Estimating TRAs from GTex

Hannah Meyer

2021-06-09

Contents

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 specific manner. In the following, we use a gene expression data set of about 50k transcripts across 30 tissues (GTEx v8) 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

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

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

Setup

```
library(tidyverse)
box::use(tra=./tra)
datadir <- "~/data/tra/gtex"
text_size <- 7
title_size <- 9
```

Download files from GTex: all genes quantified

```
gtex=https://storage.googleapis.com/gtex_analysis_v8/annotations
datadir=~/data/tra/gtex

cd $datadir
wget $gtex/GTEx_Analysis_v8_Annotations_SampleAttributesDS.txt
wget $gtex/GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm.gct.gz
gunzip $gtex/GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm.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_Analysis_v8_Annotations_SampleAttributesDS.txt"),
                                data.table=FALSE) %>%
                    as_tibble

samples_tpm <- data.table::fread(file.path(datadir,
                                "GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_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))

write_csv(tpm_attr,
          file.path(datadir,
                    "GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm_attr.csv")
)

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")

write_csv(tpm_annotated,
          file.path(datadir,
                    "GTEx_Analysis_2017-06-05_v8_RNASeQCv1.1.9_gene_tpm_gct_annotated.csv")
)
```

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
saveRDS(mean_expression, "GTEx_Analysis_v8_gene_mean_expression.Rdata")
```

GTex data: only refseq genes expressed in xx tissues

```
gtex_genes <- read_csv(file.path(datadir,"GTEx_Genes.csv"))
colnames(gtex_genes)[1] <- "ID"
gtex_genes <- gtex_genes %>%
    drop_na(ID) %>%
    column_to_rownames("ID") %>%
    t %>%
    as.data.frame %>%
    rownames_to_column("ID") %>%
    mutate(tissue = gsub("(.*)_GTEX.*", "\\1", ID))

gtex_genes_mean <- gtex_genes %>%
    select(-ID) %>%
    group_by(tissue) %>%
    summarize(across(.fns=mean)) %>%
    ungroup %>%
    column_to_rownames("tissue") %>%
    t

gtex_transcripts <- read_csv(file.path(datadir,"GTEx_Transcripts.csv"))
colnames(gtex_transcripts)[1] <- "ID"
gtex_transcripts <- gtex_transcripts  %>%
    drop_na(ID) %>%
    column_to_rownames("ID") %>%
    t %>%
    as.data.frame %>%
    rownames_to_column("ID") %>%
    mutate(tissue = gsub("(.*)_GTEX.*", "\\1", ID))
gtex_transcripts_mean <- gtex_transcripts %>%
    select(-ID) %>%
    group_by(tissue) %>%
    summarize(across(.fns=mean)) %>%
    ungroup %>%
    column_to_rownames("tissue") %>%
    t
```

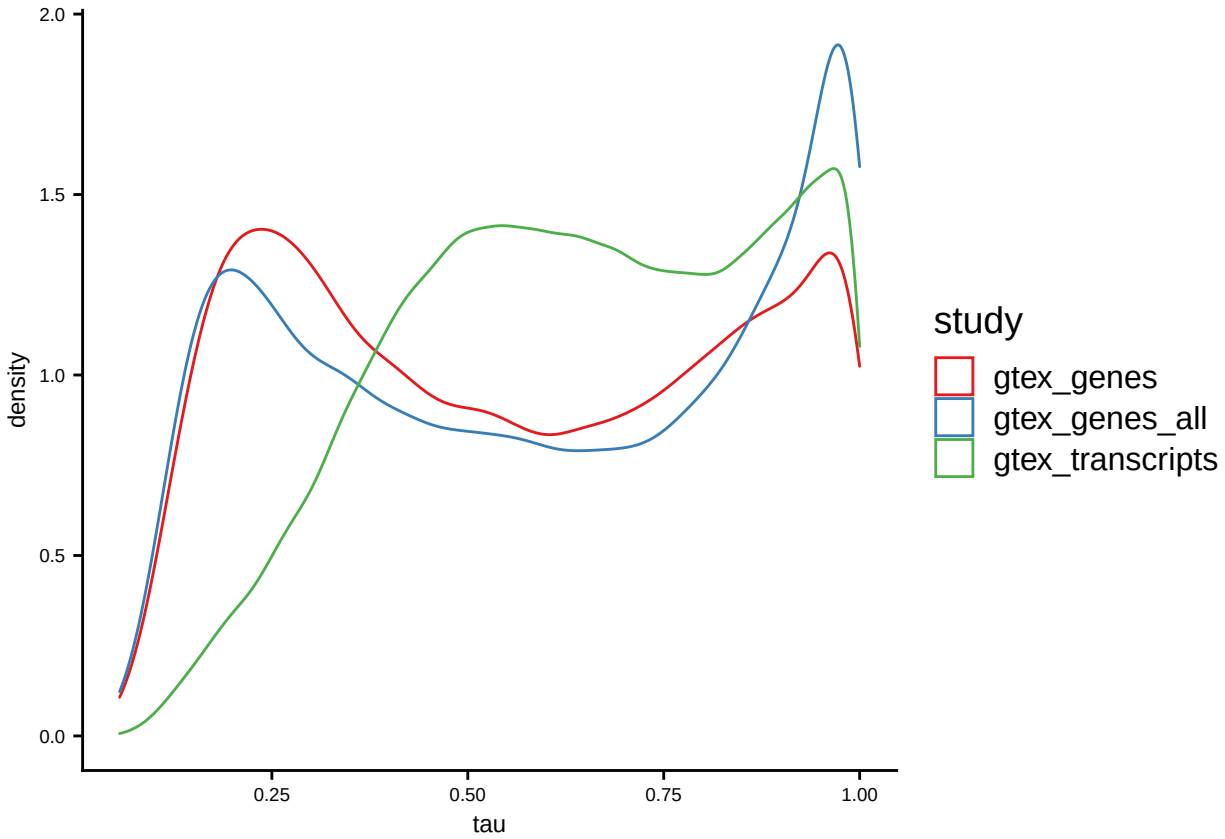
Compute list of tissue-specific genes

We follow the processing processing procedure described in the Kryuchkova-Mostacci 2017 benchmarking study:

- All genes where the highest gene expression < 1 TPM are set as not expressed;
- count data are log-transformed after addition of 1 to avoid zero counts.

```
filter_expression <-  function(mean_expression,  tpm=1) {

    gtex_max <- apply(mean_expression, 1, tra$fmax)
    mean_expression_filter <- mean_expression[gtex_max > tpm,]
    mean_expression_filter <- mean_expression_filter + 1

    return(mean_expression_filter)
}
```

```
get_tra <- function(mean_expression_filter, study, tpm=1) {

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

    write.table(select(gtex, -study),
            file=file.path(datadir, paste0(study, "_tau.csv")),
            sep=",", col.names=TRUE, row.names=FALSE, quote=FALSE)

    return(gtex)
}

gtex_genes_filter <- filter_expression(gtex_genes_mean)
gtex_transcripts_filter <- filter_expression(gtex_transcripts_mean)
gtex_all_filter <- filter_expression(gtex_all_mean)

gtex_genes_tau <- get_tra(gtex_genes_filter, "gtex_genes")
gtex_transcripts_tau <- get_tra(gtex_transcripts_filter , "gtex_transcripts")
gtex_all_tau <- get_tra(gtex_all_filter, "gtex_genes_all")

gtex_combine <- rbind(gtex_genes_tau,gtex_transcripts_tau, gtex_all_tau)
p <- ggplot(gtex_combine, aes(x=tau))
p + geom_density(aes(color=study)) +
    scale_color_brewer(type='qual', palette = "Set1") +
    cowplot::theme_cowplot() +
    theme(axis.text = element_text(size=text_size),
          axis.title = element_text(size=title_size))
```




```
tt <- tibble(GTEx = c("All GTEx Genes", "RefSeq Genes", "RefSeq Transcripts"),
       total= c(nrow(gtex_all_mean), nrow(gtex_genes_mean), nrow(gtex_transcripts_mean)),
       expressed = c(nrow(gtex_all_tau),
                                nrow(gtex_genes_tau),
                                nrow(gtex_transcripts_tau)),
       tra_0.7 = c(sum(gtex_all_tau$tau >= 0.7),
                               sum(gtex_genes_tau$tau >= 0.7),
                               sum(gtex_transcripts_tau$tau >= 0.7)),
       tra_0.8 = c(sum(gtex_all_tau$tau >= 0.8),
                               sum(gtex_genes_tau$tau >= 0.8),
                               sum(gtex_transcripts_tau$tau >= 0.8))
)
kableExtra::kable(tt, format="latex",
                  col.names = c("GTEx", "Total", "Expressed: TPM $\\ge$ 1",
                                "TRA: $\\tau \\ge$ 0.7",
                                "TRA: $\\tau \\ge$ 0.8"))
```

GTEx	Total	Expressed: TPM $\ge$ 1	TRA: $\tau \ge$ 0.7	TRA: $\tau \ge$ 0.8
All GTEx Genes	56200	32347	13413	10687
RefSeq Genes	40481	26131	9587	7083
RefSeq Transcripts	194360	118256	50875	35599

Find tissue-specificity of tissue-specific genes

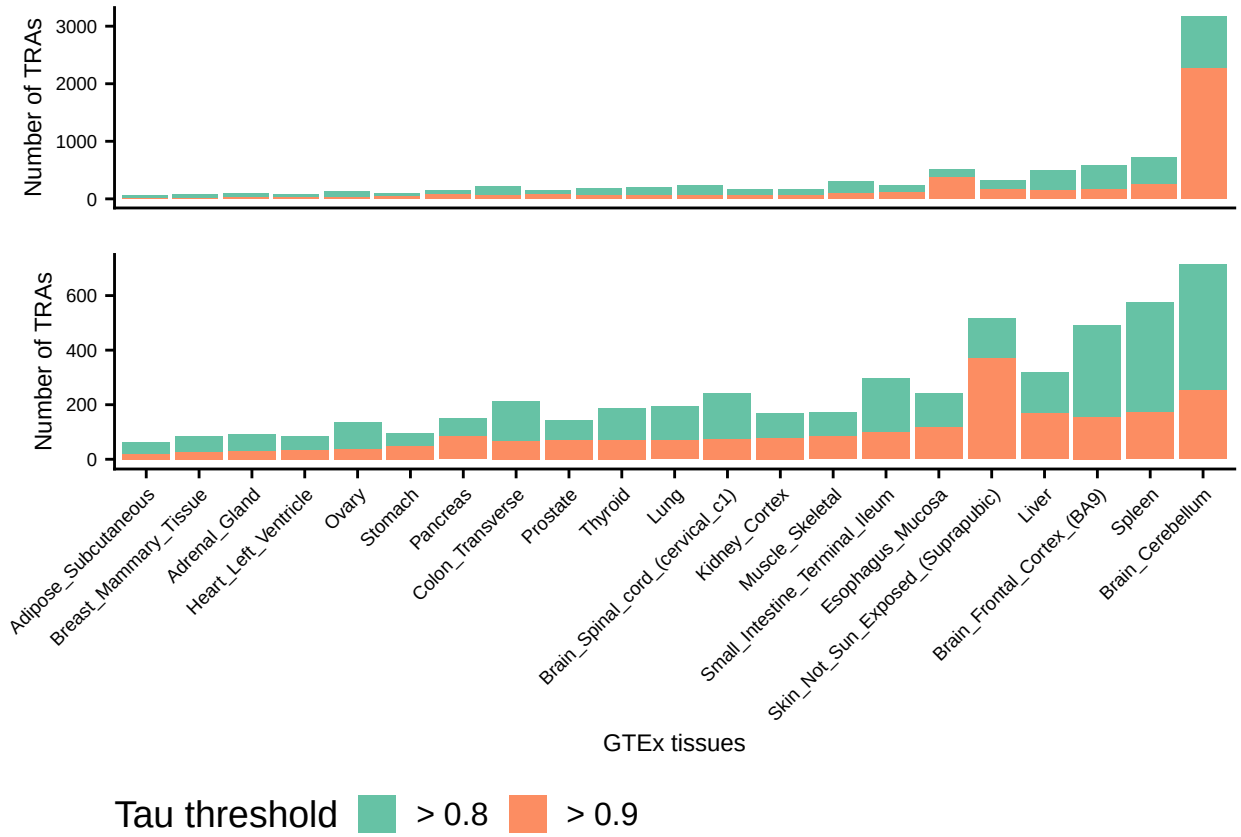
- use binarizing approach described by Yanai 2005

```
find_tissue_spec <- function(mean_expression_filter, feature_tau, study,
                             tauthr=0.8) {
    feature_tau <- feature_tau[feature_tau$tau > tauthr,]
    mean_expression_tra <- mean_expression_filter %>%
        as.data.frame %>%
        rownames_to_column("full") %>%
        filter(full %in% feature_tau$full) %>%
        column_to_rownames("full") %>%
        as.matrix

    expression_tra_tissues <- apply(mean_expression_tra, 1,
                                    tra$find_tra_tissues) %>%
        t %>%
        as.data.frame %>%
        rownames_to_column("full") %>%
        left_join(feature_tau) %>%
        select(-full, -study) %>%
        select(id, tau, everything()) %>%
        as_tibble()

    write.table(expression_tra_tissues,
            file=file.path(datadir, paste0(study, "_binarized_expression.csv")),
            sep=",", col.names=TRUE, row.names=FALSE, quote=FALSE)

    return(expression_tra_tissues)
}

gtex_genes_tissues <- find_tissue_spec(gtex_genes_filter, gtex_genes_tau,
                                       "gtex_genes")
gtex_transcripts_tissues <- find_tissue_spec(gtex_transcripts_filter, gtex_transcripts_tau,
                                       "gtex_transcripts")
gtex_genes_all_tissues <- find_tissue_spec(gtex_all_filter, gtex_all_tau,
                                       "gtex_genes_all")
```

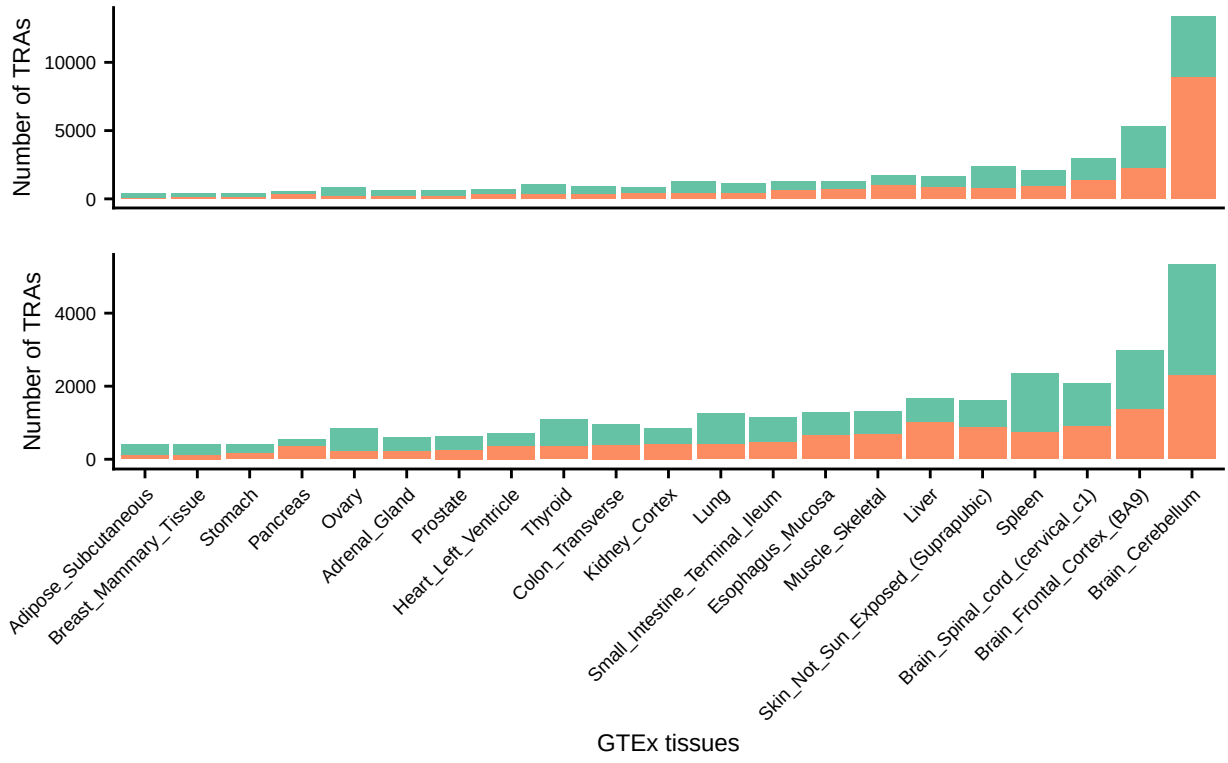
- visualise distribution of TRAs per tissue

```
visualise_tra_per_tissue <- function(expression_tra_tissues) {
    tras_per_tissue <- expression_tra_tissues %>%
        mutate(thr=case_when(tau < 0.8 ~ "> 0.7",
                             tau >= 0.8 & tau < 0.9 ~ "> 0.8",
                             tau >= 0.9 ~ "> 0.9")) %>%
        pivot_longer(-c(id, tau, thr),
                     names_to="tissue", values_to="status") %>%
        filter(status != 0) %>%
        group_by(tissue, thr) %>%
        summarise(tras = n(), .groups='drop') %>%
        arrange(tras) %>%
        mutate(tissue = fct_inorder(tissue))

    p_all_tissues <- ggplot(tras_per_tissue) +
        geom_bar(aes(x=tissue, y=tras, fill=thr), stat='identity') +
        scale_fill_manual(values=c('#66c2a5','#fc8d62','#8da0cb'), guide=FALSE) +
        labs(x="GTEx tissues",
             y="Number of TRAs",
```

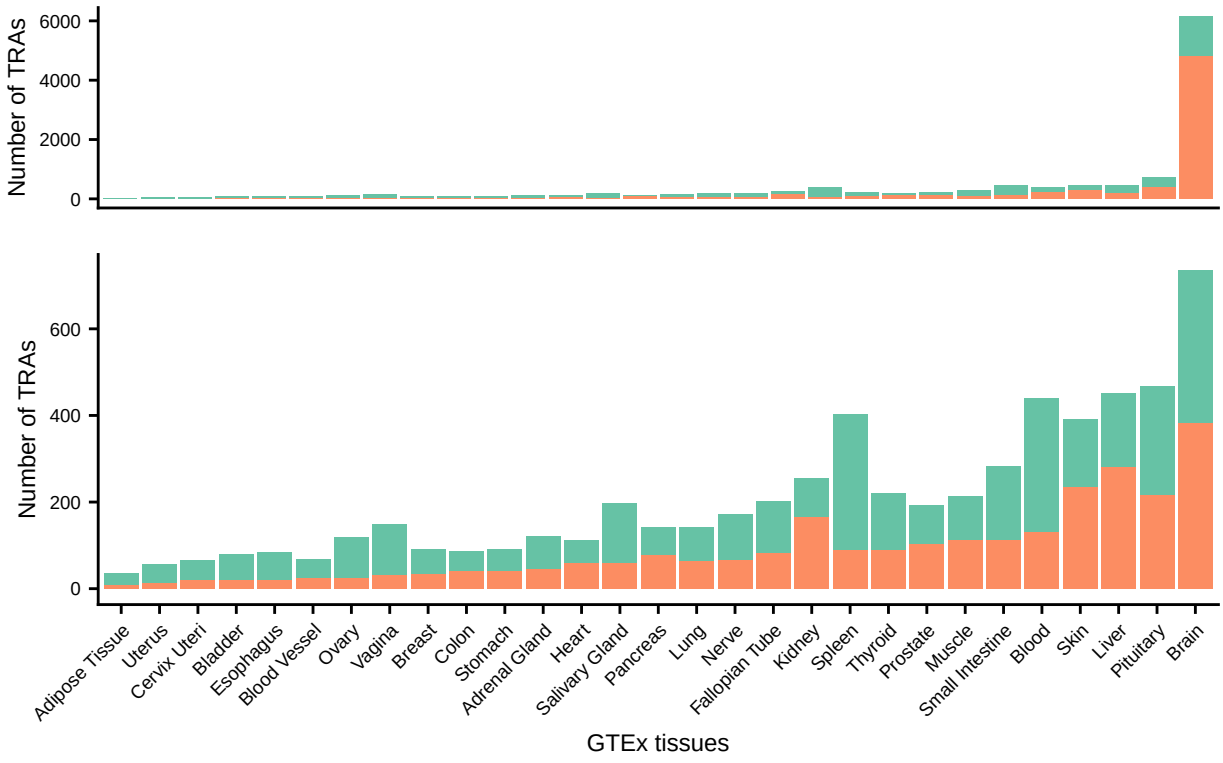
```
            fill="Tau threshold") +
        cowplot::theme_cowplot() +
        theme(axis.text.x = element_blank(),
              axis.title.x = element_blank(),
              axis.ticks.x = element_blank(),
              axis.text.y = element_text(size=text_size),
              axis.title.y = element_text(size=title_size))

    p_tissues_no_testes <- ggplot(filter(tras_per_tissue, tissue != "Testis")) +
        geom_bar(aes(x=tissue, y=tras, fill=thr), stat='identity') +
        scale_fill_manual(values=c('#66c2a5','#fc8d62','#8da0cb')) +
        labs(x="GTEx tissues",
             y="Number of TRAs",
             fill="Tau threshold") +
        cowplot::theme_cowplot() +
        theme(axis.text.x = element_text(angle=45, hjust = 1, vjust = 1),
              axis.text = element_text(size=text_size),
              axis.title = element_text(size=title_size),
              legend.position = "bottom")
    cowplot::plot_grid(p_all_tissues, p_tissues_no_testes,
                       nrow=2,
                       align="v",
                       rel_heights = c(1, 2.5),
                       axis="lr"
                       )
}
gtex_genes_tra_per_tissue <- visualise_tra_per_tissue(gtex_genes_tissues)
gtex_transcripts_tra_per_tissue <- visualise_tra_per_tissue(gtex_transcripts_tissues)
gtex_all_tra_per_tissue <- visualise_tra_per_tissue(gtex_genes_all_tissues)

gtex_genes_tra_per_tissue
```




```
gtex_transcripts_tra_per_tissue
```



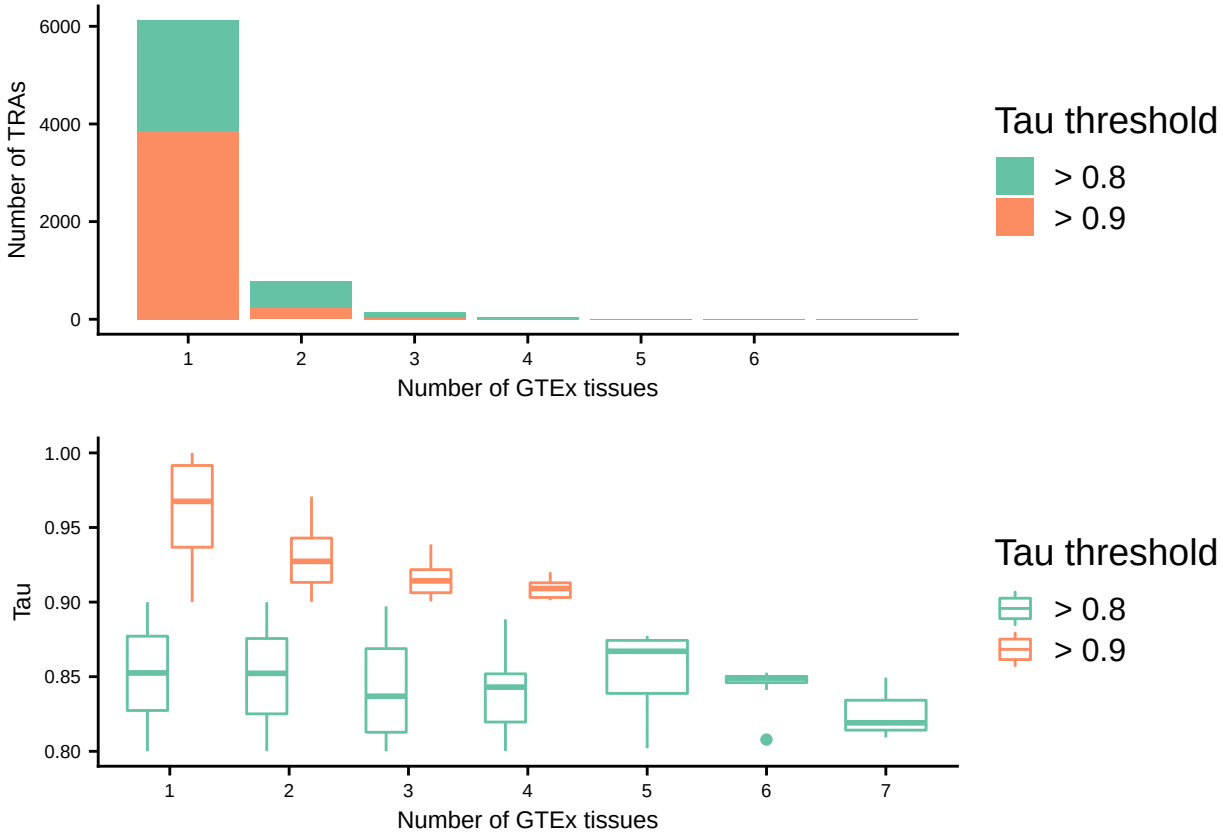

```
gtex_all_tra_per_tissue
```



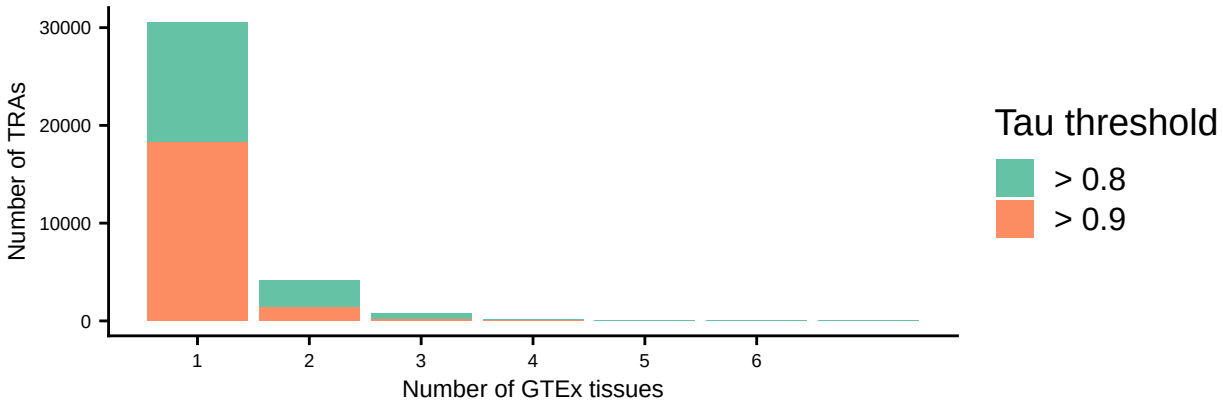
- visualize how many tissues a given TRA is assigned
- visualize relationship between tau and number of tissues expressing TRA

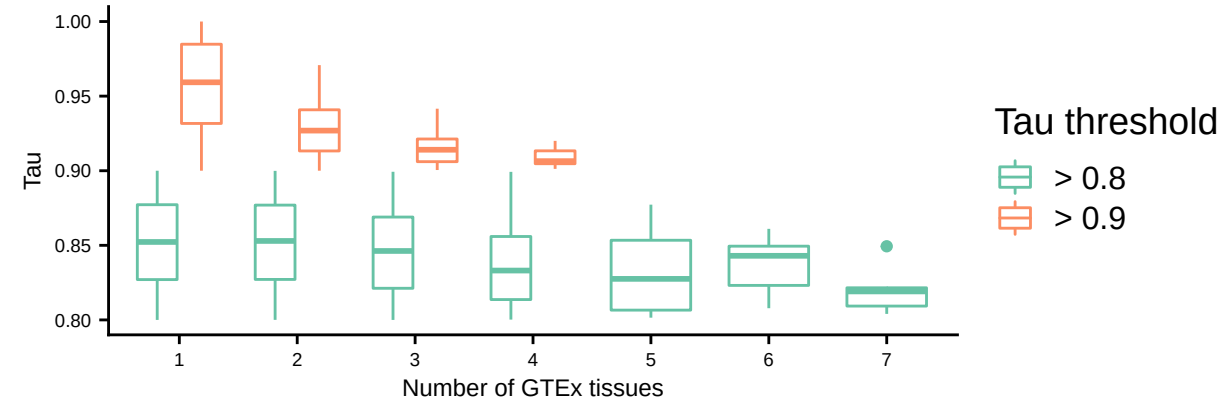
```
visualise_tissue_per_tra <- function(expression_tra_tissues) {
    tissue_per_tra <- expression_tra_tissues %>%
        mutate(thr=case_when(tau < 0.8 ~ "> 0.7",
                             tau >= 0.8 & tau < 0.9 ~ "> 0.8",
                             tau >= 0.9 ~ "> 0.9")) %>%
        pivot_longer(-c(id, tau, thr), names_to="tissue", values_to="status") %>%
        filter(status != 0) %>%
        group_by(id, tau, thr) %>%
        summarise(tissues = n(), .groups='drop') %>%
        arrange(tissues) %>%
        mutate(id = fct_inorder(id))

    p_all_tras <-  ggplot(tissue_per_tra) +
        geom_bar(aes(tissues, fill=thr)) +
        scale_fill_manual(values=c('#66c2a5','#fc8d62','#8da0cb')) +
        scale_x_continuous( breaks=1:6)+
        labs(x="Number of GTEx tissues",
             y="Number of TRAs",
             fill="Tau threshold") +
        cowplot::theme_cowplot() +
        theme(axis.text = element_text(size=text_size),
              axis.title = element_text(size=title_size))

    p_tau_versus_tissue <-  ggplot(tissue_per_tra) +
        geom_boxplot(aes(x=as.factor(tissues), y=tau, color=thr)) +
```

```
        scale_color_manual(values=c('#66c2a5','#fc8d62','#8da0cb')) +
        labs(x="Number of GTEx tissues",
             y="Tau",
             color="Tau threshold") +
        cowplot::theme_cowplot() +
        theme(axis.text = element_text(size=text_size),
              axis.title = element_text(size=title_size))

    cowplot::plot_grid(p_all_tras,p_tau_versus_tissue,
                       nrow=2,
                       align="v",
                       axis="lr")
}

gtex_genes_tissue_per_tra <- visualise_tissue_per_tra(gtex_genes_tissues)
gtex_transcripts_tissue_per_tra  <- visualise_tissue_per_tra(gtex_transcripts_tissues)
gtex_all_tissue_per_tra  <- visualise_tissue_per_tra(gtex_genes_all_tissues)

gtex_genes_tissue_per_tra
```

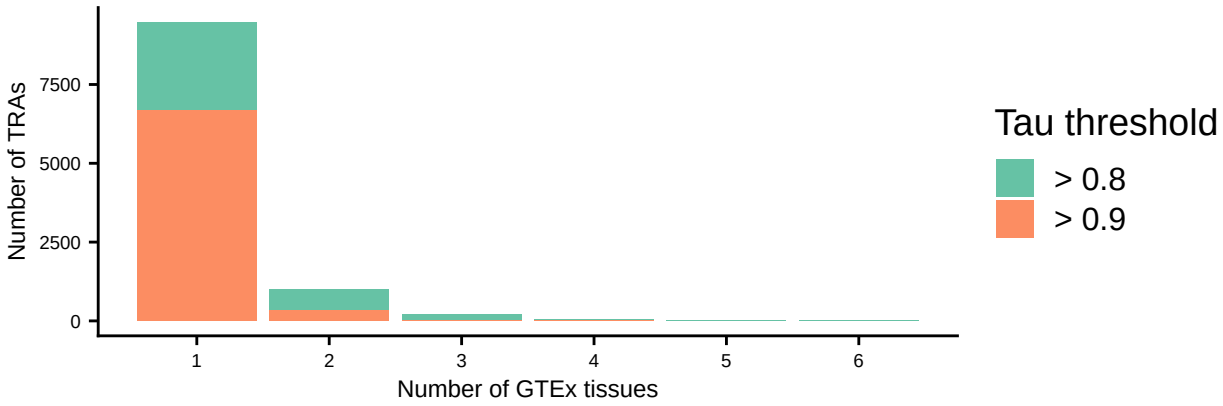


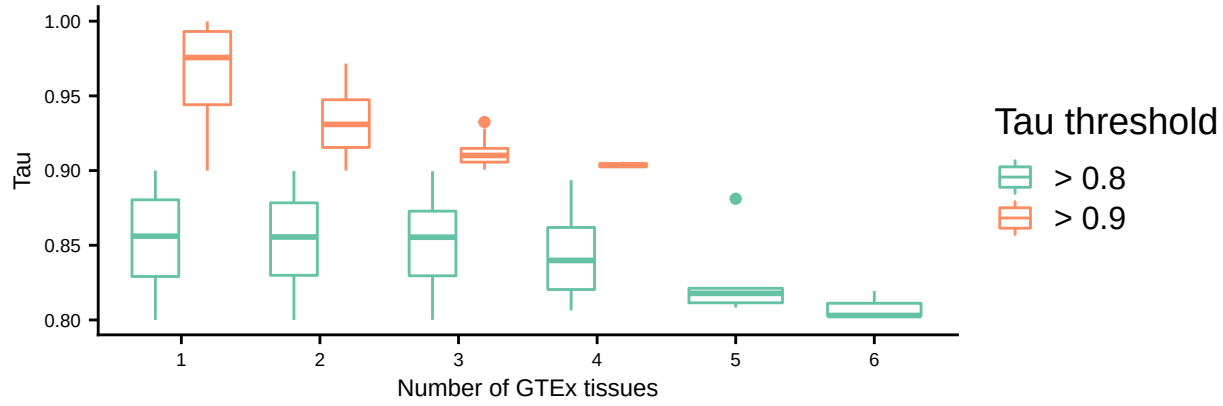

```
gtex_transcripts_tissue_per_tra
```






```
gtex_all_tissue_per_tra
```



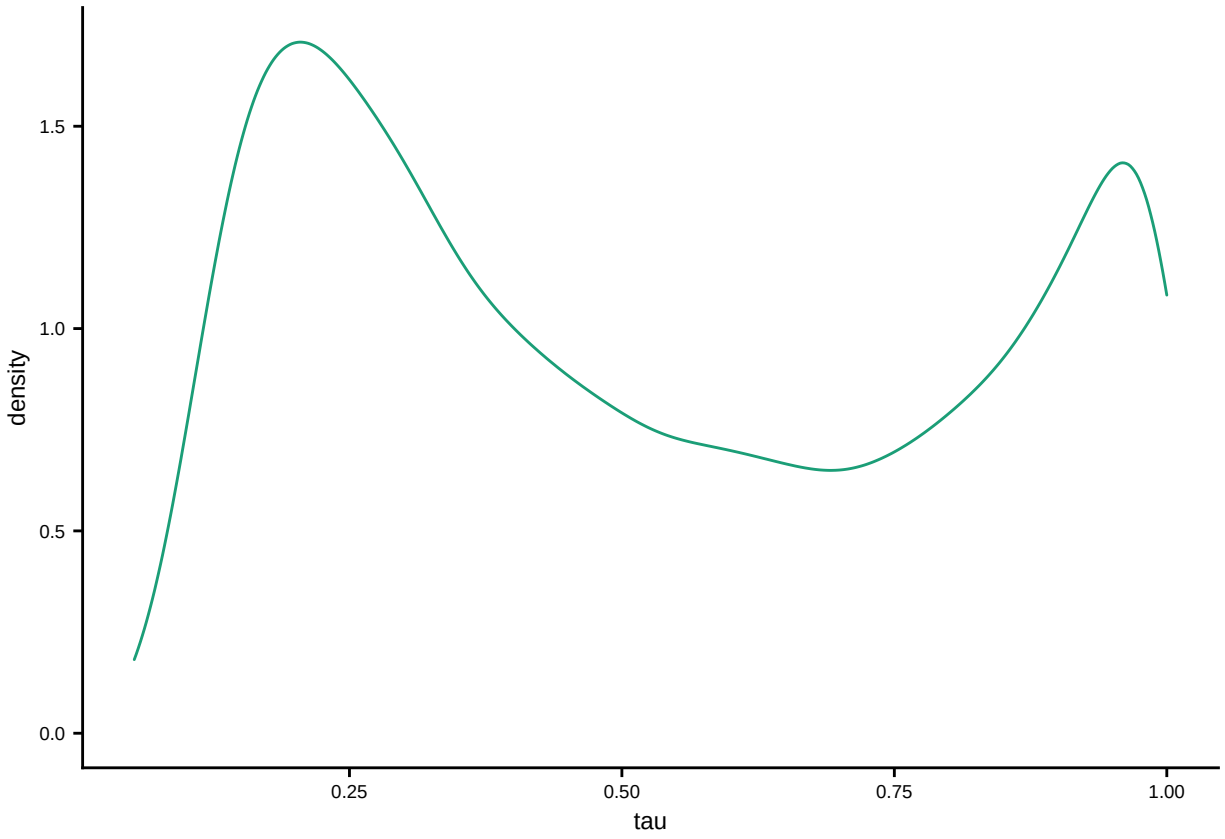


Compare to Fagerberg (2014) data

- Data set derived from Fagerberg (2014) Supplementary data;
- formatting in analogy to formating in Mostacci (2014);
- formatting 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$Ensembl.Gene.ID) > 0, ]
orgExpression <- na.omit(orgExpression[,-29])
x <- orgExpression[,-1] + 1
orgExpression[, -1] <- log2(x)
fagerberg_max <- apply(orgExpression[,-1], 1, tra$fmax)
orgExpression <- orgExpression[fagerberg_max > log2(1),]
fagerberg <- data.frame(id=orgExpression$Ensembl.Gene.ID,
                        full=orgExpression$Ensembl.Gene.ID,
                        tau=apply(orgExpression[,-1], 1, tra$tau),
                        study='fagerberg',
                        stringsAsFactors = FALSE)

fagerberg <- fagerberg %>%
```

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

p <- ggplot(fagerberg, aes(x=tau))
p + geom_density(color="#1b9e77") +
    cowplot::theme_cowplot() +
    theme(axis.text = element_text(size=text_size),
          axis.title = element_text(size=title_size))
```

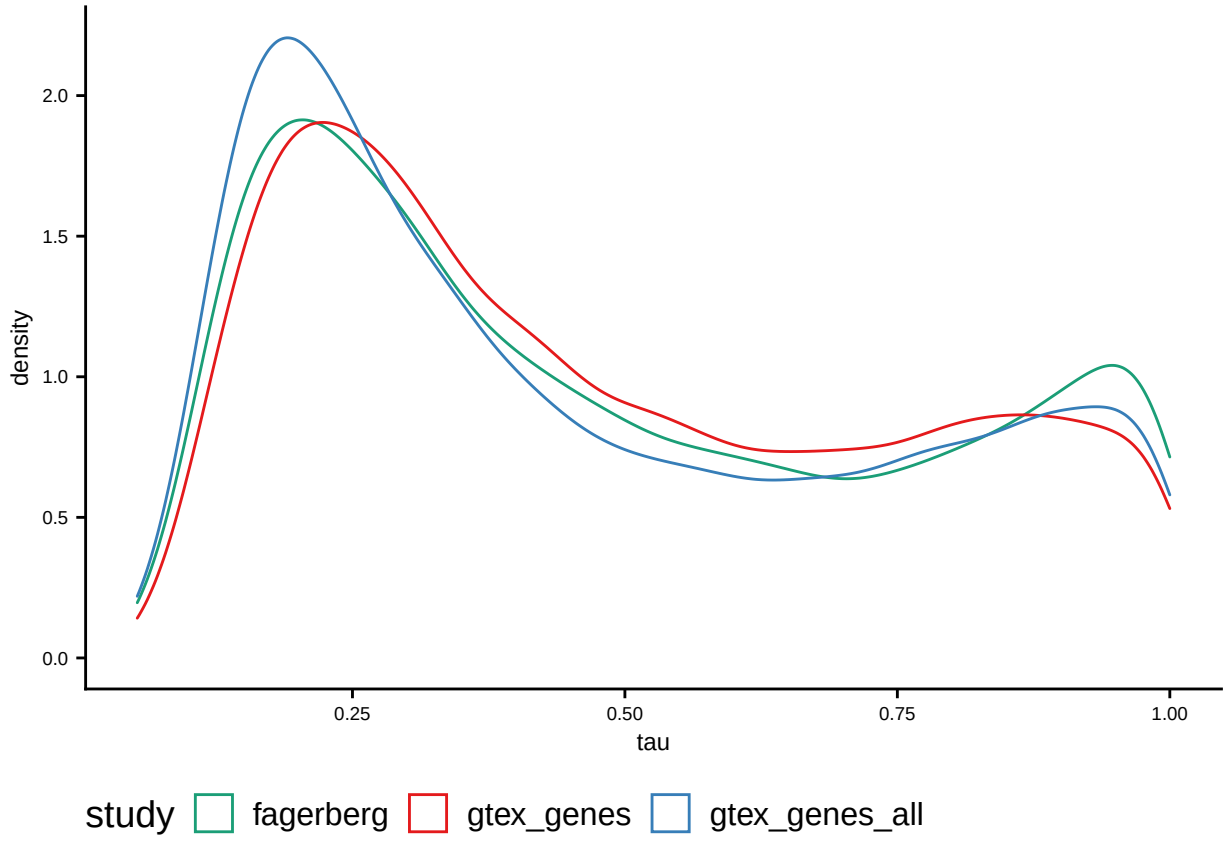


Compare the results of GTEx and Fagerberg tissue specificity

- Fagerberg 19881 genes;
- GTEx (all genes: 32347, refseq only genes 26131);
- Find common genes and compare distribution of tau;
- check overlap of TRA sets

```
all_ids <-  c(fagerberg$id, gtex_all_tau$id, gtex_genes_tau$id)
common <-  table(all_ids)[table(all_ids) == 3]

fagerberg_common <- fagerberg[fagerberg$id %in% names(common),]
gtex_all_genes_common <- gtex_all_tau[gtex_all_tau$id %in% names(common),]
gtex_genes_common <- gtex_genes_tau[gtex_genes_tau$id %in% names(common),]

combined <- rbind(fagerberg_common[, -1], gtex_all_genes_common, gtex_genes_common)
```

```
p <- ggplot(combined, aes(x=tau, color=study))
p + geom_density() +
    scale_color_manual(values=c("#1b9e77", "#e41a1c", "#377eb8"))+
    cowplot::theme_cowplot() +
    theme(axis.text = element_text(size=text_size),
          axis.title = element_text(size=title_size),
          legend.position = 'bottom')
```




```
gtex_all_genes_tissue_specific <- gtex_all_genes_common[gtex_all_genes_common$tau > 0.8,]
gtex_genes_tissue_specific <- gtex_genes_common[gtex_genes_common$tau > 0.8,]
fagerberg_tissue_specific <- fagerberg_common[fagerberg_common$tau > 0.8,]

gtex_all_genes_pct <- sum(gtex_all_genes_tissue_specific$id %in% fagerberg_tissue_specific$id)/
    nrow(gtex_all_genes_tissue_specific)

gtex_genes_pct <- sum(gtex_genes_tissue_specific$id %in% fagerberg_tissue_specific$id)/
    nrow(gtex_genes_tissue_specific)

fagerberg_all_genes_pct <- sum(gtex_all_genes_tissue_specific$id %in%
                          fagerberg_tissue_specific$id)/
    nrow(fagerberg_tissue_specific)

fagerberg_genes_pct <- sum(gtex_genes_tissue_specific$id %in%
                          fagerberg_tissue_specific$id)/
    nrow(fagerberg_tissue_specific)
```

There are 17447 common genes in the two GTEx (all genes, refseq genes) and Fagerberg dataset. Of those,

there are 3202 and 3127 genes with $\tau > 0.8$ in the GTEx all genes and refseq genes datasets, respectively, ie genes we consider as TRA, and 3565 TRAs in the Fagerberg dataset. The overlap between these GTEx and Fagerberg datasets are 0.91 and 0.82 for all genes in GTEx and 0.9 and 0.79 for refseq only genes.