Day 3

First sorted out all the github problems

I created a repository called obdsRNAseqpipeline on GitHub

Created an empty folder and then clones using this:

git clone git@github.com:rodgson543/obdsRNAseqpipeline.git

#This is my private repository

Next

We are going to go through pseudo alignment today and then do a pipeline

Alignment free method – instead of mapping/QC and counting phase, we do this all in one step: pseudo alignment

Difference is that we specify which transcript the read is from

Good for counting, v fast. matrix is output (can work on this directly)

Transcript per million (so normalised to num of transcript)

More accurate than traditional method

Limitation – cant use it to detect novel transcrpts or sequence variations, ie splicing

Good for DGE

Doesn’t produce BAM file – but can force on CL arguments

Sailfish, salmon, kallisto – Tools, widely used in RNAseq

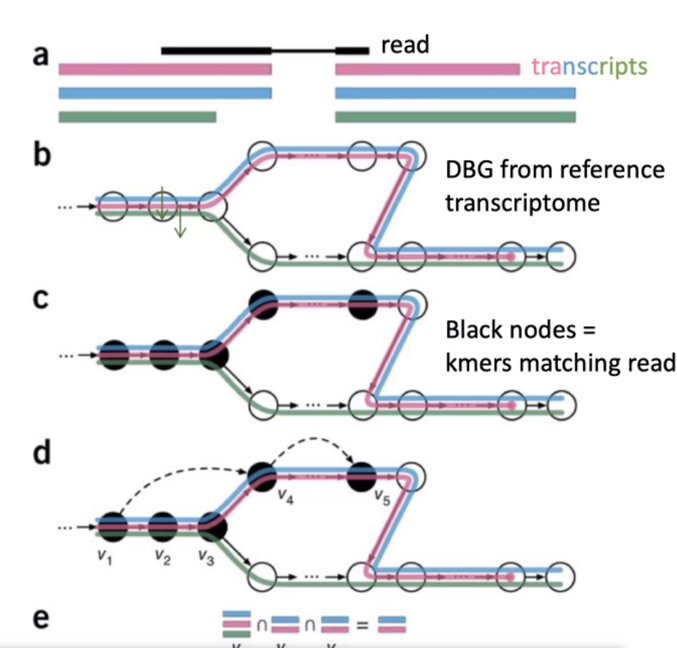
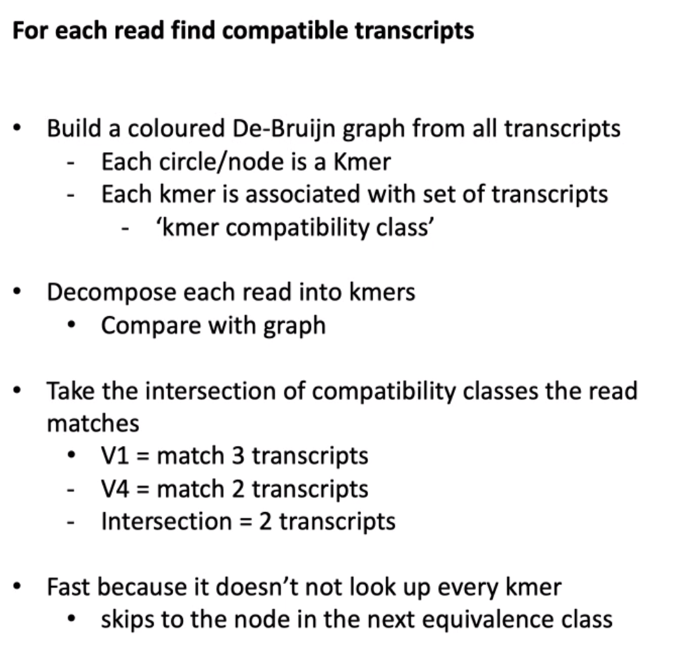
De-Bruijn lab – transcripts of genome, breaks the transcripts into kmer

Sequences of length k (kmer) default is 31

Each kmer associated with set of transcripts

See pic below

Each path through the graph is a single transcript



Kmers (black circle) – are they equivalent across sequence or exon etc – ie. Green transcript is missing some sequence (ie an exon), then join up at end. Pink transcript terminates earlier (missing an exon)

For each read – in c – black nodes are kmers

Each node is a compatibility class, take intersection – which transcript the read comes from (ie only blue or pink, not green)

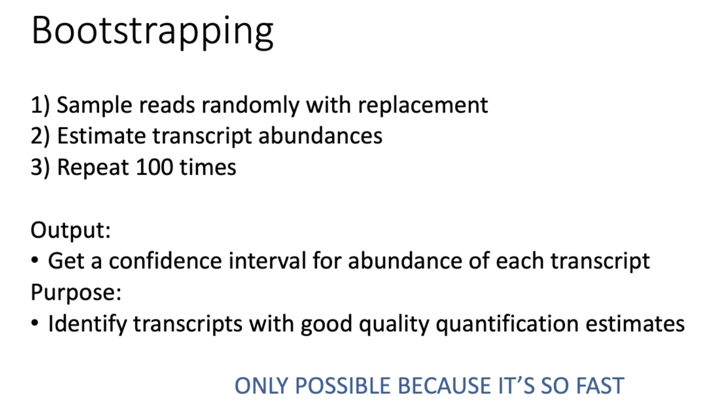
Ie. When you index a genome for mapping, before you start mapping reads for pseudoalignment, build a ref genome

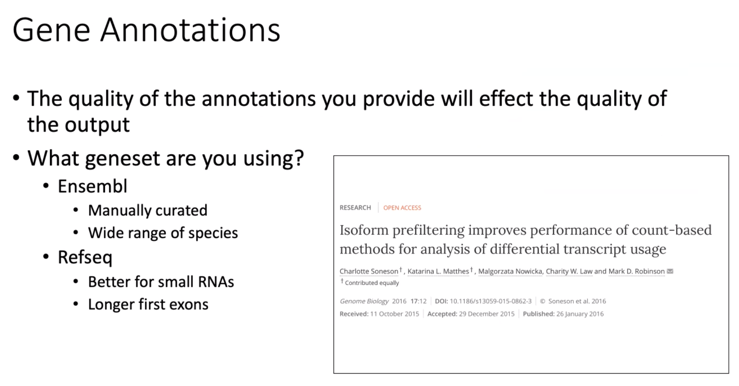
Logical splitting of the sequence into sub sequences of certain length – allows you to only really look at those kmers where they are different.

The reference will be either a ref genome or can build your own

Bootstrapping – randomly sample a subset of reads

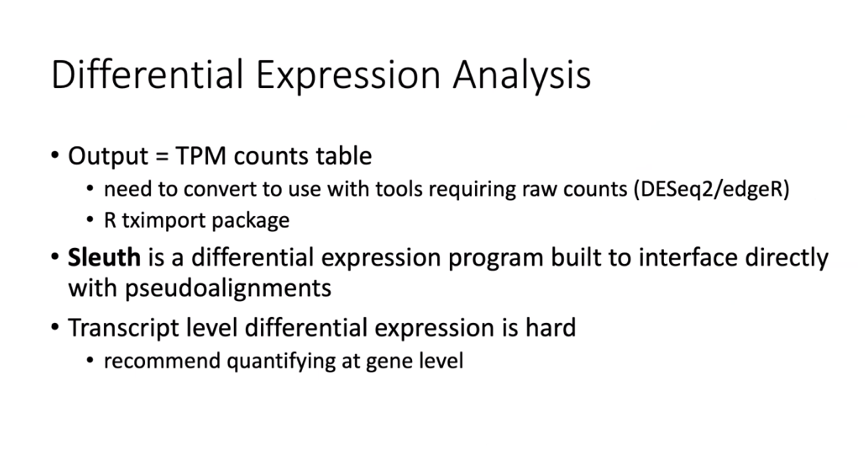
Perform 100 times, estimate transcript abundance each time (gives a confidence interval- ie erro) advantage over mapping method.



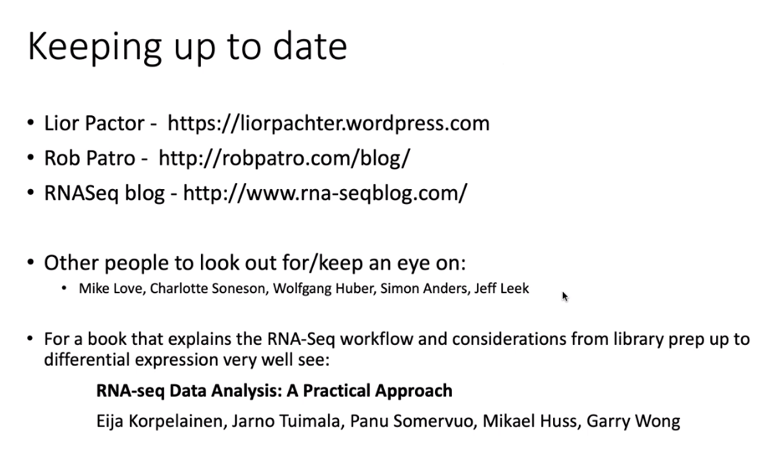


DGE analysis – from pseudoalignment (transcripts per million) have to convert TPM back to counts. Or can use sleuth (directly from kallisto – takes in tpm and confidence intervals)

Best to work at gene level



Leading the field:



Today we’re going to implement this pipeline:

How to start: look at slide 2:

<https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html>

