Introduction to Bulk RNAseq data analysis

Quantification of Gene Expression with Salmon

Quasi-mapping and gene expression quantification

So far we have QC'd our raw reads, aligned against the genome with HISAT2, and done some additional QC using the alignment results. The next step is to quantify the gene expression. One way would be to look at the alignments and use information about where genes are located on the genome to count the number of reads coming from each gene. This method is shown in the extended materials using the `featureCounts` tool.

An alternative is to use faster methods known as *quasi-mapping* or *pseudo-alignment*, followed by inferential estimation of gene expression. This approach has been developed over the past few years and has the advantages of being faster and more lightweight than full alignment (they can be run on a laptop rather than needing a high performance cluster). In addition these methods give more accurate estimates of gene expression at the transcript level and incorporate corrections for GC-content bias and sequence bias. They are also able to quantify multimapping reads in a more pragmatic manner than the standard alignment/counting approaches giving more accurate counts, particularly for genes belonging to families of genes with very similar sequences.

There are various tools that use this approach such at `Kallisto`, `Sailfish` and `Salmon`; we will use `Salmon` (Patro 2017).

You can find the full manual here:

https://salmon.readthedocs.io/en/latest/salmon.html#using-salmon

This provides much more information about various options and the reasons for using them. For most purposes we can simply use the default settings with just a few extras.

1. Indexing the transcriptome for Salmon

Salmon encompasses both alignment and quantification in a single tool. As with HISAT2, we will need a reference to map the reads to, however, in this case we will be aligning to the transcriptome rather than the genome. The transcriptome reference is a fasta file containing sequences for all transcripts. Before we can run `Salmon` we need to create an index, just as we did for `HISAT2`. For common species is possible to download pre-generated Salmon indexes, however, it is worth knowing how to create your own. This process is more computationally intensive and will not run on standard laptop or desktop.

The details of the index creation are taken from:

https://combine-lab.github.io/alevin-tutorial/2019/selective-alignment/

As well as including the transcriptome, we also want to include the genomic sequences. These will act as a *decoy* so that non-transcriptomic reads will not be erroneously counted against transcripts.

The indexing takes too long with the full transcriptome for the purposes of this training session, so we will work with the transcriptome for just genes on Chromosome 14.

The full transcriptome has been downloaded from Ensembl:

ftp://ftp.ensembl.org/pub/release-102/fasta/mus_musculus/cdna/Mus_musculus.GRCm38.cdna.all.fa.gz

Indexing the transcriptome

We are going to be giving Salmon both genomic and transcriptomic sequences to index. First we need to create a single fasta file that contains both sets of sequences. It is important to have the transcriptome sequences first in the file.

The genomic sequences are the *decoys* and we need a file that lists these so that Salmon will know which are which. In out case we are just using chr14, so we just need to create a file containing this.

Finally, we will need to provide salmon with the name of a directory in which to create the index. We don't need to make the directory, salmon will do this itself.

Exercise 1 - Create Salmon index

1. Create concatenated trancriptome/genome reference file

```
cat references/Mus_musculus.GRCm38.cdna.chr14.fa.gz \
   references/Mus_musculus.GRCm38.dna_sm.chr14.fa.gz \
   > references/gentrome.chr14.fa.gz
```

2. Create decoy sequence list from the genomic fasta

```
echo "14" > references/decoys.txt
```

3. Use `salmon index` to create the index. You will need to provide three pieces of information:
   - the **Transcript fasta file** - *references/gentrome.chr14.fa.gz*
   - the **decoys** - *references/decoys.txt*
   - the **salmon index** - a directory to write the index to, use **references/salmon_index_chr14**

Also add `-p 7` to the command to instruct salmon to use 7 threads/cores. To find the flags for the other three pieces of information use:

```
salmon index --help
```

One thing to note here is that we have not specified the `-k` parameter. This parameter sets the kmer size that the index will be based on and relates to the minimum size of kmer used for quasi-mapping. The default is 31 bases, and this is fine for read sizes over 75 bases. If your read size is less that 75, you would need to adjust this. You should set the kmer size to slightly less than half of the read length. Quasi-mapping looks for kmers that are perfect match to the reference sequence. If the kmer size is more than half the read length, then a read with a mismatch in the middle of the read will never be able to be mapped to the transcriptome, even if all other bases are a perfect match for the sequence at the location that it originated.

2. Gene expression quantification

Now that we have an index we can quickly get gene expression estimates directly from our raw fastq files.

We can use the full index here as this step is quick. The full index should already be in the references directory: `references/salmon_index`. For this exercise we'll just quantify one sample: **SRR7657883**. (We've already run salmon on the complete data set. You can see the results in the `salmon` directory. We'll use this for the differential gene expression analysis in later sessions.)

Exercise 2 - Quantify with Salmon

1. Make directory called `salmon_output`

2. Use `salmon quant` to quantify the gene expression from the raw fastq. To see all the options run `salmon quant --help-reads`. There are lot of possible parameters, we will need to provide the following:
   - **salmon index** - *references/salmon_index*
   - `-l A` - Salmon needs to use some information about the library preparation, we could explicitly give this, but it is easy enough for Salmon to **A**utomatically infer this from the data.
   - **File containing the #1 mates** - *fastq/SRR7657883.sra_1.fastq.gz*
   - **File containing the #2 mates** - *fastq/SRR7657883.sra_2.fastq.gz*
   - **Output quantification directory** - *salmon_output/SRR7657883*
   - `--gcBias` - salmon can optionally correct for GC content bias, it is recommended to always use this
   - **The number of threads to use** - *7*

Salmon creates a separate output directory for each sample analysed. This directory contains a number of files; the file that contains the quantification data is called `quant.sf`.

3. A quick summary of the results

We have generated salmon quantification results for all samples in the directory `salmon`. You can run `multiqc` on these results to get some basic statistics regarding the percentage of reads that have been counted against the transcriptome.

Exercise 3 - Run multiqc

1. Run `multiqc` on the `salmon_output` directory and create a report called `Salmon_quantification_report` in the `salmon_output` directory. Look back to the previous session if you need a reminder on how to do this.
2. Open the report and determine what percentage of the reads have been *aligned* to the transcriptome. Compare this to the QC report from the previous session.

Make transcript to gene table

Salmon quantifies gene expression at the transcript level. When we come to do our differential gene expression analysis in R, we will want to summarise this to the gene level. To do this we need a table that links transcript IDs to gene IDs. **We have already created this for you**, but, for reference, the code below was used to generate this table from the sequence headers in the transcriptome reference file.

You do not need to run this code

```
echo -e "TxID\tGeneID" > salmon_outputs/tx2gene.tsv
zcat references/Mus_musculus.GRCm38.cdna.all.fa.gz |
  grep "^>" |
  cut -f 1,4 |
  sed -e 's/^>//' -e 's/gene://' -e 's/\.[0-9]*$//' |
  tr ' ' '\t' \
  >> salmon_outputs/tx2gene.tsv
```

1. `zcat references/Mus_musculus.GRCm38.cdna.all.fa.gz` - read the zipped fasta

2. `grep "^>"` - find the sequence headers, they all start with '>'

3. `cut -f 1,4 -d ' '` - extract the 1st and 4th entries - transcript ID and gene ID

4. `sed -e 's/^>//' -e 's/gene://' -e 's/\.[0-9]*$//'` - remove the ">" from the beginning of the line, the "gene:" from the beginning of the gene ID, and the trailing ".x" number which indicates that version of the gene annotation
5. `tr ' ' '\t'` - replace spaces with tabs so that the table is tab delimited

---

References

Patro, Duggal, R. 2017. "Salmon Provides Fast and Bias-Aware Quantification of Transcript Expression." *Nature Methods* 14: 417–19. https://doi.org/10.1038/nmeth.4197.