Estimating TRAs from GTex

Hannah Meyer

2020-01-07

Contents

Tissue-restricted antigens 1
Setup 1
Functions 1
Download files from GTex 2
Load gene expression data sets 3
Format gene expression data sets 3
Compute list of tissue-specific genes 4
Compare to Fagerberg (2014) data 5
Compare the results of GTEx and Fagerberg tissue specificity 6

Tissue-restricted antigens

During T cell development in the thymus, thymocytes encounter more then 85% of all protein, including proteins that are usually expressed in a tissue specicif manner. In the following, we use a gene expression data set of about 50k transcripts across 30 tissues (GTEx v7) to estimate tissue specific genes. We use the measure τ as introduced in Yanai 2005 and benchmarked in Kryuchkova-Mostacci 2017 to determine a list of human tissue-restricted antigens:

$$\tau = \frac{\sum_{i=1}^{n}(1-\hat{x}_i)}{n-1}$$

with

$$x_i = \frac{x_i}{\max\limits_{1\leq i\leq n}\hat{x}_i}$$

where x_i is the expression of the gene in tissue i and n is the number of tissues.

Setup

```
library(tidyverse)
datadir <- "~/data/tra/gtex"
```

Functions

```
# Functions from Mostacci et al (2016) Bioinformatics

# Function requires a vector with expression of one gene in different tissues.
# Max is calculated taking in account tissues with 0 expression. 2+0+4=2
fmax <- function(x) {
    if (!all(is.na(x))) {
        res <- max(x, na.rm = TRUE)
    } else {
```

```
        res <- NA
    }
    return(res)
}
#
# Function requires a vector with expression of one gene in different tissues.
# If expression for one tissue is not known, gene specificity for this gene is
# NA; Minimum 2 tissues
tau <- function(x) {
    if (all(!is.na(x))) {
        if (min(x, na.rm = TRUE) >= 0) {
            if (max(x) != 0) {
                x <- (1 - (x / max(x)))
                res <- sum(x, na.rm = TRUE)
                res <- res / (length(x) - 1)
            } else {
                res <- 0
            }
        } else {
            res <- NA
        }
    } else {
        res <- NA
    }
    return(res)
}
```

Download files from GTex

```
gtex=https://storage.googleapis.com/gtex_analysis_v7/annotations
datadir=~/data/tra/gtex

cd $datadir
wget $gtex/GTEx_v7_Annotations_SampleAttributesDS.txt
wget $gtex/GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_reads.gct.gz
```

The following description of the Quality Control and gene expression analysis is taken 1:1 from the GTEx website

RNA-seq Alignment

Alignment to the human reference genome hg19/GRCh37 was performed using STAR v2.4.2a, based on the GENCODE v19 annotation. Unaligned reads were kept in the final BAM file. Among multi-mapping reads, one read is flagged as the primary alignment by STAR.

Quantification

Gene-level quantifications: read counts and TPM values were produced with RNA-SeQC v1.1.8 (DeLuca et al., Bioinformatics, 2012), using the following read-level filters:

1. Reads were uniquely mapped (corresponding to a mapping quality of 255 for START BAMs).

2. Reads were aligned in proper pairs.
3. The read alignment distance was <= 6 (i.e., alignments must not contain more than six non-reference bases).
4. Reads were fully contained within exon boundaries. Reads overlapping introns were not counted. These filters were applied using the "-strictMode" flag in RNA-SeQC.

QC and Sample Exclusion Process

1. RNA-seq expression outliers were identified and excluded using a multidimensional extension of the statistic described in (Wright et al., Nat. Genet. 2014). Briefly, for each tissue, read counts from each sample were normalized using size factors calculated with DESeq2 and log-transformed with an offset of 1; genes with a log-transformed value >1 in >10% of samples were selected, and the resulting read counts were centered and unit- normalized. The resulting matrix was then hierarchically clustered (based on average and cosine distance), and a chi2 p-value was calculated based on Mahalanobis distance. Clusters with >=60% samples with Bonferroni-corrected p-values <0.05 were marked as outliers, and their samples were excluded.
2. Samples with <10 million mapped reads were removed.
3. For samples with replicates, the replicate with the greatest number of reads was selected.

Expression analysis

Gene expression values for all samples from a given tissue were normalized using the following procedure:

1. Genes were selected based on expression thresholds of > 0.1 TPM in at least 20% of samples and >=6 reads in at least 20% of samples.
2. Expression values were normalized between samples using TMM as implemented in edgeR (Robinson & Oshlack, Genome Biology, 2010).
3. For each gene, expression values were normalized across samples using an inverse normal transform.

Load gene expression data sets

```
description_attr <- data.table::fread(file.path(datadir,
                                "GTEx_v7_Annotations_SampleAttributesDS.txt"),
                                data.table=FALSE) %>%
                    as_tibble

samples_tpm <- data.table::fread(file.path(datadir,
                                "GTEx_Analysis_2016-01-15_v7_RNASeQCv1.1.8_gene_tpm.gct"),
                                data.table=FALSE) %>%
                  as_tibble
```

Format gene expression data sets

Join gene ids, tissue and sample IDs with expression values.

```
tpm_attr <- samples_tpm %>%
    select(Name, Description) %>%
    mutate(ID=gsub("\\..*", "", Name))

tpm_annotated <- samples_tpm %>%
    select(-Description, -Name) %>%
```

```
    t %>%
    magrittr::set_colnames(tpm_attr$Name) %>%
    as_tibble %>%
    mutate(SAMPID=colnames(samples_tpm)[-c(1:2)]) %>%
    inner_join(select(description_attr, SAMPID, SMTS), by = "SAMPID")
```

Compute the mean expression per gene and per tissue across biological replicates

```
mean_expression <- tpm_annotated %>%
    select(-SAMPID) %>%
    group_by(SMTS) %>%
    summarise_all(mean, na.rm=TRUE)

tissues_cols <- mean_expression$SMTS
mean_expression <- t(mean_expression[,-1])
colnames(mean_expression) <- tissues_cols
```

Compute list of tissue-specific genes

We follow the processing processing procedure described in the Kryuchkova-Mostacci 2017 benchmarking study: * All genes where the highest gene expression < 1 RPKM are set as not expressed; * count data are log-transformed after addition of 1 to avoid zero counts.

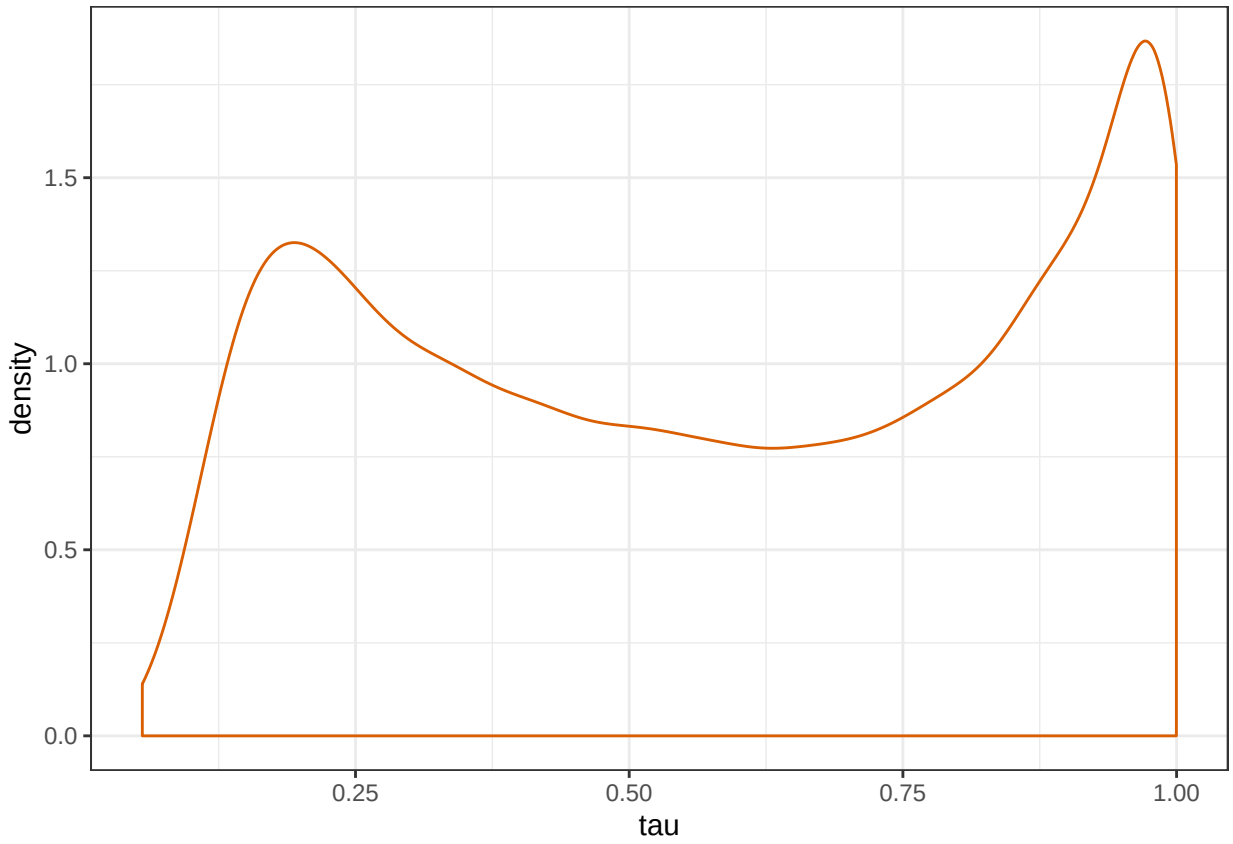
```
rpkm <- 1
gtex_max <- apply(mean_expression, 1, fmax)
mean_expression_filter <- mean_expression[gtex_max > rpkm,]
mean_expression_filter <- mean_expression_filter + 1

gtex <- data.frame(id=gsub("\\..*", "", rownames(mean_expression_filter)),
                   full=rownames(mean_expression_filter),
                   tau=apply(log2(mean_expression_filter), 1, tau),
                   study='gtex',
                   stringsAsFactors = FALSE)

gtex <- gtex %>%
    inner_join(tpm_attr, by = c("full" = "Name")) %>%
    select(Description, everything()) %>%
    rename(Gene="Description", ENSG="id") %>%
    select(-ID)
write.table(select(gtex, -study), file=file.path(datadir, "gtex_tau.csv"),
            sep=",", col.names=TRUE, row.names=FALSE, quote=FALSE)

gtex_tra <- gtex %>%
    filter(tau > 0.8) %>%
    select(Gene, ENSG)
write.table(gtex_tra, file=file.path(datadir, "gtex_tra.csv"),
            sep=",", col.names=TRUE, row.names=FALSE, quote=FALSE)

p <- ggplot(gtex, aes(x=tau))
p + geom_density(color="#d95f02") +
    theme_bw()
```



Compare to Fagerberg (2014) data

- Dataset derived from Fagerberg (2014) Supplementary data;
- formating in analogy to formating in Mostacci (2014);
- formating for zero counts adjusted to adding 1 to each count value instead of only setting 0 counts to 1.

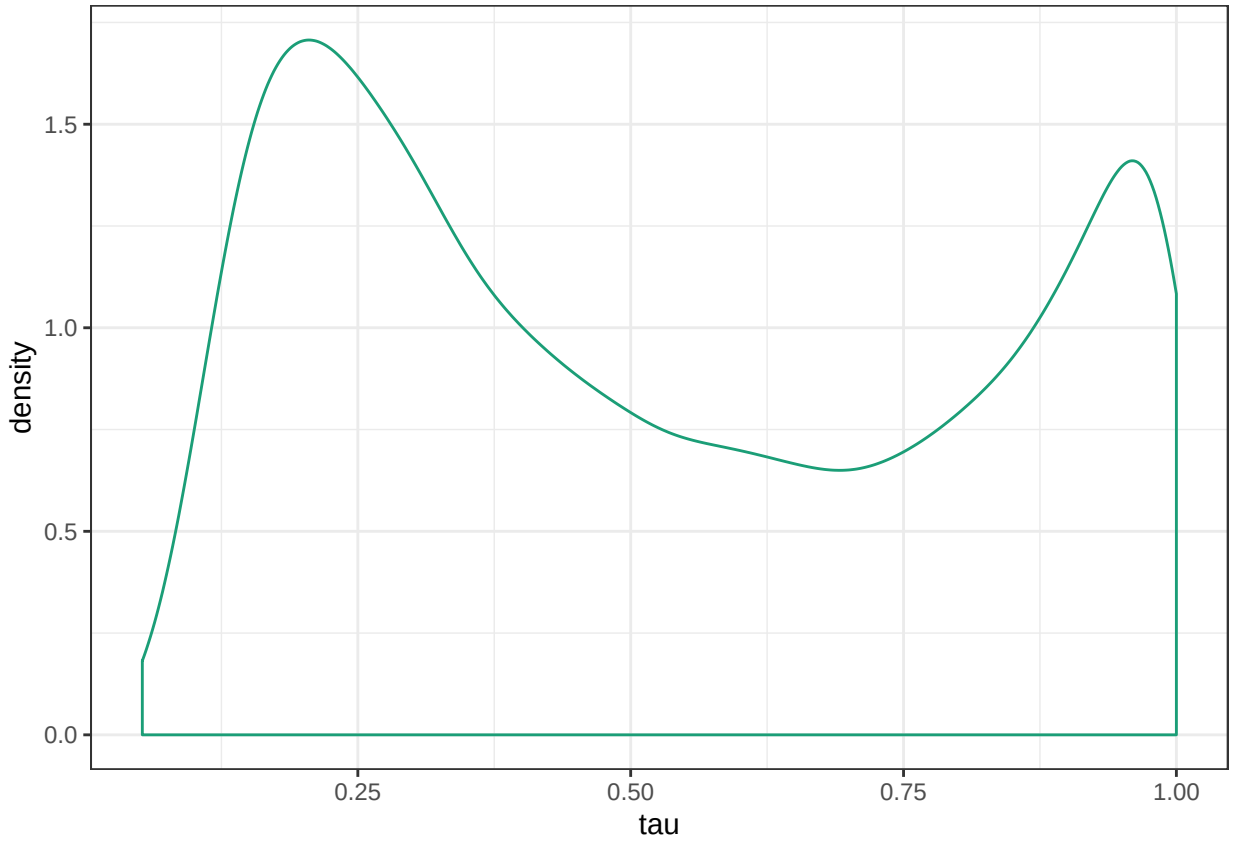
```
fb_file<- "~/data/public/2016_bioinformatics_mostacci/2014_MCP_Fagerberg.xlsx"
orgExpression <- readxl::read_xlsx(fb_file)
colnames(orgExpression)[1] <- "Ensembl.Gene.ID"

orgExpression <-
    orgExpression[regexpr("ENS", orgExpression$Ensembl.Gene.ID) > 0 |
                      regexpr("FBgn", orgExpression$Ensembl.Gene.ID) > 0 |
                      regexpr("PPAG", orgExpression$Ensembl.Gene.ID) > 0, ]
orgExpression <- na.omit(orgExpression[,-29])
x <- orgExpression[,-1] + 1
orgExpression[, -1] <- log2(x)
fagerberg_max <- apply(orgExpression[,-1], 1, fmax)
orgExpression <- orgExpression[fagerberg_max > log2(rpkm),]
fagerberg <- data.frame(id=orgExpression$Ensembl.Gene.ID,
                        full=orgExpression$Ensembl.Gene.ID,
                        tau=apply(orgExpression[,-1], 1, tau),
                        study='fagerberg',
                        stringsAsFactors = FALSE)

fagerberg <- fagerberg %>%
```

```
    left_join(tpm_attr, by = c("id" = "ID")) %>%
    select(Description, everything()) %>%
    select(-Name) %>%
    rename(Gene=Description, ENSG=id)

p <- ggplot(fagerberg, aes(x=tau))
p + geom_density(color="#1b9e77") +
    theme_bw()
```



Compare the results of GTEx and Fagerberg tissue specificity

- Fagerberg 19862 genes;
- GTEx 31473 genes;
- Find common genes and compare distribution of tau;
- check overlap of TRA sets

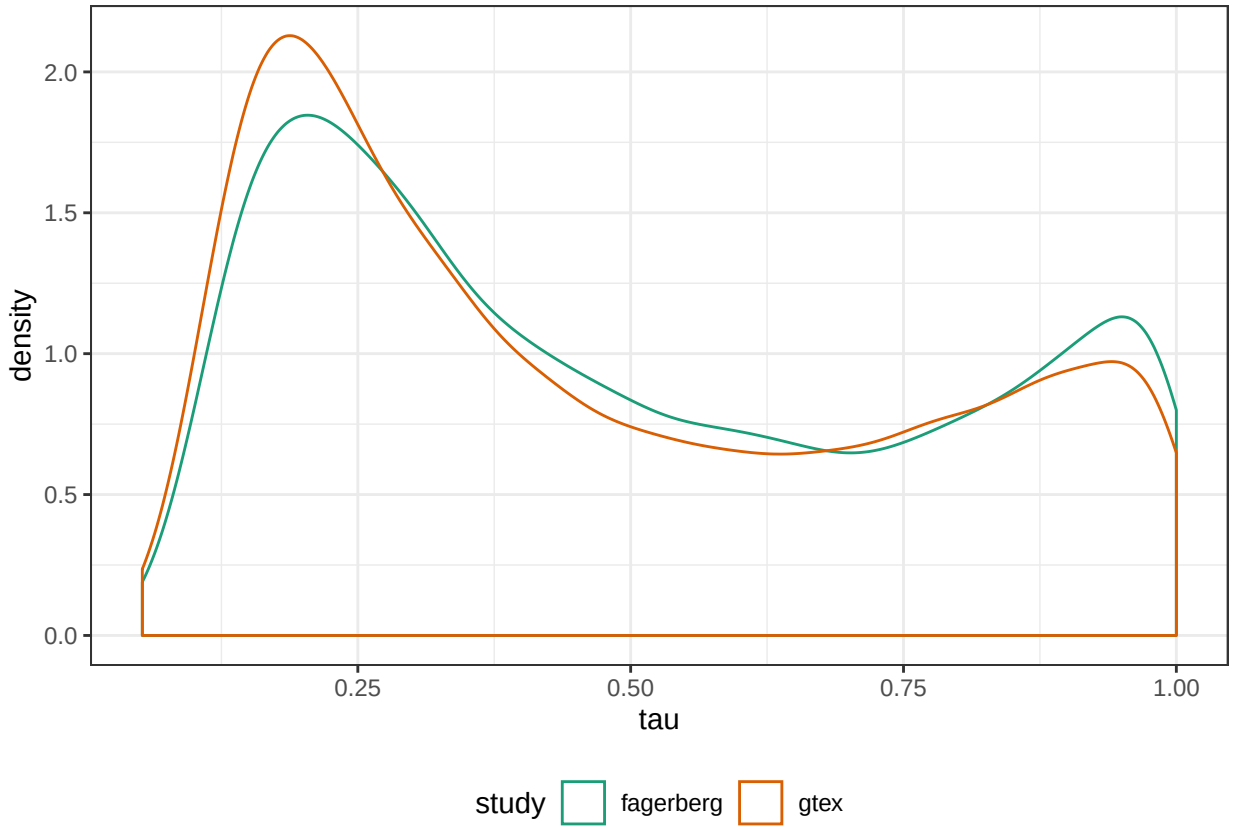
```
common <-  intersect(fagerberg$ENSG, gtex$ENSG)

fagerberg_common <- fagerberg[fagerberg$ENSG %in% common,]
gtex_common <- gtex[gtex$ENSG %in% common,]

both <- rbind(fagerberg_common, gtex_common)

p <- ggplot(both, aes(x=tau, color=study))
p + geom_density() +
    scale_color_manual(values=c("#1b9e77", "#d95f02")) +
```

```
    theme_bw() +
    theme(legend.position = 'bottom')
```




```
gtex_tissue_specific <- gtex_common[gtex_common$tau > 0.8,]
fagerberg_tissue_specific <- fagerberg_common[fagerberg_common$tau > 0.8,]

gtex_pct <- sum(gtex_tissue_specific$ENSG %in% fagerberg_tissue_specific$ENSG)/
    nrow(gtex_tissue_specific)

fagerberg_pct <- sum(gtex_tissue_specific$ENSG %in%
                         fagerberg_tissue_specific$ENSG)/
    nrow(fagerberg_tissue_specific)
```

There are 18301 common genes in the GTEx and Fagerberg dataset. Of those, there are 3608 genes with $\tau > 0.8$ in the GTEx dataset, ie genes we consider as TRA, and 4038 TRAs in the Fagerberg dataset. The overlap between these are 0.92 and 0.82 respectively.