Pseudo-aligners

RNA-seq data analysis

Johan Reimegård | 30-November-2020

Pseudoaligners only assigns read to a transcript

- Not the actual location…
  - It does it by matching k-mers between read and transcripts
  - And using statistics assign the read to a transcript
- Not to genes on a genome but to transcripts

Kmers are nucleotides of length K

- Oct4 is 1574 nt long (L = 1574)
- Kmer is 7 (K= 7)
- Oct4 will contain 1568 Kmers (L-K+1)

```
......CTTGGAACAAT............
       CTTGGAA
        TTGGAAC
         TGGAACA
          GGAACAA
           GAACAAT
```

Creates a table with all the Kmers in all transcripts

Kmer/Transcript	Oct4	Oct3	Oct2	Sox2	Sox3
CTTGGAA	TRUE	FALSE	TRUE	FALSE	FALSE
TTGGAAC	TRUE	FALSE	TRUE	FALSE	FALSE
TGGAACA	TRUE	FALSE	FALSE	FALSE	FALSE
GGAACAA	TRUE	TRUE	FALSE	FALSE	FALSE
GAACAAT	TRUE	TRUE	FALSE	FALSE	FALSE
AACAATA	TRUE	FALSE	FALSE	FALSE	FALSE

Splits up a read into the same Kmer size

`Read1 = CTTGGAACAAT`

Kmer Read1
CTTGGAA
TTGGAAC
TGGAACA
GGAACAA
GAACAAT
AACAATA

Checks in which transcripts the Kmers exist and sums them up

Kmer Read1
CTTGGAA
TTGGAAC
TGGAACA
GGAACAA
GAACAAT
AACAATA

Kmer	Oct4	Oct3	Oct2	Sox2	Sox3
CTTGGAA	TRUE	FALSE	TRUE	FALSE	FALSE
TTGGAAC	TRUE	FALSE	TRUE	FALSE	FALSE
TGGAACA	TRUE	FALSE	FALSE	FALSE	FALSE
GGAACAA	TRUE	TRUE	FALSE	FALSE	FALSE
GAACAAT	TRUE	TRUE	FALSE	FALSE	FALSE
AACAATA	TRUE	FALSE	FALSE	FALSE	FALSE

Read	Oct4	Oct3	Oct2	Sox2	Sox3
Read 1	6	2	2	0	0

Assign the read to one or many transcript

Checks which of the transcripts the number of kmers matched is least likely to happen by chance and assign it to those transcript

Read	Oct4	Oct3	Oct2	Sox2	Sox3
Read 1	6	2	2	0	0

Assign read to transcripts

Read	Oct4
Read 1	1

Add read counts to transcripts in sample

Read	Oct4	Oct3	Oct2	Sox2	Sox3
Sample 1	+1	0	0	0	0

Redo the procedure for all reads

Read2 = GATACAGATAC
6 kmers of length 7

Read	Oct4	Oct3	Oct2	Sox2	Sox3
Read 2	0	0	0	6	6

Assign read to transcripts

Read	Sox2	Sox3
Read 2	1	1

Add read counts to transcripts in sample

Read	Oct4	Oct3	Oct2	Sox2	Sox3
Sample 1	1	0	0	+1	+1

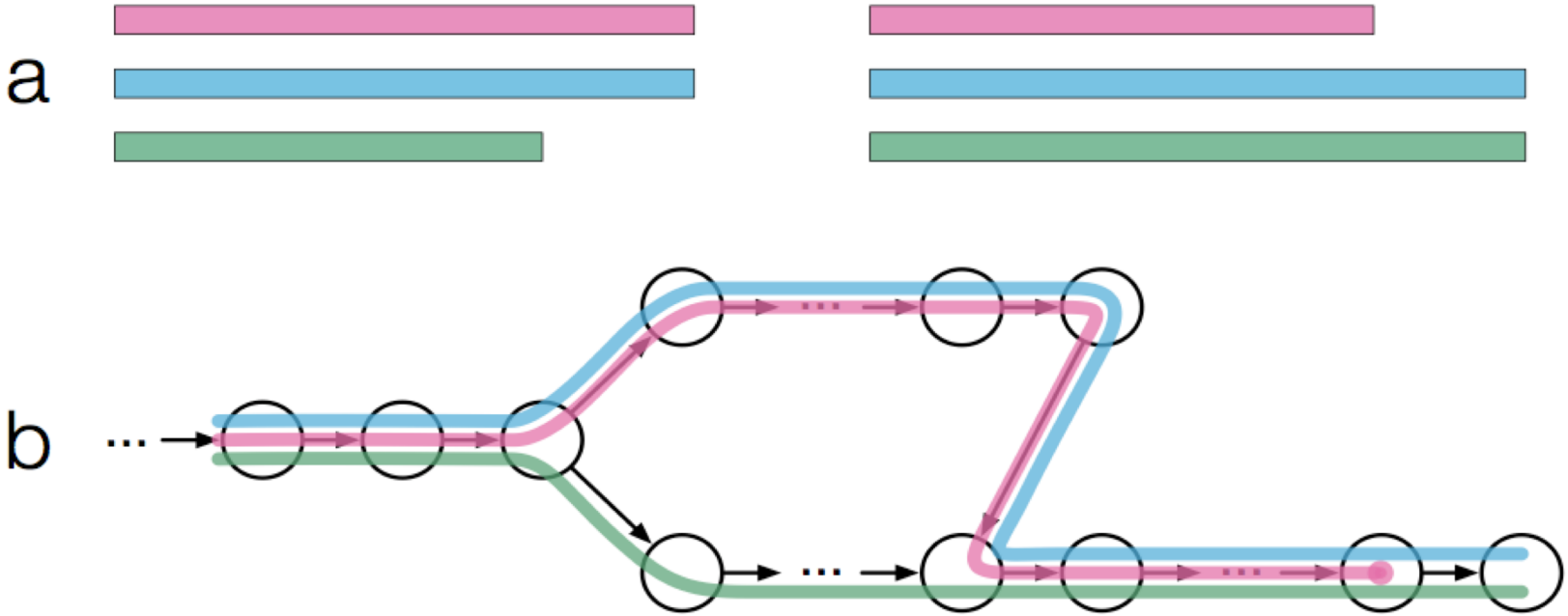
But it takes time to look up so many k-mers

Real result from Kallisto

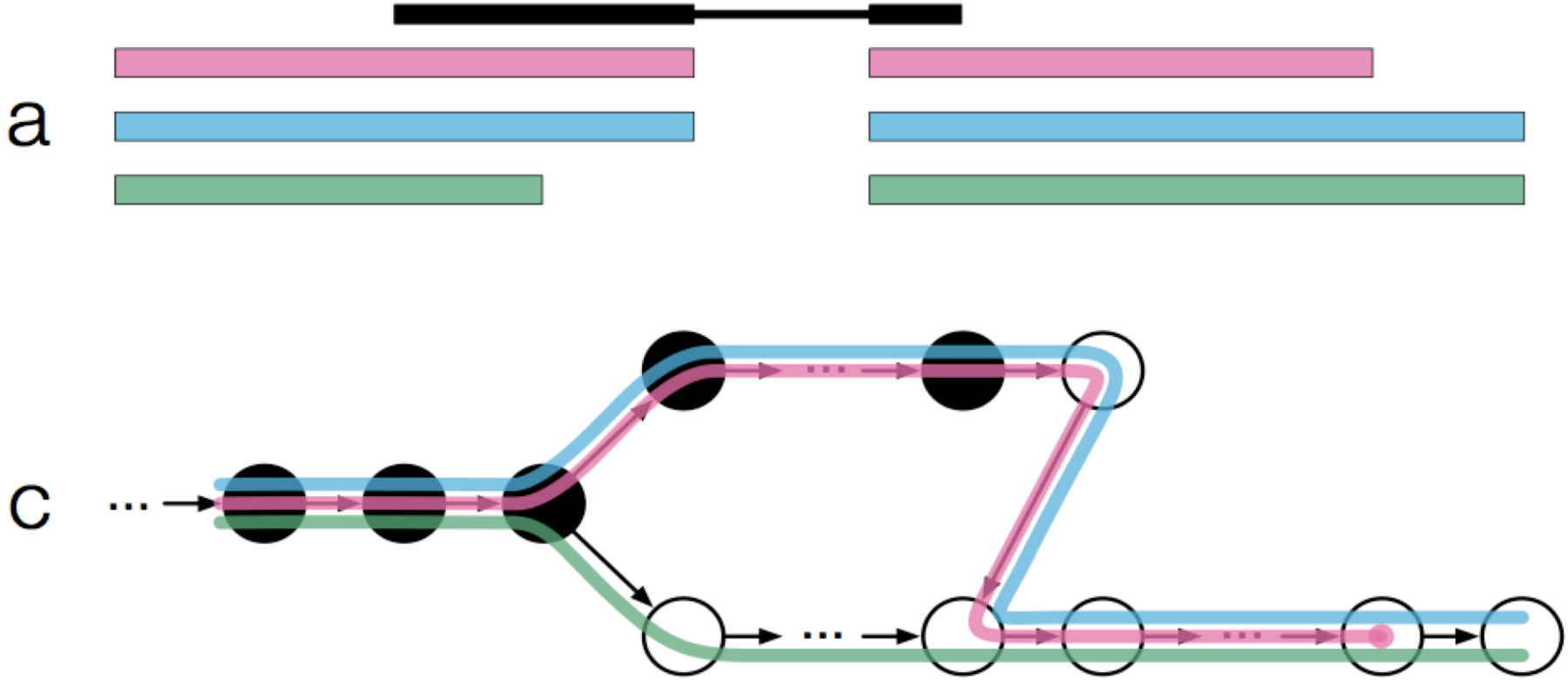
```
[quant] fragment length distribution will be estimated from the
[index] k-mer length: 31
[index] number of targets: 173,259
[index] number of k-mers: 104,344,666
```

Build de-bruin graph from kmers

In this example three isoforms are the only ones that contains a set of kmers

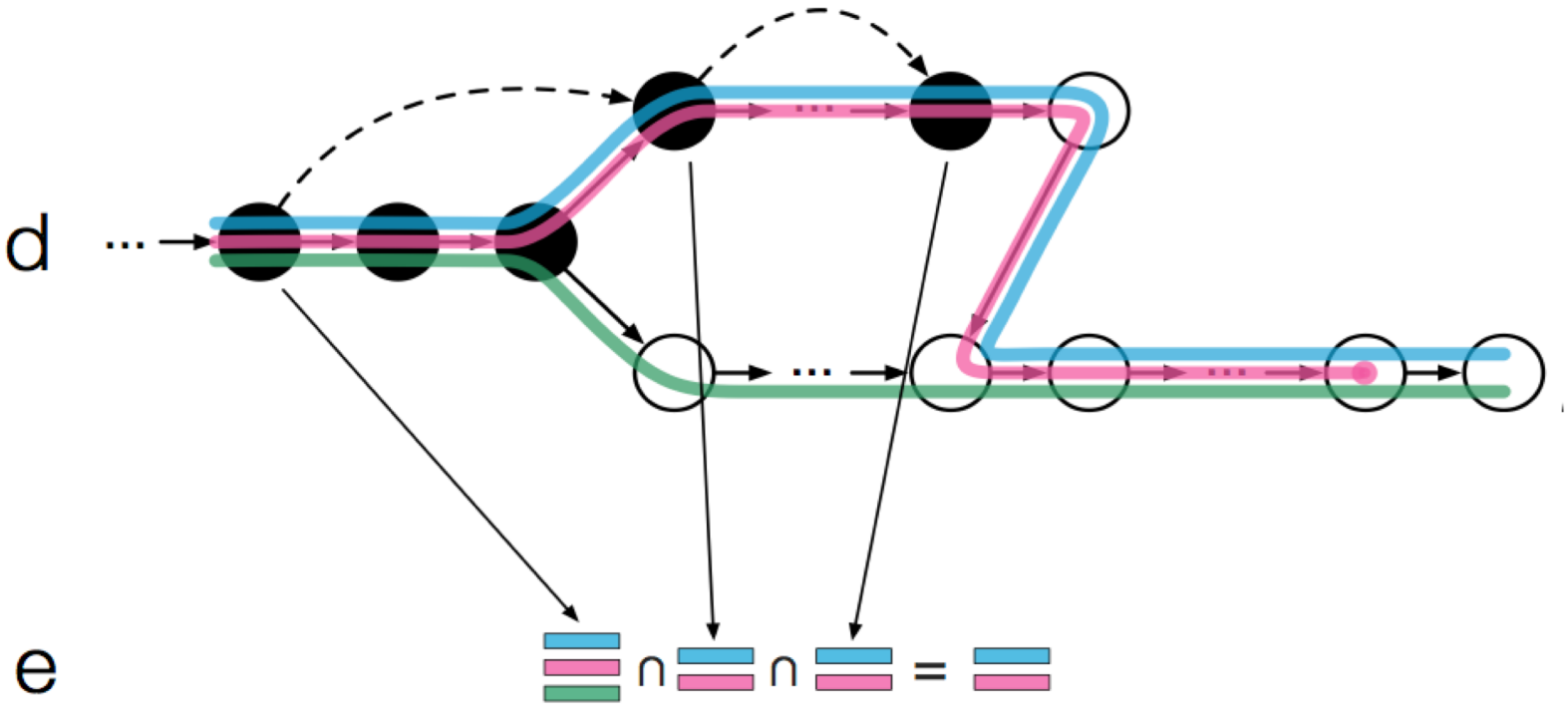


Read contains five kmers



But kmer 2,3, are redundant and can be ignored

So instead of looking up five kmers Kallisto only has to look up 3

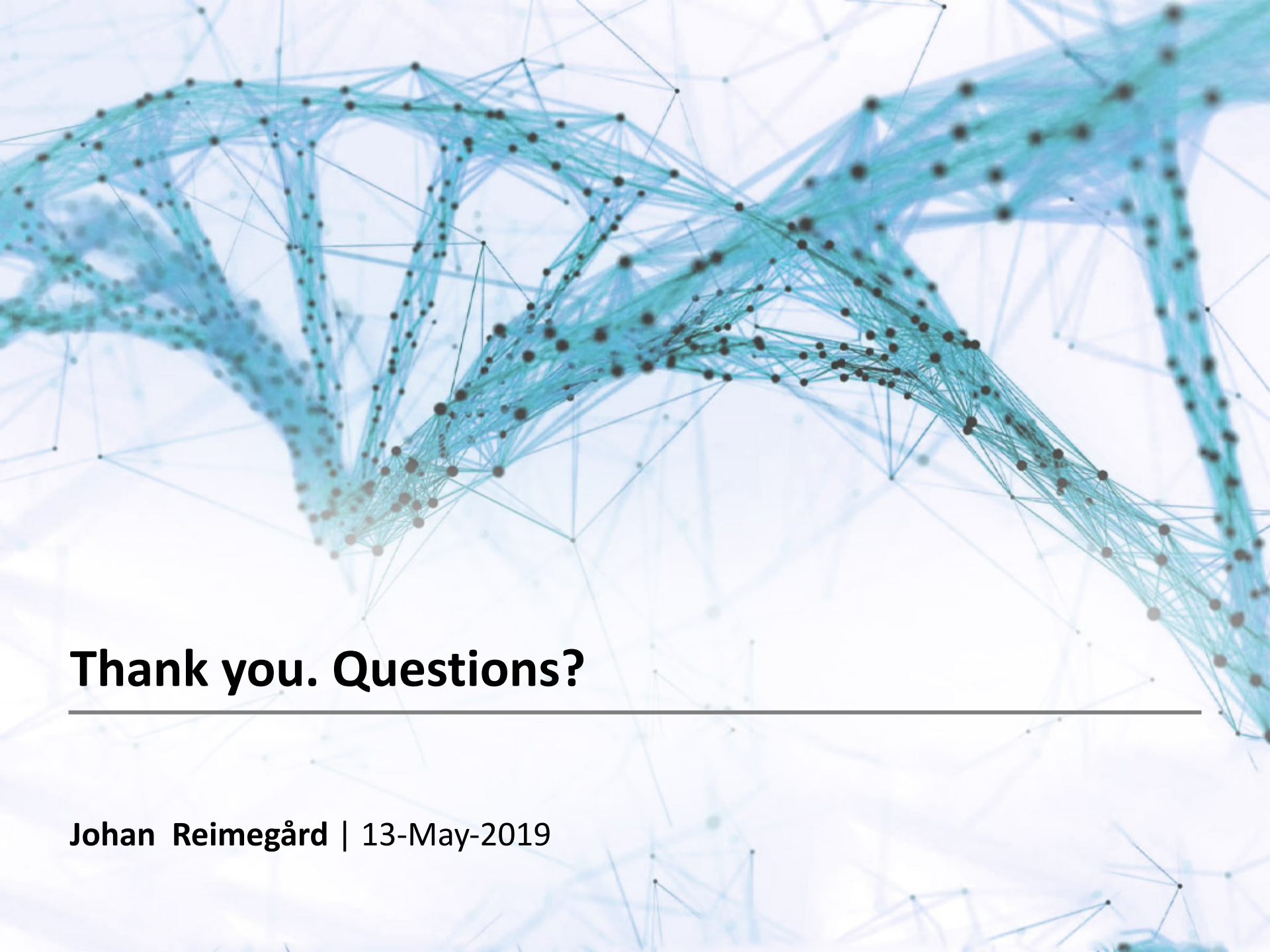


For the majority of reads, kallisto ends up performing a hash lookup for only two k-mers

So they divide it up to classes

Real result from Kallisto

```
[quant] fragment length distribution will be estimated from the
[index] k-mer length: 31
[index] number of targets: 173,259
[index] number of k-mers: 104,344,666
index] number of equivalence classes: 695,212
[quant] running in paired-end mode
[quant] will process pair 1: fastq/test.1.fastq.gz fastq/test.2.fastq.gz
[quant] finding pseudoalignments for the reads ... done
[quant] learning parameters for sequence specific bias
[quant] processed 92,206,249 reads, 82,446,339 reads pseudoaligned
[quant] estimated average fragment length: 187.018
[ em] quantifying the abundances ... done
[ em] the Expectation-Maximization algorithm ran for 1,521 rounds
[bstrp] number of EM bootstraps complete: 100
```



Thank you. Questions?

Johan Reimegård | 13-May-2019