Leveraging network information to improve population size estimation in social and environmental applications

Jessica Kunke

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2025

Reading Committee:
Tyler H. McCormick, Chair
Jon Wakefield
Carlos Cinelli

Program Authorized to Offer Degree:
Statistics

University of Washington

Abstract

Leveraging network information to improve population size estimation in social and environmental applications

Jessica Kunke

Chair of the Supervisory Committee:
Tyler H. McCormick
Department of Statistics and Department of Sociology

I.

There once was a question of size

Whose estimate none could surmise.

Whether people or fish,

Better answers we'd wish,

And through networks we see with fresh eyes!

II.

I didn't know how,

But no one else did either.

So I worked on it.

I hope you find this useful.

I sure learned a lot.

III.

Social and ecological processes contain network structures such as interpersonal relationships and the flow of water through a river system. This dissertation develops methods for using

such network information to improve population size estimates in both social (Chapter 2) and ecological (Chapters 3-4) contexts.

The first project considers the problem of estimating the size of a human subpopulation that is hard to reach through traditional survey methods, and contributes a framework for studying network scale-up method (NSUM) estimator performance when certain modeling assumptions are violated. A cost-effective approach to estimating the size or prevalence of a subpopulation that is hard to reach through a traditional survey, the NSUM makes several strong assumptions, including the random mixing assumption that any two people are equally likely to know each other. The basic NSUM involves two steps: estimating respondents' degrees or the number of people they know, then using these estimated degrees along with the number of people they report knowing in the hard-to-reach subpopulation of interest to estimate the prevalence of that subpopulation. Each of these two steps involves taking either an average of ratios or a ratio of averages, and using the ratio of averages for each step has been the most common approach. However, we developed theoretical arguments that using the average of ratios at the second, prevalence-estimation step often has lower mean squared error when the random mixing assumption is violated, which seems likely in practice; this estimator was proposed early in NSUM's development but has largely been unexplored and unused. Simulation results using an example network data set also supported these findings. On the basis of this theoretical and empirical evidence, we suggest that future surveys using a simple estimator may want to use this mixed estimator, and estimation methods based on this estimator may produce new improvements. This joint work with Ian Laga, Xiaoyue Niu, and Tyler H. McCormick is published in *Sociological Methodology* (Kunke et al., 2024).

The second project develops a class of scalable spatial stream network (S3N) models to do estimation, inference and prediction with spatial processes on stream networks on a spatial scale that was previously not feasible. Spatial process models are a standard approach to making regional estimates based on point observations, but classically they account only

for covariance based on birds' eye distance, and they are not scalable to large regions due to their computational complexity. Existing spatial stream network (SSN) models adapt such spatial processes to river networks by incorporating valid stream covariance functions, but preprocessing and estimation with these models is expensive and precludes the analysis of regions at the multi-state and national level in the United States. Our contribution is a scalable spatial stream network (S3N) model based on the SSN that uses nearest neighbor approximations and more efficient preprocessing to enable national and regional spatial process modeling on stream networks. We demonstrate the accuracy and computational efficiency of the S3N models relative to SSNs on simulated data on the Ohio River Basin stream network. This is joint work with Julian Olden and Tyler H. McCormick.

The third project applies the S3N models developed in the second project to obtain what is to our knowledge the first set of fish population size estimates for over 300 species across the entire Ohio River Basin. Estimation on this scale was previously not possible, and the approach we demonstrate can be used to estimate freshwater fish populations by species over large regions. These estimates represent a critical step for biodiversity monitoring and conservation planning, as the geographic distribution of freshwater fish species at a national scale is currently unknown. Our publicly available code makes national and regional fish population size estimation accessible to the wider research community. This is joint work with Julian Olden and Tyler H. McCormick.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

ADJACENCY MATRIX: A matrix A identifying which nodes in a network are connected with which other nodes. $A_{ij} = 1$ if i, j are connected, and $A_{ij} = 0$ otherwise.

COMID: A unique integer-valued stream reach identifier. Negative values indicate reaches added to correct topological concerns with the river network.

CONNECTED COMPONENT, OR COMPONENT: A subset G of nodes in a network that are not connected to any other nodes in the network and which do not contain any nodes that are not connected to at least one node in G.

DEGREE: The number of people in someone's social network, or the number of people someone knows, given some definition of "knowing".

HARD-TO-REACH OR HIDDEN POPULATION: A population that is not well accessed by a traditional survey because it is hard to obtain an accurate list of its members, its members distrust the institutions who attempt to survey them, its members are stigmatized and unwilling to disclose their membership, and/or the population is rare.

NETWORK: A graph consisting of nodes and edges connecting those nodes. Nodes may represent people or river junctions, for instance, and edges may represent social connections or stream segments.

NETWORK SCALE-UP METHOD: An approach for estimating the size of a human subpopulation H by conducting a traditional survey of the general population and asking respondents how many people they know in H and how many people they know in the general population. In practice, the latter question is replaced with a series of more easily answerable questions that can be used to estimate how many people each respondent knows in the general population.

PREVALENCE OF A SUBGROUP: The proportion of a population contained in the subgroup in question.

REACH: A stream segment, composed of multiple connected straight line subsegments.

STREAM OUTLET: The most downstream location within a stream network; the entire network drains to this point.

UPSTREAM DISTANCE OF A POINT: The distance from the stream outlet of the river network to the point in question.

UPSTREAM DISTANCE OF A STREAM REACH: The distance from the stream outlet of the river network to the downstream node of the reach in question.

ACKNOWLEDGMENTS

I would like to thank my dissertation committee—Tyler H. McCormick, Jon Wakefield, Carlos Cinelli, and John Choe—for their support and feedback. I am especially grateful to have had Tyler as my research advisor since my first year in the program. His curiosity and playfulness are infectious, and his ability to cut to the heart and intuition of a problem has made me a better researcher and made our research more impactful. I received excellent advice years ago from my NOAA GLERL mentor Craig Stowe about the importance in scientific research of being okay with feeling stupid, and Tyler is the mentor who has most helped me feel comfortable getting my feet wet and learning from being wrong, without which I could not have completed this thesis.

I am grateful to have had marvelous research collaborators throughout my graduate study: thank you to Maggie Xiaoyue Niu, Ian Laga, Julian Olden, Adam Visokay, Michael Baiocchi, and Kimberly Babiarz. Thank you to Shane Lubold, Aparajithan Venkateswaran, and Zachary G. Nicolaou for thoughtful discussions and input on this thesis. Thank you to the Tribal Exchange Network and the Pacific Northwest Tribal Coding Group for our rewarding work together and for everything you have taught me; special thanks to Angie Reed, Kimberly Bray, Michelle Totman, and Robert Knapp. I have also had excellent mentors on my path before my time at the University of Washington: thank you to Mei Wang, Peter McCullagh, Elisabeth Moyer, Mihai Anitescu, Carlo Graziani, Charlotte Haley, Julie Bessac, and Vishwas Rao for your support, feedback, and guidance during my master's as I was developing my path as a statistician. Thank you to Zhiming Kuang for supporting and believing in me when I needed it most.

Thank you to the supportive, fun UW community I have enjoyed being part of. Abel

Rodriguez has helped our department flourish as its chair. The department staff Ellen Reynolds, Kristine Y. Chan, Tracy Pham, Veronica Bae, and Vickie J. Graybeal have helped me navigate the program, and it has been a pleasure to get to know them and organize department events together. Ema Perkovic, Adrian Raftery, and Armeen Taeb have been excellent mentors and academic advisors to me. The UW librarians have been essential in navigating publication and software questions throughout my work. Our department has the strongest student community I've ever been part of in a department. Thank you to fellow UW crafters and musicians who have spent hours sharing the joy of non-research creative time with me. Thank you to Antonio Olivas-Martinez, Hangjun Cho, and Stephen Portillo for being constant friends and running buddies over the years. And thank you to my fellow UW students and postdocs who have become my friends.

Finally, thank you to my childhood friends, my family, my parents, and my partner, without whose support I could not have completed this degree or anything else.

DEDICATION

to Zack, Mom, and Dad with love and gratitude

Chapter 1
INTRODUCTION

Network structures appear in many social and environmental problems, and they encode valuable information. For example, personal interactions and relationships can be modeled via social networks; people live, commute to work, and access services along road networks; and freshwater fish populations live within river networks. This dissertation develops methods for using such network information to improve population size estimates in both social (Chapter 2) and ecological (Chapters 3-4) contexts.

1.1 Background

The first project considers the problem of estimating the size of a human subpopulation that is hard to reach through traditional survey methods, and contributes a framework for studying network scale-up method (NSUM) estimator performance when certain modeling assumptions are violated. A cost-effective approach to estimating the size or prevalence of a subpopulation that is hard to reach through a traditional survey, the NSUM makes several strong assumptions, including the random mixing assumption that any two people are equally likely to know each other. The basic NSUM involves two steps: estimating respondents' degrees or the number of people they know, then using these estimated degrees along with the number of people they report knowing in the hard-to-reach subpopulation of interest to estimate the prevalence of that subpopulation. Each of these two steps involves taking either an average of ratios or a ratio of averages, and using the ratio of averages for each step has been the most common approach.

The second project develops a class of scalable spatial stream network (S3N) models to do estimation, inference and prediction with spatial processes on stream networks. Spatial

process models are a classic approach to making regional estimates based on point observations, but classically they account only for covariance based on birds' eye distance, and they are not scalable to large regions due to their computational complexity. Spatial stream network (SSN) models adapt spatial processes to river networks by incorporating valid stream covariance functions, but they do not address the limited scalability of classic spatial process models. On the other hand, many methods have been developed to increase the computational efficiency of spatial processes by making various simplifying assumptions, such as using only the nearest spatial neighbors to model spatial covariance.

The third project applies the S3N models developed in the second project to propose and use a framework for estimating freshwater fish populations by species over large regions. Freshwater fish are critical for sustenance and the economy; fisheries employ millions of people, fishing is an essential and irreplaceable nutrient source for many communities, and fishing is a major industry (Lynch et al., 2022). Fish populations additionally provide valuable recreational, cultural, and ecosystem services. However, freshwater vertebrate populations have been declining twice as fast as land or ocean vertebrates (Tickner et al., 2020; Hughes, 2021), and Reid et al. (2019) document twelve threats to freshwater biodiversity that are either new or intensified since 2006. A fundamental knowledge gap in monitoring and conserving these populations is estimating their geographic distribution across the United States, particularly on the spatial scales relevant for reproduction, life cycle and environmental stressors and drivers Isaak et al. (2017).

1.2 Key Contributions

Social networks:

- We propose and illustrate a framework for evaluating the robustness of NSUM estimators when one or more assumptions are violated.

- We provide theoretical and empirical evidence that the currently most common NSUM estimator may be less robust than an alternative that is equally easy to compute.

- We suggest that future surveys using one of the basic NSUM estimators may want to use this alternative estimator.

Stream networks:

- We develop the S3N model, which we believe is the first spatial stream network model that can be applied at the scale of a HUC2 region (approximately 200,000 sq km, 170,000 stream reaches, and 9,000 observations), and for hundreds of response variables such as different species of fish.

- We provide a synthesis of existing literature on spatial stream networks, including a summary of fundamental preprocessing steps.

- We demonstrate through benchmarking that the S3N enables science at a larger scale than previously possible.

- We confirm that S3N models provide accurate model parameter estimates, perhaps even with less bias than the existing SSN models for some parameters.

- We provide open-source code for these S3N models on GitHub to make these models accessible to the wider community.

- We make the first set of fish population size estimates for over 300 fish species across the entire Ohio River Basin.

1.3 Outline of the Dissertation

Chapter 2 investigates the robustness of simple NSUM estimators for human subpopulations that are ill-served by traditional survey methods. Chapter 3 introduces scalable spatial stream network (S3N) models, and Chapter 4 applies these models to estimate fish population sizes over a larger region than previously possible. Chapter 5 concludes with comments on implications and future work.

Chapter 2

COMPARING THE ROBUSTNESS OF SIMPLE NETWORK SCALE-UP METHOD (NSUM) ESTIMATORS

2.1 Introduction

Surveys are a standard approach to estimating the size of subpopulations, or groups of people with a particular trait. Many key subpopulations of interest, however, are hard to reach[1] with standard surveys for a number of possible reasons: (1) it may be hard to get a list of the members of this subpopulation or they may be hard to contact, as in the case of the homeless subpopulation; (2) it may be hard to accurately determine their membership in the subpopulation because their group status is stigmatized, as in the case of heavy drug users; and/or (3) the subpopulation may be rare in the general population (Bernard et al., 1991; Killworth et al., 1998). The limitations of standard survey methodology in reaching these groups has motivated the development of better adapted methods.

There are generally two strategies to this estimation problem that use social networks. One is to interview an initial sample, such as a convenience sample from the hard-to-reach subpopulation of interest, then incentivize those respondents to recruit additional respondents from that subpopulation. These network sampling methods include respondent driven sampling (RDS) and snowball sampling and are often broadly called link-tracing or chain-referral methods (Salganik and Heckathorn, 2004; Handcock et al., 2014; Crawford et al., 2018). Since these methods involve surveying members of the population of interest, they have the advantage that researchers may ask additional questions to study other aspects

[1]We note that researchers and community partners have raised concerns about the term "hard-to-reach"; we use this term here to encompass communities that match one or more of the above conditions, and we echo the message of Routen et al. (2022) that the onus of reaching such populations is on researchers and funders, not on the populations themselves.

of the population in addition to estimating prevalence. For example, one could not only estimate the number of people who have been trafficked in a given region but also study how they entered trafficking, what enabled them to leave if they left, and what factors increased or decreased their vulnerability to trafficking. However, this approach is not always feasible, and when it is, it can be expensive, particularly if the survey aims to estimate sizes for multiple subpopulations.

The other strategy, the network scale-up method (NSUM), is to conduct a traditional survey with a representative sample from the general or frame population and ask respondents how many people they know in the hard-to-reach populations of interest, then use information about their personal network sizes (degrees) to scale up their data into an estimate for the general prevalence of those populations (Bernard et al., 1991, 2010). Researchers could ask respondents how many people they know (their degree) and how many people they know in the hard-to-reach group. However, people often cannot provide an accurate estimate for their degree (Killworth et al., 2003; McCormick et al., 2010), leading to inaccurate hard-to-reach population prevalence estimates. Therefore, respondents' degrees are often estimated by asking other questions that are used to estimate the degree. In the probe group method, respondents are asked how many people they know in various groups of known size that collectively are thought to be representative of the general population (Killworth et al., 1998; McCarty et al., 2001; McCormick et al., 2010). The responses to questions of the form "How many people do you know with X trait?" are known as aggregated relational data or ARD (McCormick and Zheng, 2015). This approach does not require knowing whether the respondents themselves are in the hard-to-reach populations.

In this paper we focus on the NSUM strategy. A recent literature review by Ocagli et al. (2021) suggests the majority of studies use one of the simplest estimators. However, we find that when a key model assumption is violated by the presence of barrier effects, which may often occur in practice, the mean squared error (MSE) may actually tend to be much smaller for another simple estimator that is equally easy to implement but much less commonly used. We demonstrate this through theoretical derivations as well as simulated surveys on a real

network data set.

The paper is structured as follows: Section 2.2 introduces the key estimators of interest in this study. Section 2.3 presents our framework for studying the behavior of these estimators in the presence of barrier effects. Section 2.4 details the analytical results comparing estimator bias, variance, and RMSE in this setting. Section 2.5 compares the performance of the different estimators on surveys simulated from real network data, the Facebook 100 data set. We have shared code for the analytical and simulation results on GitHub; for review, we have omitted the URL here to deidentify the manuscript for review, but we have submitted the code as supplementary files. Section 2.6 concludes with a discussion.

2.2 NSUM estimators

In this section, we introduce the estimators of interest in this study as compositions of degree and prevalence estimators. We discuss the assumptions underlying the models that led to these estimators, which estimators are currently used in practice, and why we reevaluate which estimator if any should be the standard.

The target of estimation is the prevalence r of the hard-to-reach group H in the general population, which equals the ratio of the group size N_H to the size N of the general or frame population. For each respondent in a sample of size n, let Y_{iH} represent respondent i's response to the ARD question, "How many people do you know in H?" Let D_i represent the degree of respondent i. In principle, we could determine D_i by directly asking respondents how many people they know in the general population, given some definition of what it means to know another person. Since previous studies have found these responses to be inaccurate (Killworth et al., 2003; McCormick et al., 2010), researchers often estimate the degrees instead by asking other questions. Therefore, we can decompose the estimation problem into two steps: first estimating respondents' degrees, then using these degree estimates $\hat{D}_i$ with the responses Y_{iH} to estimate the prevalence r.

The NSUM approach is based on the idea that given the response Y_{iH} and degree D_i for one person, a rough estimate of the hard-to-reach group prevalence is Y_{iH}/D_i (Bernard

et al., 1991). To obtain a better estimate, we can pool the responses and degrees from a larger sample of people and model the responses Y_{iH} as binomially distributed; henceforth we refer to this early NSUM model as the binomial model (Killworth et al., 1998). This information can be pooled in two basic ways, either the ratio of average response to average degree or the average ratio of response to degree (hereafter, we refer to the ratio of averages of a quantity as R and the average of ratios as A):

$$\hat{r}_{\mathrm{R}} = \frac{\sum_i Y_{iH}/n}{\sum_i D_i/n} = \frac{\sum_i Y_{iH}}{\sum_i D_i}, \quad \hat{r}_{\mathrm{A}} = \frac{1}{n}\sum_{i=1}^{n} \frac{Y_{iH}}{D_i}. \tag{2.1}$$

One approach for estimating degrees is to ask respondents how many people they know in k subpopulations of known size, often called probe groups or alters. Here, known size means that the number or prevalence of each probe group in the general population can be obtained from census information or other data sources. For example, respondents might be asked how many people named Jamal they know and how many firefighters they know; this would provide responses Y_{ij} for each person i about probe group $j \in \{1, 2\}$, where the two probe groups are the subsets of the general population (1) named Jamal and (2) serving as firefighters, respectively. Note that probe groups can and often do overlap; people named Jamal who serve as firefighters would be counted in both of these group sizes. Analogously to the prevalence estimates based on respondents' degrees, the degrees can be estimated based on these k probe groups by either the R or the A degree estimator:

$$\hat{D}_{i,\mathrm{R}} = N \cdot \frac{\sum_j Y_{ij}}{\sum_j N_j}, \quad \hat{D}_{i,\mathrm{A}} = N \cdot \frac{1}{k}\sum_{j=1}^{k} \frac{Y_{ij}}{N_j},$$

where N_j denotes the number of people in probe group j.

These separate degree and prevalence estimators can be combined in four ways to obtain a two-step prevalence estimator that incorporates the degree estimation step (see Table 2.1). To be explicit and concise, we will refer to these estimators by the choice of degree estimator followed by the choice of prevalence estimator; for example, we will use the name RA to refer to the estimator which plugs the ratio of averages for the degree estimates into the average of ratios for the prevalence estimates.

Table 2.1: Each of the four hard-to-reach group prevalence estimators as functions of the survey data can be viewed as a two-step estimator, a composition of a degree estimator and a prevalence estimator that is a function of the estimated degrees. Each component estimator may be the ratio of averages (R) or the average of ratios (A). We add an asterisk to the RR* estimator here and in the text to indicate that it is the current default of the four estimators. We list MLE in quotes to indicate that the RR* estimator is not itself an MLE; the R prevalence estimator is the MLE for r conditional on degrees, and the RR* uses this prevalence estimator with estimated degrees.

Name	Symbol	Degree step		Prevalence step		Other names
RR*	$\hat{r}_{\text{RR}}$	R	$\hat{D}_i = N \cdot \frac{\sum_j Y_{ij}}{\sum_j N_j}$	R	$\hat{r} = \frac{\sum_i Y_{iH}}{\sum_i \hat{D}_i}$	"MLE"
RA	$\hat{r}_{\text{RA}}$	R	$\hat{D}_i = N \cdot \frac{\sum_j Y_{ij}}{\sum_j N_j}$	A	$\hat{r} = \frac{1}{n} \sum_i \frac{Y_{iH}}{\hat{D}_i}$	PIMLE
AA	$\hat{r}_{\text{AA}}$	A	$\hat{D}_i = N \cdot \frac{1}{k} \sum_j \frac{Y_{ij}}{N_j}$	A	$\hat{r} = \frac{1}{n} \sum_i \frac{Y_{iH}}{\hat{D}_i}$	MoS
AR	$\hat{r}_{\text{AR}}$	A	$\hat{D}_i = N \cdot \frac{1}{k} \sum_j \frac{Y_{ij}}{N_j}$	R	$\hat{r} = \frac{\sum_i Y_{iH}}{\sum_i \hat{D}_i}$	(Unmentioned)

The NSUM is cost effective but also depends on fairly strong assumptions that are likely violated in many settings. The assumption that any two people are equally likely to know each other as any other two people is known as the **random mixing assumption**. The binomial model also assumes perfect visibility (each respondent knows whether each person in their network is in H), perfect recall (each respondent can enumerate everyone they know, or at least report the correct total), truthful answers, and the absence of other survey and response error. In this paper, we focus specifically on the random mixing assumption.

Violations of the random mixing assumption are called **barrier effects** and are often expected in practice. Typically, we expect that not everyone in the population is equally likely to know people in the hard-to-reach population of interest. Homophily often drives connections, and people who are more similar to people in the hard-to-reach population may be more likely to know them (McPherson et al., 2001). Additionally, if people in hard-to-reach subpopulations tend to have smaller or larger degrees than people in the general population, this necessarily violates the random mixing assumption, and there is evidence that this may be true for some subpopulations of interest (Shelley et al., 1995).

In the years since the early NSUM papers, a body of research has extended this model to handle barrier effects and relax the random mixing assumption; McCormick (2021) and Laga et al. (2021) provide detailed reviews of these methods. However, these approaches are more complex and require additional data. For example, the generalized NSUM developed by Feehan and Salganik (2016) can be used in the presence of barrier effects and imperfect visibility, but this approach requires sampling from the hard-to-reach population in addition to the original probability sample from the general population. For this reason, Feehan and Salganik (2016) also describe how correction factors can be applied to the RR* estimator if sufficient data and expert knowledge exist to estimate those factors.

A systematic literature review by Ocagli et al. (2021) of all PubMed papers from the original NSUM paper through 2020 that use NSUM ultimately includes 35 studies. Examining these papers ourselves, we found that five of them focus on methods development and another two only estimate network size, not population size. Of the remaining 28 studies, all

appear to use the RR* estimator, sometimes with adjustment factors; one compares against generalized NSUM, another compares against the AA estimator, and some are not explicit about the specific estimator or procedure. None of these studies use or consider the RA or AR estimators. The most commonly used of these four estimators therefore seems to be the RR* estimator, first proposed to our knowledge by Killworth et al. (1998). Throughout this paper, we write RR* as a reminder that this is the current default basic NSUM estimator.

The goal of the present study is to systematically evaluate the comparative performance of these estimators. An earlier study by Killworth et al. (1998) compares the RR* and RA estimators, and a more recent paper by Habecker et al. (2015) promotes the use of the AA estimator, which they call the mean-of-sums (MoS) estimator. Habecker et al. (2015) suggest potential motivations for using the AA estimator and demonstrate on an example data set that the RR* and AA estimators (as well as other modifications they propose) result in very different estimates, but they do not provide a clear theoretical or empirical evaluation of which estimator seems to be less biased. Laga et al. (2021) refer to the RA estimator as the plug-in MLE, or PIMLE. We are not aware of any literature considering or proposing the use of the AR estimator, but we evaluate it in the present study along with the other three estimators.

When the binomial model is true, both the A and R prevalence estimators with fixed or known degrees are unbiased, but the latter has a smaller variance; we provide a proof in the online supplement using reasoning about harmonic and arithmetic means. Killworth et al. (1998) compute and empirically evaluate the RR* and RA on a data set and state that the latter has an unacceptably high variance, but their theoretical analysis assumes the degrees are known; thus their theoretical analysis concerns only the (one-step) A and R prevalence estimators, not the (two-step) RR* and RA estimators. The degree estimation step in the RR* and RA estimators not only involves additional uses of the conditional proportion assumption to estimate the degree, but also accounts for the distribution of degrees rather than conditioning on them. Perhaps this is the reason that surveys tend to use the RR* if they use one of the simple estimators, and that researchers tend to start from the RR* when

they develop methods to extend the NSUM approach and relax modeling assumptions.

2.3 A framework for studying estimator behavior under barrier effects

Our question is, how do these estimators compare under nonrandom mixing, which may be common in practice? To investigate this, we assume a model for link formation in the general population, then consider the distribution of the estimators over simple random samples from that population without replacement. The binomial model can be viewed as an approximation to the Erdős-Rényi network model in which the presence or absence of a link between each pair of nodes is independently drawn from the same Bernoulli(p) distribution (see online supplement and Cheng et al. (2020) for details). In this study, we generalize this model to incorporate barrier effects by using a stochastic block model (SBM) that partitions the general population into two mutually exclusive groups, the hard-to-reach group of interest (H) and everyone else (L).

Under the two-group SBM, the probability of a link between any two nodes takes one of three values depending on the membership of the two nodes involved, and links are assumed to form independently from one another conditional on group membership. We denote the within-group probabilities by p_{HH} and p_{LL} and the between-group probability by $p_{HL} = p_{LH}$. This model will have barrier effects as long as $p_{HH} > p_{HL}$, since in that case members of H will be more connected than members of L to people in H. The dissortative condition $p_{HH} < p_{HL}$ would also create barrier effects but is typically less realistic in practice. The Erdős-Rényi model is a special case of this SBM with $p_{HH} = p_{HL} = p_{LH} = p_{LL}$. Note that whether the link probabilities are constant as a function of N (as under the constant density assumption, so that the average degree grows linearly with total network size N) or scaled by $1/N$ (so that average degree is constant with respect to N), the expressions derived in this section will not change because the additional factors of N cancel.

Many network models can be approximated by a stochastic block model with some number of blocks, perhaps many (Olhede and Wolfe, 2014), so the two-group SBM is a motivating choice for studying the impact of barrier effects. The two-group SBM may provide insights

that have more general relevance, such as behavior based on network assortativity, even if a given problem is not believed to follow a two-group SBM.

For the sake of interpretability, we start with a simple case using estimated degrees: we suppose that respondents' degrees are estimated using one probe group K. Furthermore we suppose that $K \subset L$ to avoid having to specify the prevalence of H within K, which would be necessary to compute expectations and variances. This renders the probe group unrepresentative of the general population, since it contains no one in the hard-to-reach group H, but the representativeness of probe groups can be a concern in practice and is therefore relevant to consider here (McCormick et al., 2010). Additionally, the real data example in Section 2.5 assumes more than one probe group and does not require the probe group to be disjoint from H.

For the case of a single probe group, the A and R degree estimators are identical; therefore the AA and RA estimators are equivalent when degrees are estimated using a single probe group, and our analytical results will only compare the choice of prevalence estimator. Similarly, the AR and RR* estimators are equivalent when degrees are estimated using a single probe group. However, this analysis still accounts for the additional use of the random mixing assumption in estimating the degrees, and it also accounts for the distribution of degrees instead of conditioning on them. The degree estimators are no longer identical when more than one probe group is used; therefore we save comparison with the AA and AR estimators for Section 2.5.

Killworth et al. (1998) assume a simple random sample without replacement, while Feehan (2015) and Habecker et al. (2015) propose a way to extend this to general probability survey designs. For simplicity and to stay consistent with the original estimators, in this study we assume simple random sampling without replacement.

2.4 Analytical results on estimator bias and variance

We begin by deriving approximations to the bias and variance of each estimator under a two-group stochastic block model assuming degrees are estimated using a single probe group

K contained in L. Then to facilitate interpretation, we restrict further to the case in which the within-group link probabilities are equal to the same scaling factor a times the between-group link probability p_{HL}. We present closed-form approximations for the bias and variance, and we numerically compute the bias, variance, and RMSE for the two estimators over a range of parameter values to characterize the regions in which one estimator outperforms the other. As mentioned previously, the AA and RA estimators are equivalent when degrees are estimated using a single probe group, as are the AR and RR* estimators; hence this section discusses only the RR* and the RA. However, the four estimators are distinct once there is more than one probe group, so we compare all four estimators in Section 2.5.

Under the two-group SBM, the number of people person i knows in the hard-to-reach group H and probe group K, respectively, is given by

$$Y_{iH} = \sum_{j=1}^{N_H^*} A_{ij} \sim \text{Binom}(N_H^*, p_{H\,g_i}), \quad Y_{iK} = \sum_{j=1}^{N_K^*} A_{ij} \sim \text{Binom}(N_K^*, p_{g_i\,L}),$$

where A is the adjacency matrix of the general population network; $g_i \in \{H, L\}$ denotes the group membership of person i; Y_{iH} and Y_{iK} are independent for any i, j; $N_H^* = N_H - 1$ if $i \in H$ and $N_H^* = N_H$ otherwise; and N_K^* is defined analogously to N_H^*. Henceforth we assume N_H and N_K are sufficiently large such that $N_H^* \approx N_H$ and $N_K^* \approx N_K$.

Using first-order Taylor approximations for the expectation and variance of ratios, we can estimate the expectation and variance of the estimators over the SBM superpopulation for a given sample. Since the link probability depends on the respondent's group membership, the expressions for the expectation and variance depend on n_H, the number of sample respondents that belong to the hard-to-reach group H. We then take the limit $n_H/n \to r$ as in simple random sampling without replacement (see the online supplement for further details of the derivations). In each case, the expectation is a function only of r and the three link probabilities, while the variance is also a function of n, N, and N_K:

$$
\begin{aligned}
E\left(\hat{r}_{\mathrm{RR}}\right) &\to r\frac{rp_{HH}+(1-r)p_{HL}}{rp_{HL}+(1-r)p_{LL}} \\
E\left(\hat{r}_{\mathrm{RA}}\right) &\to r\left[r\frac{p_{HH}}{p_{HL}}+(1-r)\frac{p_{HL}}{p_{LL}}\right] \\
\mathrm{Var}\left(\hat{r}_{\mathrm{RR}}\right) &\to \frac{r}{nN}\frac{\left(rp_{HL}+(1-r)p_{LL}\right)^2\left[rp_{HH}(1-p_{HH})+(1-r)p_{HL}(1-p_{HL})\right]}{\left(rp_{HL}+(1-r)p_{LL}\right)^4}+ \\
&\quad \frac{r^2}{nN_K}\frac{\left(rp_{HH}+(1-r)p_{HL}\right)^2\left[rp_{HL}(1-p_{HL})+(1-r)p_{LL}(1-p_{LL})\right]}{\left(rp_{HL}+(1-r)p_{LL}\right)^4} \\
\mathrm{Var}\left(\hat{r}_{\mathrm{RA}}\right) &\to \frac{r}{nNp_{HL}^2}\left[rp_{HH}(1-p_{HH})+(1-r)p_{LL}(1-p_{HL})\right]+ \\
&\quad \frac{r^2}{nN_Kp_{HL}^3}\left[rp_{HH}^2(1-p_{HL})+(1-r)p_{HL}^2(1-p_{LL})\right]
\end{aligned}
$$

Note that when the three link probabilities are equal, corresponding to the binomial model, the RR* estimator does not have smaller variance as suggested by Killworth et al. (1998): both estimators have the same first-order variance. We believe this is the first time this result has been shown for these estimators. In this case, the first-order approximations of both estimators' expectations equal the true prevalence; in the language of Feehan and Salganik (2016) and other literature, both the RR* and RA estimators are essentially unbiased.

Thus we have expressions for the bias and variance of each estimator under a general two-group stochastic block model when degrees are estimated using a single probe group $K \subset L$. These expressions describe how the expectation and variance depend on various parameters such as r and the link probabilities. However, we would like to be able to characterize the regions of parameter space in which each estimator performs better than the other: under which combinations of parameters is one estimator better than the other? With four parameters for the expectation and seven for the variance, it is hard to identify patterns that summarize when to use which estimator. Therefore we now analyze a slightly simpler case in which we further specify the relationships among the three link probabilities: Fix $p_{HH} = p_{LL} = ap_{HL}$ for some $0 < a < \infty$. Notice that $a = 1$ corresponds to the Erdős-

Rényi case, $a > 1$ corresponds to assortativity, and $a < 1$ corresponds to dissortativity. With these additional constraints, we can characterize the three link probabilities with just the two parameters a and p_{HL}, the latter of which we will now denote simply by p. This reduces the number of degrees of freedom by two, so that the expectation is now a function of just two parameters and the variance is a function of five.

The biases are now a function only of a and r:

$$\begin{aligned}
\text{Bias}\left(\hat{r}_{\text{RR}}\right)(a,r) &\to r\left[\frac{(a-1)(2r-1)}{(1-r)a+r}\right] \\
&= \begin{cases} > 0 & \{a>1\}\cap\{r>0.5\} \text{ or } \{a<1\}\cap\{r<0.5\}, \\ = 0 & \{a=1\}\cup\{r=0.5\}, \\ < 0 & \text{else}, \end{cases} \\
\text{Bias}\left(\hat{r}_{\text{RA}}\right)(a,r) &\to r\left[\frac{(a-1)\big[(a+1)r-1\big]}{a}\right] \\
&= \begin{cases} > 0 & \{a>1\}\cap\{a>\frac{1-r}{r}\} \text{ or } \{a<1\}\cap\{a<\frac{1-r}{r}\}, \\ = 0 & \{a=1\}\cup\{a=\frac{1-r}{r}\}, \\ < 0 & \text{else}. \end{cases}
\end{aligned}$$

The RR* is unbiased if and only if the Erdős-Rényi case holds ($a = 1$) or the hard-to-reach population prevalence is exactly 50%. The RA is unbiased if and only if the Erdős-Rényi case holds ($a = 1$) or $a = (1-r)/r$. It is unlikely that the parameters take these exact values such that the estimators are exactly unbiased, but these conditions serve as boundary cases to define regions in which one estimator or the other has smaller bias.

Now we approximate the variances. Denoting the prevalence of the probe group K by $r_K := N_K/N$, the dependence simplifies to effectively five parameters: r, a, p, r_K, and nN, since the dependence on sample size n and population size N is only through their product.

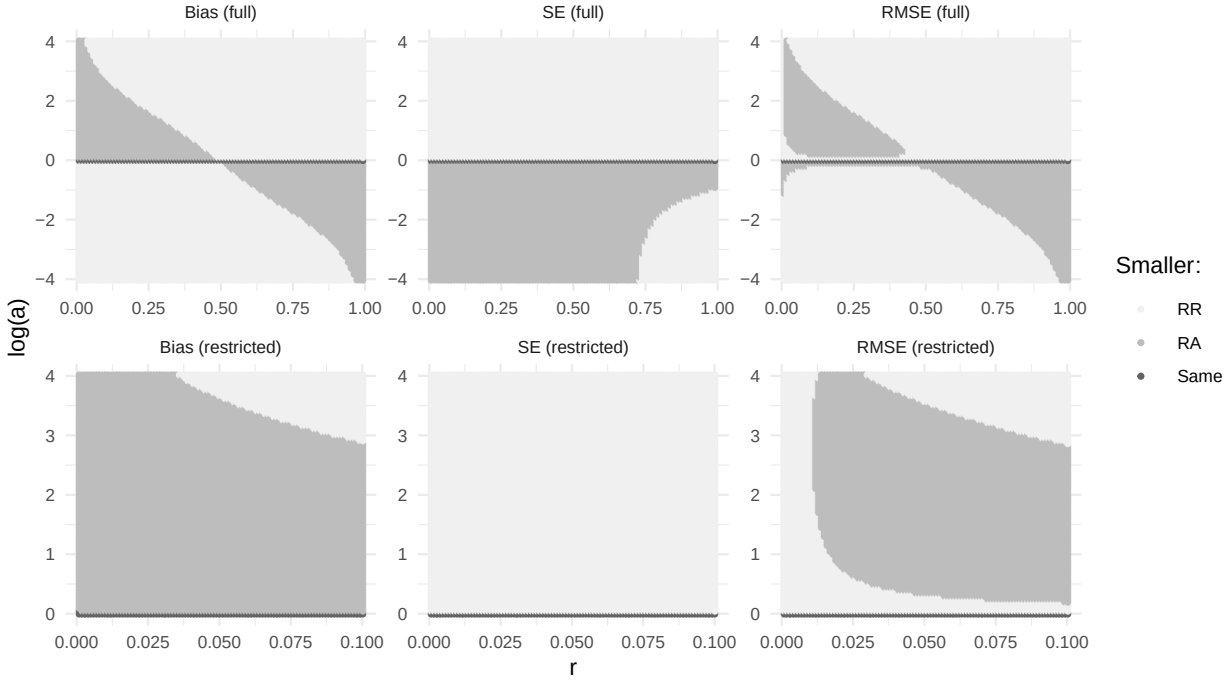


Figure 2.1: Comparing estimator bias (left panel), variance (center panel), and RMSE (right panel) as a function of a and r. The darkest regions indicate the combinations of a and r for which the RA has the lower value of the quantity examined in that subplot. Note the log scale (base-e) on a, such that the x-axis corresponds to the Erdős-Rényi case $a = 1$ in which the two estimators have the same bias and variance. Here, $p = 0.01$, $n \cdot N = 500,000$, and $r_K := N_K/N = 0.1$. The top row shows results over a wider range of parameter values while the bottom row shows results for a smaller range of parameter values thought to reflect most practical settings, namely $a > 1$ (assortative) and prevalence smaller than 10%. Under assortativity and for prevalences less than 10 or 20%, the RA often has smaller bias and RMSE than the RR*, and the choice of estimator should depend on the specific populations and study.

$$\mathrm{Var}\left(\hat{r}_{\mathrm{RR}}\right) \to \frac{r}{nNp}\left[\frac{(r+(1-r)a)^2\left(ra+(1-r)-\left[ra^2+(1-r)\right]p\right)}{\left[r+(1-r)a\right]^4} + \frac{r}{r_K}\frac{(ra+(1-r))^2\left(r+(1-r)a-\left[r+(1-r)a^2\right]p\right)}{\left[r+(1-r)a\right]^4}\right]$$

$$\mathrm{Var}\left(\hat{r}_{\mathrm{RA}}\right) \to \frac{r}{nNp}\left(pa(1-a)r+(1-p)a+\frac{1}{r_K}\left[\left([1-p]a^2+pa-1\right)r^2+(1-pa)r\right]\right)$$

To characterize the conditions under which each estimator outperforms the other, we evaluate the approximate bias and variance expressions above over a range of parameter values and visualize the regions in which each estimator has lower bias, lower variance, and lower RMSE. In the top row of Figure 2.1 we show the results for $\log(a)$ ranging from -4 to 4 in increments of 0.1, and r ranging from 0.01 to 0.99 in increments of 0.02. The bottom row shows the results restricted to assortative cases with small r, with $\log(a)$ ranging from 0 to 4 in increments of 0.05, and r ranging from 0.001 to 0.1 in increments of 0.001. In both these cases we fix $p = 0.01$, $nN = 500,000$, and $r_K = 0.1$ since the variance changes with these parameters.

The leftmost panels of Figure 2.1 compare the bias of the two estimators as a function of a and r; the darkest shaded region is the region in which the RA has smaller bias in magnitude. We have already shown the two estimators have the same bias and variance and therefore the same RMSE for $a = 0$, which is the x-axis in these figures. In practice, we typically expect assortativity ($a > 1$) and a fairly small value of r, probably less than 10 or 20%; throughout most of this region of the parameter space (for any values of the other parameters, since the estimator biases depend only on a and r), the RA has smaller bias.

The middle panels of Figure 2.1 illustrate that the variance is smaller for the RR* under assortativity and generally smaller for the RA under dissortativity. The rightmost panels comparing the RMSE of the two estimators resemble the leftmost panels comparing the bias. Therefore, in the settings most likely to be practically relevant, and for sample sizes and population sizes likely to be realistic, the RA tends to have smaller RMSE. For assortative settings with small prevalence r, the RA has lower RMSE. When the assortativity is weaker,

the RA has lower RMSE for a wider range of r, and under stronger assortativity the upper bound on r for this region shrinks.

Therefore, although the RR* is the most commonly used estimator in the literature, the RA may often have lower bias and RMSE in the presence of barrier effects.

The relative importance of the bias and variance in the RMSE are determined by nN, r_K, and p. The region of parameter space in which the RA has lower RMSE expands with nN and with p and shrinks with r_K; see the online supplement for figures and additional details. The size of this region depends more strongly on nN and p than on r_K.

2.5 Facebook 100 data example

We also conduct simulations using an example data set, and the results are consistent with the analytical results in Section 2.4 even though (1) these networks are not simulated from a two-group stochastic block model and (2) we use multiple probe groups to estimate the degrees.

For these simulations we use the Facebook 100 data set, which consists of the intra-school links in the September 2005 Facebook networks of 100 colleges and universities (Traud et al., 2011, 2012). The networks range in size from 769 to 41,554 nodes, and 91% of the schools have fewer than 25,000 nodes. Similar to Feehan et al. (2022), we create candidate probe groups from the following five variables: status (such as faculty or student), gender, year, dorm, and major. We treat them as categorical variables, with an indicator for each level of each variable, and each indicator variable whose prevalence in that school network is 0.1-10% of the population is a candidate probe group for that school network. In other words, if people living in Dorm 18 at a given school represent 0.1-10% of that school's Facebook network, we include this group in the list of candidate probe groups for that school.

We simulate surveys on this complete network data set, and since we know the true prevalence of each group in the networks, we can compare the bias, standard error (SE), and root mean square error (RMSE) of the four estimators. Some covariate data is missing in this data set, but the networks as they are recorded are taken to be the true networks for

the purpose of evaluating the estimators in this study. For example, the number of people in a given major is taken to be the number of people whose major is recorded as that value.

For each school network, we select the 20 largest candidate probe groups with low assortativity (we require the assortativity coefficient magnitude to be less than 0.1) to constitute the probe groups used in estimating respondents' degrees. From the remaining candidate groups, we choose the 10 most assortative groups and the 10 least assortative groups (those with the smallest or most negative assortativity coefficients) to be the hard-to-reach groups whose prevalences we will estimate; we will refer to a choice of school network and hard-to-reach group as a case. In the results and discussion presented here, we focus on the assortative cases since we believe those to be more relevant in practice, but we comment also on the low-assortativity cases. One school has only 29 candidate groups, so we only evaluate 9 cases for this school, resulting in a total of 999 assortative cases and 990 low-assortativity cases across the 100 schools. The prevalences of the resulting set of assortative hard-to-reach groups range from 0.010 to 9.9%, 2.2% on average.

For each case, we draw 500 "survey" samples of 500 people each using simple random sampling without replacement to represent our survey respondents, and for each survey sample we compute the RR*, RA, AR, and AA estimates for that case. We compute the responses for each survey respondent directly from the network data; for example, we compute each person's response to "How many people do you know in group X?" as the number of people they are connected to in that group. We approximate the mean and SE of each estimator for each case as the sample mean and sample standard deviation across the 500 surveys for that case, then use this to estimate the bias and RMSE for each estimator.

We categorize these cases based on their degree ratio, computed by averaging the degrees of everyone in the hard-to-reach group and dividing by the average across the degrees of everyone in the general population; in this context, the hard-to-reach group is taken to be a subset of the general population and a person's degree is the number of people they know in whatever group is defined to be the general population (Feehan and Salganik, 2016). We consider three broad categories in which the degree ratio is "low" (< 0.8, 290 cases), "high"

(> 1.2, 360 cases), or near 1 (between 0.8 and 1.2, 349 cases). The assortativity coefficient of each case ranges from 0.019 to 0.87, with first, second, and third quartiles of 0.23, 0.29, and 0.34.

It is worth noting that in these simulations, some degrees are estimated to be zero; overall across the 100 networks, 2.3% of the degree estimates are zero, ranging from 0.4-5.3% for each network. As a result, the RA and AA sample estimates for a given survey are undefined when both the numerator (response) and denominator (estimated degree) are zero for a given person, and infinite when only the estimated degree is zero. For each case, we compared four different methods of handling the zero-valued degree estimates when computing the mean and standard deviation of the survey estimates, and we found that the choice of method made little difference in the performance of the estimators. Further details are provided in the online supplement. For the rest of the analysis presented here, we set infinite values to 1 and exclude undefined values.

Even when the estimated degrees are not zero, the ratio of response to estimated degree $Y_{iH}/\hat{D}_i$ is nonsensically greater than 1 if the response is greater than the estimated degree. This is true for 0.14% of the people across these 100 networks. In the simulations, we have handled this by replacing any such ratios with 1; effectively, this assumes that for anyone whose estimated degree is smaller than their response, their personal network consists only of people in the hard-to-reach population. It is not that we think this is true, since the person may just know disproportionately fewer people in the probe groups than one would expect based on their prevalences, but it is one way to handle these cases in the absence of further knowledge. Nonetheless, in these 100 networks, we do find that people with greater responses than estimated degrees tend to know many more people in the hard-to-reach group than average (22 on average, versus an average of 1.2 for everyone across the 100 networks) and simultaneously have smaller true degrees than average (36 versus 78), so the distribution of their true ratios is skewed toward 1. Omitting these nonsensical ratios instead of replacing them with 1 leads to similar results as those we report below.

The degree estimates are reasonable but not idealized. The first, second and third quar-

Table 2.2: Pairwise comparisons of the four estimators by RMSE. For example, the RA estimator has smaller RMSE than the RR* estimator in 61.5% of the cases.

		Larger RMSE		
		AA	RR*	AR
Smaller	RA	67.1%	61.5%	61.9%
RMSE	AA	-	58.4%	58.6%
	RR*	-	-	52.3%

tiles for percent error in the degree estimates are 9.1, 19.4, 33.9% for the R degree estimates and 9.3, 19.8, 35.5% for the A degree estimates, respectively. Correlation between estimated and true degrees across all the networks using the chosen probe groups is 0.976 for the R estimates, 0.974 for the A estimates. Scatterplots of estimated versus true degrees are available in the online supplement.

Overall, the RA (AA, RR*, AR) estimator has the lowest RMSE of the four estimators in 41% (21%, 20%, 18%) of the 999 cases. The RA has lower RMSE than the AA (RR*, AR) estimator in 67% (62%, 62%) of the cases (see Table 2.2). The RR* estimator tends to have the lowest SE (46% of the cases) while the AA estimator tends to have the highest (55% of the cases). The RA estimator tends to have the lowest bias (43% of cases) while the AR and RR* tend to have the highest (34% of cases for AR, 33% for RR*).

Among cases with low degree ratios, thought to be more relevant to some hard-to-reach populations, the RA estimator has lower RMSE than the RR* (AR, AA) estimator in 80% (80%, 64%) of the cases. Figure 2.2 illustrates that while the RA tends to have larger variance than the RR* estimator in these cases, it tends to have smaller bias and RMSE.

Figure 2.3 shows that the RA estimator tends to have lower RMSE than the RR* esti-

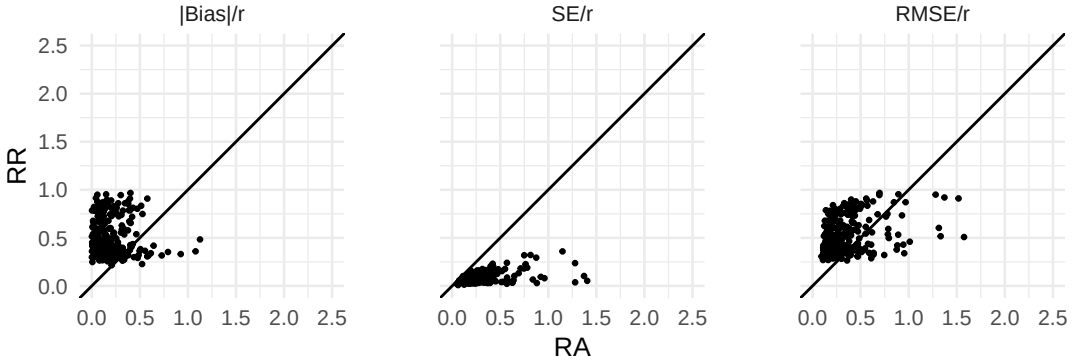


Figure 2.2: Comparing the absolute-value bias (left panel), SE (center panel), and RMSE (right panel), all standardized by the true prevalence, of the RR* and the RA in the 290 "low" degree ratio cases (< 0.8) from the Facebook 100 simulations. We standardize these metrics by the true prevalence of each case since the true prevalence varies widely across cases. Each point represents the average across 500 surveys of size 500 for one combination of school network and hard-to-reach group. The diagonal line is the one-to-one line; points above the line have lower values for the RA than the RR*. Among these cases, the RA tends to have smaller bias, larger variance, and smaller RMSE than the RR* estimator.

mator when the degree ratio is high as well, and higher RMSE when the degree ratio is near 1. Feehan and Salganik (2016) demonstrate that the bias of the RR* estimator is a function of the degree ratio, increasing as the degree ratio diverges from 1. In these simulations, the RA estimator tends to have lower bias than the RR* estimator in the low- and high-degree ratio cases. Therefore, researchers that are confident the degree ratio is near 1 for the hidden and general population of interest to them may want to use the RR*, but otherwise the RA may have lower RMSE.

We consider how these pairwise comparisons change with respect to various other factors. When the cases are grouped by true prevalence or by the percent of people in the network whose estimated degrees are zero, within each group the RMSE and bias still tend to be

lower for the RA estimator than the other estimators. Among the low-assortativity cases, we see the opposite trend in which the RA estimator tends to have worse RMSE than the RR* estimator, consistent with the analytical results presented above.

When the cases are grouped by an estimate of p_{HL}, computed by dividing the number of observed $H-L$ edges by the possible number of $H-L$ edges ($N_H(N-N_H)$), within each group the RMSE and bias still tend to be lower for the RA estimator than the other estimators except for the lowest quartile of the estimated probabilities (our estimates for p_{HL} range from 9.4×10^{-5} to 8.8×10^{-2}, and the first quartile is as large as 3.0×10^{-3}). This agrees with our analytical result above that the region in which the RA estimator has lower RMSE shrinks as this link probability decreases.

In these Facebook 100 simulations, the RA has smaller bias, variance, and RMSE than the AA estimator in more than half the cases, even among cases with degree ratios near 1. Recall that the AA estimator uses the same size estimator as the RA but uses the average of ratios for the degree estimator as well (Table 2.1). The smaller variance is not surprising since the A estimator has greater variance than the R estimator as demonstrated in the online supplement. Yet this choice appears to increase not only the variance but the bias. We suspect the increased bias follows from the fact that the probe groups are intended to be collectively but not necessarily individually representative of the general population (McCormick et al., 2010). If the set of chosen probe groups satisfy this property, then taking the ratio of averages keeps the numerator and denominator representative while taking the average of ratios changes the relative weighting of the probe groups.

In the set of networks and choices of probe groups here, the A and R degree estimates are very comparable. In these simulations, the percent error is at least 0.5% smaller for the R estimator than the A estimator in 49.8% of the cases, and the reverse is true in 42.2% of the cases. The distributions of their percent errors are very similar, and the two estimators are highly correlated with each other. When the two estimators do not perform so similarly, the difference in performance between the RA and AA estimators (and between the RR* and AR estimators) could be greater.

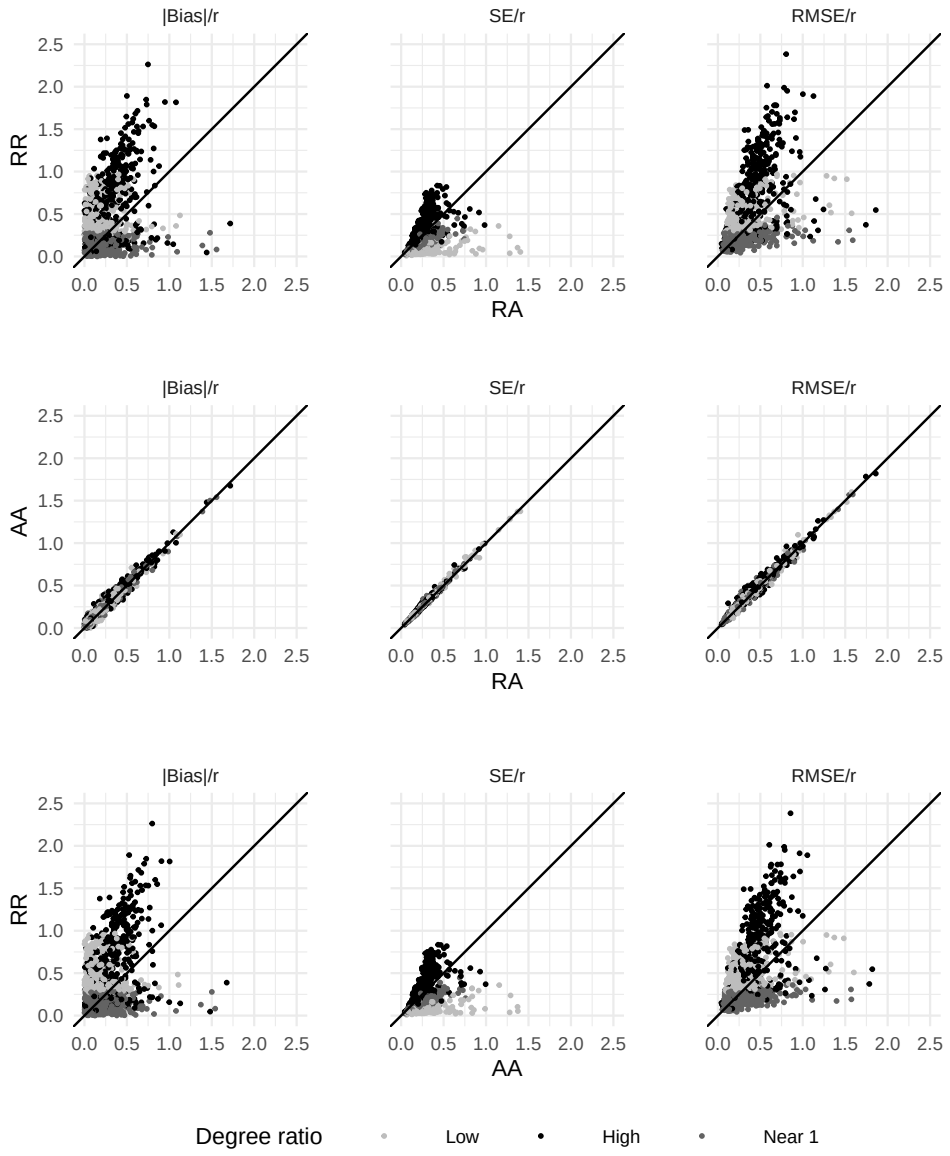


Figure 2.3: As in Figure 2.2 except comparing three of the four estimators for all 999 cases. Points representing cases with low, near-1 and high degree ratios are shown in light grey, dark grey, and black, respectively. The RA and AA estimators tend to have lower bias and RMSE than the RR* and AR estimators for cases with low and high degree ratios.

2.6 Discussion

We have presented theoretical evidence that over what seems likely to be a realistic range of true prevalence, sample size, and population size, the RA estimator often has lower bias and RMSE than the RR* estimator under a two-block stochastic block model with assortativity. We have also provided empirical evidence on real networks that the RA estimator may often have lower bias and RMSE than the RR* estimator under assortativity. Together, these results suggest that using the ratio-of-averages degree estimator and the average-of-ratios size estimator often has lower bias and RMSE than the most commonly used NSUM basic scale-up estimator, which uses the ratio of averages for both the degree and size estimators.

We have also shown empirical evidence that the RA estimator often has lower bias, RMSE, and variance than the AA (mean-of-sums) estimator, suggesting that using the average-of-ratios size estimator with the ratio-of-averages degree estimator may be better than using the average-of-ratios size estimator for both the size and degree estimators. We have provided some reasoning for these findings.

Generalized NSUM and other complex methods are likely to be preferable to these simple estimators when feasible. However, in studies that use one of these simple estimators without any corrections, the RA may be the most robust choice. When the necessary data are available, it may be easier to correct bias in the RR* estimator than the RA since the RR* bias can be expressed directly in terms of the degree ratio; however, this requires having reliable estimates of the degree ratio. When ad hoc corrections to the simple estimators are applied to the RR*, it may be helpful to also compute the RA to help bound the result; it is easily done and does not require additional data or computational power.

The theoretical and empirical findings presented here raise the question whether it may be more robust to develop methods based on the RA instead of the RR*. This may depend on how successfully we can characterize the bias of the RA and correct for it. It may turn out that we can more readily understand and correct for the bias of the RR*.

These findings do not indicate that the RA estimator is always best, but rather that

it outperforms the other estimators by enough of a margin in enough potentially relevant parameter combinations that it is worth considering (1) as a default choice in the absence of further information and (2) as the basis for future method development. The evidence in favor of one estimator over another still depends on the range of parameter values expected to be practically relevant. The more we know about the typical network and sample size, assortativity, approximate prevalence, typical degree sizes, and the fraction of the population that is in one or more of the probe groups in hard-to-reach population studies, the better we can constrain the relative performance of the estimators.

Additionally, aside from rarity (low prevalence) in some of the cases, the Facebook 100 network data do not represent subpopulations we expect to be hidden, stigmatized, or hard to reach for other reasons. Comparing the robustness of these estimators on additional data sets beyond our initial example, particularly using network data sets involving hard-to-reach populations, would provide more direct evidence for evaluating these estimators.

We have interpreted the simulation results in terms of the proportion of cases in which one estimator performs better than another, or the distribution of relative estimator performance across the cases. The set of cases is not a sample from a distribution and should be interpreted with caution.

The zero-valued degree estimates that arose in our simulations beg the question of how to handle these cases in practice. We considered four different methods, three of which reduce the effective sample size by excluding respondents with zero-valued degree estimates and non-zero responses, zero-valued degree estimates and zero-valued responses, or both (see the online supplement for details). The impact of the reduced sample size in a given study depends on how common these two types of cases are in that particular study. In the current study, the choice of method only impacts the ranking of estimators by RMSE in a handful of simulated cases, but further research is needed to study when and why that choice matters and how greatly it affects relative estimator performance. Perhaps the fourth method which does not exclude any respondents' data is sufficient for handling zero-valued degree estimates, but it would be valuable to study how large a proportion of the respondents would have to

have zero-valued degree estimates before this method introduces too much bias. Otherwise, the reduction in effective sample size by the RA estimator may be another cost to consider when deciding which estimator to use in a given study.

In both the analytical and simulation studies in this paper, the only source of error is nonrandom mixing. In practice, of course, we expect other errors. In this study, for instance, we suppose we knew respondents' true responses; in reality, a respondent may not realize someone they know is a member of the group in question or they may not want to reveal they know people in that group, leading to underestimated responses. Other errors could tend to overestimate the response or have different effects in different contexts. Future work can examine the impact on relative estimator performance in the case that these errors impact the responses but not the degree estimates, or in the case that the errors impact both.

Combining NSUM and other methods may also prove to be a helpful strategy. Hard-to-reach populations are often studied in the context of estimating the impact of a social or epidemiological concern; for instance, estimating the number of people who have experienced labor trafficking is part of an effort to understand, intercept and prevent trafficking. Studies may benefit from using RDS and related methods to learn about possible causes and interventions, while using NSUM to do more frequent monitoring. For example, researchers and monitoring agencies could use NSUM regularly to estimate how many people are being trafficked over time, and implement less frequent RDS studies to both calibrate the NSUM estimates and learn from people who have been trafficked how they entered their situation, how they were able to leave the situation, or what prevented them from leaving.

Chapter 3

SCALABLE SPATIAL STREAM NETWORK (S3N) MODELS

This chapter contributes a scalable spatial modeling approach called S3N that accounts for spatial correlations by distance along network paths within a network, particularly for stream networks, although these methods can be generalized to other contexts besides streams. S3Ns allow for estimating and predicting at larger scales than previously possible with existing spatial models for stream networks. The code for this chapter is available at `http://github.com/jpierkunke/S3N_thesis`.

3.1 Introduction

Spatial regression models such as kriging or Gaussian processes model a response variable such as species abundance as a function of environmental factors, capturing remaining spatial dependence through some spatial process (Cressie and Wikle, 2011; Rasmussen and Williams, 2006). We consider the context in which spatial correlations are based not (or not only) on Euclidean distance but on some measure of distance along a network. The specific context we consider here is that of a stream network. There are two main challenges with this approach in the context of stream networks: accounting for stream-based spatial dependence and achieving computational scalability.

Spatial dependence between observations along a river network is thought to be a function not just of Euclidean distance, as in typical spatial process models, but also of the directed or undirected distance along the stream network (Dent and Grimm, 1999; Wyatt, 2003; Ganio et al., 2005; Peterson et al., 2006; Webster et al., 2008). For instance, waterborne chemicals and small organisms may passively flow downstream, while fish can actively swim downstream or upstream, and the tendency to swim upstream varies by species (Peterson and

Ver Hoef, 2010). While it may seem natural to simply replace one distance metric with the other, substituting stream distance for Euclidean distance in already established covariance functions, Ver Hoef et al. (2006) demonstrate that this approach does not guarantee a valid (positive definite) covariance function. Ver Hoef and Peterson (2010) develop valid stream covariance functions using moving-average constructions, and their proposed spatial stream network (SSN) models were made available first as an ArcGIS toolkit and R package (`STARS` and `SSN`) in 2014 and later as a pair of R packages (`SSNbler` and `SSN2`) in 2024 (Peterson and Ver Hoef, 2014; Ver Hoef et al., 2014; Peterson et al., 2024; Dumelle et al., 2024). Using the earlier implementation, Isaak et al. (2017) apply these methods to obtain geographic trout distributions and population estimates in the Salt River watershed on the Idaho-Wyoming border.

SSN models represent a great advance in handling the spatial dependence induced by stream networks, but they do not address the limited scalability of classic spatial process models and they introduce additional computational expense. Given n locations at which observations are available, fitting spatial process models requires $O(n^3)$ floating point operations (flops) and $O(n^2)$ storage to compute the inverse and determinant of the $n \times n$ covariance matrix (Rasmussen and Williams, 2006; Datta et al., 2016b). Isaak et al. (2017) use only 108 observations and study one species over a 2,300-km^2 area. Peterson et al. (2020) study two species over a 90,822 km^2 area with 2,000-3,000 observation locations for each species, which is the largest region and number of observations we have found in the literature and still much smaller than the multi-state and national scale necessary for conservation, and yet the authors found that the first implementation of the SSN models with the `SSN` package required five days of computation time on their dataset. With the newer `SSN2` package, they report that "some model fits [...] completed in approximately 10 minutes," consistent with the times we estimate in our own benchmarking of SSN models, but we demonstrate in this chapter that with twice or ten-fold as many observations or with a larger stream network, estimation and prediction with the newer implementation quickly become infeasible again. Additionally, spatial stream network models require more complicated preprocessing than

spatial models that only account for correlations based on Euclidean distance. Both existing implementations of SSNs require significant preprocessing time.

The two common approaches to modeling spatial processes with large datasets are low-rank approximations and sparsity assumptions, and Datta et al. (2016a) develop the sparsity approach of Vecchia (1988) into a class of well-defined spatial processes called nearest-neighbor Gaussian process (NNGP) models. Saha and Datta (2018) dramatically further reduce the estimation and prediction cost by developing a bootstrap for rapid inference on spatial covariances (BRISC), with an R package by the same name. On their example temperature data set of 105,569 locations, their machine crashed when attempting to compute the inverse of the covariance matrix for the full spatial process, but estimation and prediction using BRISC took only 51.8 minutes and 1.8 seconds, respectively. However, their models currently only allow for spatial dependence as a function of Euclidean distance.

Our contribution, the scalable spatial stream network (S3N) model, combines SSNs, NNGPs, BRISC, and network preprocessing insights to enable national and regional spatial process models on stream networks. Section 3.2 provides background on classic spatial process models. Section 3.3 introduces stream networks, stream covariance functions defined on these networks, and how SSN models work. Section 3.4 examines how NNGPs and BRISC can improve the computational efficiency of estimation, inference and prediction with SSNs. Section 3.5 highlights current inefficiencies in SSN network preprocessing and identifies the fundamental steps required for preprocessing so that they can be implemented more efficiently. Section 3.6 defines S3Ns and how they combine the insights from Sections 3.2-3.5. Section 3.7 describes the benchmarking and validation analyses we performed to assess S3Ns and SSNs, and Section 3.8 presents the results of these analyses. Section 3.9 concludes this chapter with a discussion.

3.2 Spatial process models

Given observed responses $\boldsymbol{Y} = (Y(\boldsymbol{s_1}), ..., Y(\boldsymbol{s_n}))$ at n point locations $\boldsymbol{s_1}$, ..., $\boldsymbol{s_n}$ across a stream network, we wish to estimate the geographic distribution of the response variable

across the region and estimate regional totals or other summary statistics. The basic linear model assumes the mean is a linear function of p covariates and the observational errors at different points are independent:

$$\boldsymbol{Y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} = \big(\varepsilon(\boldsymbol{s_1}), ..., \varepsilon(\boldsymbol{s_n})\big), \quad \text{with } \mathrm{E}(\boldsymbol{\varepsilon}) = \boldsymbol{0}, \tag{3.1}$$

$$\mathrm{Var}(\boldsymbol{\varepsilon}) = \sigma^2 \boldsymbol{I}, \text{ or equivalently, } \Sigma(\boldsymbol{s_i}, \boldsymbol{s_j}) := \mathrm{Cov}(Y(\boldsymbol{s_i}), Y(\boldsymbol{s_j})) = \sigma^2 \delta_{ij}. \tag{3.2}$$

To allow for spatial correlations, we can instead assume a more general form for $\mathrm{Var}(\boldsymbol{\varepsilon}) = \boldsymbol{\Sigma}$. A common way to construct this covariance is to decompose it into additive components that capture different sources or forms of covariance. For instance, if we believe the fish densities have some kind of spatial correlation and also independent observational error, we may assume

$$\Sigma(\boldsymbol{s_i}, \boldsymbol{s_j}) = C(\boldsymbol{s_i}, \boldsymbol{s_j}) + \sigma^2 \delta_{ij}, \tag{3.3}$$

where k is the function that models the spatial correlation between any two points and $\sigma^2\delta_{ij}$ is the independent observational error, where $\sigma^2\delta_{ij} = \sigma^2$ if $\boldsymbol{s_i} = \boldsymbol{s_j}$ and 0 otherwise.

Often we want to both estimate model parameters from observations and use the fitted model to make predictions at other locations. In environmental statistics, this approach is known as universal kriging and the independent error term is known as the nugget (Cressie and Wikle, 2011). When researchers believe the independent error represents observational error, they may distinguish between the observed response Y which includes this observational error and the latent process f, often the target of inference and prediction, which does not (see Figure 3.1). One approach is to recast the model in Equations 3.1 and 3.3 as a hierarchical model:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{f} + \boldsymbol{\varepsilon}, & \mathrm{E}(\boldsymbol{\varepsilon}) &= \boldsymbol{0}, & \mathrm{Var}(\boldsymbol{\varepsilon}) &= \sigma^2 \boldsymbol{I}, \quad \boldsymbol{\Sigma} := \mathrm{Var}(\boldsymbol{Y}) = \boldsymbol{C} + \sigma^2 \boldsymbol{I}, \\ \boldsymbol{f} &= \boldsymbol{X\beta} + \boldsymbol{w}, & \mathrm{E}(\boldsymbol{w}) &= \boldsymbol{0}, & \mathrm{Var}(\boldsymbol{w}) &= \boldsymbol{C} \text{ where } \boldsymbol{C}_{ij} = C(\boldsymbol{s_i}, \boldsymbol{s_j}). \end{aligned} \tag{3.4}$$

Here, $\boldsymbol{f} = \big(f(\boldsymbol{s_1}), ..., f(\boldsymbol{s_n})\big)$ is the vector of latent process realizations at the n locations, and $\boldsymbol{w} = \big(w(\boldsymbol{s_1}), ..., w(\boldsymbol{s_n})\big)$ is the vector of realizations of the spatial process w at the n locations. The use of $\boldsymbol{C}$ for the latent covariance matrix and $\boldsymbol{\Sigma}$ for the response covariance

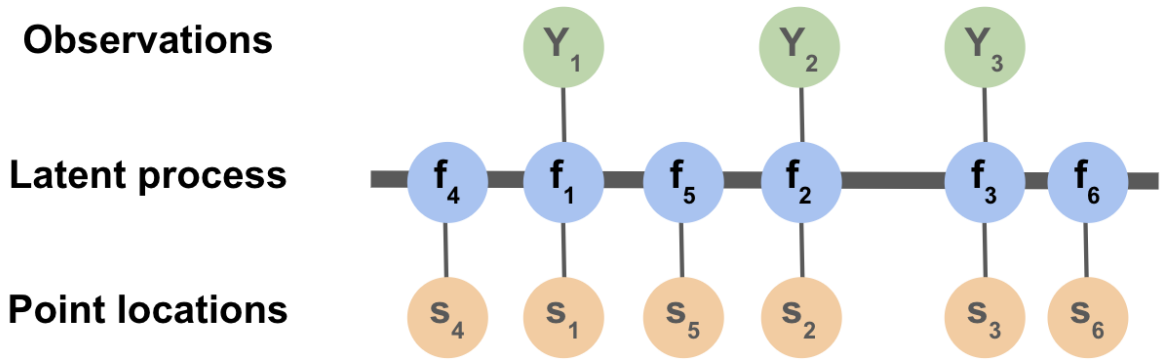


Figure 3.1: Observations are made at points s_1 through s_n, and the goal is to infer the latent process f and predict the value of f at points s_1 through s_k. Here $n = 3$ and $k = 6$. Adapted from Figure 2.3 of Rasmussen and Williams (2006).

matrix is a common notation and is consistent with the notation of Datta et al. (2016a) and Finley et al. (2019), two major references throughout this chapter.

If $\boldsymbol{Y}$ is assumed to have a multivariate normal distribution, then Model 3.4 is also referred to as a Gaussian process and is completely specified by the choice of covariates and the specification of the covariance function C (Rasmussen and Williams, 2006). This Gaussian process model takes the form

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{f} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(0, \sigma^2 \boldsymbol{I}), \\ \boldsymbol{f} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}, \quad \boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{C}), \;\; \boldsymbol{C}_{ij} = C(\boldsymbol{s_i}, \boldsymbol{s_j}). \end{aligned} \tag{3.5}$$

It is common in practice to assume some parametric structure for the covariance function; let $\boldsymbol{\theta}$ denote the vector of covariance parameters for the latent process. Under (spatial) stationarity, the assumption that the spatial process is translationally invariant, the covariance is a function of $\boldsymbol{s_i}$ and $\boldsymbol{s_i}$ only through their vector difference:

$$C(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}) = C_1(\boldsymbol{s_i} - \boldsymbol{s_j} \mid \boldsymbol{\theta}).$$

If we further assume isotropy, or rotational invariance, then the covariance is a function of

$\boldsymbol{s_i}$ and $\boldsymbol{s_j}$ only through their distance according to some relevant distance metric d:

$$C(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}) = C_2\big(d(\boldsymbol{s_i}, \boldsymbol{s_j}) \mid \boldsymbol{\theta}\big). \tag{3.6}$$

Typically $\boldsymbol{s_i} \in \mathbb{R}^2$ for all i, and d is the standard Euclidean distance metric. For example, if we assume exponential spatial covariance, we obtain the covariance function

$$C(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}) = \sigma_e^2 \exp\big(-d(\boldsymbol{s_i}, \boldsymbol{s_j})/\lambda_e\big), \quad \text{where } \boldsymbol{\theta} = \big(\sigma_e^2, \lambda_e\big). \tag{3.7}$$

Since both the observations and latent realizations are multivariate normal, their joint distributions are also multivariate normal. Suppose we have observations $\boldsymbol{Y} = (Y(\boldsymbol{s_1}), ..., Y(\boldsymbol{s_n}))$ at a set of locations $\mathcal{D}_1 = \{\boldsymbol{s_1}, ..., \boldsymbol{s_n}\}$ with covariates $\boldsymbol{X}_1$, and we wish to predict the latent process $\boldsymbol{f} = (f(\boldsymbol{r_1}), ..., Y(\boldsymbol{r_k}))$ at another set of locations $\mathcal{D}_2 = \{\boldsymbol{r_1}, ..., \boldsymbol{r_k}\}$ with covariates $\boldsymbol{X}_2$, where the two sets of locations $\mathcal{D}_1$ and $\mathcal{D}_2$ may overlap. Let $\boldsymbol{\Sigma}_{YY} := \text{Cov}(\boldsymbol{Y}, \boldsymbol{Y})$, $\boldsymbol{\Sigma}_{Yf} := \text{Cov}(\boldsymbol{Y}, \boldsymbol{f})$, and so on. The joint distribution of $\boldsymbol{Y}$ and $\boldsymbol{f}$, then, is given by

$$\begin{bmatrix} \boldsymbol{Y} \\ \boldsymbol{f} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \boldsymbol{\beta}, \begin{bmatrix} \boldsymbol{\Sigma}_{YY} & \boldsymbol{\Sigma}_{Yf} \\ \boldsymbol{\Sigma}_{fY} & \boldsymbol{\Sigma}_{ff} \end{bmatrix} \right). \tag{3.8}$$

Therefore, the conditional distribution of $\boldsymbol{f}$ given $\boldsymbol{Y}$, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is

$$\boldsymbol{f} \mid \boldsymbol{Y}, \boldsymbol{X}_1, \boldsymbol{X}_2 \sim N\left(\boldsymbol{M}, \boldsymbol{Q}\right), \text{ where} \tag{3.9}$$

$$\boldsymbol{M} = \boldsymbol{X}_2\boldsymbol{\beta} + \boldsymbol{\Sigma}_{fY}\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{Y}, \quad \boldsymbol{Q} = \boldsymbol{\Sigma}_{ff} - \boldsymbol{\Sigma}_{fY}\boldsymbol{\Sigma}_{YY}^{-1}\boldsymbol{\Sigma}_{Yf}. \tag{3.10}$$

Of the four covariance block matrices that appear in Equation 3.8, note that the nugget covariance term $\sigma^2\boldsymbol{I}$ appears only in $\boldsymbol{\Sigma}_{YY}$.

Alternatively, if we are only interested in modeling the response process itself and not the latent process, we can model $\boldsymbol{Y}$ marginally (here we combine $\boldsymbol{\varepsilon}$ and $\boldsymbol{w}$ since there is no longer a need to separate them):

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}, \quad \boldsymbol{w} \sim N(0, \boldsymbol{\Sigma}), \quad \boldsymbol{\Sigma}_{ij} = C(\boldsymbol{s_i}, \boldsymbol{s_j}) + \sigma^2\delta_{ij}. \tag{3.11}$$

In this case, suppose we have observations $\boldsymbol{Y_1} = (Y(\boldsymbol{s_1}), ..., Y(\boldsymbol{s_n}))$ at a set of locations $\mathcal{D}_1 = \{\boldsymbol{s_1}, ..., \boldsymbol{s_n}\}$ with covariates $\boldsymbol{X}_1$, and we wish to predict the response process $\boldsymbol{Y_2} =$

$(f(\boldsymbol{r_1}), ..., Y(\boldsymbol{r_k}))$ at another set of locations $\mathcal{D}_2 = \{\boldsymbol{r_1}, ..., \boldsymbol{r_k}\}$ with covariates $\boldsymbol{X}_2$. Let $\boldsymbol{\Sigma}_{11} := \text{Cov}(\boldsymbol{Y_1}, \boldsymbol{Y_1})$, $\boldsymbol{\Sigma}_{12} := \text{Cov}(\boldsymbol{Y_1}, \boldsymbol{Y_2})$, and so on; then the joint distribution of $\boldsymbol{Y_1}$ and $\boldsymbol{Y_2}$ is

$$\begin{bmatrix} \boldsymbol{Y_1} \\ \boldsymbol{Y_2} \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \boldsymbol{\beta}, \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{bmatrix} \right), \tag{3.12}$$

and the conditional distribution of $\boldsymbol{Y_2}$ given $\boldsymbol{Y_1}$, $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is

$$\boldsymbol{Y_2} \mid \boldsymbol{Y_1}, \boldsymbol{X}_1, \boldsymbol{X}_2 \sim N\left(\boldsymbol{M}, \boldsymbol{Q}\right), \text{ where} \tag{3.13}$$

$$\boldsymbol{M} = \boldsymbol{X}_2\boldsymbol{\beta} + \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{Y_1}, \quad \boldsymbol{Q} = \boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}. \tag{3.14}$$

3.3 Spatial stream network (SSN) models

Stream flow can induce different spatial covariance patterns that are not accounted for by Euclidean distance alone. Pollutants and some smaller aquatic species tend to move with the flow of water, and fish can swim with or against the flow. In each case, the relevant correlation length scale for flow-induced correlations is not the bird's eye distance between two points but the "fish's eye distance"[1], the length of the flow path between the points. Therefore, we want the latent covariance function $C(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta})$ in Equation 3.6 to include one or more variance components that are functions of stream distance. This section synthesizes the developments in spatial stream network (SSN) models over the past two decades since their initial development by Cressie et al. (2006) and Ver Hoef et al. (2006). Section 3.3.1 defines and describes stream networks. Section 3.3.2 defines stream distance and stream covariance functions, how the covariance functions are defined, and how they can be combined within an SSN. Section 3.3.3 explains the value of spatial weights in some of the covariance functions and how they are computed. Section 3.3.4 summarizes the SSN model.

---

[1]Thank you to Zachary G. Nicolaou for this term.

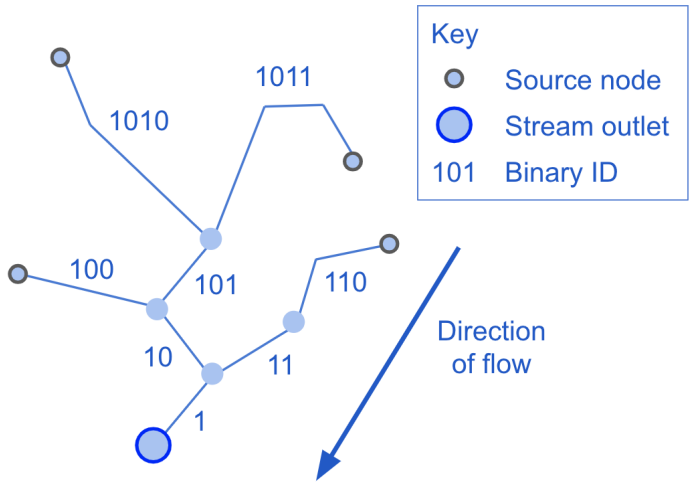
3.3.1 Stream networks

To understand this approach in further detail, we need to first mathematically define the stream network. We start by describing the topological assumptions and constraints of these models, and Figure 3.2 illustrates them for a simple example:

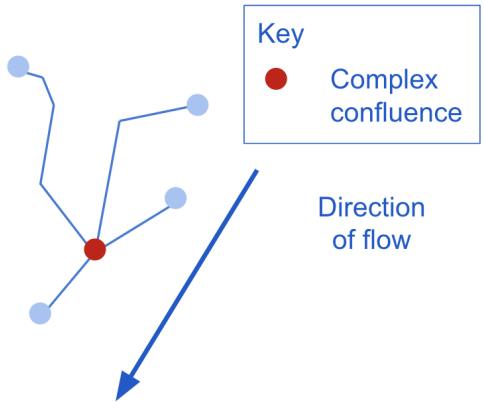
- In reality, streams have time-dependent and spatially varying width and volume, but Ver Hoef et al. (2006) model the streams as line segments connected in a directed tree graph and account for the width and volume later in other ways. Ecologists and hydrologists refer to a stream section or segment as a **stream reach**. Each reach consists of one or more line segments connected end to end with no branching; this structure is called a **linestring** in GIS terminology. Figure 3.2a shows three reaches that consist of more than one segment and five reaches that consist of a single segment each. A reach with no upstream neighbors is a **source reach** and its upstream endpoint is a **source node**; the flow of water through the network begins at source nodes and ends at **stream outlets**, which may refer to a reach with no downstream neighbors or to the downstream node of such a reach. The small stream network in Figure 3.2a has four source nodes and one stream outlet.

- Also in reality, streams can fork and merge in structures called braided streams, but for the purposes of these models Ver Hoef et al. (2006) assume the streams can only merge, not fork, as they flow downstream toward the stream outlets. Therefore if a stream network consists of a single connected component, it will have exactly one stream outlet which will be the further downstream point in the network, and if it consists of two or more components that are self-connected but are disjoint from one another, then it will have exactly as many stream outlets as it has such components.

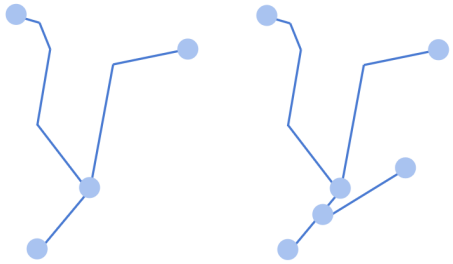
- Ver Hoef et al. (2006) additionally assume that at most two stream reaches may converge at any given point in space. This allows us to assume that each reach has at most two direct upstream neighbors, which allows us to uniquely identify stream reaches with



(a) Diagram of a small example stream network.



(b) Example of a complex confluence.



(c) Two ways to remove the complex confluence shown in Figure 3.2b.

Figure 3.2: A simple stream network example to illustrate terminology and topological constraints.

a **binary ID** as we will see shortly. Confluences that violate this condition by merging three or more reaches are called **complex confluences** and must be manually edited before fitting these models. Figure 3.2b-3.2c show an example of a complex confluence as well as two ways to remove it. The approach shown on the left side of 3.2c involves removing one of the reaches flowing into the confluence, while the right side involves moving the downstream node of one of the reaches flowing into the confluence; the latter approach requires splitting the downstream reach into two reaches.

In any finite stream network, there are many ways to assign a unique identifier to each stream reach for the purpose of indexing. Peterson and Ver Hoef (2014) insightfully proposed a specific identifier called the binary ID which not only uniquely indexes each reach but also encodes information about network structure. Binary IDs are assigned by starting at stream outlet reaches and working upstream to the source reaches in the following manner:

1. Each stream outlet reach receives a binary ID of 1. If the network consists of $K > 1$ self-connected components that are disjoint from one another, assign each component a network ID $1, ..., K$ and apply the following steps in parallel within each component to assign binary IDs; together the network ID and binary ID uniquely identify each reach.

2. Each stream outlet (binary ID = 1) has at most two direct upstream neighbors; label one 10 and the other 11. In other words, right-append either 0 or 1 to the stream outlet's binary ID to form the upstream neighbors' binary IDs.

3. For each of these upstream neighbors i, label its upstream neighbors in a similar manner by right-appending either 0 or 1 to the binary ID of stream outlet i. In the example shown in Figure 3.2a, the two upstream neighbors of reach 10 are labeled 100 and 101 while the single upstream neighbor of 11 is labeled 110.

4. Repeat Step 3 working upstream along each branch to each source reach. Figure 3.2a shows the result of this labeling process on a simple example network.

3.3.2 Stream distance and covariance functions

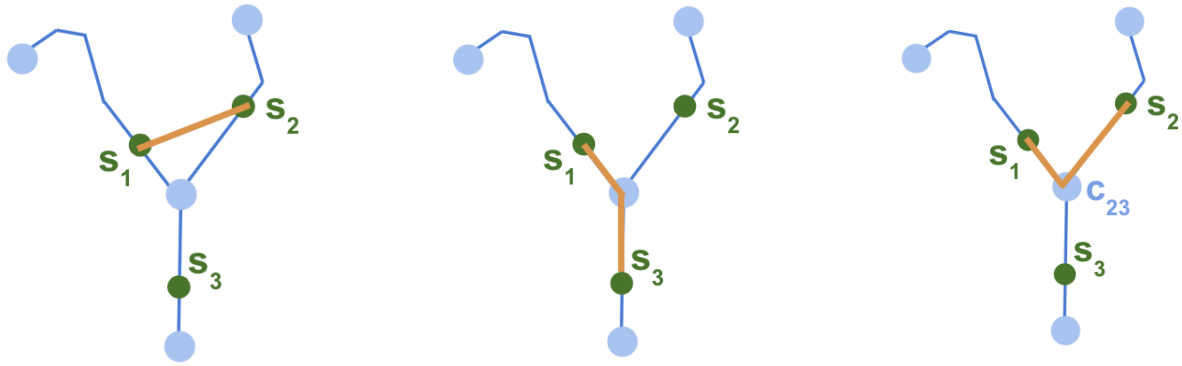
Generally, we may define the stream distance between two point locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ as the total length of the stream reaches that form the path connecting them. It will be helpful to distinguish between pairs of points that are **flow-connected**, meaning that there is a path in the stream network by which water flows from one point to the other as illustrated in Figure 3.3, and pairs of points that are not flow-connected. Each point location $\boldsymbol{s}_i$ can be thought of as a point in $\mathbb{R}^2$, either a pair of projected coordinates or a latitude-longitude pair: $\boldsymbol{s}_i = (x, y) \in \mathbb{R}^2$; this coordinate system is relevant for computing the Euclidean distance between two locations, but less so for stream distance. Alternatively, we can identify each point $\boldsymbol{s}_i$ by a different pair: the binary ID b_i of the reach containing $\boldsymbol{s}_i$ and a nonnegative real number d_i equal to the **upstream distance** of $\boldsymbol{s}_i$, defined to be the length of the flow path from $\boldsymbol{s}_i$ to its stream outlet. In this coordinate system, $\boldsymbol{s}_i = (b, d) \in \mathbb{R}^+ \times 2^\infty$ for all i.

We can then define the **stream distance** between two locations as

$$h(\boldsymbol{s}_i, \boldsymbol{s}_j) = \begin{cases} |U(\boldsymbol{s}_i) - U(\boldsymbol{s}_j)| & \text{if } \boldsymbol{s}_i, \boldsymbol{s}_j \text{ are flow-connected,} \\ U(\boldsymbol{s}_i) + U(\boldsymbol{s}_j) - 2U(\boldsymbol{c}_{ij}) & \text{else,} \end{cases}$$

where $\boldsymbol{c}_{ij}$ is the **nearest common junction** of points $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ (see Figure 3.3c for an example) and $U(\boldsymbol{s}_i)$ is the upstream distance of point $\boldsymbol{s}_i$. For points that are not flow-connected, we will sometimes also need the two distances from each point to their nearest common junction separately. In this case we define a to be the shorter of these two distances and b to be the longer:

$$a(\boldsymbol{s}_i, \boldsymbol{s}_j) := \min\big(U(\boldsymbol{s}_i), U(\boldsymbol{s}_j)\big) - U(\boldsymbol{c}_{ij}), \quad b(\boldsymbol{s}_i, \boldsymbol{s}_j) := \max\big(U(\boldsymbol{s}_i), U(\boldsymbol{s}_j)\big) - U(\boldsymbol{c}_{ij}).$$



(a) Euclidean distance between two points.

(b) Stream distance between flow-connected points.

(c) Stream distance between flow-unconnected points.

Figure 3.3: Euclidean and stream distances are represented here by the lengths of the orange paths.

Note the concise relationship between h, a and b:

$$h(\boldsymbol{s}_i, \boldsymbol{s}_j) = \begin{cases} b(\boldsymbol{s}_i, \boldsymbol{s}_j) - a(\boldsymbol{s}_i, \boldsymbol{s}_j) & \text{if } \boldsymbol{s}_i, \boldsymbol{s}_j \text{ are flow-connected,} \\ b(\boldsymbol{s}_i, \boldsymbol{s}_j) + a(\boldsymbol{s}_i, \boldsymbol{s}_j) & \text{else.} \end{cases}$$

As previously mentioned, Ver Hoef et al. (2006) demonstrate that simply substituting stream distance into covariance functions developed for Euclidean distance can lead to covariance matrices that are not positive definite. Instead, they develop covariance functions of stream distance using the approach from Yaglom (1987) and Ver Hoef and Barry (1996) of convolving moving average functions $g(\cdot \mid \boldsymbol{\theta})$ with a white noise process $W(\cdot)$:

$$Z(s \mid \boldsymbol{\theta}) = \int_{-\infty}^{\infty} g(x - s \mid \boldsymbol{\theta}) dW(x).$$

Ver Hoef and Peterson (2010) develop two classes of covariance functions which they refer to as **tail-up** and **tail-down**. For tail-up covariance functions, the moving average function is non-zero only upstream of the point in question, while for tail-down covariance functions,

the moving average function is non-zero only downstream of the point in question. As shown in Figure 3.4, we can visualize this as a shaded region alongside each stream reach, where the width of the region at each point along the stream is proportional to the magnitude of the moving average function at that point along the reach. Visualized this way, the names tail-up and tail-down correspond to whether the "tails" of the shaded regions point upstream or downstream.

Spatial correlation between two points occurs if and only if their moving average functions overlap:

$$C(s_i, s_j \mid \theta) = \mathrm{Cov}\big(Z(\boldsymbol{s}_i \mid \theta), Z(\boldsymbol{s}_j \mid \theta)\big) = \int_{-\infty}^{\infty} g(x - s_i \mid \boldsymbol{\theta}) g(x - s_j \mid \boldsymbol{\theta}) dx$$

As a result, spatial dependency with tail-up covariance occurs only between flow-connected points, while both flow-connected and flow-unconnected pairs of points can be spatially correlated with tail-down covariance (see Figure 3.4). Tail-up models are therefore useful for modeling waterborne chemicals and small organisms that move passively with the stream flow. For a given stream distance between two points, tail-down correlations cannot be much stronger if the points are flow-connected relative to when they are flow-unconnected. Hence, a tail-down covariance function can capture correlations from fish species that tend to swim upstream, while combining tail-up and tail-down covariance components allows us to better model fish species that swim downstream more often than upstream.

For a given choice of moving average function $g(\cdot \mid \boldsymbol{\theta})$[2], the tail-up covariance between $Z(\boldsymbol{s}_i)$ and $Z(\boldsymbol{s}_j)$ is

$$C(\boldsymbol{s}_i, \boldsymbol{s}_j \mid \boldsymbol{\theta}) = \begin{cases} \int_0^{\infty} g(x \mid \boldsymbol{\theta})^2 dx & h(\boldsymbol{s}_i, \boldsymbol{s}_j) = 0, \\ \int_{h(\boldsymbol{s}_i, \boldsymbol{s}_j)}^{\infty} g(x \mid \boldsymbol{\theta}) g\big(x - h(\boldsymbol{s}_i, \boldsymbol{s}_j) \mid \boldsymbol{\theta}\big) dx & \boldsymbol{s}_i, \boldsymbol{s}_j \text{ are flow-connected (FC)}, \\ 0 & \text{else}, \end{cases}$$

[2]Note that Ver Hoef and Peterson (2010) only consider moving average functions with positive support.

while the tail-down covariance function is

$$C(\boldsymbol{s}_i, \boldsymbol{s}_j \mid \boldsymbol{\theta}) = \begin{cases} \int_{-\infty}^{-h(\boldsymbol{s}_i,\boldsymbol{s}_j)} g(-x \mid \boldsymbol{\theta}) g\big(-x - h(\boldsymbol{s}_i, \boldsymbol{s}_j) \mid \boldsymbol{\theta}\big) dx & \boldsymbol{s}_i, \boldsymbol{s}_j \text{ FC}, \\ \int_{-\infty}^{-b(\boldsymbol{s}_i,\boldsymbol{s}_j)} g(-x \mid \boldsymbol{\theta}) g\Big(-x - \big(b(\boldsymbol{s}_i, \boldsymbol{s}_j) - a(\boldsymbol{s}_i, \boldsymbol{s}_j)\big) \mid \boldsymbol{\theta}\Big) dx & \text{else.} \end{cases}$$

Ver Hoef and Peterson (2010) and Ver Hoef et al. (2019) provide convenient summaries of various moving average functions and their corresponding tail-up and tail-down covariance functions. Here, we provide just two examples of functions that seem to be more common in current practice:

- The moving average function $g(x \mid \boldsymbol{\theta}) = \theta_1 \exp(-x/\theta_2)\mathbb{1}(0 \leq x)$ yields the exponential tail-up and tail-down covariance functions C_u and C_d, respectively:

$$C_u(\boldsymbol{s}_i, \boldsymbol{s}_j \mid \boldsymbol{\theta}) = \begin{cases} \theta_1 \exp(-h/\theta_2) & \boldsymbol{s}_i, \boldsymbol{s}_j \text{ FC}, \\ 0 & \text{else}, \end{cases}$$

$$C_d(\boldsymbol{s}_i, \boldsymbol{s}_j \mid \boldsymbol{\theta}) = \theta_1 \exp(-h/\theta_2).$$

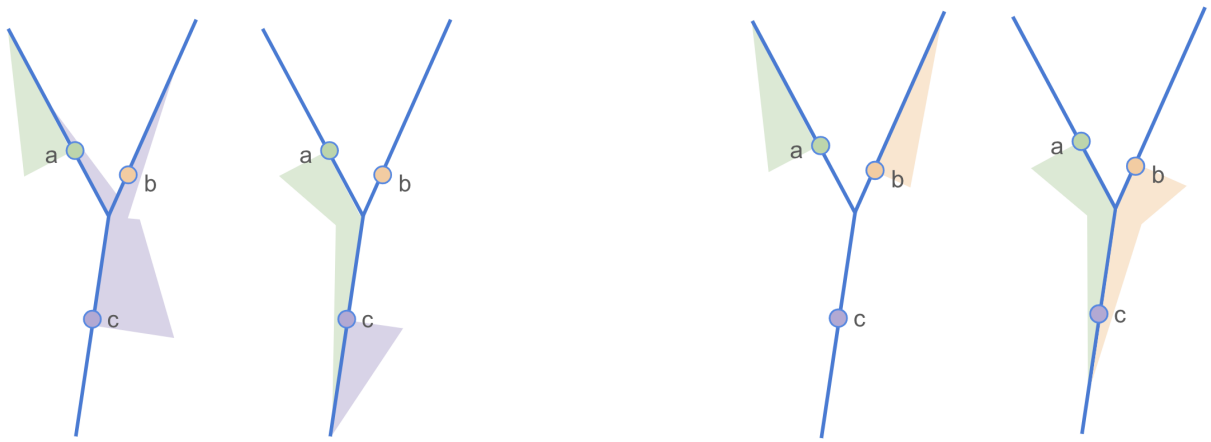
- The moving average function $g(x \mid \boldsymbol{\theta}) = \theta_1 \mathbb{1}(0 \leq x/\theta_2 \leq 1)$ yields the linear-with-sill tail-up and tail-down covariance functions C_u and C_d, respectively:

$$C_u(\boldsymbol{s}_i, \boldsymbol{s}_j \mid \boldsymbol{\theta}) = \theta_1 \left(1 - \frac{h}{\theta_2}\right) \mathbb{1}\left(\frac{h}{\theta_2} \leq 1\right),$$

$$C_d(\boldsymbol{s}_i, \boldsymbol{s}_j \mid \boldsymbol{\theta}) = \begin{cases} \theta_1 \left(1 - \frac{h}{\theta_2}\right) \mathbb{1}\left(\frac{h}{\theta_2} \leq 1\right) & \boldsymbol{s}_i, \boldsymbol{s}_j \text{ FC}, \\ \theta_1 \left(1 - \frac{b}{\theta_2}\right) \mathbb{1}\left(\frac{b}{\theta_2} \leq 1\right) & \text{else.} \end{cases}$$

3.3.3 Spatial weights for tail-up covariance

One consideration for tail-up models is how stream branching should impact the covariance. In Figure 3.3a, suppose that points $\boldsymbol{s_1}$ and $\boldsymbol{s_2}$ are the same distance upstream, but the reach containing $\boldsymbol{s_1}$ is a source reach with no stream network upstream of it, while the reach



(a) Flow-connected points a and c.

(b) Flow-unconnected points a and b.

Figure 3.4: Tail-up (left of each panel) and tail-down (right of each panel) covariances for (a) flow-connected and (b) flow-unconnected points. Adapted from Ver Hoef et al. (2019).

containing $\boldsymbol{s_2}$ has a large fraction of the network upstream of it. Then under the spatial stream model described so far, the variance would be greater at $\boldsymbol{s_2}$ than at $\boldsymbol{s_1}$. To make the covariance function stationary, Ver Hoef et al. (2006) propose weighting the reaches by some measure A of stream size or influence such as flow volume, watershed area, or Shreve's stream order. These weights are computed in three steps:

1. Since confluences of three or more reaches are not permitted in stream network models, a given reach either shares its downstream node with a single other reach or with no reaches; in the latter case, either it is a stream outlet or it flows directly into a single reach directly downstream. The segment proportional influence or **segment PI** of each reach i is

$$w_i = \begin{cases} A_i/(A_i + A_j) & \text{if reaches } i, j \text{ share a downstream node,} \\ 1 & \text{if no other reach shares a downstream node with reach } i. \end{cases}$$

2. For each reach i, let D_i be the set of all reaches downstream of and including reach i. Then the additive function value[3] or **AFV** Ω_i of reach i is defined by

$$\Omega_i = \prod_{k \in D_i} \omega_k.$$

Perhaps another name for this quantity could be the cumulative influence.

3. Finally, the spatial covariance **weight** π_{ij} for any pair of reaches i, j is equal to

$$\pi_{ij} = \begin{cases} 1 & i = j, \\ \sqrt{\Omega_i / \Omega_j} & i, j \text{ FC and } i \text{ upstream of } j, \\ \sqrt{\Omega_j / \Omega_i} & i, j \text{ FC and } j \text{ upstream of } i, \\ 0 & i, j \text{ not FC}, \end{cases}$$

or equivalently (and more conveniently for computation),

$$\pi_{ij} = \begin{cases} 1 & i = j, \\ \sqrt{\min(\Omega_i, \Omega_j) / \max(\Omega_i, \Omega_j)} & i, j \text{ FC}, \\ 0 & i, j \text{ not FC}. \end{cases}$$

Some consequences of these definitions: segment PIs sum to 1 for converging reaches, and $\Omega_i = \omega_i = 1$ for any stream outlet i.

3.3.4 The SSN model

As a result, the overal SSN model takes the following form, modified from 3.5 to add tail-up and/or tail-down stream variance components C_u and C_d in addition to an optional Euclidean

[3]While this may seem to be a misnomer given that the definition involves multiplication instead of addition, note that segment PIs for all reaches converging at a given node sum to 1, and therefore the AFV is also additive: the AFV of a given reach equals the sum of the AFVs of any and all reaches directly upstream. See Ver Hoef and Peterson (2010) for further discussion of the historical origins of this term.

distance variance component C_e:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{f} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(0, \sigma^2 \boldsymbol{I}), \\ \boldsymbol{f} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}, \quad \boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{C}), \text{ where} \\ \boldsymbol{C}_{ij} &= \pi_{ij} C_u(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_u) + C_d(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_d) + C_e(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_e). \end{aligned} \tag{3.15}$$

3.4 Nearest neighbor Gaussian process (NNGP) models

SSNs make remarkable advances in capturing stream-based spatial covariances, but they do not reduce the computational expense of spatial process models. Estimating the parameters of these models requires storing the covariance matrix and computing its inverse and determinant. The full covariance matrices for estimation and prediction with n observation point locations and q prediction point locations are $n \times n$ and $q \times q$, respectively. Computing the full likelihood for estimation requires $O(n^2)$ storage, and computing the inverse and determinant of the covariance matrix using the Cholesky decomposition requires $O(n^3)$ floating point operations (flops). Estimation with these models thus becomes prohibitive on current personal computers for $n \gtrsim 10^4$.

To reduce the computational cost of estimating SSN models, we turn to literature on fast approximations of spatial process models. Many branches of these methods have been developed, but we focus on nearest neighbor Gaussian processes since these turn the approximation into valid generative models in their own right. This key distinction enables not only efficient estimation but also efficient inference and prediction. Section 3.4.1 introduces the idea of the nearest neighbor approximation. Section 3.4.2 summarizes how NNGPs extend Vecchia's approximation to a valid generative model. Section 3.4.3 explains how the current frequentist implementation of NNGP model estimation, inference, and prediction in the `BRISC` R package works and provides computational benefits over the existing Bayesian implementation, the `spNNGP` R package.

3.4.1 Vecchia's approximation

One approach to reducing computational expense is sparse nearest neighbor approximations. The full likelihood is the joint probability of the values of the Gaussian process at all observation and prediction points, and it can be expressed as a product of conditional densities using the general product rule:

$$\begin{aligned} p(\boldsymbol{Y}) =& p\big(Y(\boldsymbol{s}_1)\big) \cdot p\big(Y(\boldsymbol{s}_2) \mid Y(\boldsymbol{s}_1)\big) \cdot p\big(Y(\boldsymbol{s}_3) \mid Y(\boldsymbol{s}_2), Y(\boldsymbol{s}_1)\big) \cdots \\ & p\big(Y(\boldsymbol{s}_n) \mid Y(\boldsymbol{s}_{n-1}), \ldots, Y(\boldsymbol{s}_1)\big) \\ =& p\big(Y(\boldsymbol{s}_1)\big) \prod_{i=2}^{n} p\big(Y(\boldsymbol{s}_i) \mid \boldsymbol{Y}(H(\boldsymbol{s}_i))\big), \end{aligned} \tag{3.16}$$

where $H(\boldsymbol{s}_i) := \{\boldsymbol{s}_{i-1}, \ldots, \boldsymbol{s}_1\}$ is the conditioning set for $\boldsymbol{s}_i$ and $\boldsymbol{Y}(H(\boldsymbol{s}_i))$ is the vector of responses (fish densities, in our case) at the locations in $H(\boldsymbol{s}_i)$. Equation 3.16 holds for any ordering of the locations s_i. Vecchia (1988) proposed approximating the likelihood by conditioning each point on at most some fixed number m of its nearest neighbors, replacing $H(\boldsymbol{s}_i)$ with a subset $M(\boldsymbol{s}_i) \subseteq H(\boldsymbol{s}_i)$ of m or fewer points that are nearest to $\boldsymbol{s}_i$ by Euclidean distance. Specifically, for $i \leq m$, $M(\boldsymbol{s}_i) = H(\boldsymbol{s}_i)$ and $|M(\boldsymbol{s}_i)| = i \leq m$. For $i > m$, $M(\boldsymbol{s}_i)$ is the set of m nearest neighbors by Euclidean distance to $\boldsymbol{s}_i$, and $|M(\boldsymbol{s}_i)| = m$. The resulting likelihood approximation is

$$p(\boldsymbol{Y}) \approx p\big(Y(\boldsymbol{s}_1)\big) \prod_{i=2}^{n} p\big(Y(\boldsymbol{s}_i) \mid \boldsymbol{Y}(M(\boldsymbol{s}_i))\big). \tag{3.17}$$

This approximate likelihood only requires storing and computing with n matrices of size at most $m \times m$, reducing storage from $O(n^2)$ to $O(nm^2)$ and computational cost in flops from $O(n^3)$ to $O(nm^3)$.

3.4.2 Extending Vecchia's approximation to a Gaussian process

A limitation of likelihood approximation methods such as that of Vecchia (1988) in Equation 3.17 above is that they need not extend to arbitrary locations unless they are shown to correspond to some underlying process. The main alternative class of approaches, low rank

models, are still too computationally expensive for large n and perform poorly in the presence of high spatial correlation, both of which are critical for regression on large spatial datasets. However, Datta et al. (2016a) proved that the nearest neighbor approximation of Vecchia (1988) corresponds to a process of its own. Consider a zero-mean Gaussian process such as w in Model 3.5 with covariance function C. Applying Vecchia's approximation in Equation 3.17 to this process yields

$$p(\boldsymbol{w}) \approx p\big(w(\boldsymbol{s}_1)\big) \prod_{i=2}^{n} p\big(w(\boldsymbol{s}_i) \mid \boldsymbol{w}(M(\boldsymbol{s}_i))\big). \tag{3.18}$$

Applying the conditional distribution results from Equations 3.9 and 3.10, we have

$$\begin{aligned} &w(\boldsymbol{s}_i) \mid \boldsymbol{w}(M(\boldsymbol{s}_i)) \sim N(\boldsymbol{a_i}, d_i) \text{ where} \\ &\quad \boldsymbol{a_i} = \boldsymbol{C}\big(M(\boldsymbol{s_i}), M(\boldsymbol{s_i})\big)^{-1} \boldsymbol{C}\big(M(\boldsymbol{s_i}), \boldsymbol{s_i}\big), \\ &\quad d_i = C\big(\boldsymbol{s_i}, \boldsymbol{s_i}\big) - \boldsymbol{C}\big(\boldsymbol{s_i}, M(\boldsymbol{s_i})\big) \boldsymbol{C}\big(M(\boldsymbol{s_i}), M(\boldsymbol{s_i})\big)^{-1} \boldsymbol{C}\big(M(\boldsymbol{s_i}), \boldsymbol{s_i}\big) \\ &\qquad = C\big(\boldsymbol{s_i}, \boldsymbol{s_i}\big) - \boldsymbol{C}\big(\boldsymbol{s_i}, M(\boldsymbol{s_i})\big) \boldsymbol{a_i}. \end{aligned} \tag{3.19}$$

In matrix form, Vecchia's approximation applied to the latent process corresponds to the model

$$\boldsymbol{w} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{\eta}, \quad \boldsymbol{\eta} \sim N(\boldsymbol{0}, \boldsymbol{D}),$$

where $\boldsymbol{A}$ is the lower triangular matrix formed by stacking the $\boldsymbol{a_i}$ for all locations and $\boldsymbol{D} = \text{diag}(d_1, ..., d_n)$.[4] Therefore,

$$\boldsymbol{w} = (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{\eta} \sim N\left(\boldsymbol{0}, (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{D} (\boldsymbol{I} - \boldsymbol{A})^{-T}\right).$$

Thus, the Vecchia approximation yields a new Gaussian process model with covariance matrix $\tilde{\boldsymbol{C}} := (\boldsymbol{I} - \boldsymbol{A})^{-1} \boldsymbol{D} (\boldsymbol{I} - \boldsymbol{A})^{-T}$ which approximates the original Gaussian process. The Cholesky factor of $\tilde{\boldsymbol{C}}$, $\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}(\boldsymbol{I} - \boldsymbol{A})$, requires only $O(nm^3)$ flops and $O(nm^2)$ storage, compared to $O(n^3)$ flops and $O(n^2)$ storage for the Cholesky factor of the covariance matrix for the full

[4] Our use of $\boldsymbol{A}$ and $\boldsymbol{D}$ here corresponds to that of Finley et al. (2019). Datta et al. (2016a) and Datta (2022) refer to matrices $\boldsymbol{A}$ and $\boldsymbol{D}$ as $\boldsymbol{B}$ and $\boldsymbol{F}$, respectively. Saha and Datta (2018) use the same notation for $\boldsymbol{A}$ as Finley et al. (2019), but their $\boldsymbol{D}$ is the inverse of $\boldsymbol{D}$ in Finley et al. (2019).

original process. Therefore, $\tilde{\boldsymbol{C}}$ is not sparse, but its inverse is sparse, which is the essential property for efficient parameter estimation.

Alternatively, if we are only interested in modeling the response process itself and not the latent process, we can apply Vecchia's approximation directly to the response $\boldsymbol{Y}$ instead of the latent process $\boldsymbol{w}$. This precludes prediction of the latent process, but it reduces the computational burden. In this case, the covariance of the nearest neighbor Gaussian process approximates $\boldsymbol{\Sigma}$ instead of $\boldsymbol{C}$, so $\tilde{\boldsymbol{\Sigma}} := (\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{D}(\boldsymbol{I} - \boldsymbol{A})^{-T}$. We obtain the model by replacing the latent process $\boldsymbol{w}$ with the response process $\boldsymbol{Y}$ and replacing the latent covariance function C with the response covariance function Σ throughout Equations 3.18 and 3.19.

3.4.3 Fast estimation and inference with `BRISC`

Estimation and inference for NNGP models can be performed using Bayesian or frequentist methods. The `spNNGP` R package provides full posterior distributions using MCMC, while the `BRISC` R package estimates parameter values using maximum likelihood estimation and estimates their confidence intervals using a bootstrap since MLE asymptotic guarantees are limited in the infilling asymptotic paradigm typical of many spatial problems. Saha and Datta (2018) demonstrate that `BRISC` is much faster than `spNNGP` for both estimation and prediction while providing comparable accuracy. Therefore, we have opted to adapt `BRISC` for our implementation of S3N models.

Confidence intervals for the fixed effect parameters can be obtained for free with model estimation as $\left(X^T\hat{\tilde{\Sigma}}^{-1}X\right)^{-1}$. However, the Bootstrap for Rapid Inference on Spatial Covariances—or BRISC (Saha and Datta, 2018), after which the `BRISC` package is named—allows for the estimation of covariance parameter confidence intervals as well, a feature that is not currently available in SSNs. BRISC is based on a parametric bootstrap proposed by Olea and Pardo-Igúzquiza (2011), in which the data are decorrelated before sampling and recorrelated afterward by multiplication with either the Cholesky factor or its inverse. This bootstrap procedure requires $O(n^3)$ operations to compute the Cholesky factors of both the covari-

ance and the precision matrices. However, the NNGP offers a cost-effective approximation: BRISC applies the NNGP to the response process and simply replaces the full Cholesky factor for $\boldsymbol{\Sigma}$ with the NNGP Cholesky factor $\tilde{\boldsymbol{L}} = \boldsymbol{D}^{-1/2}(\boldsymbol{I}-\boldsymbol{A}) = (\boldsymbol{I}-\boldsymbol{A})\boldsymbol{D}^{-1/2}$ introduced previously, where $\tilde{\boldsymbol{\Sigma}} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{L}}^T$:

1. Decorrelate the data using $\tilde{\boldsymbol{L}} = (\boldsymbol{I}-\boldsymbol{A})\boldsymbol{D}^{-1/2}$:

$$\boldsymbol{Z}^* = \tilde{\boldsymbol{L}}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}).$$

2. For $i = 1, ..., B$:

   (a) Generate a bootstrap sample $\boldsymbol{Z}_{\boldsymbol{i}}^*$ from $\boldsymbol{Z}^*$.

   (b) Recorrelate the data using $\tilde{\boldsymbol{L}}^{-1} = (\boldsymbol{I}-\boldsymbol{A})^{-1}\boldsymbol{D}^{1/2}$:

$$\boldsymbol{Y}_{\boldsymbol{i}}^{\boldsymbol{r}} = \tilde{\boldsymbol{L}}^{-1}\boldsymbol{Z}_{\boldsymbol{i}}^* + \boldsymbol{X}\boldsymbol{\beta}.$$

   Since $\tilde{\boldsymbol{L}}^{-1}$ is not readily available and would require additional computation, Saha and Datta (2018) propose and use an $O(n)$-time algorithm to recorrelate the data.

   (c) Estimate covariance parameters using the bootstrap data $\boldsymbol{Y}_{\boldsymbol{i}}^{\boldsymbol{r}}$.

3. Estimate the standard errors of the parameters as the standard deviations of the parameter estimates across all B bootstrap samples.

The implementation of the `BRISC` package further contributes to the computational cost savings, since it is implemented primarily in C and uses LAPACK for efficient matrix and vector computation. NNGPs and BRISC dramatically improve the computational scalability of Gaussian process models, but to our knowledge they have not yet been implemented for distance measures other than Euclidean distance.

3.5 Stream network preprocessing

So far we have examined statistical models and methods for making estimation, inference and prediction computationally feasible. However, preprocessing is also an expensive component

of SSNs. Both implementations of the SSN preprocessing—as the `STARS` ArcGIS toolkit in 2014, which consists of a set of Python scripts, and as the `SSNbler` R package in 2024—are a computational bottleneck in the application of SSN models as network size increases. In `STARS`, the first preprocessing step of building the stream network includes a nested loop over reaches that dominates the runtime for this step, rendering it $O(r^2)$ where r is the number of reaches. There are a number of other unnecessary loops and steps that increase runtime even if the nested loop were vectorized. `SSNbler` computes pairwise Euclidean distances between all upstream nodes and all downstream nodes in order to identify which stream reaches are connected, which makes the computational cost at least $O(r^2)$. Therefore, a necessary step in making SSNs scalable is identifying the fundamentally required preprocessing steps and eliminating as many additional expensive steps as possible. In this section, we reconsider SSN preprocessing in this light.

Fitting and using models that capture these spatial stream correlations requires substantial preprocessing to define and compute the relevant stream network in order to compute the pairwise stream distances between pairs of prediction/observation points. While computing the Euclidean distance between two points requires only their Cartesian coordinates, computing the stream distance between two points requires knowledge of stream network structure between the two points: does water flow directly from one point to another, or does water flow from both points to meet at some common junction further downstream? Computing the lengths of these paths requires knowing how different segments of the stream network are connected. In other words, we need to know the adjacency matrix of these stream segments.

Computing the stream segment adjacency matrix requires comparing the downstream node of each segment with the upstream nodes of other segments. This involves uniquely identifying the reach upstream and downstream nodes as entities distinct from the segments themselves, mapping reaches to their upstream and downstream nodes, and then mapping downstream nodes of each segment to upstream nodes of other segments. We also need to know the length of each segment and where each point is located relative to the endpoints

of the segment containing it in order to compute the distance between points.

In summary, these are the key preprocessing steps necessary to compute the pairwise stream distances among all pairs of observation and/or prediction points and to fit an SSN:

1. Build the stream network:
   (a) Uniquely identify all upstream and downstream endpoints (nodes) of the reaches.
   (b) Map reaches to their upstream and downstream nodes.
   (c) Compare the downstream node of each reach to the upstream nodes of all other reaches to identify the direct downstream neighbors of each reach.
   (d) Compute the stream reach adjacency matrix.
   (e) Identify stream outlets, which will be the starting points for computing upstream distances and spatial covariance weights throughout the network.
2. Compute the upstream distances of the reaches and points:
   (a) Compute the upstream distance of each reach i as the upstream distance of its downstream node by adding together the lengths of the reaches that connect reach i to the stream outlet node.
   (b) Compute the upstream distance of each point as the upstream distance of the reach i containing it plus the additional distance between the point and the downstream node of reach i.
3. If using a tail-up covariance function, compute the additive function values necessary to later compute the spatial weights:
   (a) Compute the segment proportional influence (segment PI) for each reach.
   (b) Compute the AFV for each reach as the product of the relevant segment PIs.

3.6 Our approach: Scalable spatial stream network (S3N) models

We combine SSNs, NNGPs, BRISC, and more efficient preprocessing to produce a scalable spatial stream network (S3N) model implementation. We reduce the computational cost of fitting and using SSNs by approximating them with the corresponding NNGPs, fitting them with maximum likelihood estimation, and using BRISC for inference. Whereas Vecchia (1988) and Saha and Datta (2018) choose the neighbor sets $M(s_i)$ to be the m nearest neighbors of s_i among $s_1, ..., s_{i-1}$ with respect to Euclidean distance, we instead choose these neighbor sets based on stream distance. Our estimation and prediction code is built upon the existing `BRISC` R package, which uses maximum likelihood and BRISC for estimation and inference on NNGP models.

The S3N model we propose is a combination of the SSN and NNGP models, applying the NNGP to the response process:

$$\begin{aligned}
\boldsymbol{Y} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}, \quad \boldsymbol{w} \sim N\left(0, \tilde{\boldsymbol{\Sigma}}\right), \\
w(\boldsymbol{s}_i) \mid \boldsymbol{w}(M(\boldsymbol{s}_i)) &\sim N(\boldsymbol{a_i}, d_i), \\
\boldsymbol{a_i} &= \boldsymbol{\Sigma}\big(M(\boldsymbol{s_i}), M(\boldsymbol{s_i})\big)^{-1}\boldsymbol{\Sigma}\big(M(\boldsymbol{s_i}), \boldsymbol{s_i}\big), \\
d_i &= \Sigma\big(\boldsymbol{s_i}, \boldsymbol{s_i}\big) - \boldsymbol{\Sigma}\big(\boldsymbol{s_i}, M(\boldsymbol{s_i})\big)\boldsymbol{a_i}, \\
\Sigma\big(\boldsymbol{s_i}, \boldsymbol{s_j}\big) &= \pi_{ij} C_u(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_u) + C_d(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_d) + C_e(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_e) + \tau^2\delta_{ij}.
\end{aligned} \tag{3.20}$$

For now, we focus on the tail-up component of the latent covariance since this is likely to capture most of the spatial correlation not accounted for through covariates:

$$C_d(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_d) := 0, \quad C_e(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_e) := 0 \quad \text{for all } i, j.$$

We reduce the computational cost of SSN preprocessing by implementing the fundamental preprocessing steps we identified above with minimal overhead. We use the coordinate pair of each upstream and downstream node, pasted together as a string, as the unique identifier of that node. We compute the stream reach adjacency matrix using the `igraph` R package and avoid computing pairwise Euclidean or stream distances until they are needed for estimation.

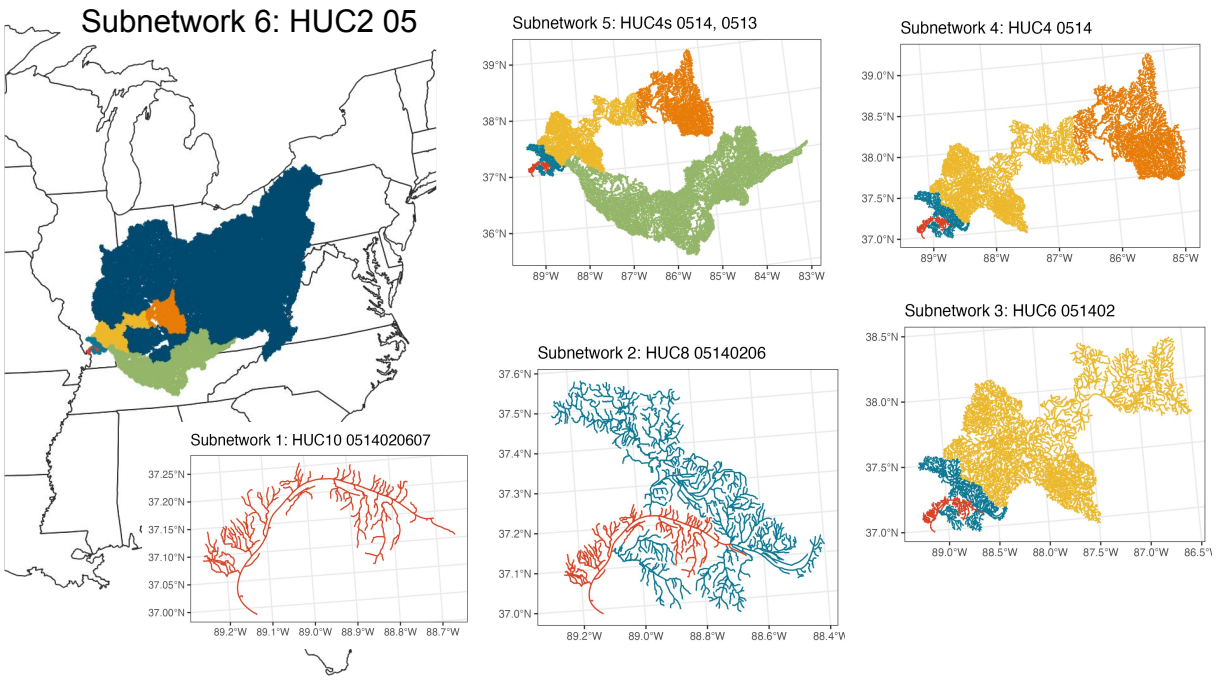


Figure 3.5: The six subnetworks of the Ohio River Basin used for benchmarking. Network i is a subnetwork of Network $i+1$ for $i = 1, ..., 5$. Networks 1-3 were also used for model validation.

For rowwise computations, we use vectorized R functions instead of looping and we avoid creating additional objects by making much of this additional stream network information part of the streams data frame itself.

3.7 Evaluation methods

We present some initial benchmarking and validation results of S3Ns against SSNs using both existing implementations of SSN preprocessing, `STARS` and `SSNbler`. Since `STARS` is implemented in ArcGIS, we benchmarked it by adding timing steps to the Python scripts that constitute the `STARS` toolkit. Since `STARS` is less computationally efficient than the newer `SSNbler`, we limit our benchmarking of `STARS` to the first preprocessing step—building the

		# Points		# Replications	
Network	# Reaches	Prediction	Observation	S3N	SSN
1	284	283	142	50	50
2	1,273	1,267	634	50	50
3	7,146	7,123	3,562	50	50
4	11,540	11,515	5,758	10	10
5	30,748	30,698	10,000	10	2
6	169,092	168,875	10,000	10	NA

Table 3.1: Description of benchmark networks and the number of replications used to benchmark S3N and `SSNbler` on each network. Only 2 replications were used on Network 5 because of the longer runtime required by `SSNbler`. `SSNbler` crashed on Network 6.

stream network—and to a single network. For `SSNbler` versus S3N, we present benchmarking and validation results for preprocessing and estimation in RStudio using the R function `Sys.time()`.

For benchmarking and validating S3N versus `SSNbler`, we chose six nested stream networks of increasing size for the benchmarking, ranging from 284 to 169,092 stream reaches; these networks are shown in Figure 3.5 and the details of these networks are summarized in Table 3.1. For prediction points, we used the midpoint of every reach in the network except for any short reaches that were added to correct topological issues in the network, making the number q of prediction points for each network slightly less than the number of reaches. For observation locations, we drew a random sample of size $n = \min(\text{ceiling}(q/2), 10000)$ of the prediction points. For each network, we simulated responses at the observation points from the following SSN model with exponential tail-up covariance, an intercept and a single

covariate:

$$\begin{aligned} \boldsymbol{Y} &= \boldsymbol{f} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N(0, \tau^2 \boldsymbol{I}), \\ \boldsymbol{f} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}, \quad \boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{C}), \\ \boldsymbol{C}_{ij} &= \pi_{ij}\sigma^2 \exp(-h_{ij}/\lambda). \end{aligned} \tag{3.21}$$

The fixed effect parameters for this model will be referred to as β_1 and β_2, while τ^2 is the independent variance, σ^2 is the exponential tail-up spatial variance scaling parameter, and λ is the range parameter representing the characteristic distance for spatial correlations. We simulated from this model with $\beta_1 = 0.5$, $\beta_2 = -44$, $\tau^2 = \sigma^2 = 5$, and $\lambda = 0.1$.

For both S3N and SSN, we simulated 50 datasets from this model for each of the three smallest networks and 10 for Network 4, and we benchmarked each model once on each dataset; we refer to these simulated datasets as replications. For Network 5, we used 10 replications for S3N but only ran SSN on the first two of those replications due to the longer runtime and higher memory requirements. `SSNbler` crashed on Network 6, but we benchmarked S3N with 10 replications. For the three smallest networks, we completed S3N and SSN preprocessing, pairwise distances, and estimation on every replication, while for the three largest networks we performed only the preprocessing steps.

We have made an effort to fairly compare S3N and `SSNbler/SSN` by matching steps with similar purposes, but we acknowledge that some of the steps of one method may include tasks that appear elsewhere in the other code, or implementation-specific tasks that appear nowhere in the other code. Table 3.2 shows the model-to-model mapping we use for benchmarking. We could instead extract more specific tasks from each model, but this would require recoding some of the implementations (for instance, extracting just the components of a function that do particular tasks even if that task is not currently available as its own function), and since our main purpose is to see what the practical computational expense of each approach is, it seems more relevant to leave the implementations intact and accept some compromise in matching steps between the two models. We have experimented with both approaches and found that they ultimately lead to similar findings.

In Table 3.2, note that SSNs currently only provide confidence intervals (CIs) for fixed

Task	S3N functions	SSN functions
Build stream network	`configure_stream_network()` - `add_upstream_dnstream_nodes()` - `get_stream_graphs()` - `add_stream_source_outlet_component()`	`lines_to_lsn()`
Compute stream updist	`compute_stream_updist_vars()`	`updist_edges()` `afv_edges()`

Compute site updist	`prep_to_compute_pwdist()`	`sites_to_lsn()`
		`updist_sites()`
		`afv_sites()`

Assemble SSN object	NA	`ssn_assemble()`
Compute pairwise distances	`compute_pwdists_preds_obs(obs_only = T)`	`ssn_create_distmat()`
Estimate model parameters	`BRISC_estimation_stream()`	`ssn_lm()`
Estimate CIs	`BRISC_bootstrap()`	NA

Table 3.2: Mapping S3N steps to SSN steps for benchmarking. Functions indented with a dash (-) are subcomponents of the function they are nested under.

effects, while S3Ns offer a bootstrap to provide CIs for covariance parameters as well.

3.8 Results

3.8.1 Preprocessing with S3N versus STARS

The Polyline to Landscape tool (hereafter PTL) of the STARS toolkit builds the stream network, which is the first preprocessing step as identified in Section 3.5. The runtime of PTL on the Ohio River Basin stream network (Network 6 of the benchmarking networks) is

PTL Step	Pre. Step	Runtime PTL	Runtime S3N
Building Hydro Relationships	1a, 1b	24 min	1 min
Building Landscape Network Relationships	1c, 1d, 1e	3.3 hrs	0.5 sec
Sorting Relationship Table Downstream	NA	2 min	NA
Creating Landscape Network Features Classes	NA	17 min	NA

Table 3.3: Benchmarking results for the steps of the Polyline to Landscape (PTL) tool of the STARS toolkit, the preprocessing (Pre.) steps they correspond to as they are enumerated in Section 3.5, and the runtimes of S3N for those corresponding preprocessing steps. The fourth PTL step is required for their particular implementation but does not accomplish fundamentally required preprocessing steps, and there is no corresponding step in S3N; hence the preprocessing step and S3N runtime columns are NA for the last row of the table.

approximately 4 hours. PTL has four main steps, whose names and runtimes are summarized in Table 3.3 along with the runtimes of the corresponding steps in S3N.

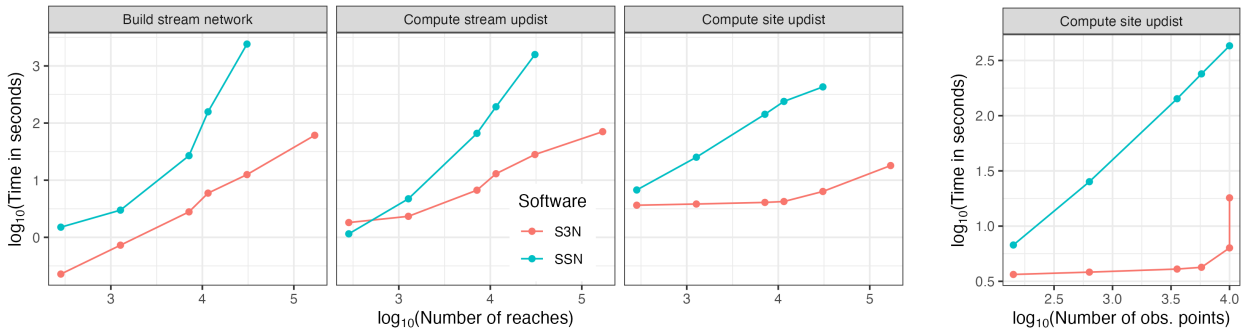
The first PTL step, called building hydro relationships, loops over stream reaches to extract the upstream and downstream nodes, adds these points with their coordinates to nodexy if they are not yet in the list, and creates a list of upstream nodes and a list of downstream nodes in the order in which the reaches are looped over, so that the ith upstream and downstream nodes correspond to the ith reach. This accomplishes preprocessing steps 1a and 1b as outlined in Section 3.5. This step took 24 minutes on the Ohio River Basin network (approximately Network 6). The analogous step in S3N is `add_upstream_dnstream_nodes()`, which takes approximately one minute (50-70 seconds) on the same network.

The second PTL step, building landscape network relationships, is the rate-limiting step. PTL loops over the list of downstream nodes which are one-to-one with the list of reaches,

adds a row to a node relationships table to represent this reach and its upstream and downstream nodes, and then enters a second loop over all upstream nodes to determine which if any other reaches are directly downstream of the one in question. If so, then PTL adds an entry for each of these reaches to a relationships table. In a separate loop over reaches, the code determines which ones are sink reaches and adds them to a sink list. This accomplishes preprocessing steps 1c through 1e as outlined in Section 3.5. This step took 3.3 hours on the Ohio River Basin network. The analogous steps in S3N are `get_stream_graphs()` and `add_stream_source_outlet_component()`, which together take 0.4-0.8 seconds.

The third PTL step, sorting relationship table downstream, starts from each sink reach and works its way upstream in a loop, finding any and all reaches upstream of the current reach, adding those and the current reach as upstream and downstream nodes in new lists, deleting them from the old feature lists to reduce search time. PTL then loops through the network again to flip the order of these lists so that they are ordered downstream instead of upstream. This step is unnecessary once the stream adjacency matrix is computed, so there is no corresponding step in S3N. Instead, S3N starts at each sink and works its way upstream only when computing upstream distances and additive function values, and it computes them all in the same procedure. This step happens later in STARS preprocessing and we did not end up timing it because we started developing our own code once we knew that the earlier step of building the stream network was prohibitive.

The fourth PTL step, creating landscape network features classes, creates the points layer, copies various objects to the geodatabase, and stores and reformats results for their landscape network (LSN) object. These steps may be required for their particular implementation but they are not fundamentally required preprocessing steps, and there is no corresponding step in S3N. S3N similarly has its own implementation-specific additional steps which have no analogue in STARS/SSN, but the total runtime is much less than STARS/SSN.



(a) (b)

Figure 3.6: Benchmarking results for S3N and SSN preprocessing. Note that for the largest network size (number of reaches) tested here, `SSNbler` failed while building the stream network and is therefore not shown here. Runtimes for computing site upstream variables depend not only on (a) the number of reaches but (b) the number of observation sites and are therefore shown as a function of each of these network properties.

3.8.2 Preprocessing with S3N versus `SSNbler`

Figure 3.6 and Table C.1 summarize the benchmarking results between S3N and `SSNbler`. Running `SSNbler` on Network 5 required increasing the memory limit in RStudio, and `SSNbler` failed on Network 6 due to integer overflow while computing pairwise Euclidean distances between all upstream nodes and all downstream nodes in order to identify which stream reaches are connected; while this is not a fundamental preprocessing step, it is part of the implementation of `SSNbler`. On Network 5, the runtime for each of the three preprocessing steps was greater for `SSNbler` than S3N by a factor of 50-200 (Table C.1). Additionally, SSN required over 48 GB memory while preprocessing Network 5. The runtime for building the stream network with the `SSNbler` code appears to grow at a rate of at least $O(r^3)$ where r is the number of reaches. By contrast, building the stream network with S3N is approximately linear in r.

On Network 3, the largest model for which we benchmarked estimation as well as preprocessing, estimation required only 0.83 seconds for S3N compared to 912.45 seconds (15.3 minutes) with SSN. The total computation time including preprocessing, computing pairwise distances, and estimating the model was 2.02 minutes for S3N, of which 1.78 minutes was devoted to computing pairwise distances, and 19.7 minutes for SSN. Computing pairwise distances for S3N can be made more efficient by rewriting the code in base R or C and by using a smart search for nearest neighbors; see Section 3.9 for further details.

Figures 3.7 and C.1 compare S3N and SSN parameter estimates on Networks 1-3 against each other and against the true values, showing strong mutual agreement. The patterns are similar across the networks, so we show the results for Network 3 in this chapter (Figure 3.7) and reserve the figures for the two smaller networks for the appendix (Figure C.1). Both models are accurate and seem to be reasonably unbiased. Both models improve with increasing network size, as expected since we simulated more observations for the larger networks. Interestingly, S3N generally recovers the true stream covariance parameters better than SSN even though the data were simulated from a full SSN, not a nearest neighbor process. SSN tends to have a longer tail than S3N for these parameters, perhaps due to an accumulation of precision errors.

3.9 Discussion

This chapter develops a scalable spatial stream network (S3N) model that enables estimation, inference, and prediction on a larger scale than previously possible. S3Ns combine spatial stream networks (SSNs), nearest neighbor Gaussian processes (NNGPs), the bootstrap for rapid inference on spatial covariances (BRISC), and efficient preprocessing. SSNs provide the theoretical foundation and key implementation insights for handling spatial stream correlations, NNGPs and BRISC enable scalable estimation and prediction, and our own reimplementation of SSN preprocessing in R makes the preprocessing feasible for large areas. We demonstrate through benchmarking and some analysis of computational complexity that the S3N pipeline of preprocessing, estimation and prediction is feasible for the entire Ohio River

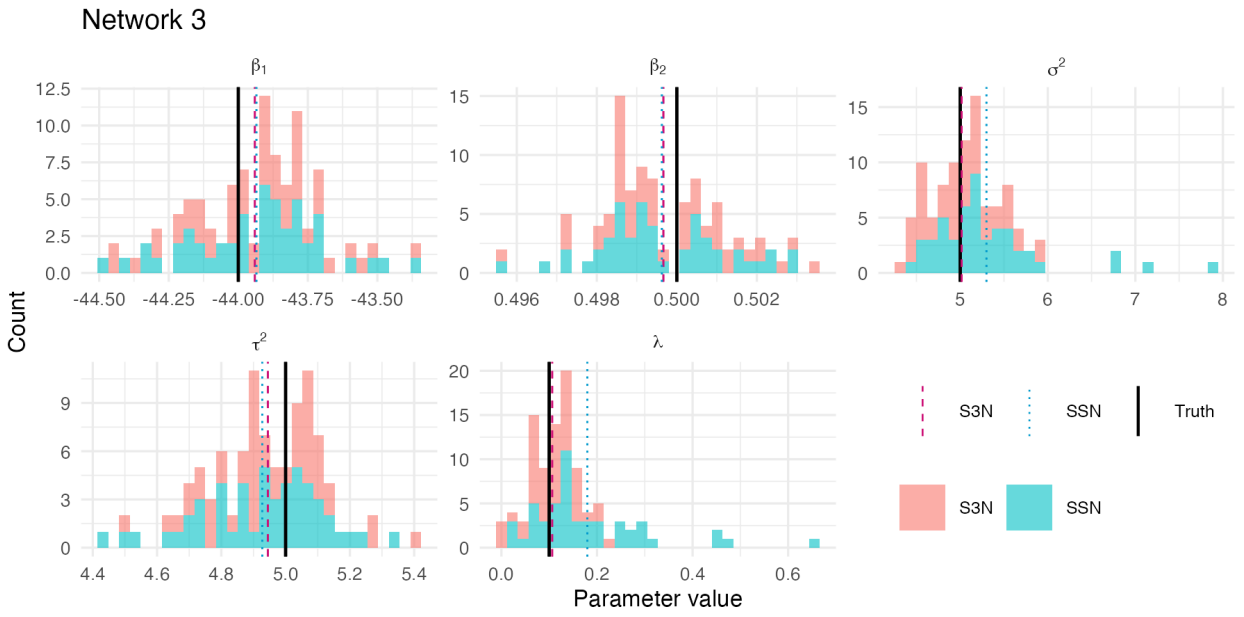


Figure 3.7: Model validation on Network 3. The solid black line indicates the true parameter values while the dashed and dotted lines represent the S3N and SSN averages, respectively, over 50 simulations.

Basin and even larger regions not yet attempted. Validation results indicate S3N correctly recovers parameter values, perhaps even with less bias and variance than SSN for the spatial stream covariance parameters.

Note that the computational cost of SSNs and S3Ns is impacted by the number of reaches in the stream network, the depth and shape of the stream network, and the number of observation points. Fitting these models to a stream network that covers a larger area but consists of fewer reaches and fewer observations will likely be less computationally intensive, not more.

The S3N implementation can be made more flexible to handle a wider variety of data sources. Currently the S3N code assumes there is at most one prediction point and one observation point per reach, that the observation points coincide with the prediction points, and that points are already snapped to the stream network, meaning that the points are

exactly on the stream reaches. These assumptions are reasonable when working with the National Stream Internet flowlines and prediction points as in this chapter and the next, but often do not apply for other data sources. In practice, observation points do not occur exactly on stream flowlines either because they are recorded on the banks of streams or because of geolocation precision errors in the flowlines and/or the observation points. Based on our experience in developing these models so far, the modifications to avoid making these three assumptions should not appreciably increase the computational time.

On the other hand, the current implementation of SSN handles the prediction points and observation points as separate layers and does not seem to utilize any overlap between them to make preprocessing more efficient. For instance, if there are points that are both observation and prediction locations, the preprocessing for these points is performed twice: adding these points to the network, computing upstream distances and AFVs, and computing pairwise distances.

The computational bottleneck in the current S3N implementation is computing the pairwise distances among observation points for estimation and prediction and between prediction and observation points for prediction. This could be made more efficient by using depth-first and breadth-first searches to first compute distances for the closest points and stop once the desired number of neighbors have been identified. Additionally, we plan to rewrite the code to minimize or eliminate use of tidyverse in favor of base R and C.

Future work includes implementing tail-down and Euclidean covariance components, which require additional considerations. Adding tail-down covariance requires computing distances $a(\boldsymbol{s_i}, \boldsymbol{s_j})$ and $b(\boldsymbol{s_i}, \boldsymbol{s_j})$ separately for pairs of flow-unconnected points. When both tail-up or tail-down and Euclidean covariances are included in the model, S3N models should ideally account for both the nearest neighbors with respect to Euclidean distance, $M_e(\boldsymbol{s_i})$, and those with respect to stream distance, $M_s(\boldsymbol{s_i})$. We can let $M(\boldsymbol{s_i}) = M_e(\boldsymbol{s_i}) \cup M_s(\boldsymbol{s_i})$. Initially, we could choose the m_e nearest neighbors with respect to Euclidean distance and the m_s nearest neighbors with respect to stream distance. An alternative is to choose the m_e nearest neighbors with respect to Euclidean distance and all observed locations that fall

within some radius r_s with respect to stream distance.

The current S3N implementation applies the nearest neighbor process to the responses, not the latent process. This functionality could easily be added. While the computational savings over the full SSN model will not be great for the latent process as for the response process, this option would enable users to make predictions at the latent level instead of the response level if they prefer. Another direction for future work is to extend S3Ns to non-Gaussian distributions such as binomial, Poisson, zero-inflated Poisson, and negative binomial. This would better adapt S3Ns for modeling ecological count data.

It would also be valuable to consider which ordering to use in defining nearest neighbors for the NNGP. The quality of the Vecchia nearest-neighbor approximation depends on the order or index assigned to the locations (Guinness, 2018). The most common ordering method indexes the points by coordinates, such as ordering the points from east to west or in order by the sum of their coordinates in $\mathbb{R}^2$ as measured from some reference point. Alternatives include maximum-minimum distance, middle-out, and random. In maximum-minimum distance, the first point s_1 is chosen to be the location closest to the center of the points by averaging the coordinates or using some other measure of centrality, and each subsequent point s_j is chosen such that j maximizes the minimum distance between s_j and one of the previous points s_k for $k < j$.

Initially, we could use the following procedure for ordering the locations: (1) choose a component uniformly at random, (2) draw a permutation of the locations in this component uniformly at random, and (3) repeat steps 1 and 2 until all the locations have been ordered. Alternatively, Guinness (2018) find that the middle-out method appears to perform best (lowest KL divergence) in both three and four dimensions with fewer than 60 neighbors, suggesting this might be a good initial choice. We could choose the component ordering using a middle-out method based on Euclidean distance, starting with the component whose "center" by some measure of centrality is closest to the center of all the locations, then choosing the sequence of components using the middle-out method also based on the centers of the components. The ordering of locations within stream network components could

be based on ascending stream order, perhaps analogous to the sorted coordinates method for Euclidean distance, or a stream analogue to MMD or middle-out instead of drawing a permutation uniformly at random. Additionally, we could use two different orderings for the Euclidean and stream-based covariance components if the likelihood factors into a Euclidean-based component and a stream-based component. Future work should compare a few methods empirically and also explore theoretical arguments for using one method or another.

Chapter 4

LARGE-SCALE FRESHWATER FISH POPULATION SIZE ESTIMATES USING AN S3N MODEL

This chapter applies the scalable stream network (S3N) model approach from the previous chapter to obtain what are, to our knowledge, the first population size estimates for 309 species of fish across the entire Ohio River Basin. This demonstrates that S3N models enable science on a scale not previously possible. Toward this aim, this chapter proposes a framework for efficiently applying S3N models to estimate geographical distributions and regional estimates separately for many species at once. The code for this chapter is available at `http://github.com/jpierkunke/S3N_thesis`, and we will soon release the `S3N` R package to help make national and regional fish population size estimation more broadly accessible.

4.1 Introduction

Globally, freshwater bodies account for only 1% of Earth's surface but 51% of known fish species (Hughes, 2021). Freshwater biodiversity is critical for ecosystem function, individual livelihoods, commercial income, outdoor recreation, and fresh water potability, among other key benefits (Lynch et al., 2022). Yet almost one-third of freshwater fish are threatened with extinction (Tickner et al., 2020). The literature documents 12 new or worsened emerging threats to freshwater biodiversity in the last two decades, including climate change, harmful algal blooms, and microplastic pollution, and Reid et al. (2019) find that several of these stressors are still poorly understood.

A fundamental knowledge gap which is critical for addressing these questions is the geographic distribution of freshwater fish species. How does fish population size vary across species and across space, and how do land use, temperature, water quality and other envi-

ronmental drivers shape species occurrence? Teams of scientists from many organizations conduct surveys to count the number of fish in segments of river at a given time. This data can then be modeled using a spatial process to scale up to geographic distributions, make regional estimates, and examine the relationship between fish abundance and environmental variables.

The S3N models we develop in the previous chapter enable us to estimate fish populations over wider regions and for more species than was previously feasible. In this chapter, we develop an efficient framework for making national and regional fish population size estimates by species for many species, and we produce estimates for over 300 species of fish across the Ohio River Basin, a dense river network across an area of approximately 200,000 sq km, using observations at approximately 9,000 locations. Producing these regional estimates and interpreting model parameters involves model estimation, inference, and prediction. Section 4.2 describes the various data sources involved in this application. Section 4.3 specifies the S3N model and how we use it to make these regional fish population estimates. Section 4.4 summarizes the resulting estimates of fish population sizes and model parameters, as well as runtimes and diagnostics of model performance. Section 4.5 concludes with a discussion.

4.2 Data

4.2.1 Environmental covariate data

We use nine environmental covariates to capture topography, land use, hydrology, and temperature, all key drivers of fish abundance: mean elevation; total basin area; percentage of land with low, medium, or high development; percentage land used for agriculture (pasture/hay or crops); total annual runoff; baseflow index; mean annual temperature; hydrological alteration index; and floodplain integrity. We use hydrological alteration index data and floodplain integrity data from McManamay et al. (2022) and Morrison et al. (2023), respectively, and all other variables come from NHDPlus version 2.1 Wieczorek et al. (2018). In our region of interest, there were 806 missing values for hydrological alteration index and 299

for floodplain integrity, representing 0.48% and 0.18% of the stream network. We imputed these values by using the value from the nearest stream reach whose value was not missing.

4.2.2 Stream network data

The National Stream Internet (NSI) provides flowline and prediction point shapefiles for streams across the United States (Nagel et al., 2017). The flowlines are high-resolution digital maps of streams that were developed for use with spatial stream network models to ensure that topological assumptions of the spatial stream network models are met. The mapped streams are subdivided into segments called reaches, and these reaches have a unique identifier called a COMID, which we believe is an abbreviation of Component ID. Some of the environmental data are matched to stream locations by a different identifier, the hydrological unit code (HUC), so we match COMIDs to HUCs using a publicly available database.

This study focuses on the Ohio River Basin, also known as Region 5, which has 169,463 stream reaches in the NSI database. Figure 4.1 plots the flowlines on a map of the Eastern portion of the United States; the river network is dense enough at this scale that it appears as a nearly solid object. The stream outlet is on the southwest side of the network.

Figure 4.1: Region 5 NSI flowlines.

Of the 169,463 stream reaches in the NSI database for this region, 217 are short 20- to 40-meter segments that were added as part of their procedure to address gaps, complex confluences, and other topological errors or complexities incompatible with SSN models. These segments are assigned the same COMID as the neighboring reach directly upstream, which makes the COMID no longer a unique reach identifier. Therefore they add an additional field for each reach called the dupli-

cate COMID to indicate whether the reach was added
during this procedure (`DUP_COMID = 1`) or not (`DUP_COMID = 0`). To create a unique reach identifier, we reassign the COMID to a unique negative value for each reach with `DUP_COMID = 1` so that COMID is again unique. There is one complex confluence left in the network, so we remove the smallest upstream branch of the confluence, both the reach involved in the confluence and all nine reaches upstream of it. For simplicity in this initial analysis, we also select just the major connected component of the network, leaving us with 169,094 reaches or 99.78% of the NSI Region 5 network.

4.2.3 Fish count data

Scientists conduct surveys to count the number of fish of different species present in a given area at a given time. The most common method is electrofishing, in which fish are temporarily stunned by an electric field in the water to allow the scientists to count them. In small streams shallow enough to wade, the electrofishing units are mounted on the bank or carried via backpack, while in larger rivers and lakes electrofishing is typically conducted by boat or raft (Figure 4.2).

In this work we use three national fish observation datasets: Chen et al. (2023), Giam and Olden (2016), and Strecker et al. (2011). These three datasets are themselves compilations of data from the national and regional United States Environmental Protection Agency (EPA) Environmental Monitoring and Assessment Program (EMAP), the EPA National Riverse and Streams Assessment (NRSA), the United States Geological Survey National Water Quality Assessment Program (USGS NAWQA), state natural resource agencies and environmetal programs for biomonitoring, and sampling efforts by university researchers. The Chen et al. (2023) dataset contains 1,332,551 observations of 776 unique species at 60,047 unique points on 42,150 unique stream reaches. The Giam and Olden (2016) dataset contains 87,864 observations and the Strecker et al. (2011) dataset contains 1,939 observations.

The Chen et al. (2023) dataset contains both count data and presence/absence data, and the observations across the datasets come from many surveys using a variety of survey

(a) Backpack electrofishing.

(b) Boat electrofishing.

Figure 4.2: Electrofishing methods. Photo credit: SARDI. Source: Bucater et al. (2025).

methods. For our analysis, we use only count observations and only observations collected via electrofishing methods, and we drop observations with unknown electric gear because they seem to be duplicates of other measurements. For each species on each stream reach for which we have fish survey data in at least one of these three datasets, we use the most recent observation in our analysis. We also use only data after 1990 since this captures the bulk of the data and reduces the time span represented by the data in this analysis.

One challenge in modeling fish populations across large regions is the range in stream sizes. Isaak et al. (2017) are able to use densities of fish per 100 m stream length instead of fish per water volume or per cross-sectional river area because they focus on larger fish in smaller streams. To account for this in our study, we assume the length of stream sampled within each survey is 20 times the mean bankfull channel width of the stream reach in which the sampling took place. If this length is less than 100 meters, we replace it with 100 meters, and if the length is greater than 1000 meters, then we replace it with 1000 meters. This is consistent with federal and most state fish monitoring electrofishing protocols (Hauer and Lamberti, 2006). We can then express the observed counts as densities by dividing each

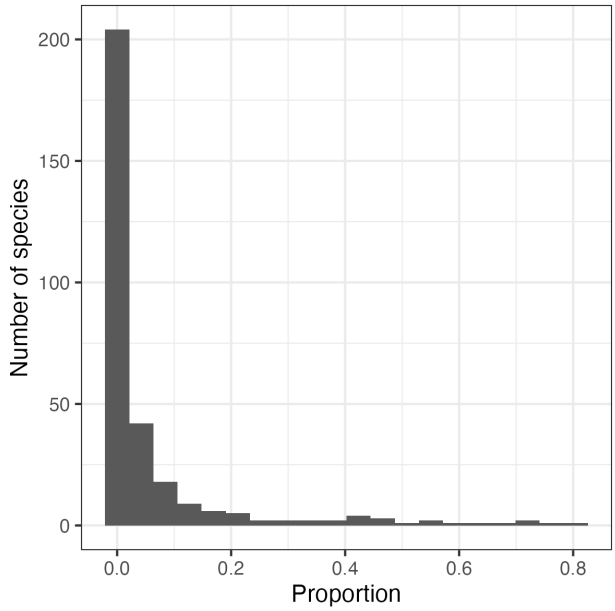


Figure 4.3: Proportion of COMIDs with any fish observations that have observations of a given species.

count by its corresponding estimated sampling length.

Figure 4.3 shows the proportion of COMIDs with any fish observations (of any species) that have observations of a particular species. The proportion of COMIDs with observations of at least one fish species that have observations of a given species varies by species from 0.0001 to 0.81 (mean 0.062; quartiles 0.0008, 0.0058, 0.044). As numbers instead of proportions, these values range from 1 to 7186 (mean 557; quartiles 7, 52, 393).

4.3 Spatial model

Given observed fish densities $\boldsymbol{Y} = (Y(\boldsymbol{s_1}), ..., Y(\boldsymbol{s_n}))$ in fish per meter at n point locations $\boldsymbol{s_1}$, ..., $\boldsymbol{s_n}$ across a region, we wish to estimate the geographic distribution of fish across the region and make regional population estimates by species. We fit a scalable spatial stream

network (S3N) model with an exponential tail-up covariance:

$$
\begin{aligned}
\boldsymbol{Y} &= \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{w}, \quad \boldsymbol{w} \sim N\left(0, \tilde{\boldsymbol{\Sigma}}\right), \\
w(\boldsymbol{s}_i) \mid \boldsymbol{w}(M(\boldsymbol{s}_i)) &\sim N(\boldsymbol{a_i}, d_i), \\
\boldsymbol{a_i} &= \boldsymbol{\Sigma}\big(M(\boldsymbol{s_i}), M(\boldsymbol{s_i})\big)^{-1}\boldsymbol{\Sigma}\big(M(\boldsymbol{s_i}), \boldsymbol{s_i}\big), \\
d_i &= \Sigma\big(\boldsymbol{s_i}, \boldsymbol{s_i}\big) - \boldsymbol{\Sigma}\big(\boldsymbol{s_i}, M(\boldsymbol{s_i})\big)\boldsymbol{a_i}, \\
\Sigma\big(\boldsymbol{s_i}, \boldsymbol{s_j}\big) &= \pi_{ij} C_u(\boldsymbol{s_i}, \boldsymbol{s_j} \mid \boldsymbol{\theta}_u) + \tau^2\delta_{ij} \\
&= \pi_{ij}\sigma_u^2 \exp\big(-h(\boldsymbol{s_i}, \boldsymbol{s_j})/\lambda_u\big) + \tau^2\delta_{ij}.
\end{aligned}
\tag{4.1}
$$

We assume a normal distribution to model fish densities rather than using a zero-inflated Poisson or a negative binomial distribution to model the fish counts because the adaptation of scalable spatial processes to generalized linear models is only now becoming possible. To our knowledge, Finley et al. (2022) develop the first and only NNGP application to use a non-Gaussian distribution, specifically the binomial distribution. Fixed-rank kriging, another approach to scalable spatial processes, only became adapted for non-Gaussian distributions in 2024 (Sainsbury-Dale et al., 2024). Our first priority in this work is to make the method scalable while accounting for stream network structure, so we save the extension to generalized linear models for future work.

As a result of using a normal distribution, the S3N model admits the possibility of negative predicted fish densities. However, we are ultimately interested in aggregating results over large regions, and in the next section we evaluate the nature and extent of negative densities in our results. Before computing regional population estimates, negative densities are set to zero, so negative predicted densities are interpreted as an indication from the model that no or few fish are expected to be present in that location. We examine model diagnostics to evaluate whether this interpretation is justified.

Preprocessing the Region 5 stream network and computing pairwise stream distances can be done once for all 309 species, while estimation, inference, and prediction must be repeated for each species. Configuring the stream network is performed twice, once to assess the network for topological concerns and examine how many components of the network

there are, and again after refining the network to remove a complex confluence and restrict the network to the single largest component for simplicity. As previously noted, configuring the stream network still leaves us with the vast majority (99.78%) of the network.

After this step, we compute the upstream distance and additive function value (AFV) for each reach, map observation and prediction locations to the stream reaches, and compute their upstream distances and AFVs. After this preprocessing and before estimation, we compute the pairwise distances between all observation points, observation-observation neighbor variables required for estimation and prediction, pairwise distances between each prediction point and all observation points, and prediction-observation neighbor variables required for prediction. These steps, particularly the prediction-observation pairwise distances, are highly parallelizable, so we compute the pairwise distances between prediction and observation points in 33 parallel batches of 5000 or fewer prediction points each.

For each species, we estimate model parameters using maximum likelihood estimation and estimate confidence intervals for both fixed effects and covariance parameters using BRISC, a nearest neighbor parametric bootstrap. To produce Region 5 population size estimates for each species, we obtain the predictive mean fish density per 100 meters at the midpoint of each stream reach, multiply this by the length of the stream reach and scale units appropriately to obtain an estimate for the mean number of fish in each stream reach, then sum across the reaches of the network to obtain a regional population estimate for that species. We use the estimated environmental fixed effects to examine the relationship between these variables and fish abundance.

We perform the entire analysis from preprocessing through prediction and visualization on a personal laptop (MacBook Pro 2020, 1.4 GHz Quad-Code Intel Core i5, with only 8 GB memory) with the exception of the pairwise distance calculations, which we run on a research cluster.

Step	Runtime	
	In seconds	In minutes
Configure stream network	124	2.1
Compute stream updist and AFV	78	1.3
Add obs, preds to LSN	22	
Compute obs-obs dist on cluster	1190	20
Compute preds-obs dist on cluster	1580	26
Estimation, average per species	5.7	
Inference, average per species per rep	4.2	
Prediction, average per species	3.3	

Table 4.1: Runtimes for each step of the analysis.

4.4 Results

4.4.1 Computational cost

Table 4.1 summarizes the runtimes for each step of the Region 5 analysis, and histograms of estimation and prediction runtimes across species are available in Figure D.1 of the appendix. The time indicated for configuring the stream network, 2.1 minutes, includes initially configuring the network, identifying and removing the complex confluence, and reconfiguring the network. In total the preprocessing took 3.8 minutes.

Computing the pairwise distances among observations and between prediction and observation locations is currently the rate limiting step of the S3N model, but these steps can be parallelized. Computing the observation pairwise distances as well as generating the nearest neighbor variables required for BRISC took 20 minutes. Computing the pairwise distances between prediction and observation points in 33 parallel batches of 5000 or fewer prediction

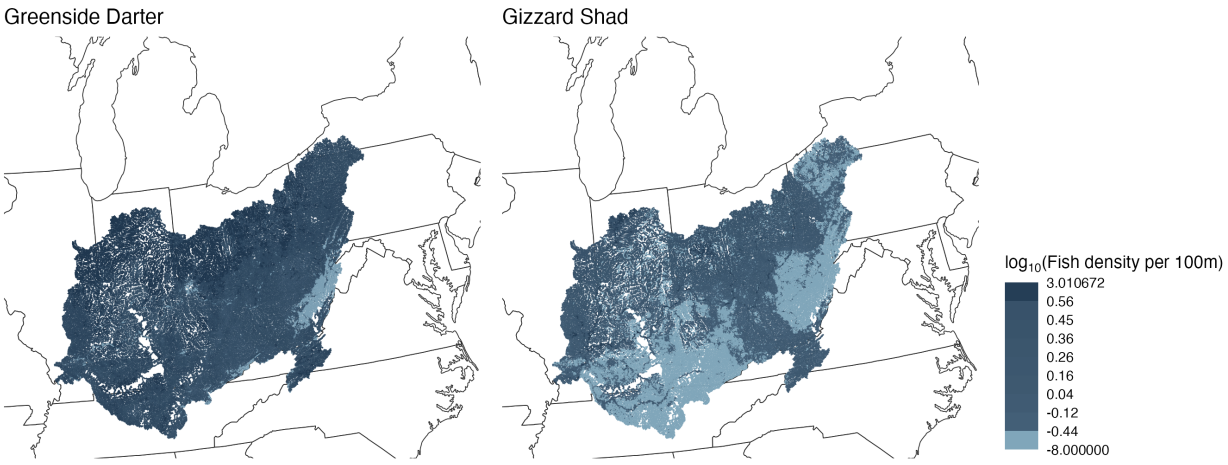


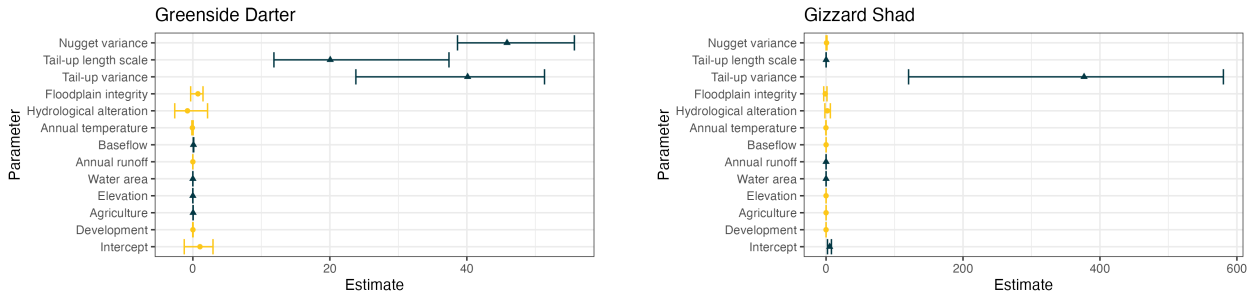
Figure 4.4: Maps of predictive mean fish density (fish per 100m stream length) for two example species. For these plots, densities less than 10^{-8} were set to 10^{-8} before taking the logarithm. To determine color scale increments, we computed percentiles, in increments of 10%, of the combined fish densities from the two species, then kept only the unique values.

points each took 26 minutes.

Estimating model parameters and predicting fish densities across Region 5 took 5.7 and 3.3 seconds on average per species. Inference with the bootstrap takes approximately as long as estimation for each bootstrap replication (4.2 seconds). We tested the bootstrap with a few different numbers of replications and found 20 seemed sufficient. The total runtimes for estimation, inference and prediction for all 309 species with 20 bootstrap reps per species were 29 minutes, 7.3 hours, and 17 minutes, respectively. If confidence intervals for the variance parameters are not of interest, then the longest step, the bootstrap, can be omitted.

4.4.2 Population size estimates and interpretation of model parameters

Figure 4.4 shows maps of the mean predicted fish density for two example species, greenside darter and gizzard shad. These maps demonstrate that the model can capture a range of



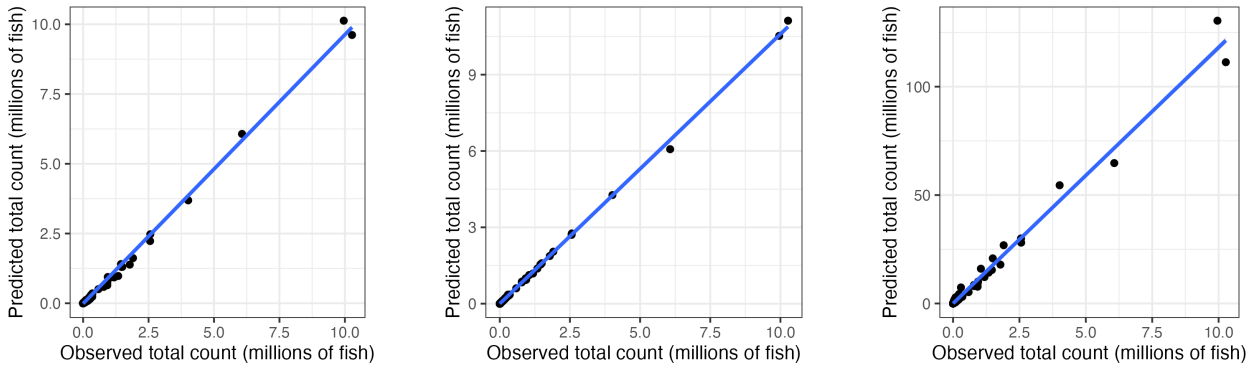
(a) (b)

Figure 4.5: Fixed effect and covariance parameter estimates and confidence intervals for each of the example species. Statistically significant coefficients are shown in dark blue with triangles for point estimates, while other coefficients are shown in yellow with circles for point estimates.

geographic distributions and spatial variation. The greenside darter's density is relatively uniform, while gizzard shad are most prevalent along the northern portion of the region with a branch to the southeast. Figure 4.5 summarizes the estimates and confidence intervals for the fixed effects and covariance parameters. The tail-up characteristic length scale was found to be roughly 60-120 m for gizzard shad and 20 km for greenside darter, and statistically significant for both species. This seems consistent with the fact that gizzard shad tend to live in lakes and reservoirs while the greenside darter tends to inhabit swift streams. Water area is statistically significant for both species. Baseflow and elevation are significant for greenside darter, and annual runoff is significant for gizzard shad.

4.4.3 Model diagnostics

Figure 4.6 shows scatterplots of observed versus predicted counts. The ratio of predicted to observed counts ranges from 0.2 to 1.02 at COMIDs where the species in question was observed, from 0.57 to 3.23 at COMIDs where any fish were observed, and from 1.09 to 153



(a) Counts where this species was observed.

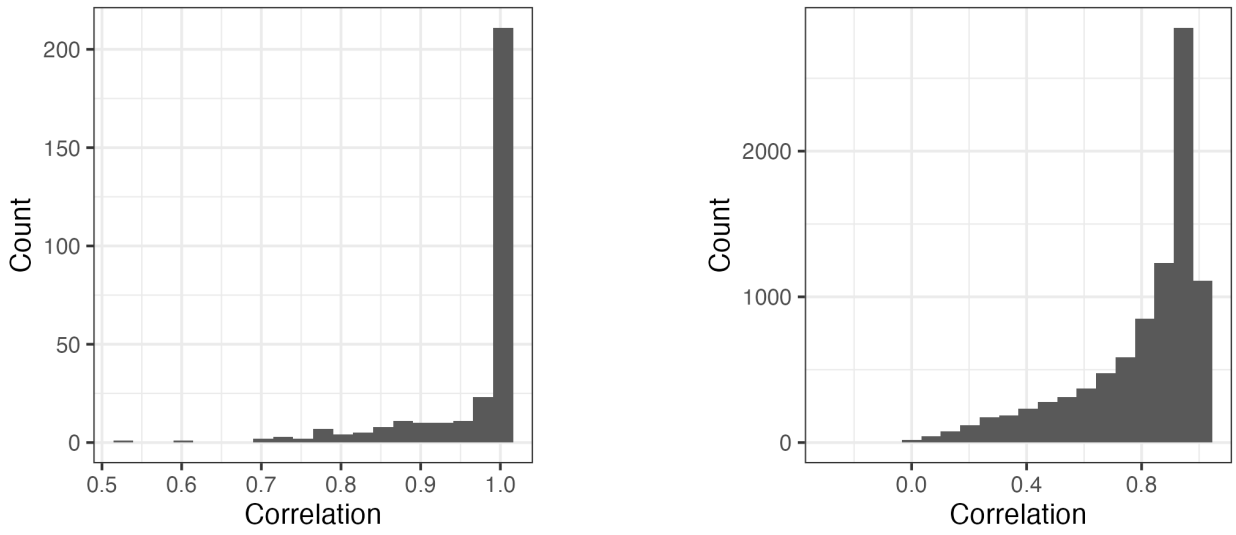
(b) Counts on COMIDs at which any fish were observed.

(c) Counts across Region 5, all COMIDs.

Figure 4.6: Observed versus predicted fish counts.

for the entire region. Additionally, Figure 4.7a shows the correlations between predicted and observed densities across all sites for each species, resulting in 309 values, each of which is a correlation of 8924 pairs of values. These correlations range from 0.52 to 1. Figure 4.7b shows the correlations between predicted and observed densities for all species at each site, resulting in 8924 values, each of which is a correlation of 309 pairs of values. These correlations by site range from -0.28 to 1. These plots provide a sanity check that the model predictions seem broadly consistent with the data.

Since we are using a Gaussian process to model the fish densities, some predicted densities are negative, but we argue that the regional population estimates are reasonable. Of the 309 species, 200 (65%) have no negative predicted densities. The mean and max proportion of predicted densities for each species that are negative are 0.0015 and 0.0792, respectively. Figure 4.8 shows that the negative densities tend to be much smaller in magnitude than the positive densities, and that the median negative density is typically higher for species for which the median positive density is also high. Therefore, the negative densities seem to correspond to locations and species for which the model predicts densities near zero, and



(a) (b)

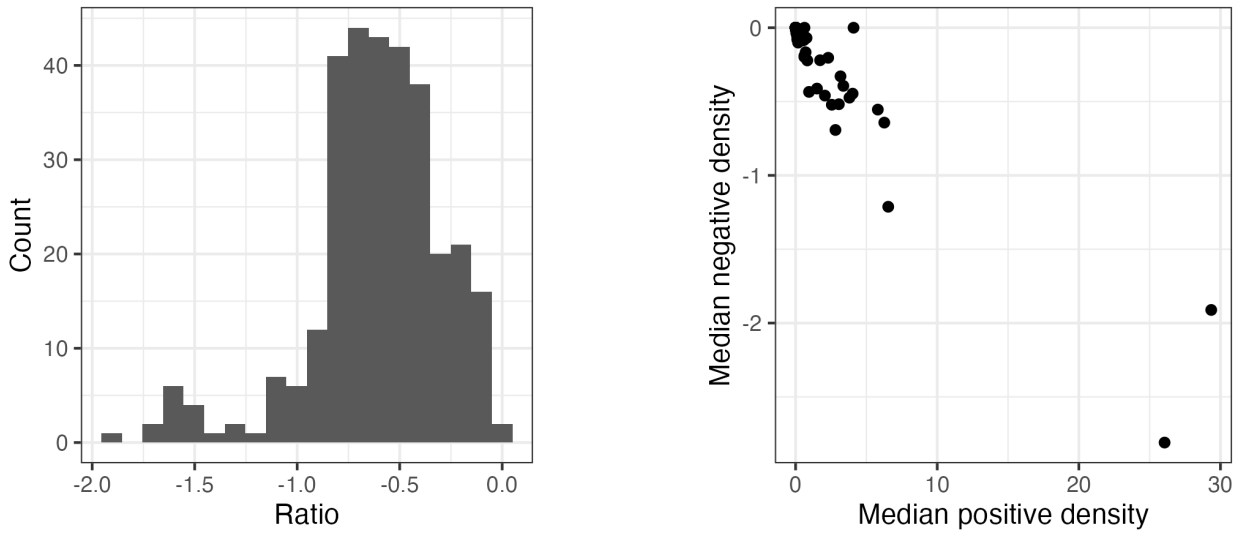
Figure 4.7: Correlations between observed and predicted densities (a) for each species across all observation points and (b) for each observation point across all species.

it seems reasonable to replace the negative densities with zero for the purpose of obtaining regional estimates.

4.5 Discussion

To our knowledge, we have produced the first regional fish population estimates by species over the entire Ohio River Basin stream network using the S3N model developed in the previous chapter. These estimates were not feasible with previously available models. These initial large-scale estimates represent a critical step toward mapping the geographic distribution of freshwater fish species and understanding the role of environmental drivers at large scales.

In future work, we plan to extend this analysis to make national estimates. This will require incorporating expert knowledge in regions with few or no fish survey observations. It



(a) (b)

Figure 4.8: Negative densities are rare and small. Panel a shows the distribution of the ratio of median negative density to median positive density by species, while Panel b displays these medians as a scatterplot.

would also be valuable to include tail-down covariance since this may enable better modeling of species that swim upstream. We will work on implementing a zero-inflated Poisson distribution in the S3N family for improved modeling of count survey data. In future research, if a dataset with higher temporal resolution is available, the S3N model applied here can also be extended from a spatial process to a spacetime process.

Chapter 5

CONCLUSION

5.1 Summary of Contributions

We have presented evidence from theoretical analyses and simulation studies that suggests a less commonly used simple NSUM population size estimator may be more robust to barrier effects than the currently most common NSUM estimator, and more robust than another estimator advocated in the literature. The analysis and simulation framework we proposed and used to evaluate these estimators can be adapted to analyze estimator robustness when other assumptions are violated.

We combine spatial stream networks (SSNs), nearest neighbor Gaussian processes (NNGPs), the bootstrap for rapid inference on spatial covariances (BRISC), and more computationally efficient stream network preprocessing to create a class of scalable spatial stream network (S3N) models. SSNs allow us to capture correlations due to stream network structure, while approximating the full process with an NNGP greatly reduces the computational expense of estimating model parameters. BRISC enables inference of the covariance parameters, whereas existing SSN models provide confidence intervals only for the fixed effects. We also streamline the preprocessing, currently a bottleneck in SSN models, by identifying essential steps and avoiding additional unnecessary computation as much as possible. We demonstrate the scalability of preprocessing, estimation and prediction with S3Ns, which makes them feasible for regions at least on the order of hundreds of thousands of square kilometers. We provide open-source code for these models to the broader community.

To our knowledge, we have produced the first regional fish population estimates by species over the entire Ohio River Basin stream network using the S3N model we developed. This network extends into 14 different states, covers 200,000 sq km, and consists of approximately

170,000 reaches with approximately 9,000 observation point locations. We present and discuss regional estimates and interpret model parameters for a few example species. We also present runtimes and diagnostics and discuss model performance.

5.2 Future Work

5.2.1 Social networks

A clear next step in the context of social networks is identifying ways to correct the bias of the basic NSUM estimators when we are limited to basic NSUM methods. GNSUM has already provided ways to correct these biases when additional data are available or when the researchers have the resources to gather the necessary additional data, but in the absence of those options, what can applied researchers do? It may be easier to correct bias in the RR^* estimator than the RA since the RR^* bias can be expressed directly in terms of the degree ratio. When ad hoc corrections to the simple estimators are applied to the RR^*, it may be helpful to also compute the RA to help bound the result; it is easily done and does not require additional data or computational power.

It would be valuable to evaluate the robustness of the basic NSUM estimators on data sets involving hard-to-reach populations if a reasonable ground truth is available or if synthetic data may be generated to simulate specific types of hard-to-reach populations.

Methods for handling zero-valued degree estimates in practice require further investigation and development. In this work, we considered four different methods. The impact of the reduced sample size in a given study depends on what proportion of respondents typically have (1) zero-valued degree estimates and non-zero responses and/or (2) zero-valued degree estimates and zero-valued responses. Further research is needed to study when and why that choice matters and how greatly it affects relative estimator performance.

Future work can examine the impact on relative estimator performance of other sources of error, especially in combination. It is valuable to consider the case that these errors impact the responses but not the degree estimates, as well as the case that the errors impact both.

Combining NSUM and other methods may be a useful strategy. Studies may benefit from using link-tracing methods at regular but infrequent intervals to learn about possible causes and interventions, while using NSUM to do more frequent monitoring in between.

5.2.2 River networks

The next major extensions to implement in the S3N model code include adding tail-down and Euclidean covariance components; extending S3Ns beyond normal distributions to generalized linear models; incorporating expert knowledge to handle regions with few or no observations; and testing the effectiveness of different methods for ordering the observations in the NNGP and perhaps allowing the user to specify one of a few different orderings.

Chapter 6

REFERENCES

Bibliography

Bernard, H. R., T. Hallett, A. Iovita, E. C. Johnsen, R. Lyerla, C. McCarty, M. Mahy, M. J. Salganik, T. Saliuk, O. Scutelniciuc, G. A. Shelley, P. Sirinirund, S. Weir, and D. F. Stroup (2010). Counting hard-to-count populations: the network scale-up method for public health. *Sexually Transmitted Infections 86*(Suppl. 2), 1368–4973.

Bernard, H. R., E. C. Johnsen, P. D. Killworth, and S. Robinson (1991). Estimating the size of an average personal network and of an event subpopulation: Some empirical results. *Social Science Research 20*(2), 109–121.

Bucater, L., C. Bice, G. Giatas, B. Zampatti, and Q. Ye (2025). *Electrofishing: What is it and how does it work?* Commonwealth Environmental Water Holder (CEWH) Flow – Monitoring, Evaluation and Research Program. https://flow-mer.org.au/electrofishing-what-is-it-and-how-does-it-work/. Accessed 2025-04-04.

Chen, K., S. R. Midway, B. K. Peoples, B. Wang, and J. D. Olden (2023). Shifting taxonomic and functional community composition of rivers under land use change. *Ecology 104*(11), e4155.

Cheng, S., D. J. Eck, and F. W. Crawford (2020). Estimating the size of a hidden finite set: Large-sample behavior of estimators. *Statistics Surveys 14*(none), 1–31.

Crawford, F. W., J. Wu, and R. Heimer (2018). Hidden population size estimation from respondent-driven sampling: A network approach. *Journal of the American Statistical Association 113*(522), 755–766. PMID: 30828120.

Cressie, N., J. Frey, B. Harch, and M. Smith (2006). Spatial prediction on a river network. *Journal of Agricultural, Biological, and Environmental Statistics 11*(2), 127–150.

Cressie, N. and C. K. Wikle (2011). *Statistics for spatio-temporal data*. John Wiley & Sons.

Datta, A. (2022). Nearest-neighbor sparse cholesky matrices in spatial statistics. *Wiley Interdisciplinary Reviews: Computational Statistics 14*(5), e1574.

Datta, A., S. Banerjee, A. O. Finley, and A. E. Gelfand (2016a). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association 111*(514), 800–812.

Datta, A., S. Banerjee, A. O. Finley, and A. E. Gelfand (2016b). On nearest-neighbor Gaussian process models for massive spatial data. *Wiley Interdisciplinary Reviews: Computational Statistics 8*(5), 162–171.

Dent, C. L. and N. B. Grimm (1999). Spatial heterogeneity of stream water nutrient concentrations over successional time. *Ecology 80*(7), 2283–2298.

Dumelle, M., E. E. Peterson, J. M. Ver Hoef, A. Pearse, and D. J. Isaak (2024). SSN2: The next generation of spatial stream network modeling in R. *Journal of Open Source Software 9*(99), 6389.

Feehan, D. M. (2015). *Network Reporting Methods*. Ph. D. thesis, Princeton University.

Feehan, D. M. and M. J. Salganik (2016). Generalizing the network scale-up method: A new estimator for the size of hidden populations. *Sociological Methodology 46*(1), 153–186.

Feehan, D. M., V. H. Son, and A. Abdul-Quader (2022). Survey methods for estimating the size of weak-tie personal networks. *Sociological Methodology 52*(2), 193–219.

Finley, A. O., A. Datta, and S. Banerjee (2022). spnngp r package for nearest neighbor gaussian process models. *Journal of Statistical Software 103*, 1–40.

Finley, A. O., A. Datta, B. D. Cook, D. C. Morton, H. E. Andersen, and S. Banerjee (2019). Efficient algorithms for bayesian nearest neighbor gaussian processes. *Journal of Computational and Graphical Statistics 28*(2), 401–414.

Ganio, L. M., C. E. Torgersen, and R. E. Gresswell (2005). A geostatistical approach for describing spatial pattern in stream networks. *Frontiers in Ecology and the Environment 3*(3), 138–144.

Giam, X. and J. D. Olden (2016). Environment and predation govern fish community assembly in temperate streams. *Global Ecology and Biogeography 25*(10), 1194–1205.

Guinness, J. (2018). Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics 60*(4), 415–429.

Habecker, P., K. Dombrowski, and B. Khan (2015, 12). Improving the network scale-up estimator: Incorporating means of sums, recursive back estimation, and sampling weights. *PLOS ONE 10*(12), 1–16.

Handcock, M. S., K. J. Gile, and C. M. Mar (2014). Estimating hidden population size using respondent-driven sampling data. *Electronic Journal of Statistics 8*(1), 1491–1521.

Hauer, F. R. and G. Lamberti (2006). *Methods in stream ecology* (2nd ed.). Academic press.

Hughes, K. (2021). *The world's forgotten fishes*. World Wide Fund for Nature (WWF).

Isaak, D. J., J. M. Ver Hoef, E. E. Peterson, D. L. Horan, and D. E. Nagel (2017). Scalable population estimates using spatial-stream-network (SSN) models, fish density surveys, and national geospatial database frameworks for streams. *Canadian Journal of Fisheries and Aquatic Sciences 74*(2), 147–156.

Killworth, P. D., E. C. Johnsen, C. McCarty, G. A. Shelley, and H. R. Bernard (1998). A social network approach to estimating seroprevalence in the United States. *Social Networks 20*, 23–50.

Killworth, P. D., C. McCarty, H. R. Bernard, E. C. Johnsen, J. Domini, and G. A. Shelley (2003). Two interpretations of reports of knowledge of subpopulation sizes. *Social networks 25*(2), 141–160.

Killworth, P. D., C. McCarty, H. R. Bernard, G. A. Shelley, and E. C. Johnsen (1998). Estimation of seroprevalence, rape, and homelessness in the United States using a social network approach. *Evaluation Review 22*(2), 289–308.

Kunke, J. P., I. Laga, X. Niu, and T. H. McCormick (2024). Comparing the robustness of simple network scale-up method estimators. *Sociological Methodology 54*(2), 385–403.

Laga, I., L. Bao, and X. Niu (2021). Thirty years of the network scale-up method. *Journal of the American Statistical Association 116*(535), 1548–1559.

Lynch, A. J., R. I. Arthur, C. Baigun, J. E. Claussen, K. Kangur, A. A. Koning, K. J. Murchie, B. J. Myers, G. L. Stokes, R. W. Tingley, and S.-J. Youn (2022). Societal values of inland fishes. In T. Mehner and K. Tockner (Eds.), *Encyclopedia of Inland Waters* (2nd ed.)., pp. 475–490. Oxford: Elsevier.

McCarty, C., P. D. Killworth, H. R. Bernard, E. C. Johnsen, and G. A. Shelley (2001). Comparing two methods for estimating network size. *Human organization 60*(1), 28–39.

McCormick, T. H. (2021, 01). The Network Scale-Up Method. In *The Oxford Handbook of Social Networks*. Oxford University Press.

McCormick, T. H., M. J. Salganik, and T. Zheng (2010). How many people do you know?: Efficiently estimating personal network size. *Journal of the American Statistical Association 105*(489), 59–70. PMID: 23729943.

McCormick, T. H. and T. Zheng (2015). Latent surface models for networks using aggregated relational data. *Journal of the American Statistical Association 110*(512), 1684–1695.

McManamay, R. A., R. George, R. R. Morrison, and B. L. Ruddell (2022). Mapping hydrologic alteration and ecological consequences in stream reaches of the conterminous United States. *Scientific Data 9*(1), 450.

McPherson, M., L. Smith-Lovin, and J. M. Cook (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology 27*(1), 415–444.

Morrison, R. R., K. Simonson, R. A. McManamay, and D. Carver (2023). Degradation of floodplain integrity within the contiguous United States. *Communications Earth & Environment 4*(1), 215.

Nagel, D., S. Wollrab, S. Parkes-Payne, E. Peterson, D. Isaak, and J. Ver Hoef (2017). National Stream Internet hydrography datasets for spatial-stream-network (SSN) analysis. Technical report, Rocky Mountain Research Station, US Forest Service Data Archive, Fort Collins, CO.

Ocagli, H., D. Azzolina, G. Lorenzoni, S. Gallipoli, M. Martinato, A. S. Acar, P. Berchialla, D. Gregori, I. S. Group, et al. (2021). Using social networks to estimate the number of COVID-19 cases: The INCIDENT (Hidden COVID-19 Cases Network Estimation) study protocol. *International Journal of Environmental Research and Public Health 18*(11), 5713.

Olea, R. A. and E. Pardo-Igúzquiza (2011). Generalized bootstrap method for assessment of uncertainty in semivariogram inference. *Mathematical Geosciences 43*, 203–228.

Olhede, S. C. and P. J. Wolfe (2014). Network histograms and universality of blockmodel approximation. *Proceedings of the National Academy of Sciences 111*(41), 14722–14727.

Peterson, D. J., E. E. Peterson, J. Ver Hoef, D. Nagel, S. Wollrab, G. Chandler, D. Horan, and S. Parkes-Payne (2020). Analysis of spatial stream networks for salmonids. Technical report, Technical report to the Bonneville Power Administration, BPA Contract.

Peterson, E., M. Dumelle, A. Pearse, D. Teleki, and J. M. Ver Hoef (2024). *SSNbler: Assemble SSN objects in R*. R package version 1.1.0.

Peterson, E. and J. Ver Hoef (2014). STARS: An ArcGIS toolset used to calculate the spatial information needed to fit spatial statistical models to stream network data. *Journal of Statistical Software 56*, 1–17.

Peterson, E. E., A. A. Merton, D. M. Theobald, and N. S. Urquhart (2006). Patterns of spatial autocorrelation in stream water chemistry. *Environmental Monitoring and Assessment 121*, 571–596.

Peterson, E. E. and J. M. Ver Hoef (2010). A mixed-model moving-average approach to geostatistical modeling in stream networks. *Ecology 91*(3), 644–651.

Rasmussen, C. E. and C. K. Williams (2006). *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT press.

Reid, A. J., A. K. Carlson, I. F. Creed, E. J. Eliason, P. A. Gell, P. T. Johnson, K. A. Kidd, T. J. MacCormack, J. D. Olden, S. J. Ormerod, et al. (2019). Emerging threats and persistent conservation challenges for freshwater biodiversity. *Biological reviews 94*(3), 849–873.

Routen, A., C. Bambra, A. Willis, and K. Khunti (2022). Hard to reach? language matters when describing populations underserved by health and social care research. *Public Health 205*, e28–e29.

Saha, A. and A. Datta (2018). BRISC: Bootstrap for rapid inference on spatial covariances. *Stat 7*(1), e184.

Sainsbury-Dale, M., A. Zammit-Mangion, and N. Cressie (2024). Modeling big, heterogeneous, non-gaussian spatial and spatio-temporal data using frk. *Journal of Statistical Software 108*, 1–39.

Salganik, M. J. and D. D. Heckathorn (2004). Sampling and estimation in hidden populations using respondent-driven sampling. *Sociological Methodology 34*, 193–239.

Shelley, G. A., H. R. Bernard, P. Killworth, E. Johnsen, and C. McCarty (1995). Who knows your HIV status? What HIV+ patients and their network members know about each other. *Social Networks 17*(3), 189–217.

Strecker, A. L., J. D. Olden, J. B. Whittier, and C. P. Paukert (2011). Defining conservation priorities for freshwater fishes according to taxonomic, functional, and phylogenetic diversity. *Ecological Applications 21*(8), 3002–3013.

Tickner, D., J. J. Opperman, R. Abell, M. Acreman, A. H. Arthington, S. E. Bunn, S. J. Cooke, J. Dalton, W. Darwall, G. Edwards, et al. (2020). Bending the curve of global freshwater biodiversity loss: an emergency recovery plan. *BioScience 70*(4), 330–342.

Traud, A. L., E. D. Kelsic, P. J. Mucha, and M. A. Porter (2011). Comparing community structure to characteristics in online collegiate social networks. *SIAM Review 53*(3), 526–543.

Traud, A. L., P. J. Mucha, and M. A. Porter (2012). Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications 391*(16), 4165–4180.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology 50*(2), 297–312.

Ver Hoef, J., E. Peterson, D. Clifford, and R. Shah (2014). SSN: An R package for spatial statistical modeling on stream networks. *Journal of Statistical Software 56*, 1–45.

Ver Hoef, J. M. and R. P. Barry (1996). Constructing and fitting models for cokriging and multivariable spatial prediction. *Journal of Agricultural, Biological, and Environmental Statistics 1*(3), 297–322.

Ver Hoef, J. M., E. Peterson, and D. Theobald (2006). Spatial statistical models that use flow and stream distance. *Environmental and Ecological statistics 13*, 449–464.

Ver Hoef, J. M. and E. E. Peterson (2010). A moving average approach for spatial statistical models of stream networks. *Journal of the American Statistical Association 105*(489), 6–18.

Ver Hoef, J. M., E. E. Peterson, and D. J. Isaak (2019). Spatial statistical models for stream networks. In A. E. Gelfand, M. Fuentes, J. A. Hoeting, and R. L. Smith (Eds.), *Handbook of Environmental and Ecological Statistics*, pp. 421–444. Chapman and Hall/CRC.

Webster, R. A., K. H. Pollock, S. K. Ghosh, and D. G. Hankin (2008). Bayesian spatial modeling of data from unit-count surveys of fish in streams. *Transactions of the American Fisheries Society 137*(2), 438–453.

Wieczorek, M., S. Jackson, and G. . Schwarz (2018). Select attributes for NHDPlus version 2.1 reach catchments and modified network routed upstream watersheds for the conterminous United States. *US Geological Survey 10*, F7765D7V.

Wyatt, R. J. (2003). Mapping the abundance of riverine fish populations: integrating hierarchical Bayesian models with a geographic information system (GIS). *Canadian Journal of Fisheries and Aquatic Sciences 60*(8), 997–1006.

Yaglom, A. (1987). *Correlation Theory of Stationary and Related Random Functions.* Springer.

Appendix A

NSUM DERIVATIONS AND ADDITIONAL RESULTS

Section A.1 details the derivations of the expectation and variance of the RR and RA estimators. Section A.2 contains proofs that the one-step average-of-ratios degree estimators and prevalence estimators have greater variance than their ratio-of-averages counterparts under the binomial model. Section A.3 demonstrates that the binomial model approximates the Erdős-Rényi model; this is not a new result but is included here for completeness and convenience. Section A.4 examines the combinations of parameter values in which the RMSE is lower for the RA estimator than the RR estimator. Section A.5 presents further details on the comparison of different methods for handling zero-valued degree estimates in the Facebook 100 simulations. Section A.6 presents and discusses additional results from the Facebook 100 simulations.

A.1 Derivations for RR and RA expectation and variance

$$E(Y_{iH}) = \begin{cases} N_H p_{HH} & i \in H \\ N_H p_{HL} & i \in L \end{cases} \qquad E(Y_{iK}) = \begin{cases} N_K p_{HL} & i \in H \\ N_K p_{LL} & i \in L \end{cases}$$

$$\mathrm{Var}(Y_{iH}) = \begin{cases} N_H p_{HH}(1-p_{HH}) & i \in H \\ N_H p_{HL}(1-p_{HL}) & i \in L \end{cases} \qquad \mathrm{Var}(Y_{iK}) = \begin{cases} N_K p_{HL}(1-p_{HL}) & i \in H \\ N_K p_{LL}(1-p_{LL}) & i \in L \end{cases}$$

$$\sum_{i=1}^{n} E(Y_{iH}) = N_H \left(n_H p_{HH} + n_L p_{HL}\right) \qquad \sum_{i=1}^{n} E(Y_{iK}) = N_K \left(n_H p_{HL} + n_L p_{LL}\right)$$

$$\sum_{i=1}^{n} \mathrm{Var}(Y_{iH}) = N_H \big[n_H p_{HH}(1-p_{HH}) + n_L p_{HL}(1-p_{HL})\big]$$

$$\sum_{i=1}^{n} \mathrm{Var}(Y_{iK}) = N_K \left[n_H p_{HL}(1 - p_{HL}) + n_L p_{LL}(1 - p_{LL}) \right]$$

First-order Taylor approximations for the mean and variance of a ratio of random variables are given by $E(A/B) \approx E(A)/E(B)$ and, if A and B are independent, $\mathrm{Var}(A/B) \approx [E(B)^2 \mathrm{Var}(A) + E(A)^2 \mathrm{Var}(B)]/[E(B)^4]$. For handling more than one probe group and allowing probe groups and the hidden group to overlap with each other, one can either assume approximate independence or include the covariance term in the variance approximation; for now we consider the case of a single probe group K that is disjoint from H.

Taking the expectation over the SBM superpopulation for a given sample,

$$\begin{aligned}
E\left(\hat{r}_{\mathrm{RR}}\right) &= E\left(\frac{N_K}{N} \frac{\sum_{i=1}^{n} Y_{iH}}{\sum_{i=1}^{n} Y_{iK}} \right) \\
&\approx \frac{N_K}{N} \frac{E\left(\sum_{i=1}^{n} Y_{iH}\right)}{E\left(\sum_{i=1}^{n} Y_{iK}\right)} && \text{1st order Taylor approximation} \\
&= \frac{N_K}{N} \frac{\sum_{i=1}^{n} E\left(Y_{iH}\right)}{\sum_{i=1}^{n} E\left(Y_{iK}\right)} \\
&= \frac{N_K}{N} \frac{n_H N_H p_{HH} + n_L N_H p_{HL}}{n_H N_K p_{HL} + n_L N_K p_{LL}} \\
&= \frac{N_H}{N} \frac{n_H p_{HH} + n_L p_{HL}}{n_H p_{HL} + n_L p_{LL}} \\
&= r \frac{n_H p_{HH} + n_L p_{HL}}{n_H p_{HL} + n_L p_{LL}}.
\end{aligned}$$

Similarly, for the RA estimator,

$$\begin{aligned}
E\left(\hat{r}_{\mathrm{RA}}\right) &= E\left(\frac{N_K}{N} \frac{1}{n} \sum_{i=1}^{n} \frac{Y_{iH}}{Y_{iK}} \right) \\
&= \frac{N_K}{N} \frac{1}{n} \sum_{i=1}^{n} E\left(\frac{Y_{iH}}{Y_{iK}} \right) \\
&\approx \frac{N_K}{N} \frac{1}{n} \sum_{i=1}^{n} \frac{E\left(Y_{iH}\right)}{E\left(Y_{iK}\right)} \\
&= \frac{N_K}{N} \left[\frac{n_H}{n} \frac{N_H p_{HH}}{N_K p_{HL}} + \frac{n_L}{n} \frac{N_H p_{HL}}{N_K p_{LL}} \right] \\
&= r \left[\frac{n_H}{n} \frac{p_{HH}}{p_{HL}} + \frac{n_L}{n} \frac{p_{HL}}{p_{LL}} \right].
\end{aligned}$$

If $n_H/n \to r$, as in simple random sampling without replacement, then

$$E\left(\hat{r}_{\mathrm{RR}}\right) \to r\frac{rp_{HH}+(1-r)p_{HL}}{rp_{HL}+(1-r)p_{LL}},$$
$$E\left(\hat{r}_{\mathrm{RA}}\right) \to r\left[r\frac{p_{HH}}{p_{HL}}+(1-r)\frac{p_{HL}}{p_{LL}}\right].$$

$$\begin{aligned}
\mathrm{Var}\left(\hat{r}_{\mathrm{RR}}\right) &= \mathrm{Var}\left(\frac{N_K}{N}\frac{\sum_{i=1}^n Y_{iH}}{\sum_{i=1}^n Y_{iK}}\right) \\
&= \frac{N_K^2}{N^2}\mathrm{Var}\left(\frac{\sum_{i=1}^n Y_{iH}}{\sum_{i=1}^n Y_{iK}}\right) \\
&\approx \frac{N_K^2}{N^2}\cdot\frac{E(\sum_{i=1}^n Y_{iK})^2\mathrm{Var}(\sum_{i=1}^n Y_{iH})+E(\sum_{i=1}^n Y_{iH})^2\mathrm{Var}(\sum_{i=1}^n Y_{iK})}{E(\sum_{i=1}^n Y_{iK})^4} \qquad \text{(A.1)} \\
&= \frac{N_K^2}{N^2}\cdot\frac{\left(\sum_{i=1}^n E[Y_{iK}]\right)^2\sum_{i=1}^n \mathrm{Var}(Y_{iH})+\left(\sum_{i=1}^n E[Y_{iH}]\right)^2\sum_{i=1}^n \mathrm{Var}(Y_{iK})}{\left(\sum_{i=1}^n E[Y_{iK}]\right)^4} \qquad \text{(A.2)} \\
&= \frac{N_K^2}{N^2}\Bigg(\frac{N_K^2\left(n_Hp_{HL}+n_Lp_{LL}\right)^2 N_H\left[n_Hp_{HH}(1-p_{HH})+n_Lp_{HL}(1-p_{HL})\right]}{N_K^4\left(n_Hp_{HL}+n_Lp_{LL}\right)^4}+ \\
&\qquad \frac{N_H^2\left(n_Hp_{HH}+n_Lp_{HL}\right)^2 N_K\left[n_Hp_{HL}(1-p_{HL})+n_Lp_{LL}(1-p_{LL})\right]}{N_K^4\left(n_Hp_{HL}+n_Lp_{LL}\right)^4}\Bigg) \\
&= r\Bigg(\frac{1}{N}\frac{\left(n_Hp_{HL}+n_Lp_{LL}\right)^2\left[n_Hp_{HH}(1-p_{HH})+n_Lp_{HL}(1-p_{HL})\right]}{\left(n_Hp_{HL}+n_Lp_{LL}\right)^4}+ \\
&\qquad \frac{r}{N_K}\frac{\left(n_Hp_{HH}+n_Lp_{HL}\right)^2\left[n_Hp_{HL}(1-p_{HL})+n_Lp_{LL}(1-p_{LL})\right]}{\left(n_Hp_{HL}+n_Lp_{LL}\right)^4}\Bigg).
\end{aligned}$$

Step (A.1) above uses the Taylor approximation and Step (A.2) holds if Y_{iG}, Y_{jG} are independent for $i \neq j$, $G = H, K$. A similar derivation for the RA estimator yields

$$\begin{aligned}
\mathrm{Var}\left(\hat{r}_{\mathrm{RA}}\right) \approx{}& \frac{r}{n^2Np_{HL}^2}\left[n_Hp_{HH}(1-p_{HH})+n_Lp_{LL}(1-p_{HL})\right]+ \\
&\frac{r^2}{n^2N_Kp_{HL}^3}\left[n_Hp_{HH}^2(1-p_{HL})+n_Lp_{HL}^2(1-p_{LL})\right].
\end{aligned}$$

If $n_H/n \to r$, then

$$\mathrm{Var}\left(\hat{r}_{\mathrm{RR}}\right) \to \frac{r}{nN}\frac{\left(rp_{HL}+(1-r)p_{LL}\right)^2\left[rp_{HH}(1-p_{HH})+(1-r)p_{HL}(1-p_{HL})\right]}{\left(rp_{HL}+(1-r)p_{LL}\right)^4} + \frac{r^2}{nN_K}\frac{\left(rp_{HH}+(1-r)p_{HL}\right)^2\left[rp_{HL}(1-p_{HL})+(1-r)p_{LL}(1-p_{LL})\right]}{\left(rp_{HL}+(1-r)p_{LL}\right)^4}$$

$$\mathrm{Var}\left(\hat{r}_{\mathrm{RA}}\right) \to \frac{r}{nNp_{HL}^2}\left[rp_{HH}(1-p_{HH})+(1-r)p_{LL}(1-p_{HL})\right] + \frac{r^2}{nN_Kp_{HL}^3}\left[rp_{HH}^2(1-p_{HL})+(1-r)p_{HL}^2(1-p_{LL})\right].$$

Note that whether the link probabilities are constant as a function of N (as under the constant density assumption, so that the average degree grows linearly with total network size N) or scaled by $1/N$ (so that average degree is constant with respect to N), the expressions for expectation and variance derived in this section will not change because the additional factors of N cancel.

A.2 Comparing the variances of A and R estimators

The two one-step degree estimators under consideration are as follows:

$$\hat{D}_{i,\mathrm{R}} = N\cdot\frac{\sum_j Y_{ij}}{\sum_j N_j}, \quad \hat{D}_{i,\mathrm{A}} = N\cdot\frac{1}{k}\sum_{j=1}^{k}\frac{Y_{ij}}{N_j}.$$

In the unlikely case that the probe groups are all the same size, then the two estimators are identical and therefore also have the same variance. We explore how the variances compare outside of this special case. Under the binomial model, $Y_{ij} \sim \mathrm{Binom}(D_i,\ N_j/N)$. Therefore,

$$\mathrm{Var}\left(\hat{D}_{i,\mathrm{R}}\right) = D_i\left[N\cdot\frac{1}{k}\frac{1}{\sum_j N_j/k} - \frac{\sum_j N_j^2}{\left(\sum_j N_j\right)^2}\right].$$

The second term within the brackets is less than $1/k$ because $N_j > 1$ for each probe group j. The first term equals $1/k$ if the average probe group size is equal to N. Therefore, the second term is negligible as long as the average probe group size is small compared to N:

$$\mathrm{Var}\left(\hat{D}_{i,\mathrm{R}}\right) \approx D_i\left[N\cdot\frac{1}{k}\frac{1}{\sum_j N_j/k}\right] = \frac{D_iN}{k}\frac{1}{\mathrm{mean}(N_j)}.$$

Following similar reasoning,

$$\mathrm{Var}\left(\hat{D}_{i,\mathrm{A}}\right) = D_i\left[\frac{N}{k}\frac{1}{k}\frac{1}{\sum_j N_j} - \frac{1}{k}\right] \approx \frac{D_i N}{k}\frac{1}{k}\frac{1}{\sum_j N_j} = \frac{D_i N}{k}\frac{1}{\mathrm{hmean}(N_j)},$$

where $\mathrm{hmean}(N_j)$ denotes the harmonic mean of the probe group sizes. Since $N_j > 0$ for all j, the harmonic mean is strictly smaller than the arithmetic mean, and therefore $\mathrm{Var}\left(\hat{D}_{i,\mathrm{R}}\right) < \mathrm{Var}\left(\hat{D}_{i,\mathrm{A}}\right)$ if the average probe group size is sufficiently smaller than N.

A similar but simpler argument demonstrates the same result for the two one-step prevalence estimators (with fixed or known degrees) by comparing the arithmetic and harmonic means of the degrees:

$$\hat{r}_{\mathrm{R}} = \frac{\sum_i Y_{iH}}{\sum_i D_i}, \quad \hat{r}_{\mathrm{A}} = \frac{1}{n}\sum_i \frac{Y_{iH}}{D_i}.$$

Again, we ignore the trivial and impractical case in which all degrees are identical. Under the binomial model, $Y_{iH} \overset{\text{indep}}{\sim} \mathrm{Binom}(D_i,\, r)$. Therefore,

$$\mathrm{Var}\left(\hat{r}_{\mathrm{R}}\right) = \frac{r(1-r)}{n}\cdot\frac{1}{\sum_i D_i/n} = \frac{r(1-r)}{n}\cdot\frac{1}{\mathrm{mean}(D_i)},$$

$$\mathrm{Var}\left(\hat{r}_{\mathrm{A}}\right) = \frac{r(1-r)}{n}\cdot\frac{1}{n}\sum_i \frac{1}{D_i} = \frac{r(1-r)}{n}\cdot\frac{1}{\mathrm{hmean}(D_i)}.$$

For $D_i > 0$, outside of the case that all degrees are equal, the harmonic mean is strictly smaller than the arithmetic mean of the degrees. Therefore, $\mathrm{Var}\left(\hat{r}_{\mathrm{R}}\right) < \mathrm{Var}\left(\hat{r}_{\mathrm{A}}\right)$.

A.3 Relating the binomial and Erdős-Rényi models

Given a set of nodes, a graph can be simulated from an Erdős-Rényi model by conducting an iid Bernoulli trial ℓ_{ij} for each pair of nodes to determine whether there is a link between them:

$$\ell_{ij} \overset{\text{iid}}{\sim} \mathrm{Bernoulli}(p)$$

Here we will suppose that the nodes have group memberships (either hidden or not hidden), but that these memberships do not impact link formation.

We can then derive random variables for the number of people each person i knows who are in the hidden group:

$$y_i = \sum_{j \in H, j \neq i} \ell_{ij} \sim \begin{cases} \text{Binom}\left(N_H,\ p\right) & i \notin H, \\ \text{Binom}\left(N_H - 1,\ p\right) & i \in H, \end{cases}$$

and the number of people each person i knows who are not in the hidden group:

$$z_i = \sum_{j \notin H, j \neq i} \ell_{ij} \sim \begin{cases} \text{Binom}\left(N - N_H - 1,\ p\right) & i \notin H, \\ \text{Binom}\left(N - N_H,\ p\right) & i \in H. \end{cases}$$

Let $N_H^* = N_H$ if $i \notin H$ and $N_H - 1$ if $i \in H$. Let $N_L^* = N - N_H - 1$ if $i \notin H$ and $N - N_H$ if $i \in H$. Note that $N_H^* + N_L^* = N - 1$ in either case, so $N_L^* = N - N_H^* - 1$. The distributions also simplify notationally to

$$y_i \sim \text{Binom}\left(N_H^*,\ p\right), \quad z_i \sim \text{Binom}\left(N_L^*,\ p\right).$$

These two variables are independent of one another for a given person, and their sum is that person's degree:

$$d_i = y_i + z_i.$$

Consider two people i and j. Their degrees are not completely independent because there is one potential link between them, so their degrees (sums of potential links) each include ℓ_{ij}. Additionally, their responses are not independent if they are both in the hidden group, because in that case y_i and y_j both correspond to sums that include ℓ_{ij}. Therefore, their conditional responses $y_i|d_i$ are not strictly independent of one another. However, for sufficiently large N and N_H this departure from independence is negligible.

The following derivation demonstrates that $y_i|d_i$ follows a hypergeometric distribution:

$$
\begin{aligned}
p(y_i = y \mid d_i = d) &= \frac{p(y_i = y, d_i = d)}{p(d_i = d)} \\
&= \frac{p(y_i = y, z_i = d - y)}{p(d_i = d)} \\
&= \frac{p(y_i = y)p(z_i = d - y)}{p(d_i = d)} && y_i, z_i \text{ indep} \\
&= \frac{p(y_i = y)p(z_i = d - y)}{\sum_{k=0}^{d_i} p(d_i = d \mid y_i = k)p(y_i = k)} \\
&= \frac{p(y_i = y)p(z_i = d - y)}{\sum_{k=0}^{d} p(y_i = k)p(z_i = d - k)} \\
p(y_i = y)p(z_i = d - y) &= \binom{N_H^*}{y}\binom{N_L^*}{d-y} p^y (1-p)^{N_H^* - y} p^{d-y} (1-p)^{N_L^* - d + y} \\
&= \binom{N_H^*}{y}\binom{N_L^*}{d-y} p^d (1-p)^{N-1-d}
\end{aligned}
$$

The only dependence on y appears in the binomial coefficients:

$$p(y_i = y \mid d_i = d) \propto \binom{N_H^*}{y}\binom{N_L^*}{d-y}.$$

Therefore, $y_i|d_i$ follows a hypergeometric distribution for the number y of successes out of d draws without replacement from a population of size $N - 1$ containing N_H^* many successes.

For sufficiently large N and N_H, the hypergeometric distribution of $y_i|d_i$ under the Erdős-Rényi model converges to the binomial distribution of $y_i|d_i$ under the binomial model, and the approximate independence of $y_i|d_i$ and $y_j|d_j$ for different people i and j converges to independence. In both models, the response y_i can be interpreted as building person i's personal network by drawing one person simply at random from the population and counting how many people were drawn that were in the hidden population; however, these draws are modeled without replacement under the Erdős-Rényi model and with replacement under the binomial model. Modeling without replacement seems more natural since people should not be double-counted in a given person's personal network.

The binomial model is an approximation for the conditional distribution of ARD responses in networks generated from an Erdős-Rényi model, and the approximation improves as N and N_H increase.

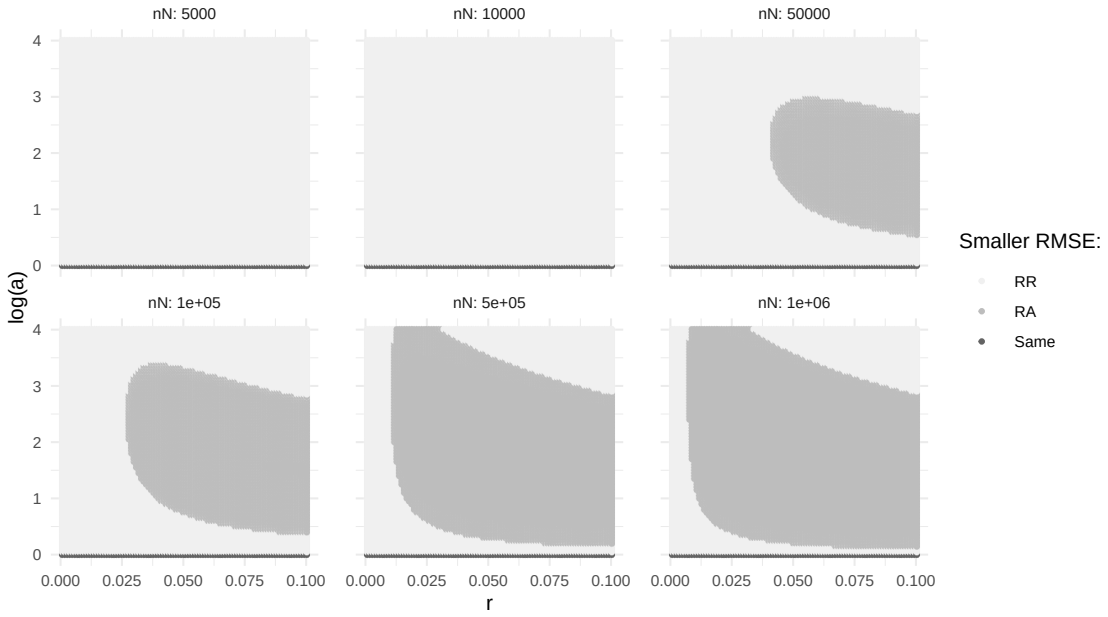


Figure A.1: The six subpanels here correspond to six different values of nN, the product of the sample size and the population size. The size of the darkest region, the region in which the RA has smaller RMSE than the RR, grows with nN (larger sample sizes and larger populations). All assortative and Erdős-Rényi simulations for each value of nN are shown; R ranges from 0.01 to 0.99, $\log(a)$ ranges from 0 to 4, $p_{HL} = 0.01$, and r_K ranges from 0.01 to 0.8.

A.4 Examining the dependence of comparative estimator performance on other parameters

Figure A.1 examines the dependence of the region in which RA has lower RMSE on the value of nN, the product of the sample size and the population size, when $p = 0.01$ and the other parameters are allowed to vary. For nN as small as five or ten thousand, the RA no longer has lower RMSE. The size of the region increases with nN, expanding to encompass smaller values of r and a wider range of a.

Figure A.2 shows the same results as Figure A.1 except for a smaller value of p, 0.001

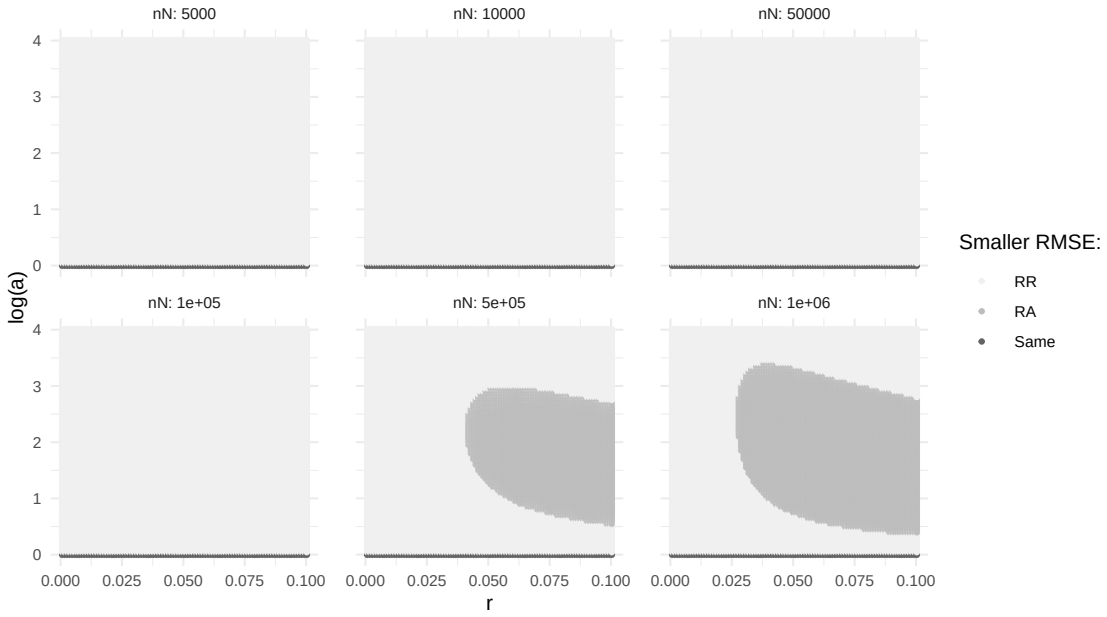


Figure A.2: As in Figure A.1 but with $p_{HL} = 0.001$ instead of 0.01. The size of the region in which the RA has smaller RMSE than the RR is smaller for smaller p.

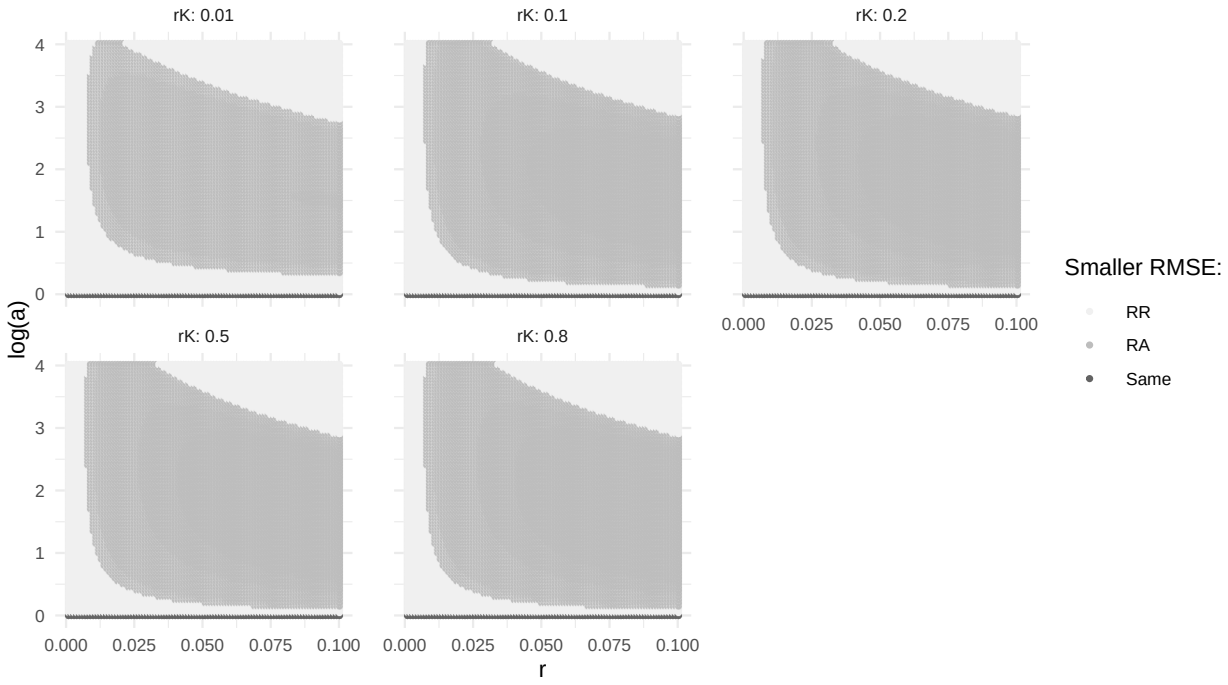


Figure A.3: The size of the region in which the RA has smaller RMSE than the RR depends on r_K, the ratio of the probe group size to the population size, to a smaller extent than on nN. The five subpanels here correspond to five different values of r_K. All assortative and Erdős-Rényi simulations for each value of nN are shown; r ranges from 0.01 to 0.99, $\log(a)$ ranges from 0 to 4, $p_{HL} = 0.01$, and nN ranges from five thousand to one million.

instead of 0.01. With smaller p, the region in which RA has smaller RMSE shrinks. Figure A.3 shows that to a smaller extent, the size of this region also depends on the value of r_K, the ratio of probe group size to population size.

A.5 Methods considered for handling zero-valued estimated degrees

We compute the RA and AA estimators four different ways to evaluate the impact of different methods for handling zero-valued degree estimates. Recall that the ratio $Y_{iH}/\hat{D}_i$ is NA when both the response Y_{iH} and degree estimate $\hat{D}_i$ are 0, and infinite when only the degree estimate $\hat{D}_i$ is 0.

- Approach 1: Set Infs to 1 and exclude NAs. This reduces the effective sample size by dropping people who have $Y_{iH} = \hat{D}_i = 0$.
- Approach 2: Set Infs to 1 and NAs to 0.
- Approach 3: Exclude Infs and NAs. This reduces the effective sample size by dropping all people with $\hat{D}_i = 0$.
- Approach 4: Exclude Infs and set NAs to 0. This reduces the effective sample size by dropping only people who have non-zero Y_{iH} but $\hat{D}_i = 0$.

The reasoning behind setting NAs to 0 is that in the simulations, the numerator $Y_{iH} = 0$ is known and without error, while the denominator $\hat{D}_i = 0$ is estimated, and the true degree must be greater than 0. Therefore, even without knowing the true degree, the ratio is known to be 0. In real life, however, there would be error in both the response Y_{iH} and the estimated degree $\hat{D}_i$. Therefore it seems Methods 1 and 3 might be the most reasonable to use in practice without further study.

Respondents with infinite-valued ratios have a non-zero response Y_{iH} divided by a zero-valued degree estimate. In simulation world, the true denominator is known to be greater than zero but its exact value could make a big difference in the estimate, especially if Y_{iH}

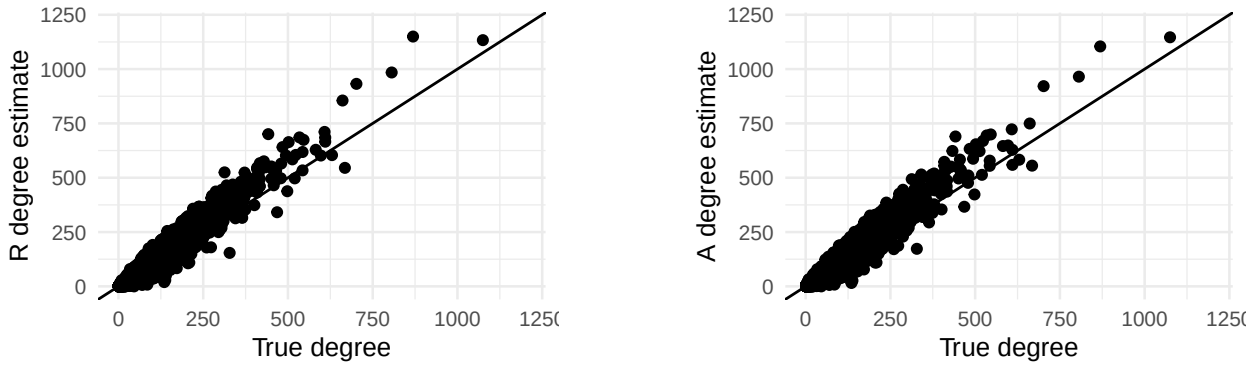
is small. For example, if $Y_{iH} = 1$, the ratio is 1/1 if the degree is 1 and 1/2 if the degree is 2. Should we actually set the Infs to NA and exclude them, computing the estimate with a reduced sample size? Degree estimates are more likely in the Facebook 100 data set to be zero when the true degree is small, and the true ratio tends to be close to 1 for these cases. These two properties could be likely in practice as well, but further research is needed to explore this.

The choice of method for handling zero-valued estimated degrees does not impact the overall findings of the paper; regardless of the method, the RA estimator has lower bias and RMSE than the other estimators in more cases than not, for instance. However, the choice of method sometimes affects which estimator has the lowest RMSE. For instance, under Method 1, the RA (AA, RR, AR) estimator has the lowest RMSE in 41% (21%, 20%, 18%) of the cases. From Method 1 to 4, the percentage increases for the RA and AA estimators and decreases for the RR and AR estimators, resulting in 43% (24%, 17%, 16%) for Method 4. This indicates that it changes the ranking of the estimators in some of the cases.

In this simulation study, then, we chose to show results using Method 1 since this seems like one of the more plausible methods in a real study.

A.6 Additional results from the Facebook 100 simulations

Figure A.4 presents scatterplots of the estimated degrees against the true degrees. The R and A estimates tend to be overestimates but are well correlated with the true degrees. Figure A.5 illustrates that the AR and RR estimators are comparable, and the RA and AA estimators tend to have lower bias and RMSE than the AR estimator for cases with low or high degree ratios.



(a) (b)

Figure A.4: Comparing the (a) R and (b) A degree estimates against the true degrees for a random sample of 10,000 people across the 100 school networks.

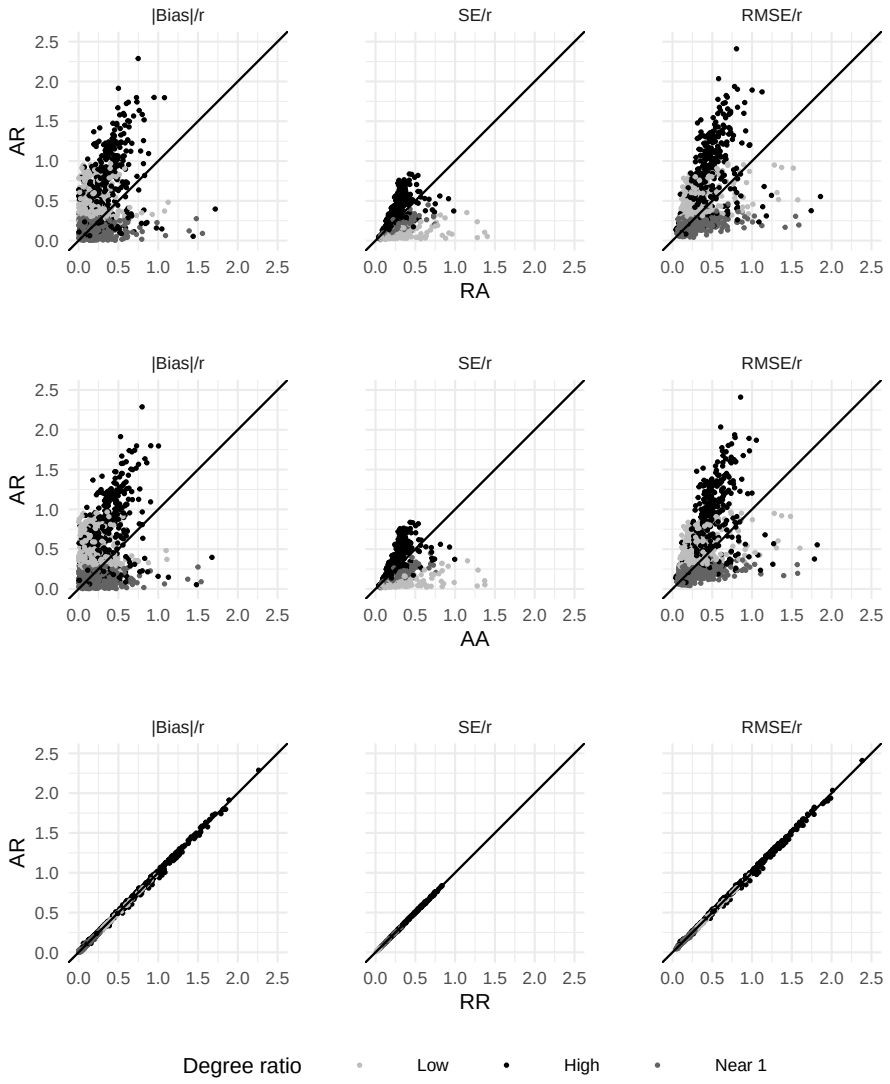


Figure A.5: Comparing the absolute-value bias (left panel), standard error (center panel), and RMSE (right panel), all standardized by the true prevalence, of the AR estimator against that of the other three estimators in all 999 cases from the Facebook 100 simulations. The diagonal line is the one-to-one line; points above the line have **higher** values for the AR estimator than the other estimator. Each point represents the average across 500 surveys of size 500 for one combination of school network and hidden group.

Appendix B

S3N MODEL DERIVATIONS

B.1 S3N likelihood

Recall that in the S3N model,

$$\boldsymbol{Y} \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2 \sim N\left(\boldsymbol{X}\boldsymbol{\beta},\ \tilde{\boldsymbol{C}}_{\boldsymbol{\theta}} + \tau^2 \boldsymbol{I}\right).$$

Therefore the log likelihood ℓ of the responses $\boldsymbol{Y}$ given the model parameters—the fixed effects $\boldsymbol{\beta}$, the latent covariance parameters $\boldsymbol{\theta} = (\sigma_u^2, \lambda_u)$, and the i.i.d. nugget variance τ^2—is given by

$$-2\ell(\boldsymbol{Y} \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2) = (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T \left(\tilde{\boldsymbol{C}}_{\boldsymbol{\theta}} + \tau^2 \boldsymbol{I}\right)^{-1} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \log\det\left(\tilde{\boldsymbol{C}}_{\boldsymbol{\theta}} + \tau^2 \boldsymbol{I}\right), \quad \text{(B.1)}$$

omitting the constant term from the factors of 2π.

Recall that $\tilde{C}_{\boldsymbol{\theta}} = (I - A)^{-1} D (I - A)^{-T}$. Factoring out σ_u^2 and solving for its optimal value in closed form allows us to optimize over only two covariance parameters instead of three. Define $\breve{F} = \sigma_u^2 F$, $\breve{C} = \sigma_u^2 C = (I - A)^{-1} \breve{D} (I - A)^{-T}$, and $\alpha = \tau^2 / \sigma_u^2$. Then Equation B.1 becomes

$$-2\ell(\boldsymbol{Y} \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2) = \frac{1}{\sigma_u^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T \left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}\right)^{-1} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \log\det\left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}\right) + n \log\left(\sigma_u^2\right). \quad \text{(B.2)}$$

Computing the first partial derivatives of ℓ with respect to σ_u^2 and $\boldsymbol{\beta}$ and setting them to zero yields the MLEs $\hat{\sigma}_u^2$ and $\hat{\boldsymbol{\beta}}$:

$$\hat{\sigma}_u^2 = \frac{1}{n}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T \left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}\right)^{-1} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}), \quad \text{(B.3)}$$

$$\hat{\boldsymbol{\beta}} = \left[\boldsymbol{X}^T \left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}\right)^{-1} \boldsymbol{X}\right]^{-1} \boldsymbol{X}^T \left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}\right)^{-1} \boldsymbol{Y}. \quad \text{(B.4)}$$

Substituting Equations B.3 and B.4 back into Equation B.2 yields

$$-2\ell(\boldsymbol{Y} \mid \boldsymbol{\beta}, \boldsymbol{\theta}, \tau^2) = n \log \left((\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T \left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I} \right)^{-1} (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) \right) + \log \det \left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I} \right), \quad \text{(B.5)}$$

omitting the constant term $n - n \log n$.

Finley et al. (2019) provide algorithms for the efficient computation of $\boldsymbol{C}^{-1}$ and quadratic forms $\boldsymbol{u}^T \boldsymbol{C}^{-1} \boldsymbol{v}$ for any matrix $\boldsymbol{C}$ that admits a decomposition of the form $\boldsymbol{C} = (\boldsymbol{I} - \boldsymbol{L})^{-1} \boldsymbol{G} (\boldsymbol{I} - \boldsymbol{L})^{-T}$ for some diagonal matrix $\boldsymbol{G}$ and some lower triangular matrix $\boldsymbol{L}$. If the NNGP is derived from the response process, then these algorithms can be applied directly to $\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}$ but predictions must be made for the response process instead of the latent process. This is how the `BRISC` R package is currently implemented.

If instead the NNGP is derived from the latent process so that predictions can be made for the latent process, the algorithms from Finley et al. (2019) can be applied to $\tilde{\boldsymbol{C}}_{\boldsymbol{\theta}}$ and $\breve{\boldsymbol{C}}_{\boldsymbol{\theta}}$ but not to $\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}$. Instead, Finley et al. (2019) note that by the Sherman-Woodbury-Morrison identity,

$$\left(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I} \right)^{-1} = \alpha^{-1} - \alpha^{-2} \boldsymbol{\Omega}^{-1}, \quad \text{(B.6)}$$

where $\boldsymbol{\Omega} = \breve{\boldsymbol{C}}_{\boldsymbol{\theta}}^{-1} + \alpha^{-1} \boldsymbol{I}$ is conveniently sparse similar to $\tilde{\boldsymbol{C}}_{\boldsymbol{\theta}}^{-1}$ and $\breve{\boldsymbol{C}}_{\boldsymbol{\theta}}^{-1}$. Additionally,

$$\log \det(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}} + \alpha \boldsymbol{I}) = n \log \alpha + \log \det(\breve{\boldsymbol{C}}_{\boldsymbol{\theta}}) + \log \det(\boldsymbol{\Omega}). \quad \text{(B.7)}$$

Appendix C

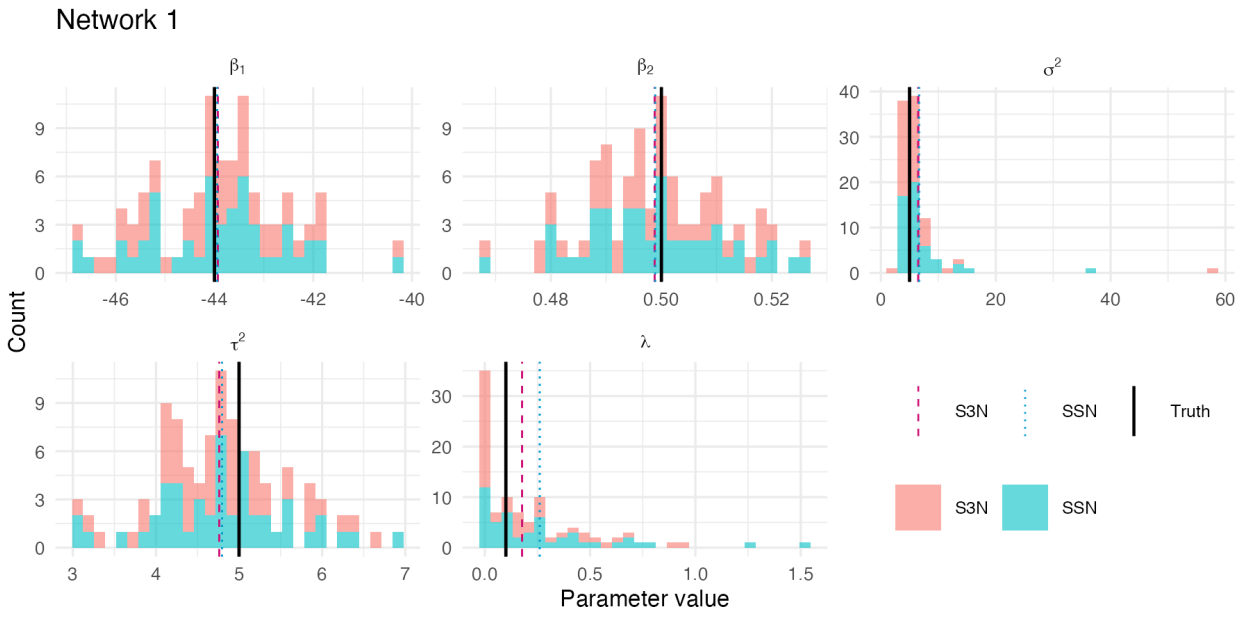
S3N ADDITIONAL RESULTS

	Build stream networks			Compute stream updist			Compute site updist		
Network	S3N	SSN	Ratio	S3N	SSN	Ratio	S3N	SSN	Ratio
1	0.227	1.51	6.62	1.81	1.15	0.635	3.65	6.74	1.85
2	0.729	2.99	4.11	2.33	4.73	2.03	3.83	25.2	6.59
3	2.79	26.9	9.63	6.68	66.1	9.89	4.08	142	34.9
4	5.94	157	26.5	13.0	193	14.9	4.24	239	56.5
5	12.5	2412	192	28.1	1582	56.2	6.35	430	67.7
6	61.1	NA	NA	70.7	NA	NA	18.0	NA	NA

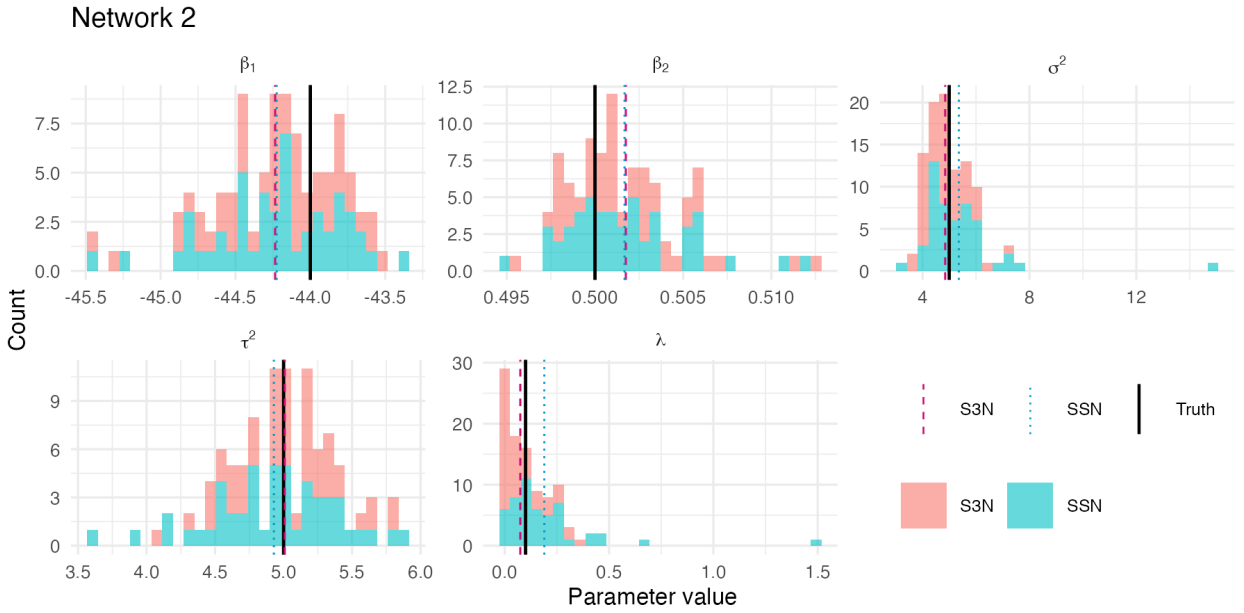
Table C.1: Benchmarking results for stream networks of various sizes. The S3N and SSN columns report average time in seconds over all simulations for each network (50 for both models Networks 1-3, 10 for S3N Networks 4-6, 10 for SSN Network 4, 2 for SSN Network 5). Ratio = SSN/S3N. The NAs in the SSN and Ratio columns for Network 6 reflect the fact that SSN preprocessing with `SSNbler` crashed for this network.

	Compute pwdists		Estimation		Total	
Network	S3N	SSN	S3N	SSN	S3N	SSN
1	2.74	0.58	0.02	1.65	8.46	13.12
2	5.20	0.57	0.18	8.47	12.27	44.34
3	106.68	14.47	0.83	912.45	121.06	1181.28

Table C.2: Benchmarking results for stream networks of various sizes. The S3N and SSN columns report average time in seconds over all 50 simulations for each network. Total times are the sum of runtimes for all functions in Table 3.2, including preprocessing steps.



(a)

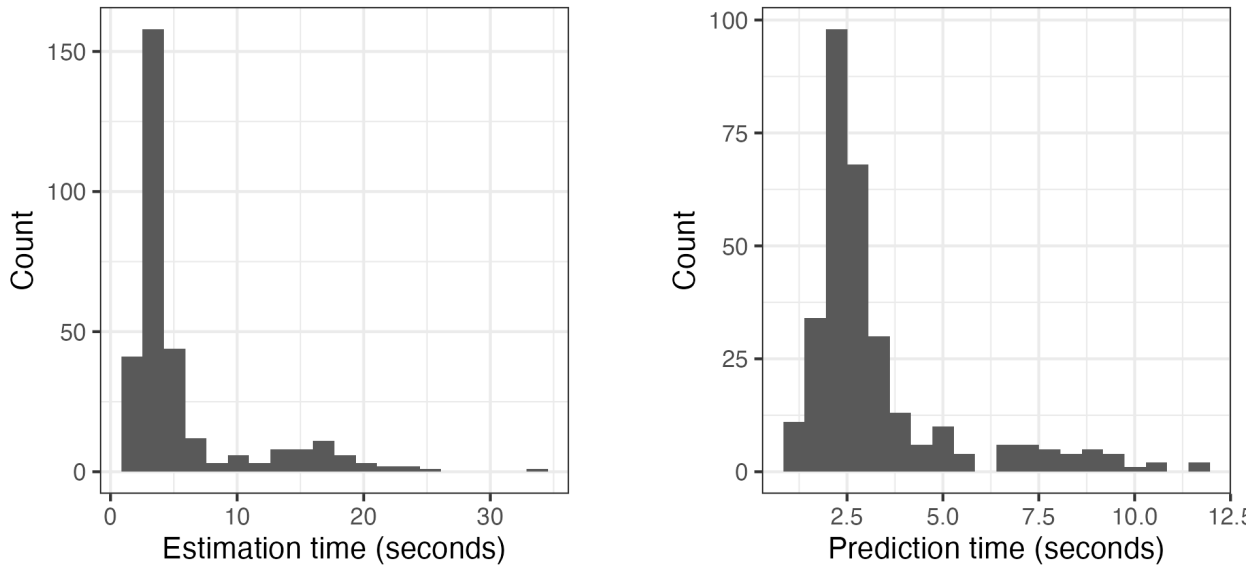


(b)

Figure C.1: Model validation on Networks 1 and 2.

Appendix D

REGION 5 ADDITIONAL RESULTS



(a) Runtime for estimating model parameters.

(b) Runtime for predicting fish densities.

Figure D.1: Computational expense for estimation and prediction.

	Greenside Darter				Gizzard Shad			
Parameter	Estimate	Lower	Upper	Signif	Estimate	Lower	Upper	Signif
Intercept	1.04e+00	-1.26e+00	2.94e+00		5.29e+00	2.09e+00	7.83e+00	Yes
Development	6.73e-03	-2.66e-02	3.32e-02		-1.98e-02	-9.20e-02	2.99e-02	
Agriculture	3.26e-02	2.01e-02	3.74e-02	Yes	7.93e-03	-2.29e-03	2.86e-02	
Elevation	-3.27e-03	-5.45e-03	-1.26e-03	Yes	-1.16e-03	-5.33e-03	2.72e-03	
Water Area	-2.51e-05	-4.33e-05	-1.21e-05	Yes	4.41e-05	1.20e-05	5.86e-05	Yes
Runoff	-1.56e-03	-4.73e-03	2.19e-03		-6.44e-03	-1.08e-02	-3.05e-03	Yes
Baseflow	8.22e-02	5.72e-02	1.25e-01	Yes	4.23e-02	-8.95e-03	7.90e-02	
Temperature	-7.37e-02	-1.48e-01	3.78e-02		-1.80e-01	-3.51e-01	7.77e-02	
Hydrol. Alt.	-7.91e-01	-2.65e+00	2.14e+00		1.84e+00	-1.75e+00	6.26e+00	
Flood. Int.	7.39e-01	-3.24e-01	1.48e+00		-1.16e+00	-3.39e+00	1.35e+00	
Tail-up var.	4.01e+01	2.38e+01	5.13e+01	Yes	3.77e+02	1.21e+02	5.80e+02	Yes
Tail-up len. λ	2.01e+01	1.18e+01	3.74e+01	Yes	1.18e-01	6.00e-02	9.33e-02	Yes
Nugget var. τ^2	4.59e+01	3.86e+01	5.57e+01	Yes	6.70e-01	0	1.34e+00	

Table D.1: Parameter estimates for two example species.