Salmon vs. TEtranscript

Comparision between Salmon vs. TEtranscript

Object

Instead using sequences from Repbase, can Salmon give us similar results with TEtranscripts?

Experiments setup

- Data from Ohtani et al. 2013, and GSE47006
- Comparision between control KD(EGFP KD) vs. PiWi KD
- Salmon: Take the sequences of annotated genomic location from TEtranscripts website (DM3), and use the sequences as reference of qunatification

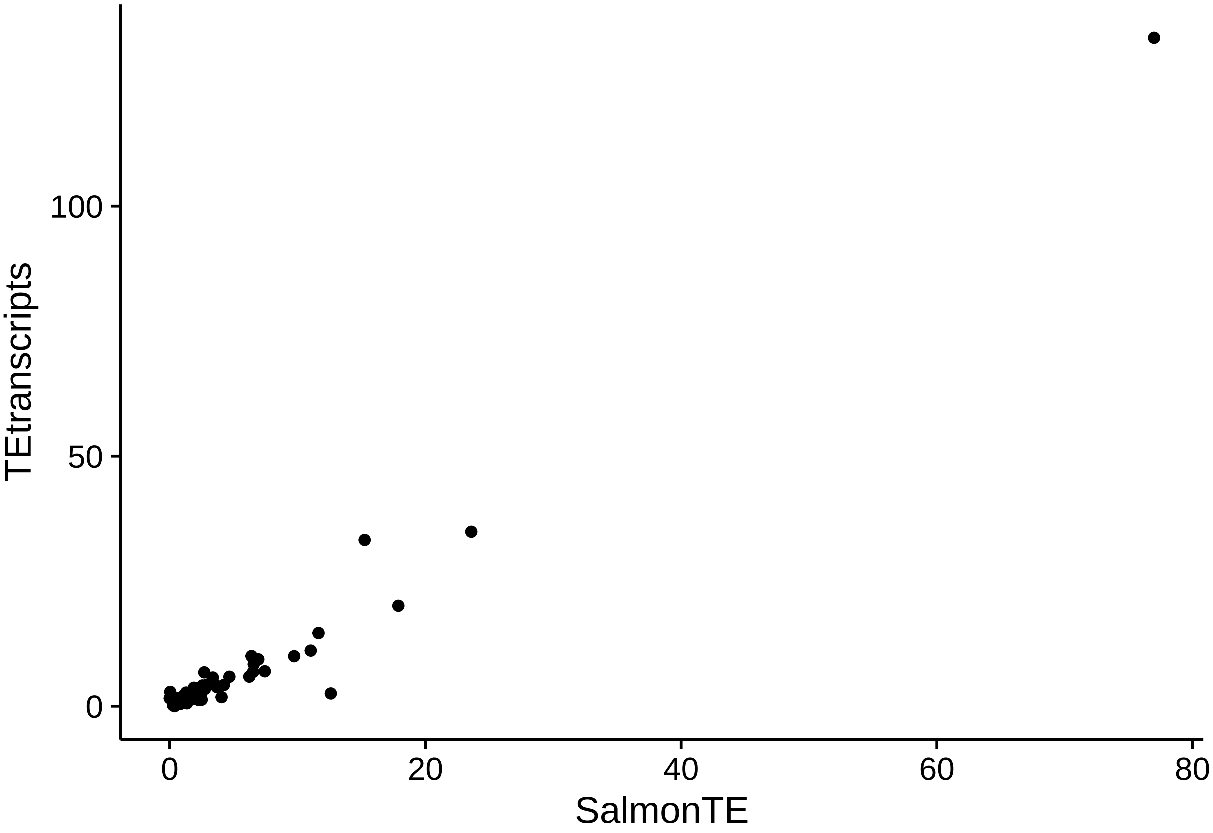
Import library and load Data

```
setwd("~/Sandbox/SalmonTE/")
library(tidyverse)
library(cowplot)
library(readxl)
library(knitr)
count.table <- read_excel("Salmon_vs_TEtranscripts.xlsx") %>%
  filter(!is.na(SRR851837_TE)&!is.na(SRR851838_TE))
#kable(head(count.table[,1:4]))
```

Scatter plot of FC

Scatter plot shows a clear linear relation between Salmon and TEtranscripts

```
count.table %>% filter(!is.na(FC)&!is.na(FC__1)) %>%
ggplot(aes_string(x="FC", y="FC__1")) +
  geom_point() +
  xlab("SalmonTE") +
  ylab("TEtranscripts")
```



Also, correlation between them is high (with Peason, and 0.84 with Spearman)

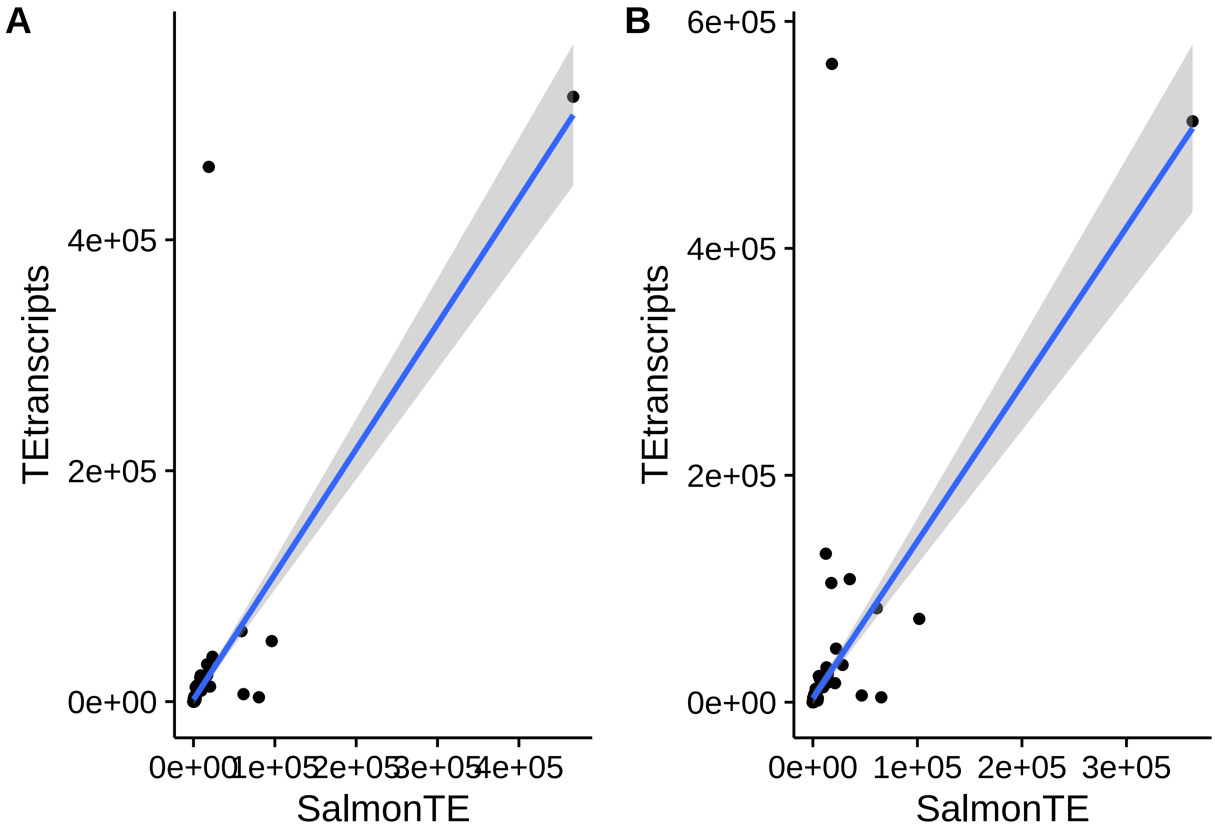
```
#cor(count.table$FC,count.table$FC__1,  use="complete.obs", method="spearman")
cor(count.table$FC,count.table$FC__1,  use="complete.obs", method="pearson")
```

```
## [1] 0.9809433
```

We also see the correlation of counts between them (A: Control KD, B: PiWi KD).

```
WT <- count.table %>%
  ggplot(aes(x=(SRR851837_Salmon), y=(SRR851837_TE))) +
   geom_point() +
  xlab("SalmonTE") +
  ylab("TEtranscripts") +
  geom_smooth(method="lm")
KO <- count.table %>%
  ggplot(aes(x=(SRR851838_Salmon), y=(SRR851838_TE))) +
   geom_point() +
  xlab("SalmonTE") +
  ylab("TEtranscripts") +
  geom_smooth(method="lm")

plot_grid(WT, KO, labels="AUTO")
```



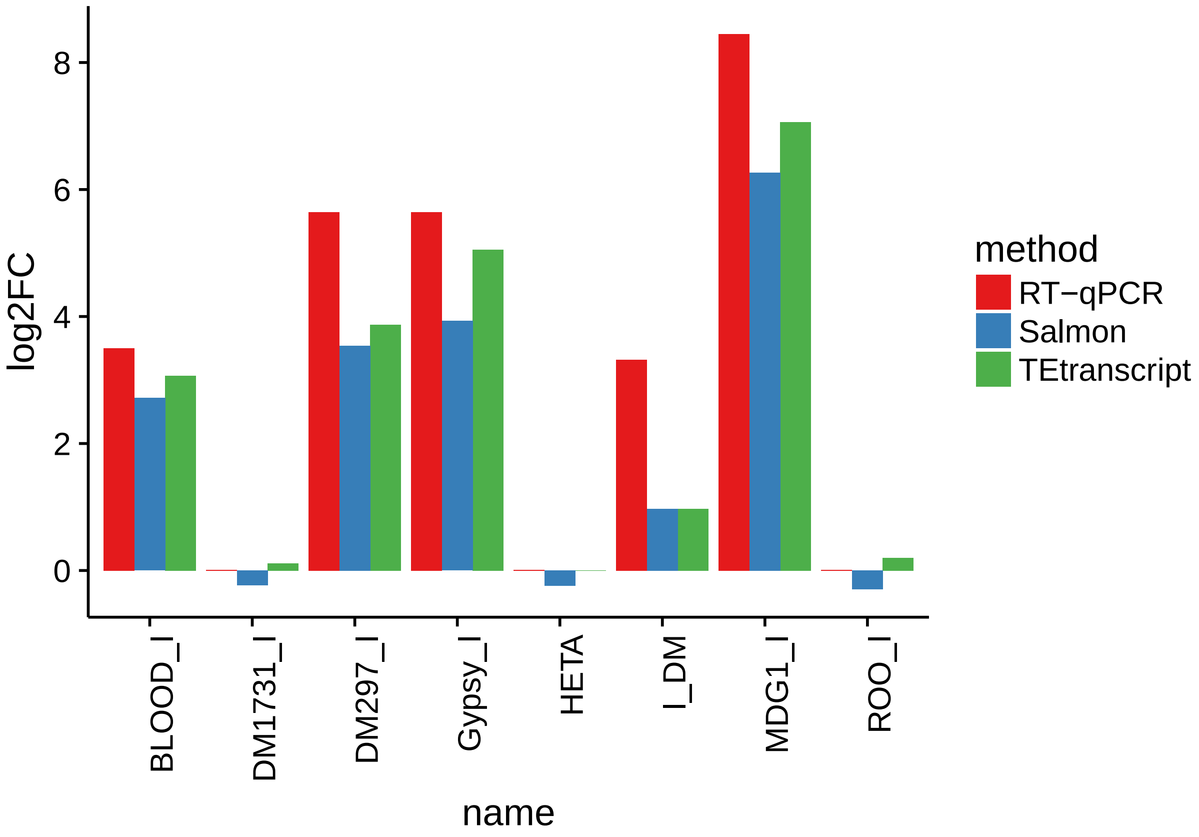
Furthermore, the correlation measurement shows they have higher simlarity between different quantifications for a same sample.

```r
kable(round(cor(count.table[,c(3,4,7,8)], use="complete.obs", method="spearman"),2))
```

	SRR851837_Salmon	SRR851838_Salmon	SRR851837_TE	SRR851838_TE
SRR851837_Salmon	1.00	0.93	0.91	0.86
SRR851838_Salmon	0.93	1.00	0.87	0.92
SRR851837_TE	0.91	0.87	1.00	0.95
SRR851838_TE	0.86	0.92	0.95	1.00

Next, I did a comparison with Fig 4 in TEtranscripts paper, and it shows that Salmon also give very similar result with TEtranscripts.

```r
tmp <- count.table %>% filter(InPaper=="Yes") %>% select(name, Log2, Log2__1)
colnames(tmp) <- c("name", "Salmon", "TEtranscript")
tmp %>% gather(method, log2FC, -name) %>%
  rbind(read.table("qpcr.tsv", header=T, sep="\t")) %>%
  ggplot(aes(x=name, y=log2FC)) +
  geom_bar(stat = "identity", aes(group=method, fill=method),
           position="dodge") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_brewer(palette = "Set1")
```



Running time comparision

- Ran on Macbook Pro (3.3 GHz i7, 16G, maximum 4 threads)
- Need to have a graph. It only takes for ~20 mins for each sample with single thread, even STAR+TEtranscripts takes more than a couple of hours.
- It would be better to try the experiments on Splicer or CRISPR server.

Discussion

- TPM of Salmon generally lower than normalized count of TEtranscripts, and I guess this can decrease $log_2 FC$ of Salmon.
- Better way to deal with the problem?

TODO

- Add results of Salmon with Repbase, which we did for ROSMAP data
- Sanitize the pipeline
- Looking for mouse Nanostring/RNAseq data in TEtranscripts

Additional Note for TEtranscripts paper and its source

Priotization

- This tool priotize TEs first, and we don't need to discard any read which is able to mapped to gene region!

reference: *Line 356-373* source code from https://github.com/mhammell-laboratory/tetoolkit/blob/master/bin/TEtranscripts

```
(annot_gene,annot_TE) = ovp_annotation(references,alignments_per_read, geneIdx, teIdx,stranded,format)

if len(alignments_per_read) > 1 : #multi read, prior to TE
    no_annot_te = parse_annotations_TE(multi_reads,annot_TE, te_counts, te_multi_counts, leftOver_te)

    if no_annot_te :
        no_annot_gene = parse_annotations_gene(annot_gene,gene_counts,leftOver_gene)
        if no_annot_gene :
            empty += 1

else : #uniq read , prior to gene
    no_annot_gene = parse_annotations_gene(annot_gene,gene_counts,leftOver_gene)
    if no_annot_gene :
        no_annot_te = parse_annotations_TE(multi_reads,annot_TE, te_counts, te_multi_counts, leftOver_t
        if no_annot_te :
            empty += 1
```

- The Fruit fly RNAseq dataset is single ended RNAseq dataset, and only use WT, and PiWi KO.
- Do we have better way to have show something with different genotype in this study?
- Can we have large scale RNAseq data to make some biological stories?

How to run star?

- I needed to use this alignment tool because TEtranscripts recommend to use it.
- Genome index building

```
STAR --runMode genomeGenerate --genomeDir dm3_genome --genomeFastaFiles chromFa/chr*.fa --sjdbGTFfile g
```

```
STAR --runThreadN 4 --genomeDir dm3_genome --readFilesIn SRR851838.fastq --outFilterMultimapNmax 100 --
STAR --runThreadN 4 --genomeDir dm3_genome --readFilesIn SRR851838.fastq --outFilterMultimapNmax 100 --
```

Note for SalmonTE pipeline

- GTF to FASTA

```
cat chromeFa/*.fa > genome.fa
gffread -w dm3_rmsk_TE.fa -g genome.fa dm3_rmsk_TE.gtf
```

- building index

```
salmon index -t dm3_rmsk_TE.fa -i dm3_rmsk_TE --type fmd
```

- running salmon

```
salmon quant -i dm3_rmsk_TE  -l A -r SRR851837/SRR851837.fastq -o SRR851837_Salmon
salmon quant -i dm3_rmsk_TE  -l A -r SRR851838/SRR851838.fastq -o SRR851838_Salmon
```

- merge TPM for each sample (merge_dup.py)

```
import sys


def get_tpm(fname):
    from collections import defaultdict
    tpm = defaultdict( float )
    with open(fname, "r") as inp:
        line = inp.readline()
        for line in inp:
            line = line.strip().split()
            name = "_".join(line[0].split("_")[:-1])
            if len(name) == 0:
                name = line[0]
            tpm[name] += float(line[3])

    return tpm

def main():
    import glob
    tb = dict()
    for s in sorted(glob.glob("*_Salmon/quant.sf")):
        sid = s.split("_")[0]
        tb[sid] = get_tpm(s)

    import pandas as pd
    with open("Salmon.out", "w") as oup:
        oup.write(pd.DataFrame(tb).to_csv(sep="\t"))


if __name__ == "__main__":
    main()
```