Inherited chromosomally integrated HHV-6 demonstrates tissue-specific RNA expression in vivo

Vikas Peddu

A thesis

Submitted in partial fulfillment of the

Requirements for the degree of

Master of Science

University of Washington

2019

Committee:

Alexander Greninger

Keith Jerome

Meei-Li Huang

Program Authorized to Offer Degree:

Laboratory Medicine

University of Washington

Abstract

Inherited chromosomally integrated HHV-6 demonstrates tissue-specific RNA expression in vivo

Vikas Peddu

Chair of the Supervisory Committee:

Dr. Alexander Greninger

Laboratory Medicine

Human herpesvirus-6A and 6B (HHV-6A, HHV-6B) are highly prevalent viruses unique in that they, besides Epstein-Barr virus, are the only human herpesviruses capable of chromosomal integration. When a chromosomal integration event happens in a germ cell the resulting progeny will have one copy of the virus in all of their cells, referred to as inherited chromosomally integrated human herpesvirus-6 (iciHHV-6). Active HHV-6 infections in those who are iciHHV-6 positive confound current DNA based PCR assays, highlighting a necessity for mRNA-based assays. As the transcriptional state of iciHHV-6 has not been defined in vivo, it is unclear which HHV-6 genes best determine actively infected iciHHV-6 individuals. Here, we screened DNA-Seq and RNA-Seq data for 650 individuals available through the Genotype-Tissue Expression (GTEx) project and identified 2 iciHHV-6A and 4 iciHHV-6B positive

candidates. When corresponding tissue-specific gene expression signatures were analyzed, the HHV-6 genes U90 and U100 were expressed in the brain, testis, breast, adrenal gland, lungs, salivary gland, esophagus, skeletal muscle, colon, tibial nerve and artery, adipose tissue, heart, skin, and thyroid. Additionally, high levels of HHV-6 gene expression were detected in the brain, testis, esophagus, and adrenal gland. We also analyzed data from the Synapse Mount Sinai Brain Bank and found similar brain tissue expression in the 4 iciHHV-6 positive samples. We found no expression of U38, a viral polymerase subunit found to be expressed in HHV-6 active infections, in GTEx whole blood RNAseq samples. However, using an RT-qPCR assay specific to the HHV-6A and -6B U38 genes, we saw that U38 was expressed in some iciHHV-6 positive PBMC samples submitted for iciHHV-6 testing.

Table of Contents

Figure List

Table List

Chapter 1: Introduction to Human herpesvirus-6

Human herpesvirus-6, an overview

Human herpesvirus 6 (HHV-6) represents two unique species: HHV-6A and HHV-6B. Primary infection occurs in 90% of children within their first two years of life and causes roseola, also known as sixth disease, and has been strongly associated with febrile seizures in children (1). HHV-6 reactivation has been observed in 56% of post hematopoietic stem cell transplant this is thought to be associated with immunosuppression (2). Those with post-transplant HHV-6 reactivation have also been observed to have a higher chance of human cytomegalovirus reactivation as well (3).

Chromosomal integration of HHV-6

As with all herpesviruses, HHV-6 establishes a lifelong latency, though it is unique in that it is the only human herpesvirus other than Epstein-Barr virus capable of chromosomal integration. The HHV-6 genome contains 8kb direct repeats flanking either end of the genome. Within these direct repeats regions are telomere-repeat-like-sequences (TRSs) containing repeats of the sequence “TTAGGG” (4). The mechanism of integration remains unknown but hypothesized to be via homologous recombination involving this region of the HHV-6 genome and the human subtelomeric repeat sequences.

U94, thought to be a possible integrase, found in both HHV-6A and HHV-6B subtypes with no known orthologue in other human herpesviruses (5). The U94 protein

is well conserved between the HHV6 subtypes with 97.5% amino acid identity, and interestingly contains several conserved regions of the Adeno associated virus 2 (AAV-2) Rep68 integrase required for Rep68 enzymatic activity. Despite this, it has been shown through U94 knockout experiments that U94 is dispensable for HHV-6 chromosomal integration (6).

Inherited chromosomally integrated HHV-6

In approximately 0.5-2% of the general population, chromosomally integrated virus can be vertically passed through the parent's germline, resulting in one copy of the HHV-6 genome in every cell in the resulting child's body. This is referred to as inherited chromosomally integrated HHV-6 (iciHHV6) (7). An extremely sensitive ddPCR based assay to detect a 1:2 ratio of HHV-6 to human genes has been described as a method of diagnosing iciHHV-6 (8).

However, iciHHV-6 presents a confounding issue for conventional DNA based PCR diagnostic assays when diagnosing HHV-6 active infections because HHV-6 DNA is always present in the cells of iciHHV-6 patients. As a result, though the previously described ddPCR assay can be used to discriminate active infections from HHV-6 positive patients, it cannot determine HHV-6 active infections in the context of iciHHV-6. To do this, an mRNA-based assay would be required. It is unclear whether integrated HHV-6 genomes are transcriptionally active, and if so, whether they exhibit tissue-specific expression.

The gene expression of HHV-6 has already been characterized through deep RNA-sequencing culture cell lines (9), and In-vivo RNA-sequencing of HHV-6 performed

by Dr. Josh Hill (10). Despite this, the in-vivo transcriptional state of iciHHV-6 remained unknown. Due to the 1% prevalence of iciHHV-6, sample collection is time consuming and tedious. This is additionally complicated by the storage requirements for RNA to prevent degradation, highlighting the necessity for either sample collection in an RNA stabilizing media, such as PAXgene RNA tubes, or rapid library preparation from the time of sampling (10). To circumvent this, we chose to analyze iciHHV-6 retrospectively through the use of massive online sequencing datasets.

Chapter 2: The use of publicly available genomic and transcriptomic datasets to detect iciHHV-6 gene expression

Rationale of filtering genomic datasets for iciHHV-6

In individuals with iciHHV-6 we would expect one copy of HHV-6 per cell, meaning one HHV-6 genome per two human genes. As such, if in sequencing datasets we find samples containing a 1:2 ratio of HHV-6 genomes to human genes we have reason to believe the sample comes from an iciHHV-6 positive donor. Though it is possible for an HHV-6 active infection to achieve this ratio, the datasets we selected are from populations in which we have little reason to suspect an HHV-6 active infection.

Description of the GTEx dataset

To resolve the need for HHV-6 gene expression data, we utilized the Genotype-Tissue Expression project (GTEx), a publicly available controlled access dataset which at the time of analysis contained whole blood whole genome, exome, and transcriptome sequencing samples from 650 recently deceased donors. Additionally, RNAseq was performed on a mean of 30 various tissue from each donor. All deceased donors used for this study were donors were otherwise healthy at their time of death and met the criteria for organ donation. The sample collection and analysis data for this project (GTEx version V6) is available through the GTEx data portal documentation page (11).

Description of the Mount Sinai Brain Bank dataset

As we were performing our GTEx analyses, HHV-6 RNA was found by Readhead et al. in an the Mount Sinai Brain Bank (MSBB) sequencing dataset, an Alzheimer's brain sequencing dataset containing whole exome and RNA sequence for 450 deceased donors (12). As this is as large dataset containing RNAseq data, we supposed there could be iciHHV-6 positive individuals here and therefore included it in our subsequent analyses.

Description of Bioinformatic analyses

Retrieval of sequences from both GTEx and MSBB required a controlled data access use application specific to either dataset. Retrieval of GTEx sequences was done using a custom pipeline written with the help of Dan Tenenbaum from Fred Hutch Scicomp using a combination of BOTO3 framework and shell scripting to run on Amazon AWS batch (https://github.com/FredHutch/sra-pipeline). Our pipeline downloaded sequences using the NCBI's SRA-toolkit Prefetch and FASTQ-dump commands and then piped alignments to Bowtie2, a sequencing alignment tool, to perform local alignments to the HHV-6A and HHV-6B reference genomes.

Due to the abundance of human genomic sequence relative to HHV-6 sequence and the presence of human repeat sequence in the HHV-6 genomes, we noticed large amounts of human repeat reads aligning to our viral reference genomes. To circumvent this, I manually trimmed repeat regions from the reference genomes before alignment. Along with the viral alignments, alignments to similarly repeat-trimmed references of the human housekeeping genes Beta-globin and EDAR were performed to determine virus

to cell ratios. These analyses are described in further detail in the methods section and outlined in figure 1 below.

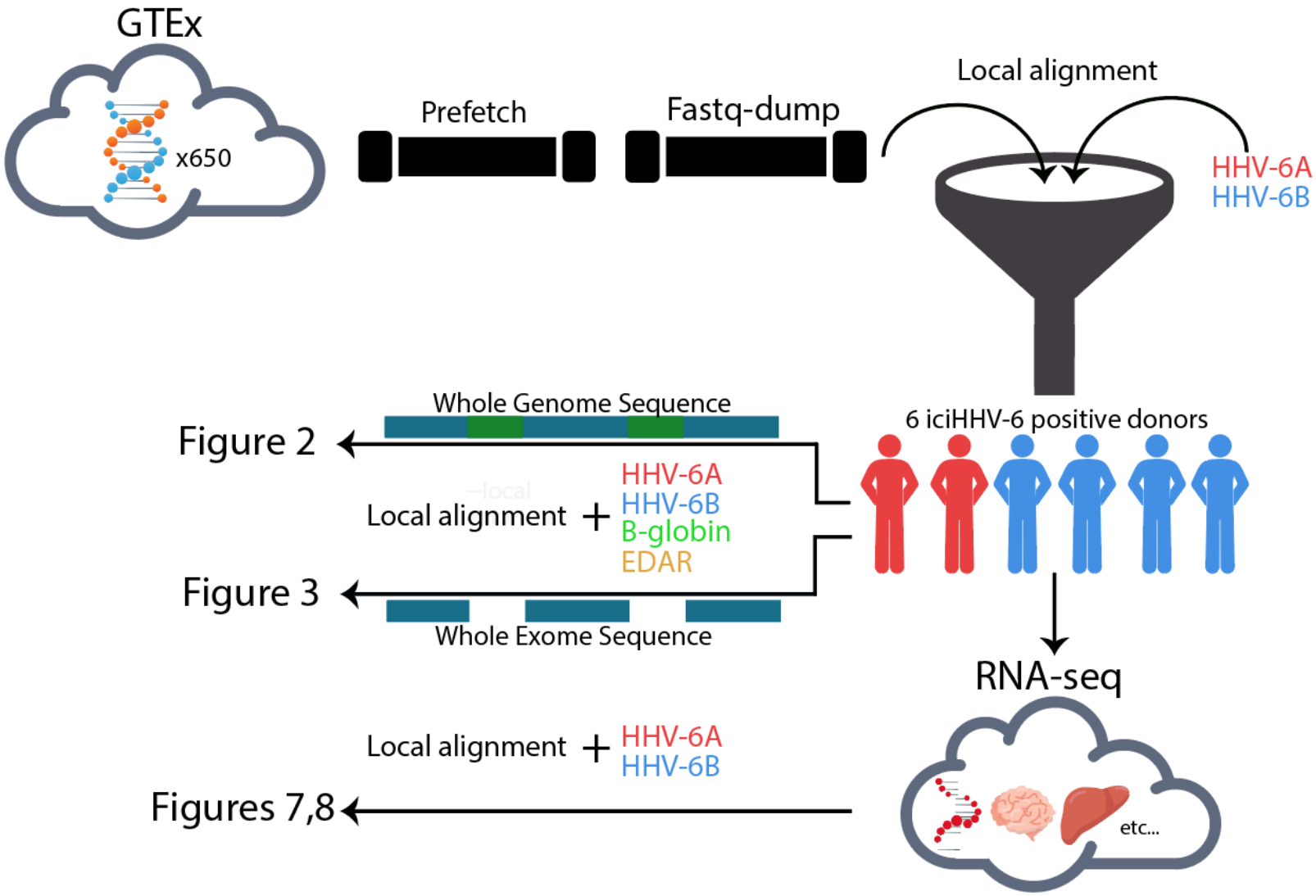

Figure 1| GTEx data was downloaded from the cloud and screened for the correct ratio of viral:cellular genes for iciHHV-6. Corresponding RNAseq files were downloaded for subsequent analyses.

Confirmation of genes found in bioinformatic analysis by PCR based methods

Three HHV-6 genes, U38, U90, and U100, were chosen for analysis by RT-qPCR and RT-ddPCR based on experiments demonstrating expression, or a lack of expression, in both in vitro and in vivo HHV-6 active infection samples, as well as our bioinformatic analyses. The gene functions are described in table 1 below.

Gene	Function
U38	Viral DNA polymerase subunit
U90	Transactivator
U100	Envelope glycoprotein Q

Table 1| Functions of genes selected for RT-qPCR and RT-ddPCR analysis

Chapter 3: Materials and Methods

GTEx data analysis

GTEx genotype data was downloaded from dbgap (June 1st, 2018) using prefetch. 650 DNA sequence SRA files were clipped, decompressed, and extracted using fastq-dump with the following flags: –W (remove tag sequences from dataset), –I (uniquely labels paired-end reads), and –split-files (splits paired-end reads into separate files). The fastq-dump output was piped to Bowtie2 (13) for alignment using the --no-unal (unaligned reads not saved in resulting SAM file) and –-local (local alignment) flags. FASTQ files were aligned to the HHV-6A (NC_001664.4), HHV-6B (AF157706.1), and HHV-7 (NC_001716.2) reference genomes with any repeat like regions manually removed.

For the 6/650 files suspected to be positive for iciHHV-6 (e.g., >25x average depth across HHV-6A/B reference genomes in DNA-Seq dataset), corresponding RNA-seq FASTQ files for all available tissues (111 total) were downloaded and aligned to HHV-6A/B reference genomes as described above. All reads were confirmed as HHV-6 using BLASTn with an Evalue < 1e-8 against the NCBI nt database (January 5, 2019). For corresponding negative controls, 100 GTEx biospecimen IDs were randomly selected and all available RNA-seq FASTQ files (1903 total) corresponding to those individuals were aligned against HHV-6A/B references as described above.

For each tissue within each GTEx donor, all reads mapping to HHV-6 were summed together respective to their genes to calculate RPKM, a normalized metric for comparing sequencing data between different datasets. Any reads that fell into regions

of overlap between two genes were counted as half of a read, ie. a read that mapped in the U3 and U4 overlap region would count as +.5 for both rather than +1 for both.

DNA-seq FASTQ reads corresponding to the six iciHHV-6 positive donors were aligned to portions of the human genes EDAR (NM_022336.4) and Beta-globin (AH001475.2) that were trimmed of human repeats as well as repeat-trimmed HHV-6A (NC_001664.4) and HHV-6B (AF157706.1) reference genomes (Supplement 2), using with the same Bowtie2 options specified above. We calculated a normalized depth of coverage by counting the number of reads aligned to the region of interest (R) divided by the total length of the sequence in megabases (B) from the sample and normalizing the highest rpkm from EDAR or beta-globin obtained to 100 ($\frac{R*30,000}{B}$).

GTEx Contamination Analyses

A random subsample of 10 million reads was taken from GTEx whole genome sequence data for all iciHHV-6 positive samples as well as the 7 RNA-seq samples that showed the highest HHV-6 gene expression. Alignment was done using Bowtie2 with the same flags described above. The frequency of insert sizes of reads aligning to HHV-6 in each sample was taken from the resulting SAM files. These same samples were also aligned to the HG38 human reference genome with the same alignment settings as mentioned before to calculate human insert sizes.

Alignment of HHV-6B RNA-seq reads to the GTEx-OXRO consensus genome was done to verify SNPs that could differentiate each donor. Alignment of reads to the HHV-6B Z29 reference genome was also done to find paired reads spanning splice junctions to demonstrate reads were RNA and not DNA.

Mount Sinai Brain Bank (MSBB) data

Whole exome BAM files aligned to hg19 were downloaded (8/2018) from the Synapse MSBB database (14). Unmapped reads were extracted via Samtools using the command “samtools view -b -f 4 <bam file>” and converted into FASTQ files via samtools bam2fq (15). FASTQ files were then combined and aligned to HHV-6A and HHV-6B reference genomes as described above. Twenty randomly selected samples that were negative for HHV-6 DNA were also screened for HHV-6 RNA. All reads were confirmed as HHV-6 using BLASTn with an Evalue<1e-8 against the NCBI nt database (January 5, 2019).

Construction of phylogenetic trees

HHV-6A and HHV-6B genomes were downloaded from NCBI GenBank (September 1st, 2018). Contiguous HHV-6B sequences between nucleotides 9,515 and 118,889 of the Z29 reference sequence (AF157706.1), corresponding to genes U4 - U77, were used for analysis due to lack of missing sequence in this region. Contiguous HHV-6A sequences between nucleotides 79,352 and 110,248 of the U1102 reference sequence (NC_001664.4) were similarly used, corresponding to genes U48 - U73. Both sets of subsequences were aligned with MAFFT using default parameters. Phylogenetic trees were constructed using the Geneious tree builder with 100 iterations.

PBMC sample collection and extraction

31 Leftover clinical PBMC samples stored in 1 ml of PBS from physician ordered iciHHV-6 tests were obtained from the University of Washington Clinical Molecular Virology Laboratory in a 6-month period between 2017-2018. Clinical IciHHV-6 testing and typing was done using a previously published ddPCR protocol (8). We were blinded from these results until all of our RT-qPCR analyses were complete. PBMCs were centrifuged at 5000g for 5 minutes and the cell pellet was stored in 300 µl of Qiagen RNAprotect at -80ºC until subsequent RNA extraction for RT-qPCR analysis.

U38 RT-qPCR and U90, U100 RT-ddPCR

Primers and probes for both HHV-6A and HHV-6B for U38, U90, and U100 were designed using the HHV-6A U1102 (NC_001664.4) and HHV-6B Z29 (AF157706.1) reference sequences. U38 primers and probes were designed to detect both HHV-6A and HHV-6B, while U90 and U100 were type specific. Analysis for U38 was performed using RT-qPCR (Qiagen Quantitec Reverse Transcription kit), while RT-ddPCR (1-Step RT-ddPCR Advanced Kit) was used for U90 and U100 in an attempt to multiplex the assays to conserve our limited quantity of clinical sample.

In order to assess for DNA contamination, all samples were run with and without the reverse transcriptase enzyme as well. To assess overall RNA quality of the sample, all samples were also run alongside the human housekeeping gene RPP30, also with and without reverse transcriptase. The primer and probe sequences for all genes are listed below in table 2, with cycling conditions in table 3.

Creation of standards for U38

As our U38 analyses depended on RT-qPCR rather than RT-ddCPR, in order to quantitate copies of RNA a standard curve would be required. RNA from HHV-6 infected culture cells was used as a template. CDNA was created using the Invitrogen Superscript III Reverse Transcriptase kit, and U38 was amplified by conventional PCR with a modified U38 forward primer containing the T7 promoter sequence (TAATACGACTCACTATAG). Amplicons were verified and purified from a gel, and then transcribed using the Thermo Fisher Megascript T7 transcription kit. From here, serial dilutions were made and quantified using both a Nanodrop and Qubit, and subsequently stored at -80ºC for long term storage.

Gene	Primer/Probe	Sequence
U38	Forward Primer	TGCCCGATTYTGAAAAAGCT
	Reverse Primer	CCTGTGGGTATTCATAAAATTTTGC
HHV-6A U90	Probe	FAM-CTCCCGCGCTTTGCACAGACG -BHQ
	Forward Primer	GATGCACCAATACTTTGGATGC
	Reverse Primer	AGAAGAAGGATCAGATGGAGAGTTG
	Probe	FAM-ACTGAGCAGCTAAAG-MGBNFQ
	Forward Primer	AGCAGGCTTTCAAAGGACACA
	Reverse Primer	CCCTCTGGAAACAACATGGAAT
	Probe	FAM-AACGTATGCAAAACTACCATC-MGB

HHV-6A U100	Forward Primer	CCGCCATTGTTTCGTATTTTC
	Reverse Primer	GAGCTACGCCAAAACTACAAAGTG
	Probe	FAM-ATCAGTTCACATAGAGTCT-MGBNFQ
HHV-6B U100	Forward Primer	ATAGCAGAGAGAAGTGAAGGCCGA
	Reverse Primer	TCCTGAACTACGCCATTTGCGATG
	Probe	JOE-TGTCTCTATGAGACAACCGCAGCTGT-BHQ1

Table 2| Primers and probes used for the RT-qPCR and RT-ddPCR analyses

Gene	Temperature	Time	
U38	50°C	20 min	**45 cycles**
	95°C	5 min	
	95°C	**15 sec**	
	60°C	**45 sec**	
U90, U100	50°C	60 min	**40 cycles**
	94°C	10 min	
	94°C	**30 sec**	
	60°C	**1 min**	
	98°C	10 min	

Table 3| Cycling conditions for Rt-qPCR and RT-ddPCR analyses

Chapter 4: Results

Screening depth of coverage to HHV-6 in DNA-Seq data reveals 6 iciHHV-6 cases among 650 individuals

From the whole genome DNA-Seq data available from 650 GTEx individuals, we determined 6 were consistent with iciHHV-6: 4 iciHHV-6B and 2 iciHHV-6A. These 6 samples had an average normalized depth of coverage across the HHV-6 genome that was approximately half (0.45 ± 0.035) that of human housekeeping genes EDAR and beta-globin, consistent with heterozygous iciHHV-6 at the approximate 1% prevalence typically found in human populations (Figure 2) (16). No evidence of chromosomally integrated HHV-7 was found (data not shown) (17)

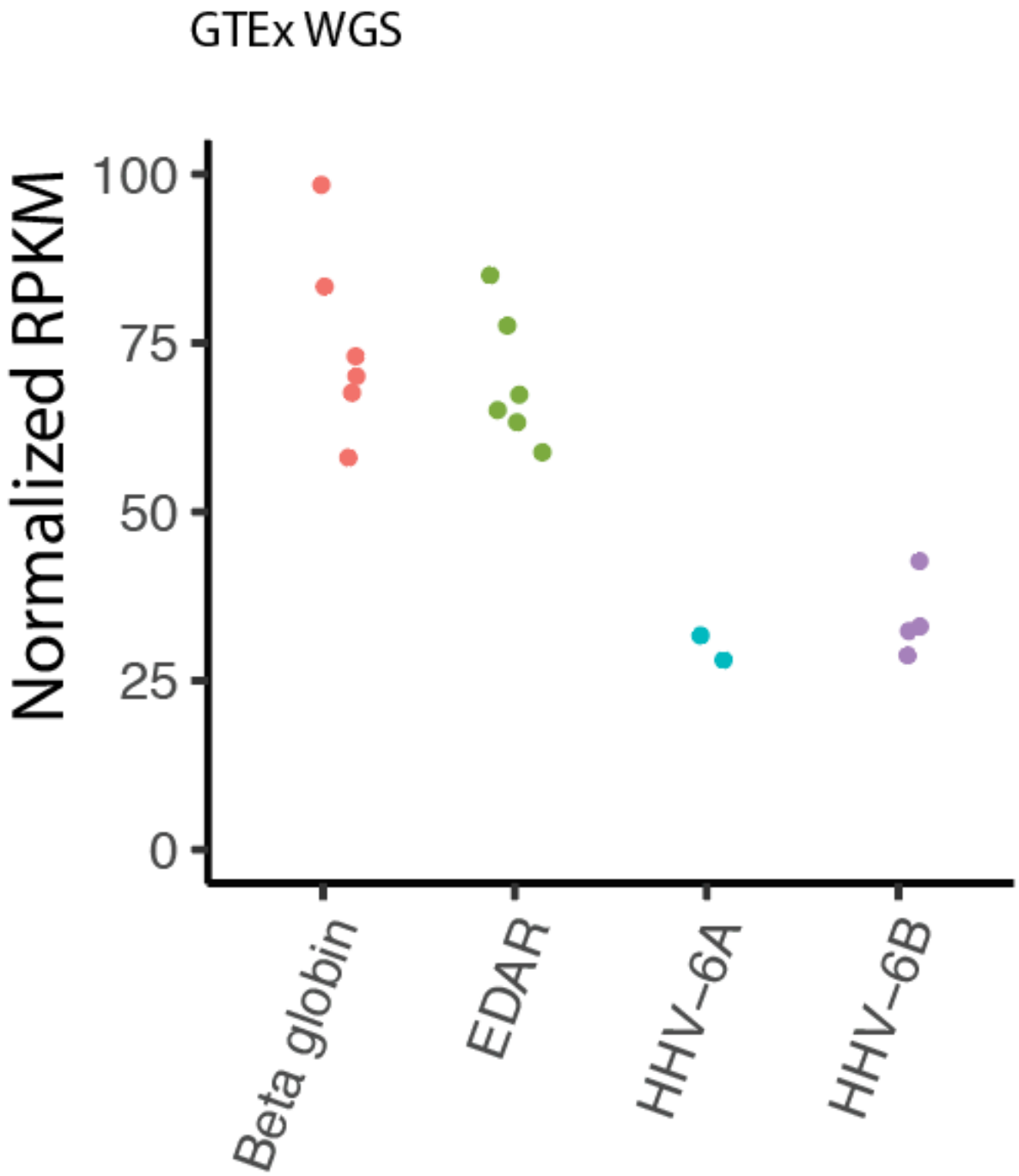

Figure 2| Normalized whole genome sequencing coverage of genes aligning to human housekeeping genes vs HHV-6A and HHV-6B for iciHHV-6 suspected donors in the GTEx dataset.

Off-target HHV-6 read coverage from whole exome sequencing also correctly detects iciHHV-6

Aligning reads from whole exome sequence (WES) revealed low but consistent coverage of HHV-6 in these 6 putative iciHHV-6-positive samples. Within the GTEx WES dataset the only samples with reads aligning to HHV-6 were the six iciHHV-6 positive samples by WGS (Figure 2). The mean depth of coverage for HHV-6 in the exome data from iciHHV-6 individuals was 0.27x, compared to 117x for beta-globin and 14.5x for EDAR. Screening of the other 603 available whole exome sequences revealed no HHV-6 sequence outside of the repeat regions, indicating a lack of iciHHV-6. We used the exome screening approach to screen the MSBB dataset of 350 individuals and found 4 iciHHV-6 positive individuals, again consistent with the expected population incidence of iciHHV-6 (Figure 3).

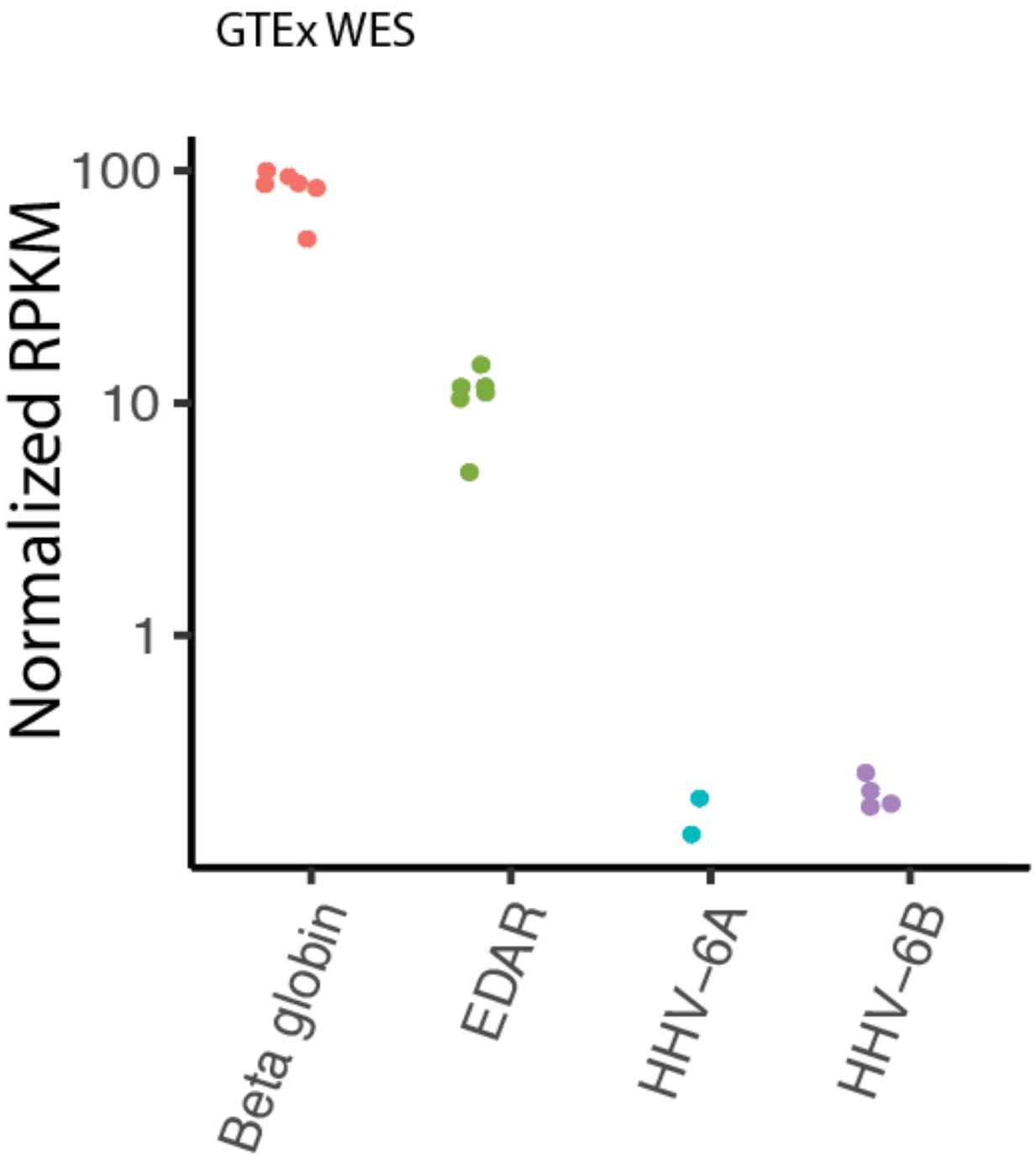

Figure 3| Normalized whole exome sequencing coverage of genes aligning to human housekeeping genes vs HHV-6A and HHV-6B for iciHHV-6 suspected donors in the

GTEx dataset.

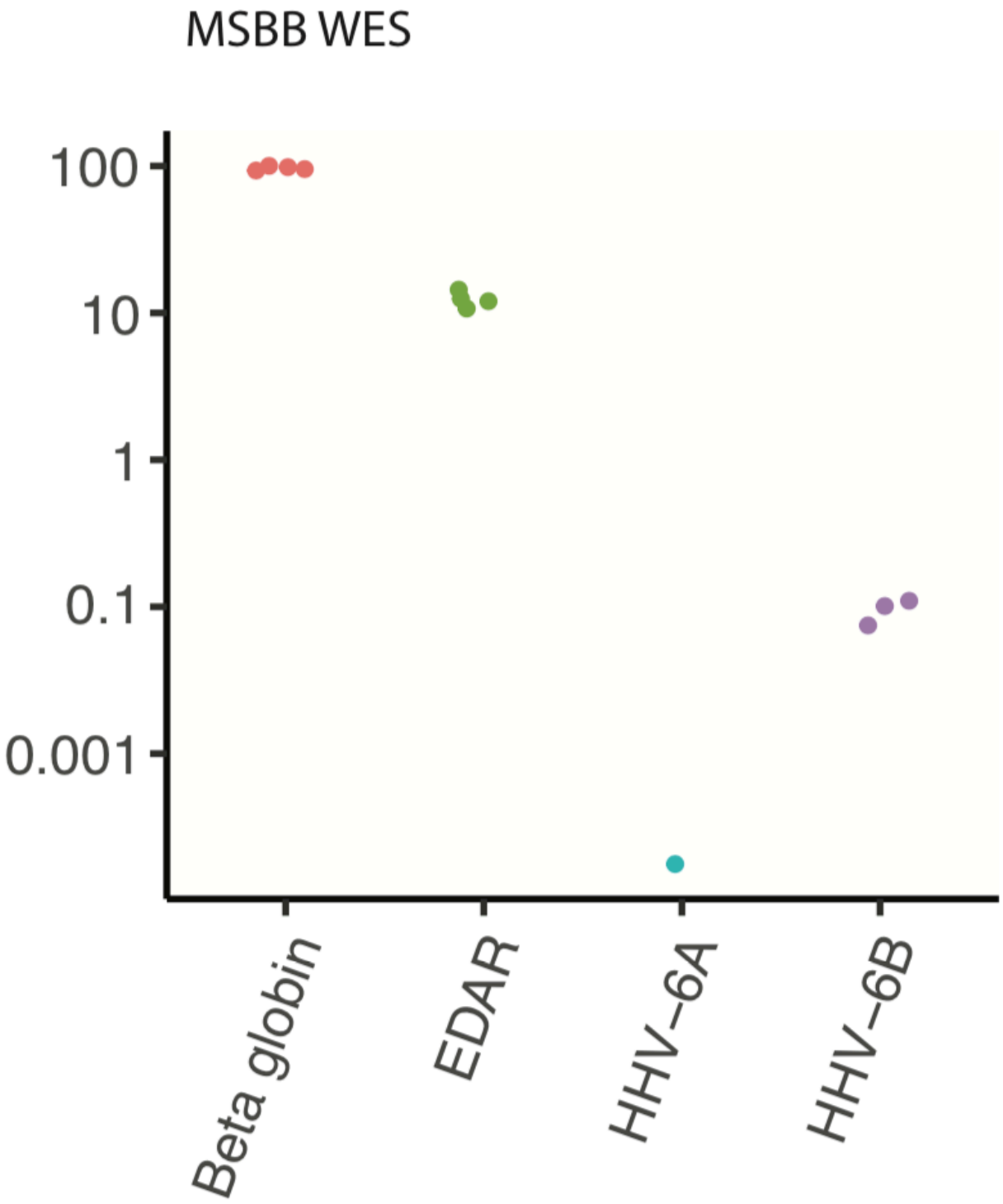

Figure 4| Normalized whole exome sequencing coverage of genes aligning to human housekeeping genes vs HHV-6A and HHV-6B for iciHHV-6 suspected donors in the MSBB dataset.

Phylogenetic trees reveal genetic high genetic similarity amongst iciHHV-6 sequences

Phylogenetic trees reveal clustering with previously deposited iciHHV-6 sequences (Figures 5 and 6). From the available genbank HHV-6B sequences there are two distinct HHV-6B clades visible: One Asian and one American/European with GTEx-OXRO falling into the former, and GTEx-13LV, -14C38, and -YF70 the latter. The American clade consists of iciHHV-6B sequences from the UK and Seattle, as well as HHV-6B sequences from New York (14). The Asian clade contains iciHHV-6B sequences from Pakistan and China, as well as HHV-6B sequences from Japan. It was not possible to build consensus sequences from the Synapse iciHHV-6 exome sequence samples due to insufficient coverage (Figures 3 and 4).

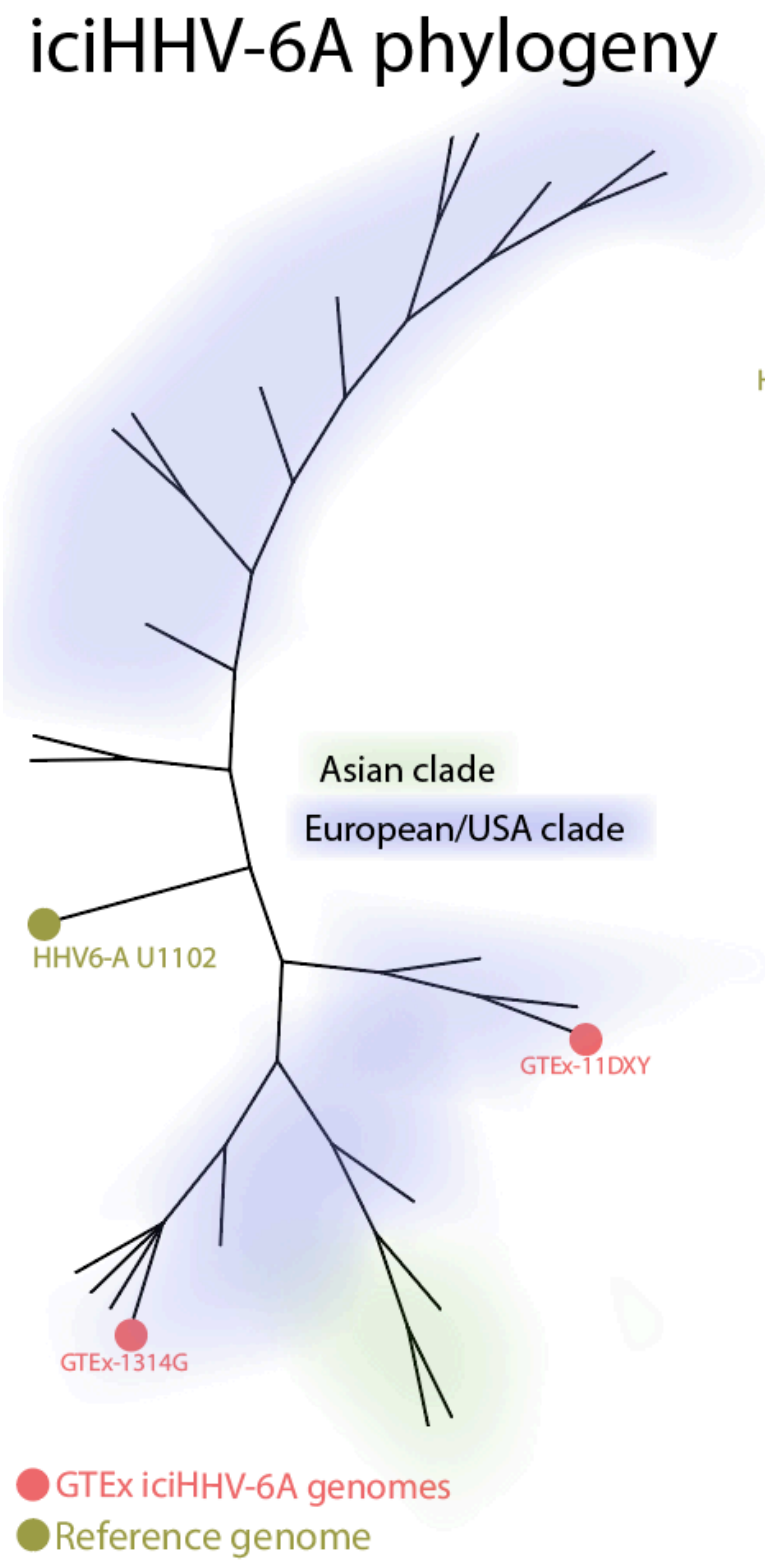

Figure 5| Phylogeny of GTEx iciHHV-6A sequences against other deposited iciHHV-6A sequences

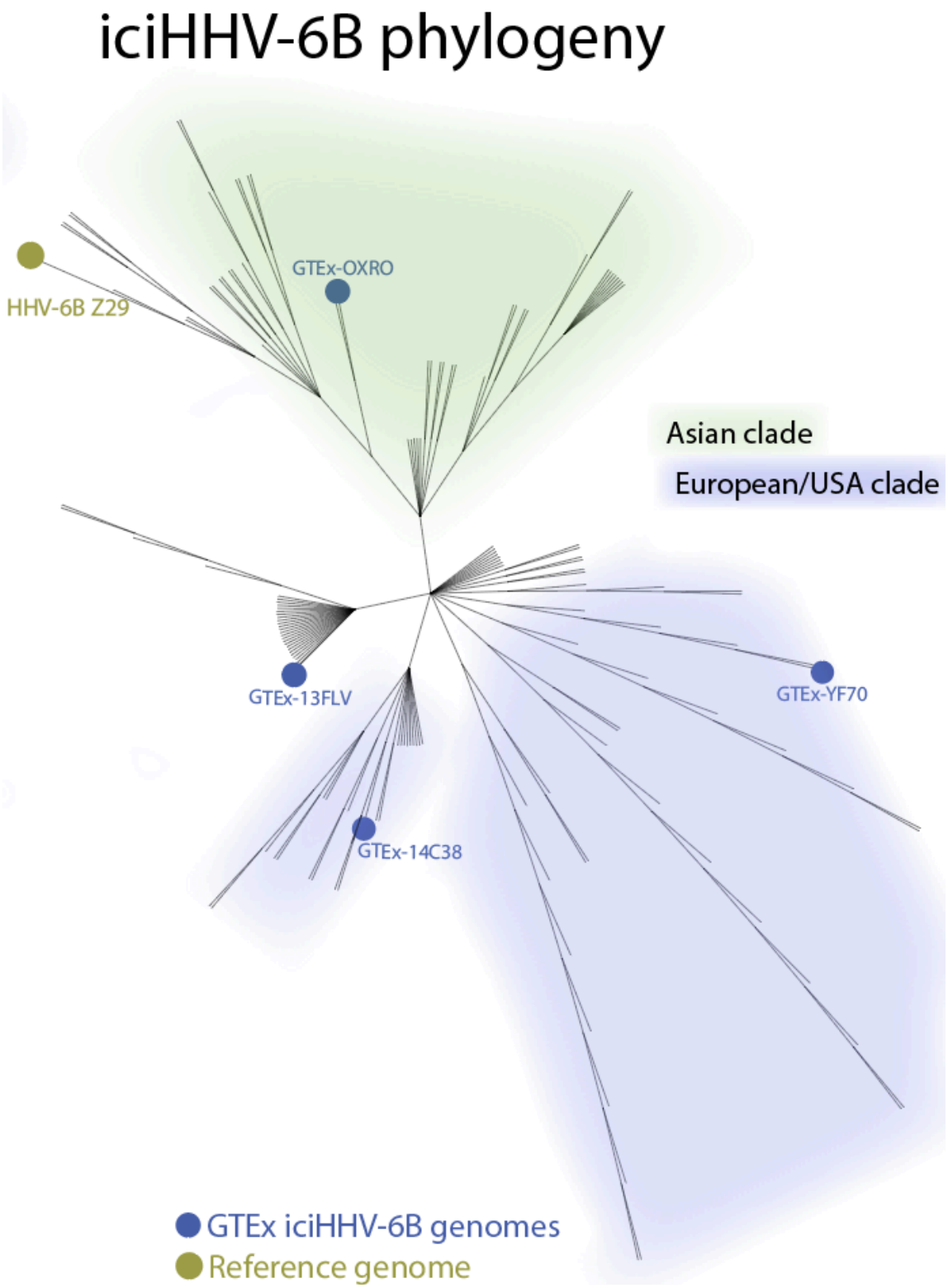

Figure 6| Phylogeny of GTEx iciHHV-6B sequences against other deposited HHV-6B and iciHHV-6B sequences

HHV-6 glycoprotein U100 and IE1 U90 genes are consistently expressed in various tissues for both iciHHV-6A and -6B

From the six individuals who tested positive for iciHHV-6 based on WGS and WES data, RNA-seq data was available from 111 tissues. Analysis of these transcriptomes showed variable tissue-specific activity with highest gene expression levels in U90-U100 genes for both iciHHV-6A and iciHHV-6B (Figure 7). In iciHHV-6B positive samples, the highest level of expression was seen in the U90 IE-1 transactivator and U100 glycoprotein Q genes. HHV-6 was most actively expressed in the testis, esophagus, and brain tissue. Expression of iciHHV-6A genes was notably higher than those of iciHHV-6B in brain tissue, as observed in the MSBB datasets (Figure 8). The RNAseq data for the 100 iciHHV-6 negative donors were also analyzed in the same way. No reads aligning to HHV-6 were found.

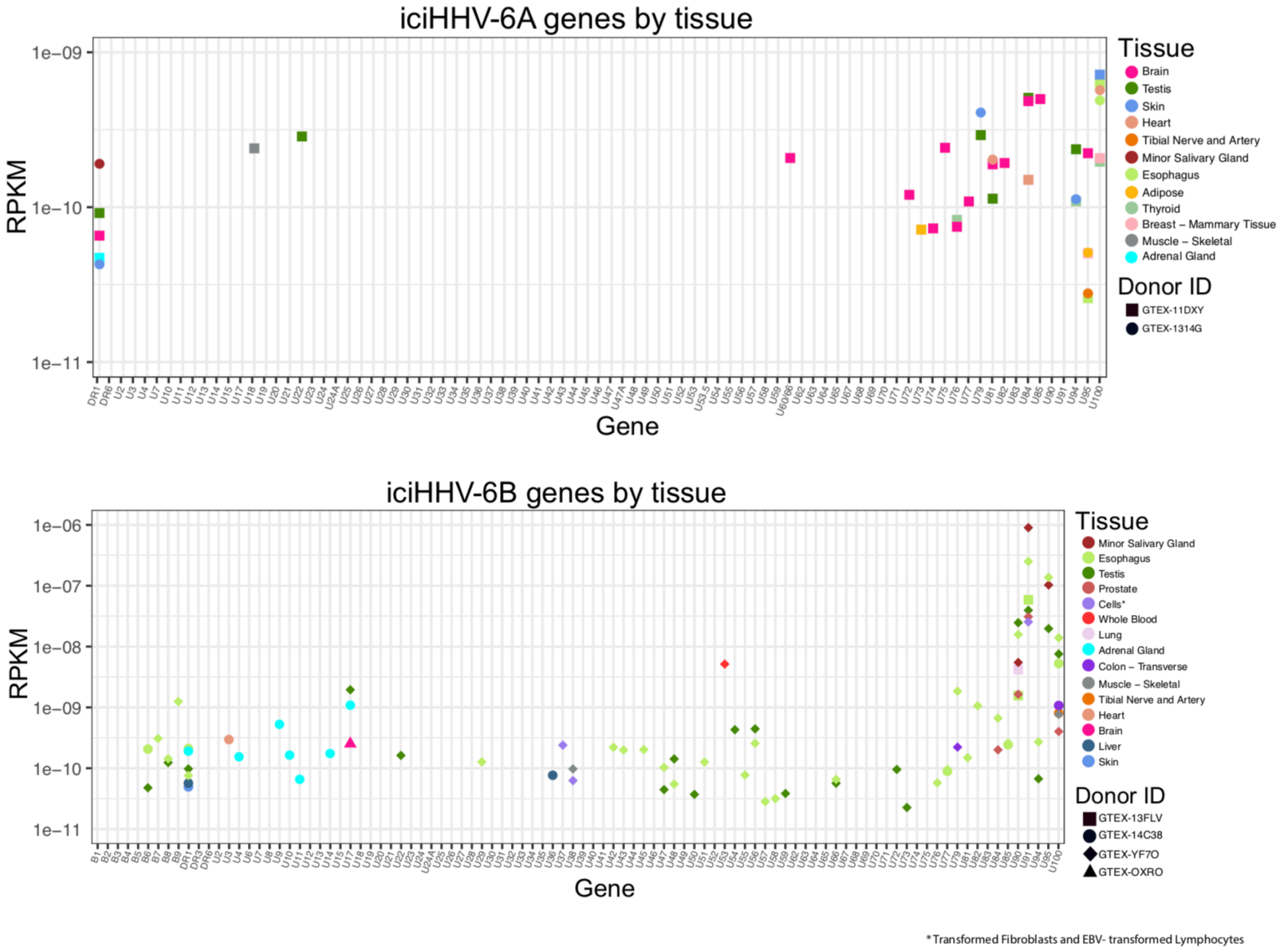

Figure 7| Normalized expression of HHV iciHHV-6A and B genes colored by tissue for the GTEx dataset.

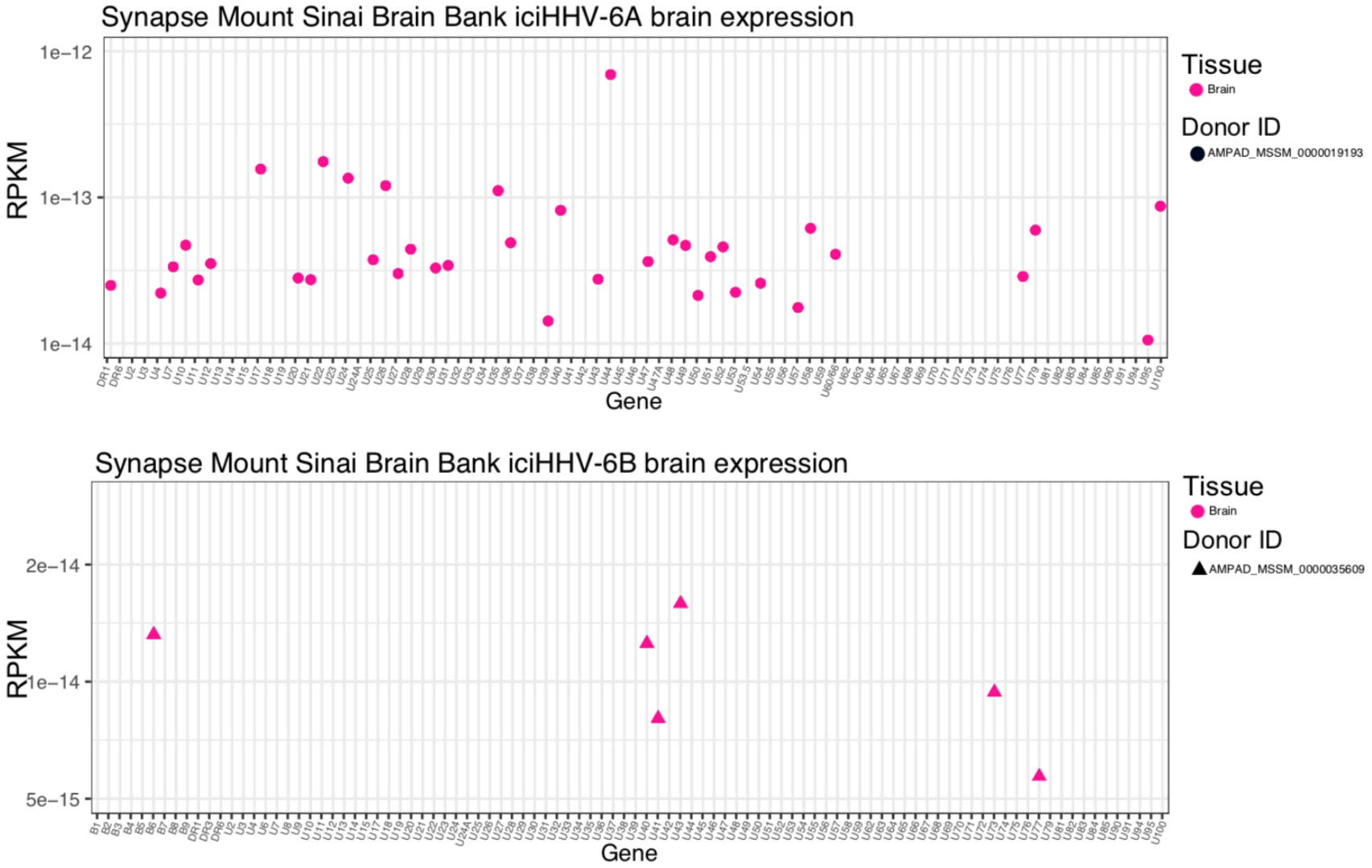

Figure 8| Normalized expression of HHV iciHHV-6A and B genes colored by tissue for the MSBB dataset.

The presence of reads spanning exons, comparison of SNPs, and insert mean sizes suggest RNAseq reads are from RNA

In the U100 gene one GTEx-YF7O read was found be on two consecutive exons, excluding the intron between. Two fragments were also found spanning two consecutive exons (Figure 9). Two GTEx-14C38 fragments were similarly found in the U100 gene as well. Three fragments from GTEx-13FLV were similarly found in the U90 gene. RNAseq reads from GTEx-OXRO were insufficient to perform the contamination analyses mentioned here.

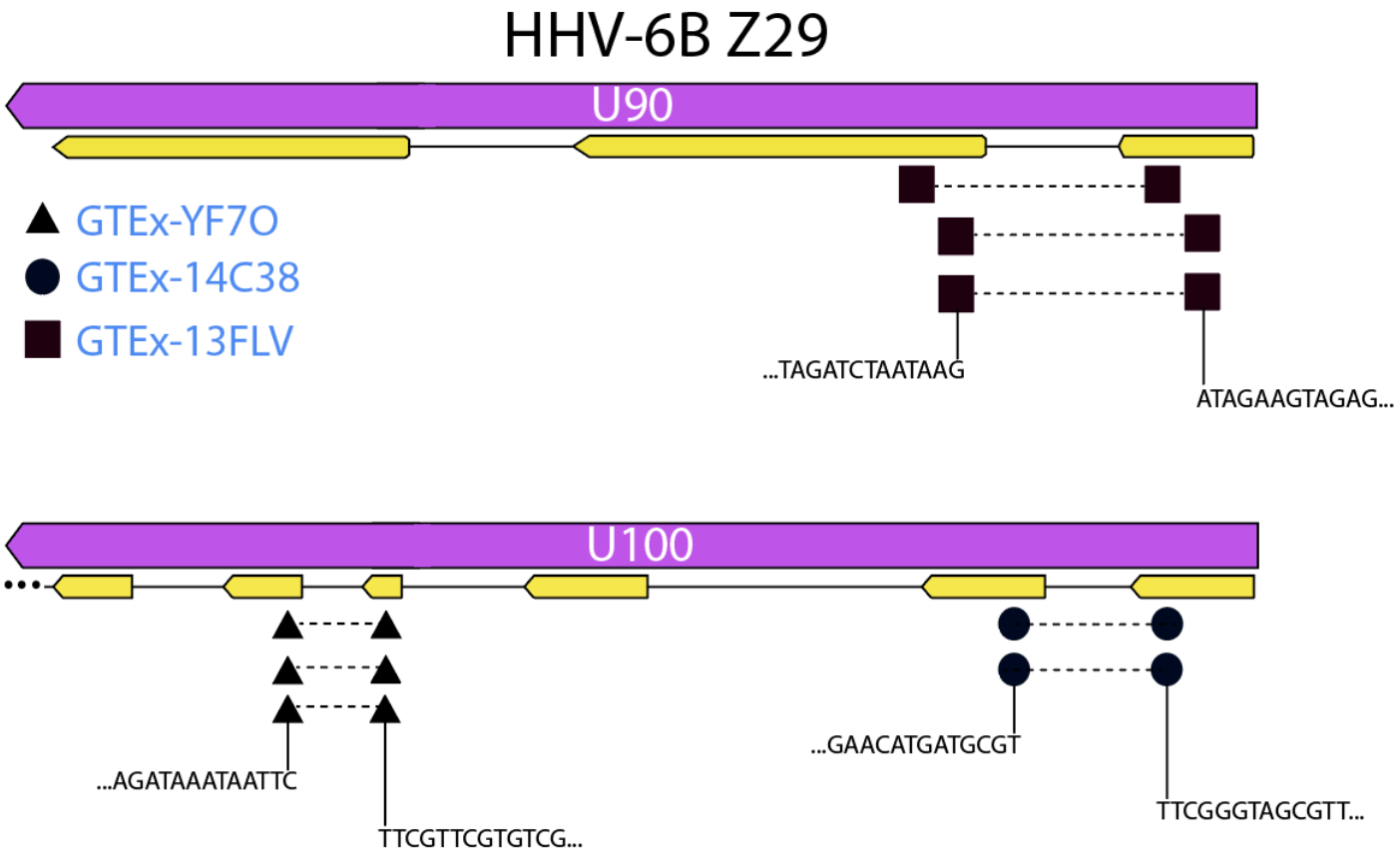

Figure 9| analysis of GTEx RNAseq reads spanning exons for 3 out of 4 samples. The remaining sample only had two fragments total, both of which were in the middle of genes

For the samples GTEx-13FLV, GTEx-14C38, and GTEx-YF7O SNPs were detected in multiple loci when aligned against the GTEx-OXRO consensus genome. SNPs found in the RNAseq with a depth of coverage of at least 2 were then manually verified against their respective WGS consensus sequences and found to be consistent (Figure 10).

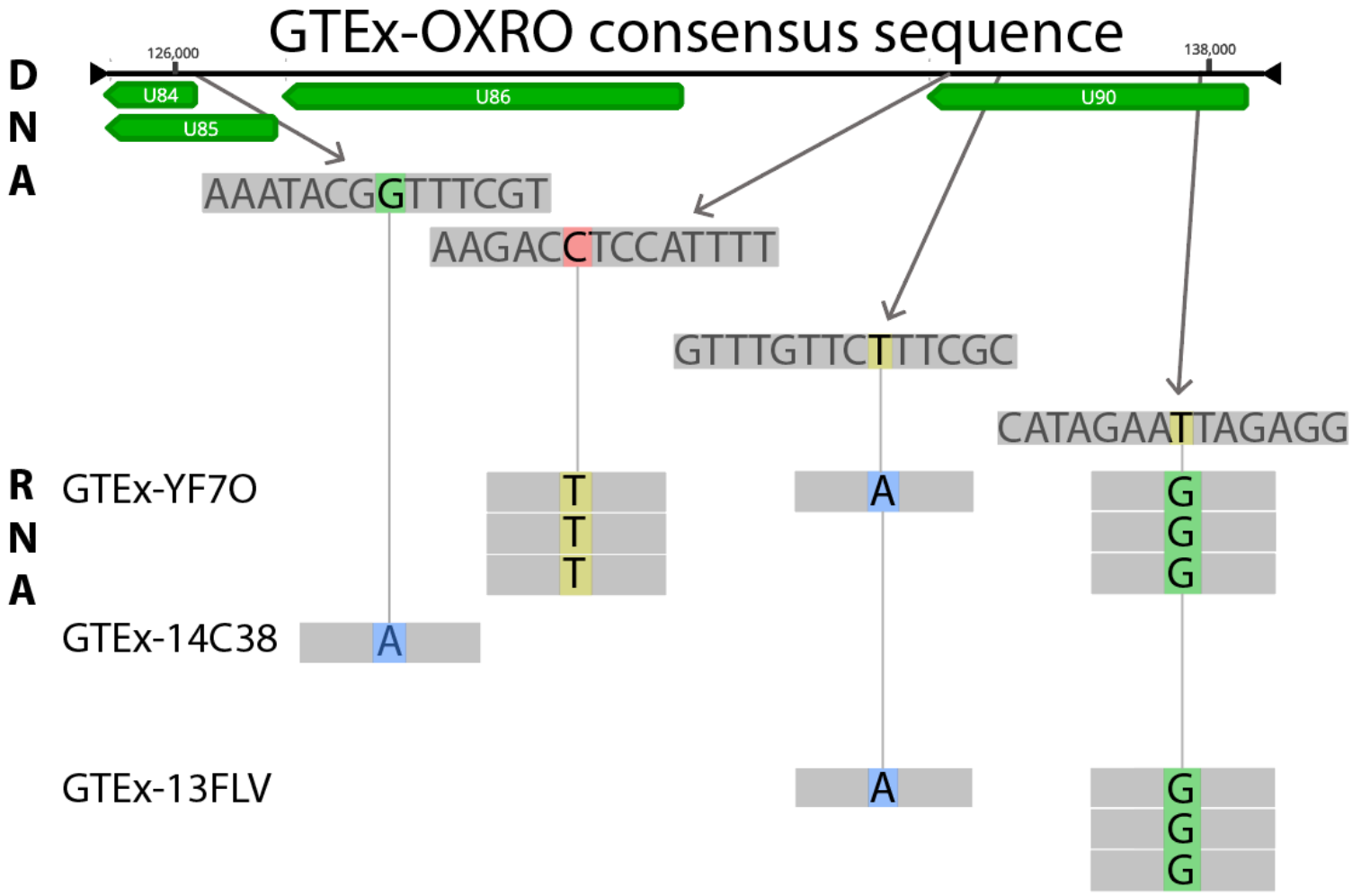

Figure 10| analysis of unique SNPs in GTEx RNAseq reads compared to the GTEx-OXRO consensus genome

U38 expression was detected in both iciHHV-6B positive and negative clinical PBMC samples

RT-qPCR for U38 was performed on 31 PBMC samples. Samples were determined as having viable RNA if there was over a two-cycle threshold difference in the with and without reverse transcriptase RPP30 treatments. Based on this, 29 samples were determined to be viable: 15 iciHHV-6 positive (6 iciHHV-6A, 9iciHHV-6B), and 9 iciHHV-6 negative but active infection positive (1 HHV-6A, 8 HHV-6B), and the remaining 5 were iciHHV-6 negative samples had no HHV-6 DNA detected at all. The quantitative results of this assay are shown below in table 4 and figure 11.

		type	U38 copies/microliter	U38 copies/mL
iciHHV-6 positive		B	153	6136
		B	81	3222
		B	0	0
		B	0	0
		B	0	0
		B	0	0
		B	0	0
		B	0	0
		B	0	0

		A	0	0
		A	0	0
		A	0	0
		A	0	0
		A	0	0
		A	0	0
iciHHV-6 negative	HHV-6 not detected by ddPCR	-	0	0
		-	0	0
		-	0	0
		-	0	0
		-	0	0
	HHV-6 detected by ddPCR	B	572	22876
		B	62	2480
		B	31	1222
		B	0	0
		B	0	0
		B	0	0
		B	0	0
		B	0	0
		A	0	0

Table 4| Distribution of clinical samples with their corresponding U38 RT-qPCR quantities.

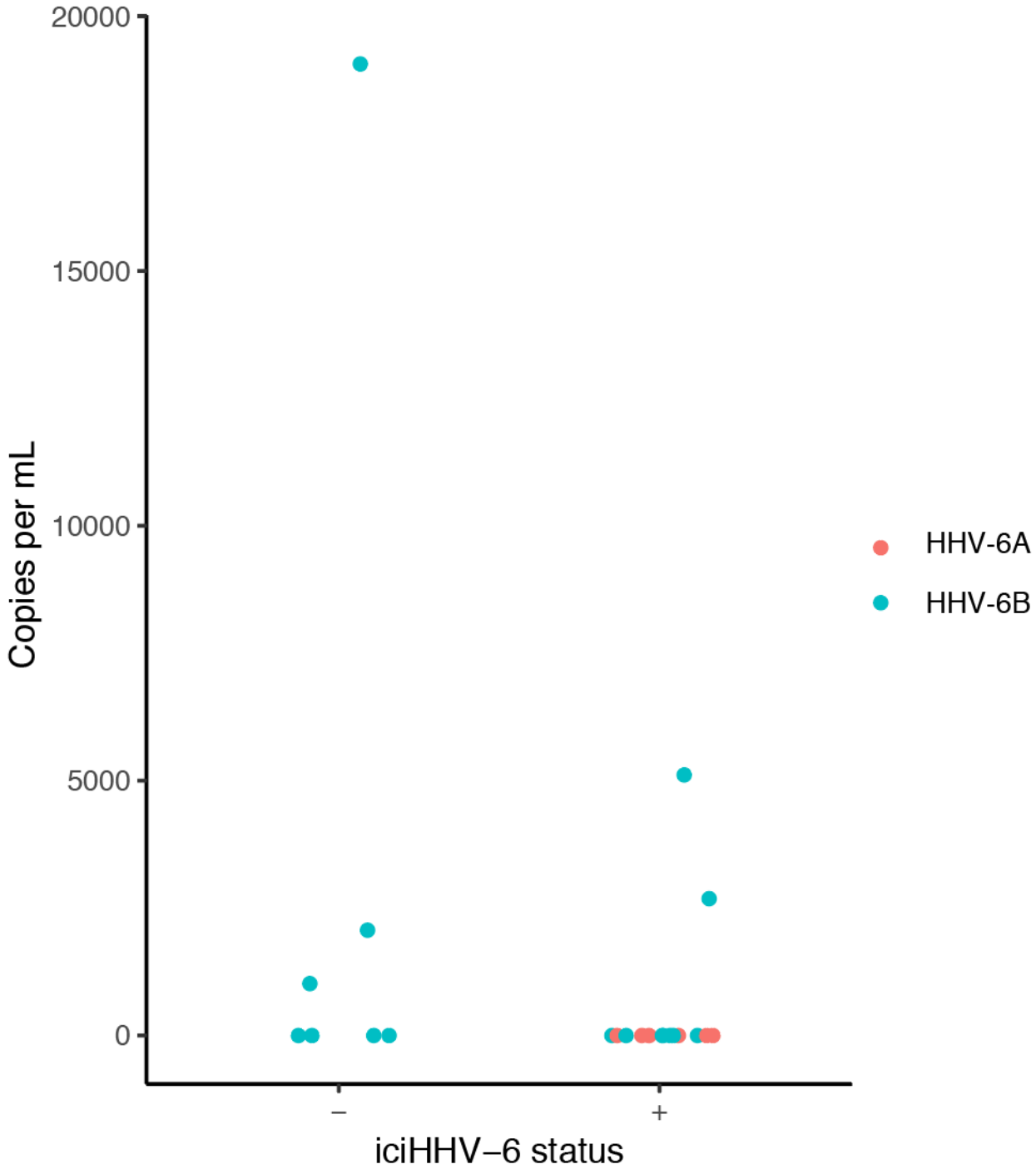

Figure 11| U38 RT-qPCR expression results for iciHHV-6 positive PBMC samples and iciHHV-6 negative but active infection positive PBMC samples. HHV-6 negative samples are not shown here.

Our U90 and U100 assays were unsuccessful in providing interpretable results

RT-ddPCR was run for the U90 and U100 genes in an attempt to use as little clinical sample as possible for each run. Unfortunately, the results for both U90 and U100 were uninterpretable, likely due to sample RNA degradation over time and repeated freeze-thaw cycles.

Chapter 5: Discussion

In the analysis of 650 available genomes we detected a 0.92% incidence of iciHHV-6 which matches with the previously reported rate of .5-2% (7). Phylogeny shows that three GTEx iciHHV-6 sequences clustered closely with previously deposited iciHHV-6B sequences. The remaining iciHHV-6B sample clustered with previously deposited Japanese iciHHV-6B sequences. Asian origin iciHHV-6B sequences have high sequence similarity (14), we believe this region specific sequence similarity, also supported by the clustering of the other three iciHHV6-B sequences, lends evidence to a founder effect for iciHHV-6B. In a similar fashion, we see close clustering in our two iciHHV-6A GTEx samples with other iciHHV-6A samples, also lending support to there *having* been a founder effect.

During our analysis of the GTEx RNAseq files we found iciHHV-6A and -6B are not equally active in all tissue. This suggests that there is tissue-dependent activity, however in most active tissue there was consistently high expression of the U90 and U100 genes relative to other genes for both iciHHV-6A and iciHHV-6B.

We also found that coverage from WES data could prove to be a reliable metric for detecting iciHHV-6 in WES sequence. In the GTEx dataset we first analyzed Whole Genome Sequence data to find possible iciHHV-6 candidates and confirmed the candidates by looking for a 1:2 ratio of coverage for housekeeping genes to HHV-6 genomes. Out of the 650 available WES samples in the GTEx dataset, the only ones that contained reads to HHV-6 were the six samples confirmed by our WGS screen (Figure 2).

Within the GTEx iciHHV-6 positive samples brain tissue was observed to be relatively highly active with 11 different genes being expressed (figure 7). Using the previously mentioned WES coverage as a screen we also screened the Synapse Mount Sinai Brain Bank (MSBB) dataset and found three iciHHV-6B sequences, and one iciHHV-6A sequence. We were unable to build consensus sequences for these files due to the relatively lower coverage of WES data as compared to WGS data.

Upon analysis of the corresponding RNA-seq reads we found that iciHHV-6 was expressed in the frontal pole, superior temporal gyrus, parahippocampal gyrus, and interior frontal gyrus. Of the three positive iciHHV-6B samples, only one was found to be expressing genes. This sample, AMPAD_MSSM_0000035609, was only active in the parahippocampal gyrus, and had only 6 genes expressed as opposed to 38 genes expressed for the iciHHV-6A sample (Figure 8). RPKM values for iciHHV-6B sequences was roughly 2-4 logs lower than were in the GTEx expression data.

Since much of the work described here was secondary data analysis, our study has a number of limitations. We were unable to obtain tissues or additional metadata from either the GTEx or MSBB datasets to confirm our work. We cannot rule out the possibility of pre-analytical errors such as trace DNA contamination being the cause of low levels of iciHHV-6 RNA expression. Where possible, we used specific HHV-6 SNPs to ensure no cross-sample contamination could account for recovery of HHV-6 reads and we confirmed every read by BLASTn analysis to NT.

From collected PBMC samples we detected expression of U38 in both iciHHV-6B positive and negative samples. In this analysis, however, it should be noted that since the collected samples were meant for a DNA-based ddPCR assay, substantial RNA

degradation is expected due to suboptimal storage conditions for RNA. Of the 15 iciHHV-6 positive samples, no iciHHV-6A samples were positive for U38 expression, but 2 iciHHV-6B samples were. Similarly, no U38 expression was detected in HHV-6A active infection samples but was in three HHV-6B active infection samples were. No HHV-6 negative samples were positive for U38 expression.

Our GTEx analysis shows low level expression of the iciHHV-6B U38 gene only in skeletal muscle tissue as well as transformed lymphocytes and fibroblasts. U38 is otherwise unexpressed in all tissue analyzed including whole blood. Though there is U38 activity in iciHHV-6 positive samples it remains unclear whether this is a reactivation of the iciHHV-6, or a superinfection with another strain of HHV-6. In order to assess this issue a possible follow up experiment with RNA-seq of similar samples could be done to compare SNPs in the expressed RNA against the iciHHV-6 positive DNA sequence.

References

1. Theodore WH, Epstein L, Gaillard WD, Shinnar S, Wainwright MS, Jacobson S. 2008. Human herpes virus 6B: A possible role in epilepsy? Epilepsia 49:1828–1837.

2. Advances in the Characterization of the T-Cell Response to Human Herpesvirus-6.

3. Crocchiolo R, Giordano L, Rimondo A, Bologna M, Sarina B, Morabito L, Bramanti S, Castagna L, Mineri R. 2016. Human Herpesvirus 6 replication predicts Cytomegalovirus reactivation after allogeneic stem cell transplantation from haploidentical donor. J Clin Virol 84:24–26.

4. Stanton R, Wilkinson GWG, Fox JD. 2003. Analysis of human herpesvirus-6 IE1 sequence variation in clinical samples. Journal of Medical Virology 71:578–584.

5. Trempe F, Gravel A, Dubuc I, Wallaschek N, Collin V, Gilbert-Girard S, Morissette G, Kaufer BB, Flamand L. 2015. Characterization of human herpesvirus 6A/B U94 as ATPase, helicase, exonuclease and DNA-binding proteins. Nucleic Acids Research 43:6084–6098.

6. Gravel A, Dubuc I, Wallaschek N, Gilbert-Girard S, Collin V, Hall-Sedlak R, Jerome KR, Mori Y, Carbonneau J, Boivin G, Kaufer BB, Flamand L. 2017. Cell Culture Systems To Study Human Herpesvirus 6A/B Chromosomal Integration. J Virol 91.

7. Agut H, Bonnafous P, Gautheret-Dejean A. 2015. Laboratory and Clinical Aspects of Human Herpesvirus 6 Infections. Clinical Microbiology Reviews 28:313–335.

8. Sedlak RH, Cook L, Huang M-L, Magaret A, Zerr DM, Boeckh M, Jerome KR. 2014. Identification of Chromosomally Integrated Human Herpesvirus 6 by Droplet Digital PCR. Clinical Chemistry 60:765–772.

9. Greninger AL, Knudsen GM, Roychoudhury P, Hanson DJ, Sedlak RH, Xie H, Guan J, Nguyen T, Peddu V, Boeckh M, Huang M-L, Cook L, Depledge DP, Zerr DM, Koelle DM, Gantt S, Yoshikawa T, Caserta M, Hill JA, Jerome KR. 2018. Comparative genomic, transcriptomic, and proteomic reannotation of human herpesvirus 6. BMC Genomics 19.

10. Hill JA, Ikoma M, Zerr DM, Basom RS, Peddu V, Huang M-L, Sedlak RH, Jerome KR, Boeckh M, Barcy S. 2018. RNA sequencing of the in vivo human herpesvirus 6B transcriptome to identify targets for clinical assays distinguishing between latent and active infections. Journal of Virology JVI.01419-18.

11. GTEx Portal.

12. Readhead B, Haure-Mirande J-V, Funk CC, Richards MA, Shannon P, Haroutunian V, Sano M, Liang WS, Beckmann ND, Price ND, Reiman EM, Schadt EE, Ehrlich ME, Gandy S, Dudley JT. 2018. Multiscale Analysis of Independent Alzheimer's Cohorts Finds Disruption of Molecular, Genetic, and Clinical Networks by Human Herpesvirus. Neuron 99:64-82.e7.

13. Langmead B, Salzberg SL. 2012. Fast gapped-read alignment with Bowtie 2. Nature Methods 9:357–359.

14. Wang M, Beckmann ND, Roussos P, Wang E, Zhou X, Wang Q, Ming C, Neff R, Ma W, Fullard JF, Hauberg ME, Bendl J, Peters MA, Logsdon B, Wang P, Mahajan M, Mangravite LM, Dammer EB, Duong DM, Lah JJ, Seyfried NT, Levey AI, Buxbaum JD, Ehrlich M, Gandy S, Katsel P, Haroutunian V, Schadt E, Zhang B. 2018. The Mount Sinai cohort of large-scale genomic, transcriptomic and proteomic data in Alzheimer's disease. Scientific Data 5:180185.

15. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. 2009. The Sequence Alignment/Map format and SAMtools. Bioinformatics 25:2078–2079.

16. Pantry SN, Medveczky PG. 2017. Latency, Integration, and Reactivation of Human Herpesvirus-6. Viruses 9.

17. Prusty BK, Gulve N, Rasa S, Murovska M, Hernandez PC, Ablashi DV. 2017. Possible chromosomal and germline integration of human herpesvirus 7. Journal of General Virology 98:266–274.

Code Appendix

Generation of gene expression figures (R):

```
 library(Biostrings)
library(Rsamtools)
library(GenomicAlignments)
library(edgeR)
library(rtracklayer)
library(xlsx)
library(ggplot2)
library(gtools)
library(plotrix)
library(foreach)
library(doParallel)
library(seqinr)
library(scales)
library(RColorBrewer)
library(svMisc)

setwd("/fh/scratch/delete30/jerome_k/vikas/Thesis_from_other_mac/redownload
ed_hhv6a/")

data <- read.xlsx("gtex_icihhv6_hhv6b_positives.xlsx", sheetIndex = 2)
datatrimmed <- data[, -c(5:9),(12:13)]
data<-data[complete.cases(data$Run), ]
data<-data[data$LibrarySelection=='cDNA',]
#reads z29 gff3, trims out repeats
gff3<-readGFF('hhv6areference.gff3',version=3,columns=NULL, tags=NULL,
filter=NULL, nrows=-1, raw_data=FALSE)
for (j in 1:3) {
  for( i in 1:nrow(gff3)){
    if(( gff3$rpt_type[i]=='TANDEM')==TRUE && is.na(gff3$rpt_type[i])==FALSE){
      print('ok')
      gff3<-gff3[-i,]

    }
  }
}

for (j in 1:3) {
  for( i in 1:nrow(gff3)){
```

```
    if(( gff3$rpt_type[i]=='TERMINAL')==TRUE &&
is.na(gff3$rpt_type[i])==FALSE){
      print('ok')
      gff3<-gff3[-i,]

    }
  }
}

genedf<-as.data.frame(gff3)


#deletes U86
gff3<-gff3[!is.na(gff3$gene),]
gff3<-gff3[!(gff3$gene=='U86'),]

#turns sam files into bam files. Merges all bam files into one big file
system('for i in *.sam; do samtools view -Sb $i > $i.bam; done ')

srr<-c()
read_id<-c()
read_seq<-c()
body_site<-c()
read_pos<-c()
sample_identifier<-c()

filelist<-list.files(pattern = "*.bam")

###takes forever but let it run against localblast later
##more reproducible
##read_ids have srr number in them so you can sort later
### run this whole shit against localblast
for( i in 1:length(filelist)){
  #progress(i)
  #print(filelist[i])
  tempbam<-scanBam(filelist[i])
  for(j in 1:length(tempbam[[1]]$qname)){
    #progress(j)
    ####setting thresholds for where to pull reads from. This avoids a bunch of
repeat reads####
    if( tempbam[[1]]$pos[j] > 10000 && tempbam[[1]]$pos[j] < 155000) {
      #print('ok')
```

```
      srr<-append(srr,strsplit(filelist[i],split = '.sam')[[1]][1])
      read_id<-append(read_id,tempbam[[1]]$qname[j])
      read_seq<-append(read_seq,tempbam[[1]]$seq[j])
      read_pos<-append(read_pos,tempbam[[1]]$pos[j])
    }
  }
}

alldata<-data.frame(srr,read_id,read_seq,read_pos)

fastaFile<-alldata[,c(2,3)]
fastaFile$read_id<-as.character(fastaFile$read_id)

system('mkdir fastas')
setwd('fastas')
for (i in 1:nrow(fastaFile)){
  print(i/nrow(fastaFile)*100)
  write.fasta(fastaFile$read_seq[i], fastaFile$read_id[i],
paste0(fastaFile$read_id[i],'.fasta'), open = "w", nbchar = 200, as.string = TRUE)

}

system("find . -path './*fasta' -prune -type f -exec cat {} + > big-fasta.txt")
######need to change this line if changing wd######
system("blastn -query
/Users/gerbix/Documents/vikas/Thesis_from_other_mac/redownloaded_hhv6a/fa
stas/big-fasta.txt -db /Users/gerbix/Downloads/blast_viruses/all_virus.fasta -
num_threads 8 -out blast_results.txt")
system("python
/Users/gerbix/Documents/vikas/Thesis_from_other_mac/hhv6b_rna/testing/blast
_hits.py
/Users/gerbix/Documents/vikas/Thesis_from_other_mac/redownloaded_hhv6a/fa
stas/blast_results.txt herpesvirus 6")
####run local blast on that file###
###run the verison of blast_hits for the local database on the imac###
###make dataframe for any reads that survive only###
##pull info like gene location etc for those positive files only###

blast_hits <- read.csv('blast_hits.csv')

colnames(blast_hits)[1]<-'read_id'
colnames(blast_hits)[2]<-'hits'
```

```
blast_hits<-blast_hits[blast_hits$hits>0,]
blast_hits$sample<-NA
for(i in 1:nrow(blast_hits)){
  #progress(i)
  blast_hits$sample[i]<-strsplit(as.character(blast_hits$read_id[i]), "[.]")[[1]][1]
}

positivebams<-unique(blast_hits$sample)

setwd("/fh/scratch/delete30/jerome_k/vikas/Thesis_from_other_mac/redownload
ed_hhv6a/")


bamslist<-c()
for (i in positivebams){
  tempbamname<-paste0(i, '.sam.bam')
  temp<-scanBam(tempbamname)
  bamslist<-append(bamslist, temp)
}

total_list<-blast_hits
total_list$start = NA
total_list$end = NA

alldata_readids<-alldata$read_id

#threshold is at least 1 hit for each read on blast viral
blast_hits = blast_hits[blast_hits$hits>0,]

#deletes any reads from the original data frame that don't have any blast hits
failslist<-c()
for( i in 1:length(alldata_readids)) {
  print(i)
  if( alldata_readids[i]  %in% blast_hits$read_id) { }
  else {
    failslist <- append(failslist,i)
  }
}

blast_trimmed_alldata<-alldata

for(failed in failslist) {
  blast_trimmed_alldata$read_id[failed]<-NA
```

```
}

blast_trimmed_alldata<-
blast_trimmed_alldata[complete.cases(blast_trimmed_alldata), ]
blast_trimmed_alldata$gene = NA
blast_trimmed_alldata$count = 0
blast_trimmed_alldata$read_pos <- as.integer(blast_trimmed_alldata$read_pos)
blast_trimmed_alldata$read_id<-as.character(blast_trimmed_alldata$read_id)
blast_trimmed_alldata$read_seq<-
as.character(blast_trimmed_alldata$read_seq)
blast_trimmed_alldata$srr<-as.character(blast_trimmed_alldata$srr)
blast_trimmed_alldata$transcriptlength <- 0

genepositions<-data.frame(gff3$gene)
colnames(genepositions)[1]<-'gene'
genepositions$gene<-as.character(genepositions$gene)
genepositions$start<-gff3$start
genepositions$end<-gff3$end


duplicatesrr<-c()
duplicatereadid<-c()
duplicatereadseq<-c()
duplicatepos<-c()
duplicategenes<-c()
duplicatecount<-c()
duplicatetranscriptlength<-c()

blasttrimmedlength = nrow(blast_trimmed_alldata)
for (i in 1:blasttrimmedlength){
  print(100*i/blasttrimmedlength)
  for(j in 1:nrow(genepositions)){
    #start and end positions for each gene in the gff3
    tempstart = as.integer(genepositions$start[j])
    tempend = as.integer(genepositions$end[j])
    temptranscriptlength = tempend - tempstart
    #if there's already a gene associaated with the read this changes the count to
.5 and adds .5 to the duplicate list for that read
    if(blast_trimmed_alldata$read_pos[i]<tempend &&
blast_trimmed_alldata$read_pos[i]>tempstart){

      if(blast_trimmed_alldata$count[i] == 1){
        duplicatesrr<-append(duplicatesrr,blast_trimmed_alldata$srr[i])
```

```
      duplicatereadid<-append(duplicatereadid, blast_trimmed_alldata$read_id[i])
      # print(blast_trimmed_alldata$read_id[i])
      duplicatereadseq<-append(duplicatereadseq,
blast_trimmed_alldata$read_seq[i])
      duplicatepos<-append(duplicatepos, blast_trimmed_alldata$read_pos[i])
      duplicategenes<-append(duplicategenes, genepositions$gene[j])
      duplicatecount<-append(duplicatecount, '.5')
      blast_trimmed_alldata$count[i] = .5
      duplicatetranscriptlength <- append(duplicatetranscriptlength,
temptranscriptlength)
    }
    #if the read hasn't been counted yet this gives it a value of 1 (only found once
so far)
    else {
      blast_trimmed_alldata$gene[i]<- genepositions$gene[j]
      blast_trimmed_alldata$count[i] = 1
      blast_trimmed_alldata$transcriptlength[i] = temptranscriptlength
    }
  }
 }
}

duplicates<-data.frame(duplicatesrr, duplicatereadid, duplicatereadseq,
duplicatepos, duplicategenes, duplicatecount,duplicatetranscriptlength)

colnames(duplicates)[1] <- 'srr'
colnames(duplicates)[2] <- 'read_id'
colnames(duplicates)[3] <- 'read_seq'
colnames(duplicates)[4] <- 'read_pos'
colnames(duplicates)[5] <- 'gene'
colnames(duplicates)[6] <- 'count'
colnames(duplicates)[7] <- 'transcriptlength'
deduplicated = rbind(blast_trimmed_alldata, duplicates)
deduplicated = deduplicated[deduplicated$count>0,]

deduplicated$rpkm <-0



#dataframe for SRR
datadf<-as.data.frame(data$Run)
colnames(datadf)[1]<-'SRR'
datadf$body_site<-data$body_site
```

```
datadf$sampleID<-data$submitted_subject_id
datadf<-datadf[complete.cases(datadf), ]

#dataframe for gene info
countdf<-data.frame(as.character(gff3$gene))
colnames(countdf)[1]<-'gene'
countdf[2]<-gff3$start
colnames(countdf)[2]<-'start'
countdf[3]<-gff3$end
colnames(countdf)[3]<-'end'
countdf<-na.omit(countdf)
countdf[4]<-0
colnames(countdf)[4]<-'count'
#######
countdf$transcriptlength<-countdf$end-countdf$start
countdf$gene<-as.character(countdf$gene)
largestgene<-c()
largeststart<-c()
largestend<-c()
uniquegenes<-unique(countdf$gene)
for(i in 1:length(uniquegenes)) {
  templargestgene<-c()
  tempstart<-c()
  tempend<-c()
  templength<-c()
  for(j in 1:nrow(countdf)){
    if(uniquegenes[i]==countdf$gene[j]){
      templargestgene<-append(templargestgene,countdf$gene[j])
      tempstart<-append(tempstart,countdf$start[j])
      tempend <- append(tempend,countdf$end[j])
      templength<-append(templength,countdf$transcriptlength[j])

    }
  }
  largestgene<-append(largestgene,templargestgene[1])
  largeststart<-append(largeststart,max(tempstart))
  largestend<-append(largestend,max(tempend))
}

countdf<-data.frame(largestgene,largeststart,largestend)
countdf$count<-0
countdf$transcriptlength<-0
```

```
colnames(countdf)[1]<-'gene'
colnames(countdf)[2]<-'start'
colnames(countdf)[3]<-'end'

count<-0
df_length<-nrow(countdf) * nrow(datadf)
totaldf<-data.frame(matrix(ncol = 7, nrow = df_length))
for (i in 1:nrow(datadf)) {
  for (j in 1:(nrow(countdf))){
    count<-count+1
    totaldf[count,1]<-as.character(datadf$SRR[i])
    colnames(totaldf)[1]<-'SRR'
    totaldf[count,2]<-as.character(datadf$sampleID[i])
    colnames(totaldf)[2]<-'sample_id'
    totaldf[count,3]<-as.character(datadf$body_site[i])
    colnames(totaldf)[3]<-'body_site'
    totaldf[count,4]<-as.character(countdf$gene[j])
    colnames(totaldf)[4]<-'gene'
    totaldf[count,5]<-as.integer(countdf$start[j])
    colnames(totaldf)[5]<-'start'
    totaldf[count,6]<-as.integer(countdf$end[j])
    colnames(totaldf)[6]<-'end'
    totaldf[count,7]<-as.character(countdf$count[j])
    colnames(totaldf)[7]<-'count'
  }
  print(100*i/nrow(countdf))
}
totaldf$transcriptlength<-totaldf$end-totaldf$start
totaldf$rpkm<-0




#aggregating counts for the same SRR and same gene
#sum goes into totaldf$count
templist<-c()
totaldf$count<-as.double(totaldf$count)
for ( i in 1:nrow(totaldf)){
  tempgene<-totaldf$gene[i]
  tempsrr<-totaldf$SRR[i]
  tempsum = deduplicated$count[(deduplicated$srr == tempsrr &
deduplicated$gene == tempgene)]
  if( identical(tempsum, character(0))){}
```

```
  else {
    print(tempsum)
    tempsum = as.double(tempsum)
    #print(tempsum)
    print(sum(tempsum))
    #print('\n')
    totaldf$count[i] <- sum(tempsum)
    templist<-append(templist,sum(tempsum))
  }
}

#rpkm calculation
rpkmdf<-totaldf[totaldf$count>0,]
rpkmdf$rpkm<-0
for(i in 1:nrow(rpkmdf)){
  total_reads_per_sample<-
data$MBases[rpkmdf$SRR[i]==as.character(data$Run)]
  temptranscriptlength<-rpkmdf$transcriptlength[i]/1000
  total_reads_per_sample<-total_reads_per_sample*1000000
  rpkmdf$rpkm[i] <-
(rpkmdf$count[i])/(temptranscriptlength*total_reads_per_sample)
}

#making graph

#df so all genes pop up on x axis
genedf<-as.data.frame(gff3$gene)
colnames(genedf)[1]<-'gene'
genedf[2]<-0
colnames(genedf)[2]<-'count'
genedf<-aggregate(genedf$count~genedf$gene,data=genedf,FUN=sum)
colnames(genedf)[1]<-'gene'
colnames(genedf)[2]<-'count'
genedf[3]<-0
colnames(genedf)[3]<-'SRR'
genedf[4]<-0
colnames(genedf)[4]<-'body_site'
genedf[5]<-0
colnames(genedf)[5]<-'sample_id'
genedf[6]<-0
colnames(genedf)[6]<-'start'
genedf[7]<-0
colnames(genedf)[7]<-'end'
```

```
genedf[8]<-0
colnames(genedf)[8]<-'transcriptlength'
genedf[9]<-0
colnames(genedf)[9]<-'count'
genedf<-genedf[,c(3,5,4,1,6,7,2)]
genedf$transcriptlength<-0
genedf$rpkm<-0

df_for_graph<-rbind(rpkmdf,genedf)


df_for_graph$gene<-as.character(df_for_graph$gene)
df_for_graph<-df_for_graph[mixedorder(df_for_graph$gene),]
df_for_graph$order<-1:nrow(df_for_graph)

colourCount<-length(unique(df_for_graph$SRR))
getPalette = colorRampPalette(brewer.pal(9, "Set1"))
tryingcolors<-(color = brewer.pal(8, "Dark2"))
tryingcolors2<-(color = brewer.pal(8, "Set1"))
tryingcolors3<-(color = brewer.pal(8, "Set2"))
coloring = c("#89C5DA", "#DA5724", "#74D944", "#CE50CA", "#3F4921",
"#C0717C", "#CBD588", "#5F7FC7",
           "#673770", "#D3D93E", "#38333E", "#508578", "#D7C1B1", "#689030",
"#AD6F3B", "#CD9BCD",
           "#D14285", "#6DDE88", "#652926", "#7FDCC0", "#C84248", "#8569D5",
"#5E738F", "#D1A33D",
           "#8A7C64", "#599861","#0000ff","#ff0000")
allcolors<-append(coloring,tryingcolors)
allcolors<-append(allcolors,tryingcolors2)
allcolors<-append(allcolors,tryingcolors3)


for( i in 1:nrow(df_for_graph)){
  if(df_for_graph$count[i]==0){
    df_for_graph$count[i]<-NA
    df_for_graph$body_site[i]<-NA
    df_for_graph$sample_id[i]<-NA
    df_for_graph$rpkm[i]<-NA
    #ordered_final_df$pos[i]<-NA
  }
}
```

```
graph_2<-ggplot(df_for_graph,aes(x=reorder(gene, order), y=rpkm)) +
geom_point(aes(shape=sample_id, color=body_site), size=4)
graph_2 +scale_colour_manual(values=allcolors) +
  coord_trans(y = "log10") +
  theme(legend.key.size = unit(.05, "cm")) +
  theme_bw() +
  scale_shape_manual(values=c(15, 16, 17, 18,20))+
  theme(axis.text.x=element_text(angle=70,vjust=.5,hjust=.9)) +
  labs(x = "Gene")

#condensing graph by body sites
condenseddf<-df_for_graph

for(i in 1:nrow(condenseddf)) {
  if( is.na(condenseddf$body_site[i])==FALSE & condenseddf$body_site[i] ==
'Pituitary'){
    condenseddf$body_site[i] <-"Brain - Pituitary"
  }
}

#condenses all brain tissue names to just brain
for(i in 1:nrow(condenseddf)) {
  if( !isEmpty(grep('Brain', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Brain'
  }
}
#condenses all esophagus tissue names to just esophagus
for(i in 1:nrow(condenseddf)) {
  if( !isEmpty(grep('Esophagus', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Esophagus'
  }
}
#condenses all skin tissue names to just skin
for(i in 1:nrow(condenseddf)) {
  if( !isEmpty(grep('Skin', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Skin'
  }
}
#condenses all Adipose tissue names to just Adipose
for(i in 1:nrow(condenseddf)) {
```

```
  if( !isEmpty(grep('Adipose', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Adipose'
  }
}

#condenses all Heart tissue names to just Heart
for(i in 1:nrow(condenseddf)) {
  if( !isEmpty(grep('Heart', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Heart'
  }
}

#condenses all tibial tissue names to just tibial nerve and artery
for(i in 1:nrow(condenseddf)) {
  if( !isEmpty(grep('Nerve - Tibial', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Tibial Nerve and Artery'
  }
}

for(i in 1:nrow(condenseddf)) {
  if( !isEmpty(grep('Artery - Tibial', condenseddf$body_site[i]))){
    print(condenseddf$body_site[i])
    condenseddf$body_site[i]<-'Tibial Nerve and Artery'
  }
}



noNAcondenseddf<-condenseddf[complete.cases(condenseddf),]


for(i in 1:nrow(noNAcondenseddf)){
  for(j in 1:nrow(noNAcondenseddf)){
    if(noNAcondenseddf$rpkm[i]>0 && i!=j &&
noNAcondenseddf$body_site[i]==noNAcondenseddf$body_site[j] &&
noNAcondenseddf$gene[i] == noNAcondenseddf$gene[j] &&
noNAcondenseddf$sample_id[j]==noNAcondenseddf$sample_id[i]){
      noNAcondenseddf$rpkm[i]<-noNAcondenseddf$rpkm[i] +
noNAcondenseddf$rpkm[j]
      noNAcondenseddf$rpkm[j]<-0
```

```
    }
  }
}
noNAcondenseddf<-noNAcondenseddf[noNAcondenseddf$rpkm>0,]
noNAcondenseddf$order<-NULL
noNAcondenseddf<-rbind(noNAcondenseddf,genedf)
noNAcondenseddf$gene<-as.character(noNAcondenseddf$gene)
noNAcondenseddf<-noNAcondenseddf[mixedorder(noNAcondenseddf$gene),]
noNAcondenseddf$order<-1:nrow(noNAcondenseddf)

write.xlsx(noNAcondenseddf, 'HHV-6a_noNacondenseddf_for_figure.xlsx')

for(i in 1:nrow(noNAcondenseddf)){
  if(noNAcondenseddf$rpkm[i]==0){
    noNAcondenseddf$rpkm[i]<-NA
  }

}
noNAcondenseddf$color<-NA
makecolor<-function(body_site, newcolor){
  for(i in 1:nrow(noNAcondenseddf)){
    if(noNAcondenseddf$body_site[i]==body_site){
      print('ok')
      noNAcondenseddf$color[i]<<-newcolor
    }
  }
}
# makecolor("Testis", "green")
# makecolor("Brain" , "Pink")
# makecolor("Adrenal Gland", 'cyan')
# makecolor('Minor Salivary Gland', 'brown')
# makecolor("Skin",'beige')
# makecolor("Muscle - Skeletal", "azure4")
# makecolor("Adipose", "darkgoldenrod1")
# makecolor("Thyroid", "coral2")
# makecolor("Heart", "red")
# makecolor("Esophagus", "darkolivegreen2")
# makecolor("Nerve - Tibial", "darkslategray4")
# makecolor("Breast - Mammary Tissue", "darkorchid")
# makecolor("Atery - Tibial" , "Black")
```

```
cols<-c("Cells, Transformed fibroblasts and EBV-transformed lymphocytes" =
"mediumpurple2",
       "Prostate" = "indianred",
       "Testis"= "chartreuse4",
       "Brain"= "deeppink",
       "Adrenal Gland"= 'cyan',
       'Minor Salivary Gland'= 'brown',
       "Skin"='cornflowerblue',
       "Muscle - Skeletal"= "azure4",
       "Adipose"= "darkgoldenrod1",
       "Thyroid" = "darkseagreen3",
       "Lung" = "thistle2",
       "Colon - Transverse" = "blueviolet",
       "Heart"= "darksalmon",
       "Esophagus"= "darkolivegreen2",
       "Nerve - Tibial"= "blueviolet",
       "Breast - Mammary Tissue"= "lightpink1" ,
       "Tibial Nerve and Artery"= "darkorange2",
       "Liver" = "steelblue4",
       "0" = 'white',
       "Whole Blood" = "firebrick1")
base_breaks <- function(n = 10){
  function(x) {
    axisTicks(log10(range(x, na.rm = TRUE)), log = TRUE, n = n)
  }
}

breaks <- 10^(-10:10)
minor_breaks <- rep(1:9, 21)*(10^rep(-10:10, each=9))


###works
##########for figure no legend#############
deduplicatedcondensedgraph<-ggplot(noNAcondenseddf,aes(x=reorder(gene,
order), y=rpkm)) + geom_point(aes(shape=sample_id,
color=as.character(body_site)), size=2)
deduplicatedcondensedgraph +scale_colour_manual(values=cols) +
  #scale_y_log10(breaks = breaks, minor_breaks = minor_breaks) +
  scale_y_log10(breaks = c(1e-11,1e-10,1e-9,1e-8,1e-7,1e-6), limits = c(1e-11,
1e-9)) +
  ##change scaled to FALSE to make the tickmarks go away but gridlines stay
  #annotation_logticks(sides='l', scaled= TRUE, alpha = .5) +
  theme(legend.key.size = unit(.05, "cm")) +
```

```
  theme_bw() +
  scale_shape_manual(values=c(1,15, 16, 17, 18))+
  theme(axis.text.x=element_text(angle=70,vjust=.5,hjust=.9, size =
5),legend.position = "none") +
  labs(x = "Gene", y= 'log(RPKM) of reads for each tissue')
##figure out how to change the ticks on the y axis
ggsave('63_gtex_HHV-6a_figure_formatted_no_legend.pdf', width=7.75,
height=3,useDingbats=FALSE)



########for figure with legend###########
deduplicatedcondensedgraph<-ggplot(noNAcondenseddf,aes(x=reorder(gene,
order), y=rpkm)) + geom_point(aes(shape=sample_id,
color=as.character(body_site)), size=2)
deduplicatedcondensedgraph +scale_colour_manual(values=cols) +
  #scale_y_log10(breaks = breaks, minor_breaks = minor_breaks) +
  scale_y_log10(breaks = c(1e-9,1e-8,1e-7,1e-6)) +
  ##change scaled to FALSE to make the tickmarks go away but gridlines stay
  #annotation_logticks(sides='l', scaled= TRUE, alpha = .5) +
  theme(legend.key.size = unit(.05, "cm")) +
  theme_bw() +
  scale_shape_manual(values=c(1,15, 16, 17, 18))+
  theme(axis.text.x=element_text(angle=70,vjust=.5,hjust=.9, size = 5),
legend.text = element_text(size=5)) +
  guides(fill = guide_legend(override.aes = list(size=10)))+
  labs(x = "Gene", y= 'log(RPKM) of reads for each tissue')
##figure out how to change the ticks on the y axis
ggsave('gtex_HHV-6A_figure_formatted_with_legend.pdf', height = 12, width =
12, useDingbats=FALSE)

legendorderdf<-data.frame(unique(noNAcondenseddf$body_site))
legendorderdf$sums<-NA
for ( i in 1:nrow(noNAcondenseddf)){
  legendorderdf$sums[i]<-
sum(noNAcondenseddf$rpkm[noNAcondenseddf$body_site==legendorderdf$uni
que.noNAcondenseddf.body_site.[i]])
  }

########for figure#########
combined_for_figure<-ggplot(noNAcondenseddf,aes(x=reorder(gene, order),
y=rpkm)) + geom_point(aes(shape=sample_id, color=as.character(body_site)),
size=4)
```

```
combined_for_figure +scale_colour_manual(values=cols) +
  scale_y_log10(breaks = breaks, minor_breaks = minor_breaks) +
  theme(legend.key.size = unit(.05, "cm")) +
  theme_bw() +
  scale_shape_manual(values=c(1,15, 16, 17, 18))+
  theme(axis.text.x=element_text(angle=75,vjust=.9,hjust=.9)) +
  labs(x = "Gene", y= 'RPKM of reads for each tissue', size = 8)+
  theme( axis.title.x = element_text(size = 8),  axis.title.y = element_text(size = 8))
##figure out how to change the ticks on the y axis
ggsave('figure_formatted_gtex_HHV-6A.pdf_full_legend', width=7.5, height=3)


###for 11DXY
DXYdf<-noNAcondenseddf[noNAcondenseddf$sample_id=='GTEX-11DXY',]
DXYdf$order<-NULL
DXYdf$color<-NULL
DXYdf<-rbind(DXYdf,genedf)
DXYdf$gene<-as.character(DXYdf$gene)
DXYdf<-DXYdf[mixedorder(DXYdf$gene),]
DXYdf$order<-1:nrow(DXYdf)
for( i in 1:nrow(DXYdf)){
  if(DXYdf$count[i]==0 | is.na(DXYdf$count[i])){
    DXYdf$count[i]<-NA
    DXYdf$body_site[i]<-NA
    DXYdf$sample_id[i]<-NA
    DXYdf$rpkm[i]<-NA
    #ordered_final_df$pos[i]<-NA
  }
}
for( i in 1:nrow(DXYdf)){
  if(DXYdf$count[i]==0 | is.na(DXYdf$count[i])){
    DXYdf$count[i]<-NA
    DXYdf$body_site[i]<-NA
    DXYdf$sample_id[i]<-NA
    DXYdf$rpkm[i]<-NA
    #ordered_final_df$pos[i]<-NA
  }
}
DXYdf_graph<-ggplot(DXYdf,aes(x=reorder(gene, order), y=rpkm)) +
geom_point(aes(color=body_site), size=2)
DXYdf_graph +
  scale_colour_manual(values=cols) +
  scale_y_log10(breaks = c(1e-10,1e-9,1e-8), limits = c(1e-10,1e-8)) +
```

```
  ##change scaled to FALSE to make the tickmarks go away but gridlines stay
  #annotation_logticks(sides='l', scaled= TRUE, alpha = .5) +
  theme(legend.key.size = unit(.05, "cm")) +
  theme_bw() +
  scale_shape_manual(values=c(1,15, 16, 17, 18))+
  theme(axis.text.x=element_text(angle=70,vjust=.5,hjust=.9, size =
5),legend.position = "none") +
  labs(x = "Gene", y= 'log(RPKM) of reads for each tissue')  +
  ggtitle('GTEx-11DXY')
ggsave('hhv6a_GTEx-11DXY.pdf', width=7.75, height=3, useDingbats=FALSE)

DXYlegendorderdf<-data.frame(unique(as.character(DXYdf$body_site)))
DXYlegendorderdf$sums<-NA
colnames(DXYlegendorderdf)[1]<-'body_site'
for( i in 1:nrow(DXYlegendorderdf)){
  DXYlegendorderdf$sums[i]<-
sum(DXYdf$rpkm[DXYdf$body_site==DXYlegendorderdf$body_site[i]],na.rm=TR
UE)

}

###for 1314G
Gdf<-noNAcondenseddf[noNAcondenseddf$sample_id=='GTEX-1314G',]
Gdf$order<-NULL
Gdf$color<-NULL
Gdf<-rbind(Gdf,genedf)
Gdf$gene<-as.character(Gdf$gene)
Gdf<-Gdf[mixedorder(Gdf$gene),]
Gdf$order<-1:nrow(Gdf)
for( i in 1:nrow(Gdf)){
  if(Gdf$count[i]==0 | is.na(Gdf$count[i])){
    Gdf$count[i]<-NA
    Gdf$body_site[i]<-NA
    Gdf$sample_id[i]<-NA
    Gdf$rpkm[i]<-NA
    #ordered_final_df$pos[i]<-NA
  }
}
for( i in 1:nrow(Gdf)){
  if(Gdf$count[i]==0 | is.na(Gdf$count[i])){
    Gdf$count[i]<-NA
    Gdf$body_site[i]<-NA
    Gdf$sample_id[i]<-NA
```

```
    Gdf$rpkm[i]<-NA
    #ordered_final_df$pos[i]<-NA
  }
}

Gdf_graph<-ggplot(Gdf,aes(x=reorder(gene, order), y=rpkm)) +
geom_point(aes(color=body_site), size=2)
Gdf_graph +
  scale_colour_manual(values=cols) +
  scale_y_log10(breaks = c(1e-10,1e-9,1e-8), limits = c(1e-10,1e-8)) +
  ##change scaled to FALSE to make the tickmarks go away but gridlines stay
  #annotation_logticks(sides='l', scaled= TRUE, alpha = .5) +
  theme(legend.key.size = unit(.05, "cm")) +
  theme_bw() +
  scale_shape_manual(values=c(1,15, 16, 17, 18))+
  theme(axis.text.x=element_text(angle=70,vjust=.5,hjust=.9, size =
5),legend.position = "none") +
  labs(x = "Gene", y= 'log(RPKM) of reads for each tissue')  +
  ggtitle('GTEx-1314G')
ggsave('hhv6b_GTEx-1314G.pdf', width=7.75, height=3, useDingbats=FALSE)

GDFlegendorderdf<-data.frame(unique(as.character(Gdf$body_site)))
GDFlegendorderdf$sums<-NA
colnames(GDFlegendorderdf)[1]<-'body_site'
for( i in 1:nrow(GDFlegendorderdf)){
  GDFlegendorderdf$sums[i]<-
sum(Gdf$rpkm[Gdf$body_site==GDFlegendorderdf$body_site[i]],na.rm=TRUE)

}


#y=sum of rpkms, x=genes
totalrpkmdf<-data.frame(as.character(unique(gff3$gene)))
colnames(totalrpkmdf)[1]<-'gene'
totalrpkmdf$sum <- 0
for( i in 1:nrow(totalrpkmdf)) {
  for(j in 1:nrow(df_for_graph)) {
    if( is.na(df_for_graph$rpkm[j])==FALSE ) {
      if( totalrpkmdf$gene[i] == df_for_graph$gene[j]) {
        totalrpkmdf$sum[i]<- totalrpkmdf$sum[i] + df_for_graph$rpkm[j]
        print(totalrpkmdf$sum[i])
      }
    }
```

```
  }
}
totalrpkmdf$order = 1:nrow(totalrpkmdf)

totalrpkmgraph<-ggplot(data = totalrpkmdf,aes(x=reorder(gene, order),
y=totalrpkmdf$sum)) +
  geom_bar(stat = "identity") +
  xlab('gene') +
  ylab('sum of RPKM per gene') +
  ggtitle('sum of RPKM per gene across all HHV-6Aexpression positive tissue')
totalrpkmgraph
```

Generation of average coverage figures (R):

```
#To find average coverage for different sample types change the lookup
excel file
library(Biostrings)
library(Rsamtools)
library(GenomicAlignments)
library(edgeR)
library(rtracklayer)
library(readxl)
library(ggplot2)
library(gtools)
library(plotrix)
library(foreach)
library(doParallel)
library(seqinr)
library(scales)
library(wesanderson)
library(RColorBrewer)

args = commandArgs(trailingOnly=TRUE)
betaglobinpath<-args[1]
edarpath<-args[2]
rpp30path<-args[3]
hhv6apath<-args[4]
hhv6bpath<-args[5]
setwd('/Users/gerbix/Documents/vikas/Thesis_from_other_mac/average_c
overage/')
data <- read.xlsx("Exomes_only.xlsx")
edarpath<-'edars_exome'
betaglobinpath<-'bglobins_exome'
hhv6apath<-'hhv6a_exome'
hhv6bpath<-'hhv6b_exome'


depthcounter<-function(paths,start,stop) {
  list<-c()
  filenames<-list.files(path=paths, pattern='*.bam$')
  for( i in 1:length(filenames)) {
    count<-0
```

```
    tempbam<-scanBam(paste0(paths,'/',filenames[i]))
    for( j in 1:length(tempbam[[1]]$pos)){
      if(tempbam[[1]]$pos[j]>start & tempbam[[1]]$pos[j]<stop) {
        count<-count+1
      }
    }
    print(count)
    list[i]<-count
    print(100*(i/length(filenames)))
  }
  combined<-c()
  combined[[1]]<-list
  combined[[2]]<-filenames
  return(combined)
}
bglobinstart<-1545
bglobinstop<-3871
bglobin<-depthcounter(betaglobinpath,bglobinstart,bglobinstop)
bglobincounts<-bglobin[1]
bglobinnames<-bglobin[2]
bglobindf<-data.frame(bglobincounts,bglobinnames)
bglobindf$type<-'Beta globin'
colnames(bglobindf)[1]<-'counts'
colnames(bglobindf)[2]<-'names'
bglobindf$depth<-(bglobindf$counts * 76)/(bglobinstop-bglobinstart)
bglobindf$total_reads<-0

for ( i in 1:nrow(bglobindf)){
  temp<-
substr(bglobindf$names[i],1,nchar(as.character(bglobindf$names[i]))-4)
  print(temp)
  bglobindf$total_reads[i]<-(data$MBases[data$Run==temp])
}

edarstart<-8390
edarstop<-10900
edar<-depthcounter(edarpath,edarstart,edarstop)
edarcounts<-edar[1]
edarnames<-edar[2]
edardf<-data.frame(edarcounts,edarnames)
```

```
edardf$type<-'EDAR'
colnames(edardf)[1]<-'counts'
colnames(edardf)[2]<-'names'
edardf$depth<-(edardf$counts * 76)/(edarstop-edarstart)
for ( i in 1:nrow(edardf)){
  temp<-substr(edardf$names[i],1,nchar(as.character(edardf$names[i]))-4)
  edardf$total_reads[i]<-(data$MBases[data$Run==temp])
}

hhv6astart<-42000
hhv6astop<-90000
hhv6a<-depthcounter(hhv6apath,hhv6astart,hhv6astop)
hhv6acounts<-hhv6a[1]
hhv6anames<-hhv6a[2]
hhv6adf<-data.frame(hhv6acounts,hhv6anames)
hhv6adf$type<-'HHV-6A'
colnames(hhv6adf)[1]<-'counts'
colnames(hhv6adf)[2]<-'names'
hhv6adf$depth<-(hhv6adf$counts * 76)/(hhv6astop-hhv6astart)
for ( i in 1:nrow(hhv6adf)){
  temp<-substr(hhv6adf$names[i],1,nchar(as.character(hhv6adf$names[i]))-
8)
  hhv6adf$total_reads[i]<-(data$MBases[data$Run==temp])
}

hhv6bstart<-42000
hhv6bstop<-90000
hhv6b<-depthcounter(hhv6bpath,hhv6bstart,hhv6bstop)
hhv6bcounts<-hhv6b[1]
hhv6bnames<-hhv6b[2]
hhv6bdf<-data.frame(hhv6bcounts,hhv6bnames)
hhv6bdf$type<-'HHV-6B'
colnames(hhv6bdf)[1]<-'counts'
colnames(hhv6bdf)[2]<-'names'
hhv6bdf$depth<-(hhv6bdf$counts * 76)/(hhv6bstop-hhv6bstart)

for ( i in 1:nrow(hhv6bdf)){
  temp<-substr(hhv6bdf$names[i],1,nchar(as.character(hhv6bdf$names[i]))-
8)
  hhv6bdf$total_reads[i]<-(data$MBases[data$Run==temp])
```

```
}

#allcombined<-rbind(bglobindf,edardf,rpp30df,hhv6adf,hhv6bdf)
allcombined<-rbind(bglobindf,edardf,hhv6adf,hhv6bdf)
allcombined$total_reads<-allcombined$total_reads*1000000
allcombined$normalized<-allcombined$depth/allcombined$total_reads
allcombined$normalized<-allcombined$normalized*300000000000
allcombined[nrow(allcombined)+1,] <- NA
allcombined[nrow(allcombined),3]<-'negative'
allcombined[nrow(allcombined),4:6]<-0

pal <- wes.palette(name = "Zissou", type = "continuous")

graph<-ggplot(allcombined,aes(x=names, y=depth, color=as.factor(type)))
+ geom_point()
graph + scale_color_brewer(palette="Dark2") + theme_minimal() +
  labs(x = "Gene")

#not normalized depth
p2 <- ggplot(allcombined, aes(x=factor(type),y=depth, color=type))+
  geom_point() + labs(title="Average Coverage") +
  theme_bw() +
  expand_limits(x = 0, y = 0)

p2

#normalized depth
p3 <- ggplot(allcombined, aes(x=factor(type),y=normalized,
color=type,size=2))+
  geom_point() + labs(title="Average Coverage") +
  theme_bw() +
  xlab('gene') +
  ylab('normalized depth') +
  scale_y_continuous(trans='log10')

p3

########formatted for figure########
```

```
p4 <- ggplot(allcombined,
aes(x=factor(allcombined$type),y=allcombined$normalized,
color=type,size=2))+
  #geom_point(size =1) +
  geom_jitter(width  = .15, size = .7 )+
  ggtitle('GTEx WES') +
  theme_classic() +
  xlab('gene') +
  ylab('normalized depth') +
  scale_y_continuous(trans='log10', breaks = c(0,10,100,1000,100000)) +
  #  ylim(0,100) +
  theme(plot.title = element_text(size=11), legend.position="none",
axis.text.x = element_text(angle = 70, hjust = 1))
p4


renormalized<-allcombined
renormalized$normalized<-renormalized$normalized/300000000000
renormalized$normalized<-renormalized$normalized/1.507799e-10




p5<- ggplot(renormalized,
aes(x=factor(renormalized$type),y=renormalized$normalized,
color=type,size=2))+
  #geom_point(size =1) +
  geom_jitter(width  = .15, size = .7 )+
  ggtitle('GTEx WES') +
  theme_classic() +
  xlab('gene') +
  ylab('normalized depth') +
  scale_y_continuous(trans='log10', breaks = c(1,10,100,1000,100000)) +
  #  ylim(0,100) +
  theme(plot.title = element_text(size=11), legend.position="none",
axis.text.x = element_text(angle = 70, hjust = 1))
p5
```

Pipeline for AWS-batch SRA download and alignments: https://github.com/FredHutch/sra-pipeline