

Acquiring, annotating, and normalizing gene expression data from GTEx

Warren Anderson
Joon Yuhl Soh
Mete Civelek

April 2, 2020

This guide provides code and documentation of analyses from Anderson et al., 2020, *Sex differences in human adipose tissue gene expression and genetic regulation involve adipogenesis*

Contents

1	Overview	2
2	Annotation data for GTEx	2
3	Acquiring read count data	2
4	Acquiring TPM data	3
5	Normalizing TPM data	4
6	References	9

List of Figures

1	Distributions of TPM before and after quantile normalization.	6
2	Distributions of TPM before and after inverse normalization.	7
3	XIST expression from the inversely normalized (left) and the original (right) GTEx data set	8

1 Overview

This vignette contains procedures for processing subcutaneous adipose tissue expression data. Similar methods were used for other tissues.

2 Annotation data for GTEx

We downloaded data from the GTEx portal (<https://www.gtexportal.org>). Here is the download information.

```
# download site
https://www.gtexportal.org/home/datasets

# download files under GTEx Analysis V8, Annotations
GTEx_Analysis_v8_Annotations_SampleAttributesDS.xlsx
GTEx_Analysis_v8_Annotations_SampleAttributesDS.txt
GTEx_Analysis_v8_Annotations_SubjectPhenotypesDS.xlsx
GTEx_Analysis_v8_Annotations_SubjectPhenotypesDS.txt
```

We next downloaded the eQTL expression to get the sample/subject identifiers used for the subcutaneous adipose eQTL analysis reported by GTEx based on their quality controls (Consortium, 2015).

```
# download site
https://www.gtexportal.org/home/datasets

# download files under GTEx Analysis V8, Single-Tissue cis-eQTL Data
GTEx_Analysis_v8_eQTL_expression_matrices.tar.gz
gunzip GTEx_Analysis_v8_eQTL_expression_matrices.tar.gz
tar -xvf GTEx_Analysis_v8_eQTL_expression_matrices.tar

# isolate the subcutaneous adipose data (located in the GTEx_Analysis_v8_eQTL_expression_matrices folder)
cd GTEx_Analysis_v8_eQTL_expression_matrices
gunzip Adipose_Subcutaneous.v8.normalized_expression.bed.gz

# get the subcutaneous subject identifiers
head -1 Adipose_Subcutaneous.v8.normalized_expression.bed | cut -f1-4 --complement > subq_subjects.txt
```

The file *subq_subjects.txt* contains identifiers for the subcutaneous adipose tissue samples used in our eQTL analysis. This file was used to get the corresponding genotypes. Now we must also process the full sample data annotation. Here we isolate sample identifiers and corresponding tissues in the file *tissue_sampleID.txt*.

```
cat GTEx_Analysis_v8_Annotations_SampleAttributesDS.txt | cut -f1,7 > tissue_sampleID.txt
```

3 Acquiring read count data

Because multiple samples from individual samples were processed, we will download the read count data and select the samples corresponding to the greatest number of reads. The sample identifiers corresponding to the columns of the read matrix are saved in *gtex_read_ids.txt*.

```
# download site
https://www.gtexportal.org/home/datasets

# download files under GTEx Analysis V8, RNA-Seq Data
GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_reads.gct.gz
gunzip GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_reads.gct.gz

# remove header and rename the file for convenience
tail -n +3 GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_reads.gct > gtex_reads.txt

# get the read data identifiers
head -1 gtex_reads.txt > gtex_read_ids.txt
```

4 Acquiring TPM data

We download and process the TPM subcutaneous adipose expression data. These data can be acquired and processed as follows.

```
# download site
https://www.gtexportal.org/home/datasets

# download files under GTEx Analysis V8, RNA-Seq Data
GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm.gct.gz
gunzip GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm.gct.gz

# remove header and rename file
tail -n +3 GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm.gct > TPM7.txt

# get list of column headings (sample IDs) in tpm file:
head -1 TPM7.txt > TPM_sampleID.txt
```

Now we need to isolate the TPM data corresponding to the subcutaneous adipose samples of interest. We use R to get the columns of interest for the TPM data set. First, we will get the sample identifiers of the TPM data set that match the subjects included in the subcutaneous adipose eQTL analysis. Then we will get the corresponding read counts for these samples. As shown below, we found that the TPM data contained multiple adipose samples for single individuals. However, the read data contained only one read set for each individual. We used the sample identifiers from the TPM data set that corresponded to samples for which read data were available.

```
library(dplyr)

# import data
subq_id = read.table("subq_subjects.txt", sep = "\t", header=F, stringsAsFactors=F) %>% t
read_id = read.table("gtex_read_ids.txt", sep = "\t", header=F, stringsAsFactors=F) %>% t
TPM_id = read.table("TPM_sampleID.txt", sep = "\t", header=F, stringsAsFactors=F,
  comment.char="") %>% t
tissue_id = read.table("tissue_sampleID.txt", sep = "\t", header=T, stringsAsFactors=F)

# get subjects from subcutaneous adipose samples
subq_samps = tissue_id %>% filter(SMTSD == "Adipose - Subcutaneous") %>% select(SAMPID)
samp_subj = sapply(subq_samps$SAMPID, function(x){
  s1 = strsplit(x, "-")[[1]]
  return(paste0(s1[1], "-", s1[2]))
})
subq_samps = subq_samps %>% mutate(subq_subj = samp_subj)

# get subq samples matching subjects used for the GTEx qQTL analysis
# note that multiple samples were processed for identical subjects
# but there are only 385 unique subjects from which the samples were acquired
subq_samps_eqtl = subq_samps[subq_samps$subq_subj %in% subq_id,]

# get the column numbers of subcutaneous samples in the read data matrix
read.inds = sapply(subq_samps_eqtl$SAMPID, function(x) which(read_id==x)) %>% unlist

# identify TPM samples with associated read data
subq_samps_eqtl_tpm = subq_samps_eqtl[subq_samps_eqtl$SAMPID %in% names(read.inds),]

# checks
dim(subq_samps_eqtl_tpm) # 385 by 2
length(unique(subq_samps_eqtl_tpm$subq_subj)) # 385
all(subq_samps_eqtl_tpm$subq_subj == subq_id) # TRUE

# select TPM columns
tpm_cols0 = match(subq_samps_eqtl_tpm$SAMPID, read_id)
tpm_cols = tpm_cols0[!is.na(tpm_cols0)]
```

```
out = c()
for(ii in 1:length(tpm_cols)){
  out = paste0(out,"$",tpm_cols[ii],",")
}
```

The vector *out* contains the column numbers of interest for the TPM data set. We pasted these numbers into the command line as follows to isolate the appropriate TPM data.

```
# isolate subcutaneous TPM data using the command line (include name and description, which are $1 and $2)
cat TPM7.txt | awk -v OFS='\t' '{print $1,$2,$3,$31,$35,$57,$85,$108,$120,$130,$193,$266,$354,$369,$406,$430,$438,$472,$490,
$503,$579,$591,$629,$670,$715,$742,$768,$792,$813,$827,$859,$873,$904,$914,$919,$968,$1009,$1023,$1055,$1064,$1077,$1105,
$1133,$1142,$1166,$1176,$1209,$1227,$1270,$1279,$1299,$1341,$1503,$1518,$1579,$1605,$1630,$1659,$1682,$1698,$1719,$1741,
$1764,$1782,$1850,$1868,$1892,$1968,$2000,$2035,$2079,$2108,$2115,$2127,$2149,$2191,$2204,$2246,$2265,$2290,$2327,$2353,
$2364,$2387,$2456,$2481,$2532,$2544,$2569,$2582,$2611,$2625,$2660,$2689,$2728,$2732,$2749,$2767,$2799,$2838,$2870,$2899,
$2947,$2971,$3001,$3016,$3067,$3095,$3126,$3147,$3189,$3199,$3239,$3266,$3284,$3333,$3365,$3401,$3420,$3436,$3460,$3489,
$3526,$3584,$3609,$3627,$3656,$3673,$3705,$3827,$3844,$3877,$3899,$3902,$3920,$3940,$3968,$3978,$4009,$4044,$4059,$4082,
$4112,$4121,$4165,$4192,$4230,$4276,$4313,$4323,$4375,$4390,$4419,$4460,$4482,$4489,$4511,$4537,$4573,$4581,$4620,$4650,
$4662,$4678,$4720,$4737,$4745,$4773,$4798,$4828,$4842,$4915,$4965,$5052,$5112,$5130,$5228,$5257,$5314,$5355,$5434,$5526,
$5539,$5564,$5570,$5588,$5604,$5636,$5656,$5678,$5699,$5710,$5740,$5799,$5823,$5875,$5896,$5919,$5947,$6003,$6044,$6057,
$6085,$6150,$6161,$6194,$6201,$6225,$6255,$6279,$6306,$6334,$6361,$6412,$6439,$6454,$6477,$6515,$6521,$6544,$6561,$6591,
$6609,$6636,$6661,$6701,$6725,$6765,$6787,$6801,$6830,$6850,$6884,$6903,$6920,$6937,$6973,$7003,$7037,$7070,$7111,$7132,
$7139,$7176,$7205,$7244,$7273,$7290,$7323,$7330,$7380,$7390,$7420,$7448,$7465,$7495,$7519,$7545,$7571,$7602,$7635,$7686,
$7710,$7743,$7769,$7797,$7824,$7861,$7899,$7914,$7950,$7975,$8002,$8022,$8032,$8067,$8130,$8176,$8195,$8218,$8244,$8270,
$8297,$8315,$8343,$8373,$8405,$8434,$8463,$8496,$8506,$9656,$9678,$9714,$9720,$9744,$9761,$9787,$9809,$10246,$10342,$10355,
$10375,$10429,$10462,$10490,$10515,$10526,$10546,$10564,$10597,$10618,$10656,$10664,$10734,$10761,$10790,$10815,$10823,
$10840,$10876,$10914,$10937,$10960,$10981,$11012,$11033,$11039,$11058,$11081,$11112,$11146,$11159,$11172,$11191,$11212,
$11218,$11250,$11274,$11303,$11326,$11351,$11430,$11454,$11485,$11504,$11526,$11544,$11559,$11576,$11589,$11619,$11635,
$11657,$11670,$11705,$11740,$11764,$11780,$11809,$11822,$11825,$11850,$11857,$11897,$11908,$11921,$11946,$11989,$12021,
$12050,$12073,$12090,$12181,$12230,$12237,$12268,$12282,$12305,$12343,$12409,$12427,$12444,$12501,$12523,$12537,$12563,
$12571,$12588,$12602,$12620,$12633,$12648,$12665,$12681,$12690,$12711,$12725,$12739,$12759,$12774,$12792,$12808,$12830,
$12853,$12868,$12887,$12892,$12900,$12957,$12964,$12989,$13021,$13053,$13068,$13085,$13104,$13122,$13151,$13172,$13180,
$13194,$13209,$13236,$13253,$13302,$13313,$13330,$13384,$13412,$13428,$13435,$13488,$13492,$13522,$13550,$13568,$13613,
$13638,$13654,$13680,$13711,$13724,$13775,$13825,$13853,$13877,$13888,$13903,$13923,$13941,$13949,$13967,$13976,$14016,
$14047,$14065,$14100,$14112,$14129,$14162,$14186,$14200,$14249,$14279,$14285,$14307,$14324,$14357,$14414,$14466,$14487,
$14514,$14522,$14544,$14555,$14579,$14606,$14628,$14643,$14667,$14690,$14709,$14782,$14816,$14892,$14905,$14985,$15005,
$15103,$15114,$15151,$15194,$15220,$15271,$15286,$15293,$15299,$15319,$15328,$15358,$15369,$15399,$15407,$15421,$15450,
$15475,$15481,$15523,$15540,$15551,$15575,$15592,$15613,$15640,$15665,$15689,$15693,$15735,$15762,$15789,$15836,$15861,
$15892,$15911,$15932,$15956,$15978,$15998,$16020,$16054,$16058,$16092,$16109,$16160,$16168,$16188,$16204,$16218,$16232,
$16306,$16324,$16346,$16369,$16400,$16424,$16449,$16463,$16515,$16531,$16563,$16596,$16632,$16666,$16681,$16707,$16732,
$16753,$16774,$16803,$16816,$16873,$16880,$16913,$16940,$16960,$16986,$17027,$17050,$17074,$17104,$17123,$17143,$17184,
$17193,$17220,$17236,$17272,$17291,$17308,$17324,$17356,$17384}' > tpm_subq.txt
```

5 Normalizing TPM data and QC

Note that we have the subcutaneous adipose TPM data of interest in *tpm_subq.txt* we evaluate the effects of applying quantile normalization, which is commonly used to bring all samples on the same scale. However, we will not continue with the quantile normalized data because TPM data are normalized for sample scale. Further normalization may distort the results. Then we will map the expression ranks of each gene on to a standard normal cdf and inverse transform the data, thereby generating a normal distribution of expression levels for each gene. These analyses were completed in R. We will verify sex annotation by plotting XIST expression with respect to sex. First, we isolate the sex identities from the subject phenotype data.

```
cat GTEX_Analysis_v8_Annotations_SubjectPhenotypesDS.txt | cut -f1,2 > sex_info.txt
```

```
# GTEX TPM normalization
library(devtools)
library(Biobase)
library(preprocessCore)
library(dplyr)
library(beeswarm)

# import TPM data
tpm_expr0 = read.table("tpm_subq.txt", sep="\t", stringsAsFactors=F, header=T,
  check.names=F)
```

```

tpm_transcript_IDs = tpm_expr0[,1:2]
tpm_expr0 = tpm_expr0[,-c(1,2)]
rownames(tpm_expr0) = tpm_transcript_IDs$Name

# import eQTL data
fname = "Adipose_Subcutaneous.v8.normalized_expression.bed"
eqtl_expr0 = read.table(fname,sep="\t",stringsAsFactors=F,header=T,comment.char="",
  check.names=F)
eqtl_transcr = eqtl_expr0$gene_id
eqtl_expr0 = eqtl_expr0[,-c(1:4)]
rownames(eqtl_expr0) = eqtl_transcr

# isolate transcripts in the TPM data set that were included in the GTEx eQTL study
tpm_expr1 = tpm_expr0[rownames(tpm_expr0) %in% rownames(eqtl_expr0),]

# perform quantile normalization to standardize across samples
tpm_quantile_norm = normalize.quantiles(as.matrix(tpm_expr1))
colnames(tpm_quantile_norm) = colnames(tpm_expr1)
rownames(tpm_quantile_norm) = rownames(tpm_expr1)

# perform inverse quantile normalization to standardize expression of each gene
tpm_inverse_norm = t(apply(as.matrix(tpm_expr1),1,function(x){
  qnorm( rank(x) / (length(x)+1) )
})) %>% as.data.frame
colnames(tpm_inverse_norm) = colnames(tpm_expr1)
rownames(tpm_inverse_norm) = rownames(tpm_expr1)

# plot distributions of TPM:
pdf("density_quantNorm.pdf")
par(mfrow=c(2,1), mar=c(5,6,4,2)+0.1)
colramp = colorRampPalette(c(3,"white",2))(20)
plot(density(tpm_expr1[,1]),col=colramp[1],lwd=3,ylim=c(0,0.2),xlim=c(-5,200),
  main="no normalization",
  xlab = "TPM", ylab = "density",cex.lab=2,cex.main=2,cex.axis=2)
for(i in 2:20){lines(density(tpm_expr1[,i]),lwd=3,col=colramp[i])}

# plot distributions of quantile normalized expression:
plot(density(tpm_quantile_norm[,1]),col=colramp[1],lwd=3,ylim=c(0,0.2),xlim=c(-5,200),
  main="quantile normalization",
  xlab = "TPM", ylab = "density",cex.lab=2,cex.main=2,cex.axis=2)
for(i in 2:20){lines(density(tpm_quantile_norm[,i]),lwd=3,col=colramp[i])}
dev.off()

```

The effect of quantile normalization on the distribution is demonstrated In Figure 1. As mentioned earlier, we will proceed with the distribution before quantile normalizaiton.

```

# plot distributions of TPM:
pdf("density_quantNorm_genes.pdf")
par(mfrow=c(2,1), mar=c(5,6,4,2)+0.1)
plot(density(tpm_quantile_norm[,1]),col=colramp[1],lwd=3,ylim=c(0,5),xlim=c(-5,15),
  main="no gene normalization",
  xlab = "TPM", ylab = "density",cex.lab=2,cex.main=2,cex.axis=2)
for(i in 2:20){lines(density(tpm_quantile_norm[,i]),lwd=3,col=colramp[i])}

plot(density(data.matrix(tpm_inverse_norm[,1])),col=colramp[1],lwd=3,
  ylim=c(0,0.6),xlim=c(-5,15),
  main="gene normalization",
  xlab = "TPM", ylab = "density",cex.lab=2,cex.main=2,cex.axis=2)

```

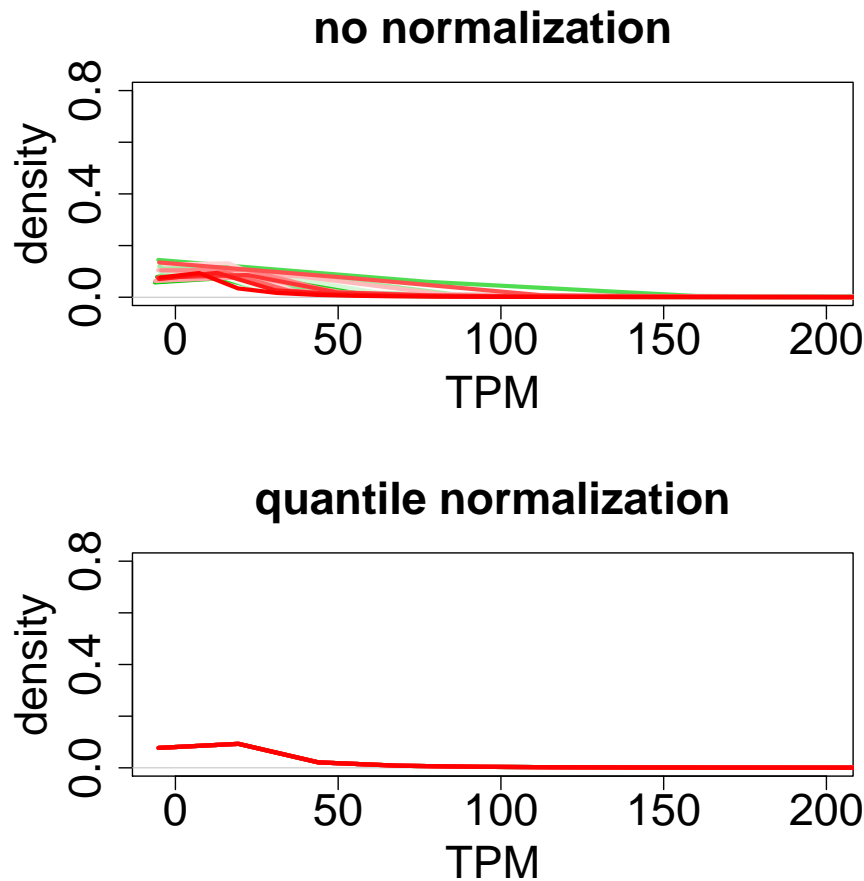


Figure 1: Distributions of TPM before and after quantile normalization.

```
for(i in 2:20){lines(density(data.matrix(tpm_inverse_norm[i,])),lwd=3,col=colramp[i])}
dev.off()
```

The effect of inverse normalization is depicted in Figure 2. The occasional divergences from the theoretical normal are presumably due to the overlapping ranks among a few elements.

To QC the data, we evaluate the *XIST* expression differences between females and males. We also plot the original *XIST* expression data that has not been inversely normalized. As shown in Figure 3, the females have substantially greater *XIST* expression.

```
# perform XIST test to see the difference between males and females
# read in the sex data for all the subjects
subj_sex = read.table("sex_info.txt", sep = "\t", header=T, stringsAsFactors=F)

# find male and female subject IDs
subj_male_ID <- subj_sex$SUBJID[subj_sex$SEX == 1]
subj_female_ID <- subj_sex$SUBJID[subj_sex$SEX == 2]

# modify the columnnames to match with the sex identification
subj_IDs <- sapply(colnames(tpm_inverse_norm),function(x){
  s1 = strsplit(x,"-")[[1]]
  return(paste0(s1[1],"-",s1[2]))
})

# find male and female indices
```

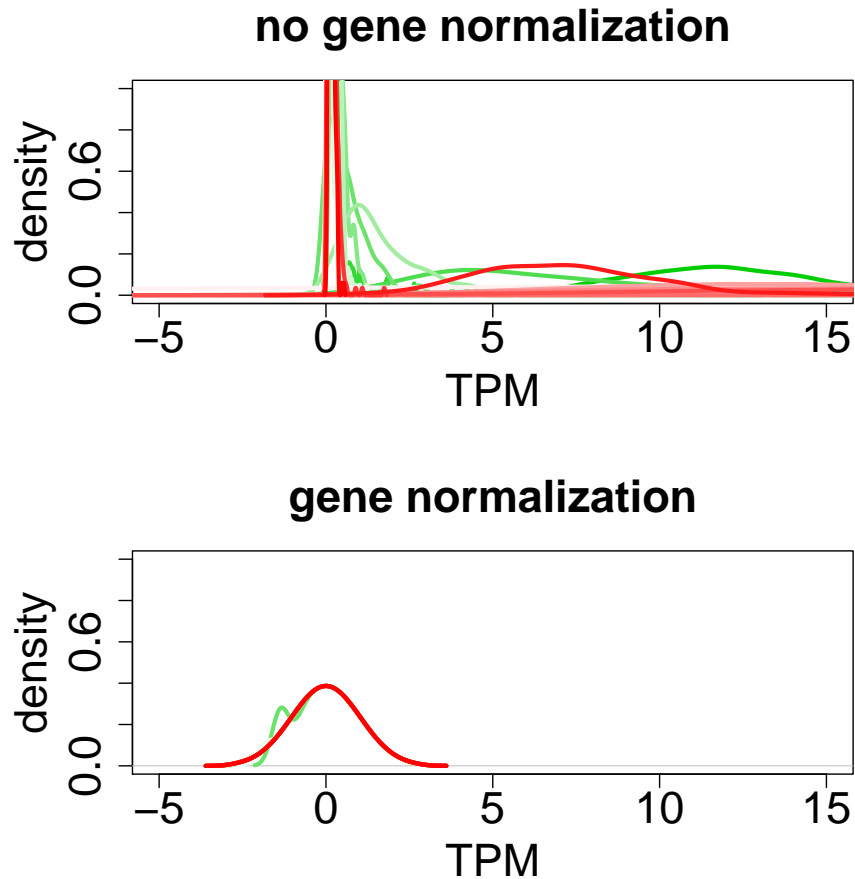


Figure 2: Distributions of TPM before and after inverse normalization.

```

subj_male_index <- subj_IDS %in% subj_male_ID
subj_female_index <- subj_IDS %in% subj_female_ID

# find the XIST index
xist_ID <- tpm_transcript_IDS$Name[tpm_transcript_IDS$Description == "XIST"]
xist_index <- rownames(tpm_inverse_norm) == xist_ID

# create a sex list that corresponds with the XIST row
sex_col <- colnames(tpm_inverse_norm)
sex_col[subj_male_index] <- "Male"
sex_col[subj_female_index] <- "Female"
table(sex_col) # female: 194 male: 387

# create a dataframe that indicates XIST values and sex identification
xist_sex_dataframe <- data.frame(XIST = c(t(tpm_inverse_norm[xist_index,])),
  sex = sex_col)

# plot Xist
pdf("gtex_XIST_mf.pdf",width=4,height=4)
beeswarm(XIST~sex, data = xist_sex_dataframe,
  cex=0.25, pch=16, col=c("red","blue"), cex.lab=1.3, xlab="")
abline(h = 0.4)
while (!is.null(dev.list())) dev.off()

```

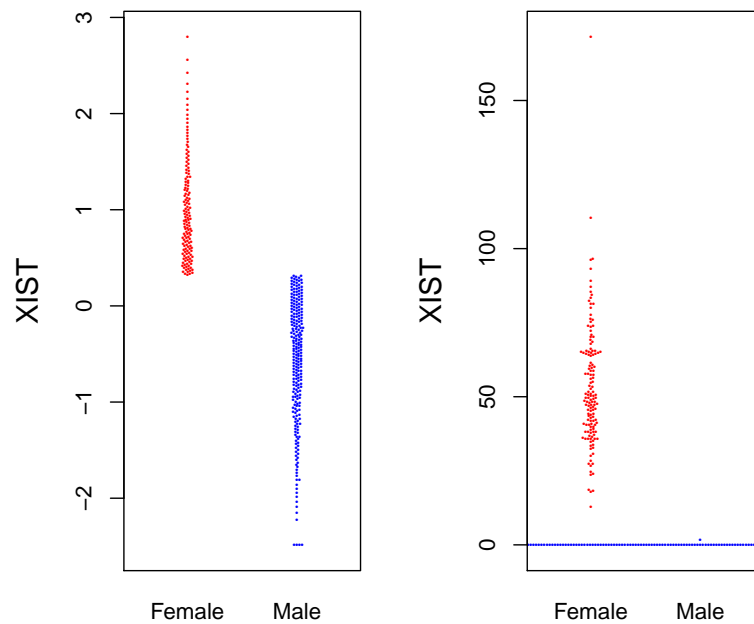


Figure 3: XIST expression from the inversely normalized (left) and the original (right) GTEx data set .

```
# perform XIST test to the data before inverse normalization to see the difference
# between males and females more distinctively
# create a dataframe that indicates XIST values and sex identification
xist_sex_dataframe_original <- data.frame(XIST = c(t(tpm_expr1[xist_index,])),
  sex = sex_col)

# plot Xist for both data for comparison.
pdf("gtex_XIST_mf_comparison.pdf",width=6,height=6)
par(mfrow=c(1,2))
beeswarm(XIST~sex, data = xist_sex_dataframe,
  cex=0.25, pch=16, col=c("red","blue"), cex.lab=1.3, xlab="")
beeswarm(XIST~sex, data = xist_sex_dataframe_original,
  cex=0.25, pch=16, col=c("red","blue"), cex.lab=1.3, xlab="")
while (!is.null(dev.list())) dev.off()
```

Finally, we export the data.

```
# output normalized data
write.table(tpm_inverse_norm,"subq_gtex_invNorm.txt",quote=F,sep="\t",
  col.names=T,row.names=T)
```


6 References

Consortium G (2015). "The Genotype-Tissue Expression (GTEx) pilot analysis: Multitissue gene regulation in humans." *Science (New York, N.Y.)*, **348**, 648–660.