The curious case of the missing T cell receptor transcripts: part 2

Implementation details

This is a follow-up describing how exactly we found our missing TCR transcripts when analyzing the thymus atlas dataset. Check out part 1 for the background. Here's what I did.

- Grab copies of the TCR segment reference from TRACER's public github page here -- look for FASTA files (ending in `.fa`) under the folder for `raw_seqs`.
- Write an R script to concatenate them, spitting out a FASTA file. Also keep track of where the "splice junction" boundaries should go, creating a GTF file to use later on in STAR index generation and a refFlat file for use in the Drop-seq tools' gene tagging step.
- Generate a tiny STAR index for the resulting tiny reference genome (or perhaps it should be called a reference "recombinome").
- Try realigning some of our data. Fail miserably. Apparently, STAR does not scale well when a high fraction of reads don't align to the reference, and most of our reads were not from TCR genes. For more on this topic, see Alex Dobin's comments on the STAR grougle goop.
- Concatenate the tiny TCR reference recombinome with the rest of the mm10 mouse reference genome. Using this as a reference allowed STAR to deal with most non-TCR reads in its typical efficient manner.
- Subset reads falling into the TCR recombinome, then send them through `DigitalExpression`, the counting utility from the Drop-seq pipeline. Since it was only the TCR-aligned reads, I set `MIN_NUM_GENES_PER_CELL=1`. I also set `READ_MQ=1` since the alignments were, not surprisingly, of low quality.

Try it yourself

You can download my simple TCR recombinomes here (human, mouse). To keep the file sizes down, those links include only the TCR recombinome, not the rest of the genome. You'll need to concatenate them onto the regular human or mouse reference genomes before using them. These can be found from the McCarroll lab (human, mouse). The code would look something like this.

- Concatenate onto the regular mm10 reference.

```
mm10=<path_to_mm10>
cat simple_recombinome.gtf  ${mm10}/mm10.gtf > mm10_plus_TCR.gtf
cat simple_recombinome.refFlat ${mm10}/mm10.refFlat > mm10_plus_TCR.refFlat
cat simple_recombinome.fa ${mm10}/mm10.fasta > mm10_plus_TCR.fa
```

- Make sure they look right.

```
less mm10_plus_TCR.gtf
less mm10_plus_TCR.refFlat
less mm10_plus_TCR.fa
```

- Feed them into STAR to build a genome index.

```
star --runMode genomeGenerate
     --sjdbOverhang 49 \
     --genomeDir        mm10_plus_TCR \
     --genomeFastaFiles mm10_plus_TCR.fa \
     --sjdbGTFfile      mm10_plus_TCR.gtf
```

- When using the Drop-seq tools, there's a step that merges two BAM files to combine alignment info with cell and molecular barcodes. For this, you'll need to build a Picard dictionary for the new reference FASTA.

```
java -Xmx6g -jar <path_to_picard1>/CreateSequenceDictionary.jar \
 REFERENCE=mm10_plus_TCR.fa \
 O=mm10_plus_TCR.dict
```

- Run the Drop-seq tools as you would normally, but using this reference. I have a wrapper that does this, but I am not ready to make it public. For now, check out the Drop-seq alignment cookbook.

- Recount the TCR reads with permissive settings. You'll need a BAM file with tags for cell barcode, molecular barcode, and gene name.

```
samtools index merged_exon_tagged.bam
samtools view -bh merged_exon_tagged.bam  chrTCR_recombinome >  tcr.bam
path/to/Drop-seq_tools/DigitalExpression \
  SUMMARY=TCR.dge.summary.txt  \
  I=tcr.bam  \
  O=TCR.dge.txt.gz  \
  MIN_NUM_GENES_PER_CELL=1 \
  READ_MQ=1
```

- If you want to save yourself some hassle down the line, specify a cell barcode whitelist in the above step (using `CELL_BC_FILE=<your_whitelist.tsv>`) instead of doing all the cells like I do in my example.

Room for improvement

This problem is complicated enough that one could write a dissertation on it. I only spent the better part of a week. So, this could no doubt be optimized much further.

If I were to stick with the same quick and dirty strategy and refine it a bit, I would focus on getting the alignment and quantification to handle ambiguity. Right now, the handling of uncertainty is far less than ideal. For instance, the default behavior of the Drop-seq tools' `TagReadWithGeneExon` is to ignore anything aligning over more than one exon. For a very simple tweak, I could run it again and pass in `ALLOW_MULTI_GENE_READS=TRUE` , or I could name the "exons" in the refFlat file by locus instead of by segment. I could also forego STAR + Dropseq tools in favor of something that fits the situation more naturally -- maybe Kallisto. If the uncertainty is somehow represented in the count matrix, it might be possible to resolve some puzzles after the fact in a simple way. For instance, if a cell has ambiguous J-segment reads that could be from TCRA or TCRG, but it clusters with gamma-delta T cells and has reads that unambiguously originate from TCRD, then it's unlikely that the ambiguous J-segment reads are from TCRA.

There's a place for quick and dirty solutions, but for someone interested in TCR repertoires, you could probably do better by incorporating known TCR biology into a customized model, as TRAPeS and TRACER do. I am not sure if those particular tools are the right choice because their locus reassembly is rather ambitious and is meant for higher-resolution data. But just to explain what I mean, TCR's should contain one V segment, one J segment, one C segment, and up to 3 D segments. It would be great to have an aligner that would say, "The V segment is `TRAV9N` , there are 2 D segments but I don't know exactly which, and the J section is `TRAJ26` or `TRAJ55` ." Partial information like this could be really helpful in defining lymphocyte subpopulations, which are frequently biased towards certain *segments*.

Acknowledgments

Thanks to ??? for reading draft of this post. (All errors are my own.)

Miscellaneous thoughts and self-congratulatory remarks

- This whole episode played out over a year ago. I am pretty pleased with my past self for leaving notes in enough detail to write it all up without a hassle.
- This topic is cool, but I have no idea who will read it. If you're reading, I'd love to hear from you!