: 7602-7611.

Calgua B, Mengewein A, Grunert A, Bofill-Mas S, Clemente-Casares P, Hundesa A, Wyn-Jones AP, López-Pila JM & Girones R (2008) Development and application of a one-step low cost procedure to concentrate viruses from seawater samples. *Journal of Virological Methods* **153**: 79-83.

Cantón R, González-Alba JM & Galán JC (2012) CTX-M Enzymes: Origin and Diffusion. *Front Microbiol* **3**: 110.

Cassini A, Högberg LD, Plachouras D*, et al.* (2019) Attributable deaths and disability-adjusted life-years caused by infections with antibiotic-resistant bacteria in the EU and the European Economic Area in 2015: a population-level modelling analysis. *The Lancet Infectious diseases* **19**: 56-66.

Centers for Disease Control and Prevention (2018) One Health Basics. Vol. 2019 ((NCEZID) NCfEaZID, ed.) p.^pp.

Centers for Disease Control and Prevention (2019) About Antibiotic Resistance. Vol. 2020 p.^pp. Centers for Disease Control and Prevention, National Center for Emerging and Zoonotic Infections Diseases (NCEZID), Division of Healthcare Quality Promotion (DHQP).

Centers for Disease Control and Prevention (2019) Antibiotic resistance threats in the United States, 2019.

Centers for Disease Control and Prevention (2021) What is Polio. Vol. 2021 p.^pp.

Centers for Disease Control and Prevention (2022) Wastewater Surveillance: A New Frontier for Public Health. Vol. 2022 p.^pp.

Centers for Disease Control and Prevention (2022) Wastewater Surveillance Data Reporting and Analytics. Vol. 2022 p.^pp.

Centers for Disease Control and Prevention (2022) National Wastewater Surveillance System (NWSS). Vol. 2022 p.^pp.

Centers for Disease Control and Prevention (2022) COVID Data Tracker: Wastewater Surveillance. Vol. 2022 p.^pp.

Centers for Disease Control and Prevention, National Center for Emerging and Zoonotic Infectious Diseases & Promotion DoHQ (2022) COVID-19: U.S. Impact on Antimicrobial Resistance, Special Report 2022. p.^pp.

Cherry JD & Krogstad P (2004) SARS: The First Pandemic of the 21st Century. *Pediatric Research* **56**: 1-5.

Chibber P, Haq SA, Ahmed I, Andrabi NI & Singh G (2020) Advances in the possible treatment of COVID-19: A review. *European Journal of Pharmacology* **883**: 173372.

Corman VM, Landt O, Kaiser M*, et al.* (2020) Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR. *Euro Surveill* **25**.

Costa PMd, Vaz-Pires P & Bernardo F (2006) Antimicrobial resistance in Enterococcus spp. isolated in inflow, effluent and sludge from municipal sewage water treatment plants. *Water Research* **40**: 1735-1740.

D'Aoust PM, Graber TE, Mercier E*, et al.* (2021) Catching a resurgence: Increase in SARS-CoV-2 viral RNA identified in wastewater 48 h before COVID-19 clinical tests and 96 h before hospitalizations. *Sci Total Environ***770**: 145319.

Demeke T & Jenkins GR (2010) Influence of DNA extraction methods, PCR inhibitors and quantification methods on real-time PCR assay of biotechnology-derived traits. *Anal Bioanal Chem* **396**: 1977-1990.

Donà V, Bernasconi OJ, Kasraian S, Tinguely R & Endimiani A (2017) A SYBR Green-based real-time PCR method for improved detection of mcr-1-mediated colistin resistance in human stool samples. *Journal of Global Antimicrobial Resistance* **9**: 57-60.

Dowdle WR, De Gourville E, Kew OM, Pallansch MA & Wood DJ (2003) Polio eradication: the OPV paradox. *Rev Med Virol* **13**: 277-291.

Dumke R, de la Cruz Barron M, Oertel R, Helm B, Kallies R, Berendonk TU & Dalpke A (2021) Evaluation of Two Methods to Concentrate SARS-CoV-2 from Untreated Wastewater. *Pathogens* **10**.

D'Costa VM, King CE, Kalan L*, et al.* (2011) Antibiotic resistance is ancient. *Nature* **477**: 457-461.

Elsayed AA, Darwish SF, Zewail MB, Mohammed M, Saeed H & Rabea H (2021) Antibiotic misuse and compliance with infection control measures during COVID-19 pandemic in community pharmacies in Egypt. *Int J Clin Pract* e14081.

Fagnant CS, Sánchez-Gonzalez LM, Zhou NA*, et al.* (2018) Improvement of the Bag-Mediated Filtration System for Sampling Wastewater and Wastewater-Impacted Waters. *Food and Environmental Virology* **10**: 72-82.

Falman JC, Fagnant-Sperati CS, Kossik AL, Boyle DS & Meschke JS (2019) Evaluation of Secondary Concentration Methods for Poliovirus Detection in Wastewater. *Food and Environmental Virology* **11**: 20-31.

Fani M, Teimoori A & Ghafari S (2020) Comparison of the COVID-2019 (SARS-CoV-2) pathogenesis with SARS-CoV and MERS-CoV infections. *Future Virology* **15**: 317-323.

Farkas K, Hillary LS, Thorpe J, Walker DI, Lowther JA, McDonald JE, Malham SK & Jones DL (2021) Concentration and Quantification of SARS-CoV-2 RNA in Wastewater Using Polyethylene Glycol-Based Concentration and qRT-PCR. *Methods Protoc* **4**.

Forés E, Bofill-Mas S, Itarte M, Martínez-Puchol S, Hundesa A, Calvo M, Borrego CM, Corominas LL, Girones R & Rusiñol M (2021) Evaluation of two rapid ultrafiltration-based methods for SARS-CoV-2 concentration from wastewater. *Sci Total Environ* **768**: 144786.

Fronhoffs S, Totzke G, Stier S, Wernert N, Rothe M, Brüning T, Koch B, Sachinidis A, Vetter H & Ko Y (2002) A method for the rapid construction of cRNA standard curves in quantitative real-time reverse transcription polymerase chain reaction. *Molecular and Cellular Probes* **16**: 99-110.

Gerrity D, Papp K, Stoker M, Sims A & Frehner W (2021) Early-pandemic wastewater surveillance of SARS-CoV-2 in Southern Nevada: Methodology, occurrence, and incidence/prevalence considerations. *Water Research X* **10**.

Gibson KE, Schwab KJ, Spencer SK & Borchardt MA (2012) Measuring and mitigating inhibition during quantitative real time PCR analysis of viral nucleic acid extracts from large-volume environmental water samples. *Water Research* **46**: 4281-4291.

Global Polio Eradication Initiative The Virus. Vol. 2021 p.^pp.

Gonzalez R, Curtis K, Bivins A, Bibby K, Weir M, Yetka K, Thompson H, Keeling D, Mitchell J & Gonzalez D (2020) COVID-19 Surveillance in Southeastern Virginia Using Wastewater-Based Epidemiology. *Water Research* **186**: 116296.

Greenhalgh T, Jimenez JL, Prather KA, Tufekci Z, Fisman D & Schooley R (2021) Ten scientific reasons in support of airborne transmission of SARS-CoV-2. *The Lancet* **397**: 1603-1605.

Guo J, Li J, Chen H, Bond PL & Yuan Z (2017) Metagenomic analysis reveals wastewater treatment plants as hotspots of antibiotic resistance genes and mobile genetic elements. *Water Research* **123**: 468-478.

Harnisz M & Korzeniewska E (2018) The prevalence of multidrug-resistant Aeromonas spp. in the municipal wastewater system and their dissemination in the environment. *Science of The Total Environment* **626**: 377-383.

Harrington A, Vo V, Papp K*, et al.* (2022) Urban monitoring of antimicrobial resistance during a COVID-19 surge through wastewater surveillance. *Science of The Total Environment* **853**: 158577.

Harris PA, Taylor R, Thielke R, Payne J, Gonzalez N & Conde JG (2009) Research electronic data capture (REDCap)—A metadata-driven methodology and workflow process for providing translational research informatics support. *Journal of Biomedical Informatics* **42**: 377-381.

Harris PA, Taylor R, Minor BL*, et al.* (2019) The REDCap Consortium: Building an International Community of Software Platform Partners. *Journal of Biomedical Informatics* **95**: 103208.

Harrison AG, Lin T & Wang P (2020) Mechanisms of SARS-CoV-2 Transmission and Pathogenesis. *Trends in Immunology* **41**: 1100-1115.

Hazard D, Kaier K, von Cube M, Grodd M, Bugiera L, Lambert J & Wolkewitz M (2020) Joint analysis of duration of ventilation, length of intensive care, and mortality of COVID-19 patients: a multistate approach. *BMC Medical Research Methodology* **20**: 206.

He X, Lau EHY, Wu P*, et al.* (2020) Temporal dynamics in viral shedding and transmissibility of COVID-19. *Nature Medicine* **26**: 672-675.

He YH, Ruan GJ, Hao H, Xue F, Ma YK, Zhu SN & Zheng B (2020) Real-time PCR for the rapid detection of vanA, vanB and vanM genes. *J Microbiol Immunol Infect* **53**: 746-750.

Hendriksen RS, Bortolaia V, Tate H, Tyson GH, Aarestrup FM & McDermott PF (2019) Using Genomics to Track Global Antimicrobial Resistance. *Frontiers in Public Health* **7**.

Hendriksen RS & Munk P & Njage P*, et al.* (2019) Global monitoring of antimicrobial resistance based on metagenomics analyses of urban sewage. *Nature Communications* **10**: 1124.

Hill M & Pollard AJ (2022) Detection of poliovirus in London highlights the value of sewage surveillance. *The Lancet* **400**: 1491-1492.

Holmes AH, Moore LSP, Sundsfjord A, Steinbakk M, Regmi S, Karkey A, Guerin PJ & Piddock LJV (2016) Understanding the mechanisms and drivers of antimicrobial resistance. *The Lancet* **387**: 176-187.

Hora PI, Pati SG, McNamara PJ & Arnold WA (2020) Increased Use of Quaternary Ammonium Compounds during the SARS-CoV-2 Pandemic and Beyond: Consideration of Environmental Implications. *Environ Sci Technol Lett*.

Hovi T, Shulman LM, Avoort HVD, Deshpande J, Roivainen M & Gourville EMD (2012) Role of environmental poliovirus surveillance in global polio eradication and beyond. *Epidemiology and Infection* **140**: 1-13.

Huijbers PMC, Flach C-F & Larsson DGJ (2019) A conceptual framework for the environmental surveillance of antibiotics and antibiotic resistance. *Environment International* **130**: 104880.

Huijbers PMC, Larsson DGJ & Flach C-F (2020) Surveillance of antibiotic resistant Escherichia coli in human populations through urban wastewater in ten European countries. *Environmental Pollution* **261**: 114200.

Hunter P (2007) Inevitable or avoidable? *EMBO reports* **8**: 531-534.

Hutchings M, Truman A & Wilkinson B (2019) Antibiotics: past, present and future. *Current Opinion in Microbiology* **51**: 72-80.

Interagency Coordination Group on Antimicrobial Resistance (2019) No Time to Wait: Securing the Future from Drug Resistant Infections. p.^pp. 28. United Nations.

Jafferali MH, Khatami K, Atasoy M, Birgersson M, Williams C & Cetecioglu Z (2020) Benchmarking virus concentration methods for quantification of SARS-CoV-2 in raw wastewater. *Science of The Total Environment*142939.

Jin L, Pruden A, Boehm AB, Alvarez PJJ, Raskin L, Kohn T & Li X (2022) Integrating Environmental Dimensions of “One Health” to Combat Antimicrobial Resistance: Essential Research Needs. *Environmental Science & Technology*.

Jones DL, Baluja MQ, Graham DW*, et al.* (2020) Shedding of SARS-CoV-2 in feces and urine and its potential role in person-to-person transmission and the environment-based spread of COVID-19. *Science of The Total Environment* **749**: 141364.

Karthikeyan S, Ronquillo N, Belda-Ferre P, Alvarado D, Javidi T, Longhurst CA & Knight R (2021) High-Throughput Wastewater SARS-CoV-2 Detection Enables Forecasting of Community Infection Dynamics in San Diego County. *mSystems* **6**.

King County Public Health (2020) Overall counts and rates by city, health reporting area, and zip code. (County PH-SaK, ed.) p.^pp.

Klymus KE, Merkes CM, Allison MJ*, et al.* (2020) Reporting the limits of detection and quantification for environmental DNA assays. *Environmental DNA* **2**: 271-282.

Kumblathan T, Piroddi N, Hrudey SE & Li XF (2022) Wastewater Based Surveillance of SARS-CoV-2: Challenges and Perspective from a Canadian Inter-laboratory Study. *J Environ Sci (China)* **116**: 229-232.

Kwak Y-K, Colque P, Byfors S, Giske CG, Möllby R & Kühn I (2015) Surveillance of antimicrobial resistance among Escherichia coli in wastewater in Stockholm during 1 year: does it reflect the resistance trends in the society? *International Journal of Antimicrobial Agents* **45**: 25-32.

La Rosa G, Iaconelli M, Mancini P, Bonanno Ferraro G, Veneri C, Bonadonna L, Lucentini L & Suffredini E (2020) First detection of SARS-CoV-2 in untreated wastewaters in Italy. *Sci Total Environ* **736**: 139652.

Lam AP & Wunderink RG (2006) Methicillin-Resistant S. aureus Ventilator-Associated Pneumonia: Strategies to Prevent and Treat. *Semin Respir Crit Care Med* **27**: 92-103.

Lambert F, Jacomy H, Marceau G & Talbot* PJ (2008) SARS- and Other Coronaviruses, Laboratory Protocols. *Methods in molecular biology (Clifton, NJ)* **454**: 93-102.

Langford BJ, So M, Raybardhan S, Leung V, Soucy JR, Westwood D, Daneman N & MacFadden DR (2021) Antibiotic prescribing in patients with COVID-19: rapid review and meta-analysis. *Clin Microbiol Infect* **27**: 520-531.

Larsson DGJ, Flach C-F & Laxminarayan R (2022) Sewage surveillance of antibiotic resistance holds both opportunities and challenges. *Nature Reviews Microbiology*.

Lekunberri I, Balcázar JL & Borrego CM (2017) Detection and quantification of the plasmid-mediated mcr-1 gene conferring colistin resistance in wastewater. *International Journal of Antimicrobial Agents* **50**: 734-736.

Lewis GD & Metcalf TG (1988) Polyethylene glycol precipitation for recovery of pathogenic viruses, including hepatitis A virus and human rotavirus, from oyster, water, and sediment samples. *Applied and environmental microbiology* **54**: 1983-1988.

Liguori K, Keenum I, Davis BC, Calarco J, Milligan E, Harwood VJ & Pruden A (2022) Antimicrobial Resistance Monitoring of Water Environments: A Framework for Standardized Methods and Quality Control. *Environmental Science & Technology*.

Lin L, Liu Y, Tang X & He D (2021) The Disease Severity and Clinical Outcomes of the SARS-CoV-2 Variants of Concern. *Frontiers in Public Health* **9**.

Liu DX, Liang JQ & Fung TS (2021) Human Coronavirus-229E, -OC43, -NL63, and -HKU1 (Coronaviridae). *Encyclopedia of Virology* 428-440.

Loge FJ, Thompson DE & Call DR (2002) PCR Detection of Specific Pathogens in Water: A Risk-Based Analysis. *Environmental Science & Technology* **36**: 2754-2759.

Majlander J, Anttila VJ, Nurmi W, Seppälä A, Tiedje J & Muziasari W (2021) Routine wastewater-based monitoring of antibiotic resistance in two Finnish hospitals: focus on carbapenem resistance genes and genes associated with bacteria causing hospital-acquired infections. *Journal of Hospital Infection* **117**: 157-164.

Mallapaty S (2020) How sewage could reveal true scale of coronavirus outbreak. *Nature* **580**: 176.

McClary-Gutierrez JS, Aanderud ZT, Al-faliti M*, et al.* (2021) Standardizing data reporting in the research community to enhance the utility of open data for SARS-CoV-2 wastewater surveillance. *Environmental Science: Water Research & Technology*.

McInnes Ross S, uz-Zaman Md H, Alam Imam T, Ho Siu Fung S, Moran Robert A, Clemens John D, Islam Md S & van Schaik W (2021) Metagenome-Wide Analysis of Rural and Urban Surface Waters and Sediments in Bangladesh Identifies Human Waste as a Driver of Antibiotic Resistance. *mSystems* **6**: e00137-00121.

McKinney KR, Gong YY & Lewis TG (2006) Environmental transmission of SARS at Amoy Gardens. *J Environ Health* **68**: 26-30; quiz 51-22.

McNemar Q (1947) Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **12**: 153-157.

Medema G, Been F, Heijnen L & Petterson S (2020) Implementation of environmental surveillance for SARS-CoV-2 virus to support public health decisions: Opportunities and challenges. *Curr Opin Environ Sci Health***17**: 49-71.

Medema G, Heijnen L, Elsinga G, Italiaander R & Brouwer A (2020) Presence of SARS-Coronavirus-2 RNA in Sewage and Correlation with Reported COVID-19

Prevalence in the Early Stage of the Epidemic in The Netherlands. *Environmental Science & Technology Letters* **7**: 511-516.

Medema G, Heijnen L, Elsinga G, Italiaander R & Brouwer A (2020) Presence of SARS-Coronavirus-2 in sewage. *medRxiv* 2020.2003.2029.20045880.

Melnick JL, Horstmann DM & Ward R (1946) The isolation of poliomyelitis virus from human extra-neural sources; comparison of virus content of blood, oropharyngeal washings, and stools of contacts. *J Clin Invest* **25**: 275-277.

Mendez II, Hermann LL, Hazelton PR & Coombs KM (2000) A comparative analysis of Freon substitutes in the purification of reovirus and calicivirus. *Journal of Virological Methods* **90**: 59-67.

Merchel Piovesan Pereira B & Tagkopoulos I (2019) Benzalkonium Chlorides: Uses, Regulatory Status, and Microbial Resistance. *Appl Environ Microbiol* **85**.

Metcalf TG, Melnick JL & Estes MK (1995) Environmental Virology - from Detection of Virus in Sewage and Water by Isolation to Identification by Molecular-Biology - a Trip of over 50 Years. *Annual Review of Microbiology* **49**: 461-487.

Miller SA, Ferreira JP & LeJeune JT (2022) Antimicrobial Use and Resistance in Plant Agriculture: A One Health Perspective. Vol. 12 p.^pp.

Miura F, Kitajima M & Omori R (2021) Duration of SARS-CoV-2 viral shedding in faeces as a parameter for wastewater-based epidemiology: Re-analysis of patient data using a shedding dynamics model. *Sci Total Environ* **769**: 144549.

Mobasseri G, Teh CSJ, Ooi PT, Tan SC & Thong KL (2019) Molecular Characterization of Multidrug-Resistant and Extended-Spectrum Beta-Lactamase-Producing Klebsiella pneumoniae Isolated from Swine Farms in Malaysia. *Microbial Drug Resistance* **25**: 1087-1098.

Monteiro S, Rente D, Cunha MV*, et al.* (2022) A wastewater-based epidemiology tool for COVID-19 surveillance in Portugal. *Science of The Total Environment* **804**: 150264.

Munir M, Wong K & Xagoraraki I (2011) Release of antibiotic resistant bacteria and genes in the effluent and biosolids of five wastewater utilities in Michigan. *Water Research* **45**: 681-693.

Munk P & Brinch C & Møller FD*, et al.* (2022) Genomic analysis of sewage from 101 countries reveals global landscape of antimicrobial resistance. *Nature Communications* **13**: 7251.

Murray PR, Rosenthal KS & Pfaller MA (2013) Chapter 54 Picornaviruses. *Medical Microbiology,*p.^pp. Elsevier Saunders, Philadelphia, PA.

Na G, Zhang K, Gao H, Li R, Jin S, Zhao F, Zhang H & Li S (2021) Occurrence and distribution characteristics of antibiotic resistance genes in sediments between urban and rural of the Liaohe River Basin, China. *Environmental Science and Pollution Research* **28**: 54002-54014.

Naughton CC, Roman FA, Alvarado AGF*, et al.* (2021) Show us the Data: Global COVID-19 Wastewater Monitoring Efforts, Equity, and Gaps. *medRxiv* 2021.2003.2014.21253564.

NIST/SEMATECH (2012) e-Handbook of Statistical Methods 1.3.5.17.2. Teitjen-Moore Test for Outliers. Vol. 2022 p.^pp.

NOAA (2022) Monthly Total Precipitation for SEATTLE BOEING FIELD, WA. (National Weather Service, ed.) p.^pp. Web.

NPR (2020) The Morning Edition. p.^pp.

Osbiston K, Oxbrough A & Fernández-Martínez LT (2021) Antibiotic resistance levels in soils from urban and rural land uses in Great Britain. *Access Microbiology* **3**.

Parvez MK & Parveen S (2017) Evolution and Emergence of Pathogenic Viruses: Past, Present, and Future. *Intervirology* **60**: 1-7.

Peccia J, Zulli A, Brackney DE*, et al.* (2020) SARS-CoV-2 RNA concentrations in primary municipal sewage sludge as a leading indicator of COVID-19 outbreak dynamics. *medRxiv* 2020.2005.2019.20105999.

Peccia J, Zulli A, Brackney DE*, et al.* (2020) Measurement of SARS-CoV-2 RNA in wastewater tracks community infection dynamics. *Nature Biotechnology* **38**: 1164-1167.

Pecson BM, Darby E, Haas CN*, et al.* (2021) Reproducibility and sensitivity of 36 methods to quantify the SARS-CoV-2 genetic signal in raw wastewater: findings from an interlaboratory methods evaluation in the US. *Environmental Science-Water Research & Technology* **7**: 504-520.

Peiser J (2020) The University of Arizona says it caught a dorm's covid-19 outbreak before it started. Its secret weapon: poop. *The Washington post*.

Perez-Cataluna A, Cuevas-Ferrando E, Randazzo W, Falco I, Allende A & Sanchez G (2021) Comparing analytical methods to detect SARS-CoV-2 in wastewater. *Science of the Total Environment* **758**.

Philo SE, Ong AQW, Keim EK, Swanstrom R, Kossik AL, Zhou NA, Beck NK & Meschke JS (2022) Development and Validation of the Skimmed Milk Pellet Extraction Protocol for SARS-CoV-2 Wastewater Surveillance. *Food Environ Virol*.

Philo SE, Keim EK, Swanstrom R*, et al.* (2021) A comparison of SARS-CoV-2 wastewater concentration methods for environmental surveillance. *Science of The Total Environment* **760**: 144215.

Poirel L, Bonnin RA & Nordmann P (2012) Genetic support and diversity of acquired extended-spectrum β-lactamases in Gram-negative rods. *Infection, Genetics and Evolution* **12**: 883-893.

Pruden A, Pei R, Storteboom H & Carlson KH (2006) Antibiotic Resistance Genes as Emerging Contaminants: Studies in Northern Colorado. *Environmental Science & Technology* **40**: 7445-7450.

Pruden A, Vikesland PJ, Davis BC & de Roda Husman AM (2021) Seizing the moment: now is the time for integrated global surveillance of antimicrobial resistance in wastewater environments. *Current Opinion in Microbiology* **64**: 91-99.

Pärnänen KMM, Narciso-da-Rocha C, Kneis D*, et al.* (2019) Antibiotic resistance in European wastewater treatment plants mirrors the pattern of clinical antibiotic resistance prevalence. *Science Advances* **5**: eaau9124.

Quach-Cu J, Herrera-Lynch B, Marciniak C, Adams S, Simmerman A & Reinke RA (2018) The Effect of Primary, Secondary, and Tertiary Wastewater Treatment Processes on Antibiotic Resistance Gene (ARG) Concentrations in Solid and Dissolved Wastewater Fractions. Vol. 10 p.^pp.

Quan PL, Sauzade M & Brouzes E (2018) dPCR: A Technology Review. *Sensors (Basel)* **18**.

Rasile B & Maas K (2021) SARS-CoV-2 Wastewater RNA Concentration and Extraction (Nanotrap and NucleoMag RNA Water). Vol. 2021 p.^pp. protocols.io.

Rački N, Dreo T, Gutierrez-Aguirre I, Blejec A & Ravnikar M (2014) Reverse transcriptase droplet digital PCR shows high resilience to PCR inhibitors from plant, soil and water samples. *Plant Methods* **10**: 42.

Rimoldi SG, Stefani F, Gigantiello A*, et al.* (2020) Presence and infectivity of SARS-CoV-2 virus in wastewaters and rivers. *Science of the Total Environment* **744**.

Rios G, Lacoux C, Leclercq V*, et al.* (2021) Monitoring SARS-CoV-2 variants alterations in Nice neighborhoods by wastewater nanopore sequencing. *The Lancet Regional Health – Europe* **10**.

Rizzo L, Manaia C, Merlin C, Schwartz T, Dagot C, Ploy MC, Michael I & Fatta-Kassinos D (2013) Urban wastewater treatment plants as hotspots for antibiotic resistant bacteria and genes spread into the environment: A review. *Science of The Total Environment* **447**: 345-360.

Robinson TP, Wertheim HFL, Kakkar M, Kariuki S, Bu D & Price LB (2016) Animal production and antimicrobial resistance in the clinic. *The Lancet* **387**: e1-e3.

Robinson TP, Bu DP, Carrique-Mas J*, et al.* (2016) Antibiotic resistance is the quintessential One Health issue. *Transactions of the Royal Society of Tropical Medicine and Hygiene* **110**: 377-380.

Roschanski N, Fischer J, Guerra B & Roesler U (2014) Development of a Multiplex Real-Time PCR for the Rapid Detection of the Predominant Beta-Lactamase Genes CTX-M, SHV, TEM and CIT-Type AmpCs in Enterobacteriaceae. *PLOS ONE* **9**: e100956.

Rousham EK, Unicomb L & Islam MA (2018) Human, animal and environmental contributors to antibiotic resistance in low-resource settings: integrating behavioural, epidemiological and One Health approaches. *Proceedings of the Royal Society B* **285**: 20180332.

RTP (2020) COVID-19. Minstra admite que Portugal entra em “fase de crescimento exponencial”. p.^pp. Rádio e Televisão de Portugal.

Russell CD & Fairfield CJ & Drake TM*, et al.* (2021) Co-infections, secondary infections, and antimicrobial use in patients hospitalised with COVID-19 during the first pandemic wave from the ISARIC WHO CCP-UK study: a multicentre, prospective cohort study. *The Lancet Microbe* **2**: e354-e365.

Ryerson A, Lang D, Alazawi M & al. e (2022) Wastewater Testing and Detection of Poliovirus Type 2 Genetically Linked to Virus Isolated from a Paralytic Polio Case - New York, March 9 - October 11, 2022. *Morb Mortal Wkly Rep* **71**: 1418-1424.

Sabet NS, Subramaniam G, Navaratnam P & Sekaran SD (2007) Detection of methicillin- and aminoglycoside-resistant genes and simultaneous identification of S. aureus using triplex real-time PCR Taqman assay. *Journal of Microbiological Methods* **68**: 157-162.

Saga T & Tamaguchi K (2009) History of Antimicrobial Agents and Resistant Bacteria. *Japan Medical Association Journal* **2**: 103-108.

Schrader C, Schielke A, Ellerbroek L & Johne R (2012) PCR inhibitors - occurrence, properties and removal. *Journal of Applied Microbiology* **113**: 1014-1026.

Sherchan SP, Shahin S, Ward LM, Tandukar S, Aw TG, Schmitz B, Ahmed W & Kitajima M (2020) First detection of SARS-CoV-2 RNA in wastewater in North America: A study in Louisiana, USA. *Science of the Total Environment* **743**.

Sheridan C (2020) Coronavirus and the race to distribute reliable diagnostics. *Nat Biotechnol* **38**: 382-384.

Shieh YSC, Wait D, Tai L & Sobsey MD (1995) Methods to remove inhibitors in sewage and other fecal wastes for enterovirus detection by the polymerase chain reaction. *Journal of Virological Methods* **54**: 51-66.

Singh DD, Sharma A, Lee H-J & Yadav DK (2022) SARS-CoV-2: Recent Variants and Clinical Efficacy of Antibody-Based Therapy. *Frontiers in Cellular and Infection Microbiology* **12**.

Smith MQRP & Ruxton GD (2020) Effective use of the McNemar test. *Behavioral Ecology and Sociobiology* **74**: 133.

Solis-Moreira J (2022) Study: New Wastewater Surveillance Method Detected SARS-CoV-2 Variants of Concern Up to 2 Weeks Before Clinical Tests. *JAMA* **328**: 914-915.

Svec D, Tichopad A, Novosadova V, Pfaffl MW & Kubista M (2015) How good is a PCR efficiency estimate: Recommendations for precise and robust qPCR efficiency assessments. *Biomolecular detection and quantification* **3**: 9-16.

Tao C-W, Hsu B-M, Ji W-T, Hsu T-K, Kao P-M, Hsu C-P, Shen S-M, Shen T-Y, Wan T-J & Huang Y-L (2014) Evaluation of five antibiotic resistance genes in wastewater treatment systems of swine farms by real-time PCR. *Science of The Total Environment* **496**: 116-121.

Taylor LH, Latham SM & woolhouse MEJ (2001) Risk factors for human disease emergence. *Philosophical Transactions of the Royal Society of London Series B: Biological Sciences* **356**: 983-989.

Thakur S & Gray GC (2019) The Mandate for a Global "One Health" Approach to Antimicrobial Resistance Surveillance. *Am J Trop Med Hyg* **100**: 227-228.

The World Health Organization (2003) Guidelines for environmental surveillance of poliovirus circulation. p.^pp. 1-28. Vaccine Assessment and Monitoring of the Department of Vaccines and Biologicals, Geneva, Switzerland.

Tiri B, Sensi E, Marsiliani V*, et al.* (2020) Antimicrobial Stewardship Program, COVID-19, and Infection Control: Spread of Carbapenem-Resistant Klebsiella Pneumoniae Colonization in ICU COVID-19 Patients. What Did Not Work? Vol. 9 p.^pp.

Tomasino MP, Semedo M, Vieira e Moreira P, Ferraz E, Rocha A, Carvalho MF, Magalhães C & Mucha AP (2021) SARS-CoV-2 RNA detected in urban wastewater from Porto, Portugal: Method optimization and continuous 25-week monitoring. *Science of The Total Environment* **792**: 148467.

Tornimbene B, Eremin S, Escher M, Griskeviciene J, Manglani S & Pessoa-Silva CL (2018) WHO Global Antimicrobial Resistance Surveillance System early implementation 2016-17. *The Lancet Infectious Diseases* **18**: 241-242.

Tyra H (2017) Tips for working with IDT Gene Fragments. Vol. 2022 p.^pp. Integrated DNA Technologies.

US Centers for Disease Control and Prevention (2020) CDC 2019-Novel Coronavirus (2019-nCoV) Real-Time RT-PCR Diagnostic Panel. Vol. 2020 p.^pp. Online.

Vennapusa AR, Somayanda IM, Doherty CJ & Jagadish SVK (2020) A universal method for high-quality RNA extraction from plant tissues rich in starch, proteins and fiber. *Scientific Reports* **10**: 16887.

Vijgen L, Keyaerts E, Moës E, Maes P, Duson G & Ranst MV (2005) Development of One-Step, Real-Time, Quantitative Reverse Transcriptase PCR Assays for Absolute Quantitation of Human Coronaviruses OC43 and 229E. *Journal of Clinical Microbiology* **43**: 5452-5456.

Vogels CBF, Brito AF, Wyllie AL*, et al.* (2020) Analytical sensitivity and efficiency comparisons of SARS-CoV-2 RT–qPCR primer–probe sets. *Nature Microbiology* **5**: 1299-1305.

Wang J, Ben W, Yang M, Zhang Y & Qiang Z (2016) Dissemination of veterinary antibiotics and corresponding resistance genes from a concentrated swine feedlot along the waste treatment paths. *Environment International* **92**: 317-323.

Wang W, Xu Y, Gao R, Lu R, Han K, Wu G & Tan W (2020) Detection of SARS-CoV-2 in Different Types of Clinical Specimens. *JAMA* **323**: 1843-1844.

Wang X, Bailey ES, Qi X, Yu H, Bao C & Gray GC (2020) Bioaerosol Sampling at a Live Animal Market in Kunshan, China: A Noninvasive Approach for Detecting Emergent Viruses. *Open Forum Infectious Diseases* **7**: ofaa134.

Water Research Foundation (2020) Wastewater Surveillance of the COVID-19 Genetic Signal in Sewersheds. p.^pp. The Water Research Foundation.

WHO (2020) COVID-19 Dashboard. p.^pp. Geneva: World Health Organization.

Williams MR, Stedtfeld RD, Guo X & Hashsham SA (2016) Antimicrobial Resistance in the Environment. *Water Environment Research* **88**: 1951-1967.

Wolter N, Jassat W, Walaza S*, et al.* (2022) Early assessment of the clinical severity of the SARS-CoV-2 omicron variant in South Africa: a data linkage study. *The Lancet* **399**: 437-446.

Woolhouse M, Ward M, Bunnik Bv & Farrar J (2015) Antimicrobial resistance in humans, livestock and the wider environment. *Philosophical Transactions of the Royal Society B: Biological Sciences* **370**: 20140083.

Woolhouse MEJ & Gowtage-Sequeria S (2005) Host range and emerging and reemerging pathogens. *Emerg Infect Diseases* **11**: 1842-1847.

World Health Organization (2020) Pneumonia of Unknown Cause - China. Vol. 2020 (News DO, ed.) p.^pp.

World Health Organization (2020, March 11) WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020. p.^pp.

World Health Organization (2021) WHO integrated global surveillance on ESBL-producing E. coli using a “One Health” approach: implementation and opportunities. p.^pp. 76. World Health Organization, Geneva.

Wu F, Zhang J, Xiao A*, et al.* (2020) SARS-CoV-2 Titers in Wastewater Are Higher than Expected from Clinically Confirmed Cases. *mSystems* **5**.

Wu Y, Guo C, Tang L*, et al.* (2020) Prolonged presence of SARS-CoV-2 viral RNA in faecal samples. *The Lancet Gastroenterology & Hepatology* **5**: 434-435.

Xiao F, Tang M, Zheng X, Liu Y, Li X & Shan H (2020) Evidence for Gastrointestinal Infection of SARS-CoV-2. *Gastroenterology* **158**: 1831-1833.e1833.

Xu S, Chen M, Feng T, Zhan L, Zhou L & Yu G (2021) Use ggbreak to Effectively Utilize Plotting Space to Deal With Large Datasets and Outliers. *Front Genet* **12**: 774846.

Ye YY, Ellenberg RM, Graham KE & Wigginton KR (2016) Survivability, Partitioning, and Recovery of Enveloped Viruses in Untreated Municipal Wastewater. *Environmental Science & Technology* **50**: 5077-5085.

Yu A, Hughes B, Wolfe M*, et al.* (2021) Estimating relative abundance of two SARS-CoV-2 variants through wastewater surveillance at two large metropolitan sites.

Zarei-Baygi A & Smith AL (2021) Intracellular versus extracellular antibiotic resistance genes in the environment: Prevalence, horizontal transfer, and mitigation strategies. *Bioresour Technol* **319**: 124181.

Zhang A, Call DR, Besser TE, Liu J, Jones L, Wang H & Davis MA (2019) β-lactam resistance genes in bacteriophage and bacterial DNA from wastewater, river water, and irrigation water in Washington State. *Water Research* **161**: 335-340.

Zhang DY, Ling HB, Huang X*, et al.* (2020) Potential spreading risks and disinfection challenges of medical wastewater by the presence of Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2) viral RNA in septic tanks of Fangcang Hospital. *Science of the Total Environment* **741**.

Zhang J-j, Dong X, Cao Y-y, Yuan Y-d, Yang Y-b, Yan Y-q, Akdis CA & Gao Y-d (2020) Clinical characteristics of 140 patients infected with SARS-CoV-2 in Wuhan, China. *Allergy* **75**: 1730-1741.

Zhou NA, Ong AQW, Fagnant-Sperati CS*, et al.* (2022) Evaluation of sampling and concentration methods for Salmonella enterica serovar Typhi detection from wastewater. *medRxiv*2022.2006.2007.22275929.

Zhou NA, Fagnant-Sperati CS, Komen E*, et al.* (2019) Feasibility of the Bag-Mediated Filtration System for Environmental Surveillance of Poliovirus in Kenya. *Food and Environmental Virology* 1-13.

Zhu L, Li R, Yan Y & Cui L (2022) Urbanization drives the succession of antibiotic resistome and microbiome in a river watershed. *Chemosphere* **301**: 134707.

Zieliński W, Korzeniewska E, Harnisz M, Hubeny J, Buta M & Rolbiecki D (2020) The prevalence of drug-resistant and virulent Staphylococcus spp. in a municipal wastewater treatment plant and their spread in the environment. *Environment International* **143**: 105914.