Mapping strategies for sequence reads (with focus on RNA-seq)

Ernest Turro

University of Cambridge

14 Sep 2016

Quantification

An important aim in genomics is working out the **contents** of a biological sample.

1. What distinct elements are in the sample?
2. How many copies of each element are in the sample?

Quantification

An important aim in genomics is working out the **contents** of a biological sample.

1. What distinct elements are in the sample?
2. How many copies of each element are in the sample?

RNA-seq:

1. What is the sequence of each distinct RNA molecule?
2. What is the concentration of each RNA molecule?

Quantification

An important aim in genomics is working out the **contents** of a biological sample.

1. What distinct elements are in the sample?
2. How many copies of each element are in the sample?

RNA-seq:

1. What is the sequence of each distinct RNA molecule?
2. What is the concentration of each RNA molecule?

ChIP-seq:

1. What is the sequence/location of each binding site?
2. How frequently is each site bound in a population of cells?

Motivation

In an ideal world...

- we would sequence each molecule of interest from start to finish without breaks
- there would be no errors in the sequences

Motivation

In an ideal world...

- we would sequence each molecule of interest from start to finish without breaks
- there would be no errors in the sequences

... and there would be an excess supply of biostatisticians

Motivation

In an ideal world...

- we would sequence each molecule of interest from start to finish without breaks
- there would be no errors in the sequences

... and there would be an excess supply of biostatisticians

In the real world...

- molecules of interest need to be selected
- DNA/RNA needs to be shattered into fragments
- fragments need to be amplified
- # reads from a fragment is hard to control (0, 1 or more times)
- different parts of a class of molecules may be sequenced different numbers of times (leads to variation in **coverage**)
- there are sequencing errors

Imperfect data

The data consist of

- 1 or 2 read sequences from each fragment
- base call qualities for each base in each read
- meta-data (e.g. read → cDNA library)

Imperfect data

The data consist of

- 1 or 2 read sequences from each fragment
- base call qualities for each base in each read
- meta-data (e.g. read → cDNA library)

On their own, unprocessed, these data are not very useful!

Imperfect data

The data consist of

- 1 or 2 read sequences from each fragment
- base call qualities for each base in each read
- meta-data (e.g. read → cDNA library)

On their own, unprocessed, these data are not very useful!

We have accumulated (prior) biological knowledge, including

- reference genome sequences
- genome annotations (gene structures, binding motifs, etc)

We must label (or **map**) reads to relate them to existing knowledge

Imperfect data

The data consist of

- 1 or 2 read sequences from each fragment
- base call qualities for each base in each read
- meta-data (e.g. read → cDNA library)

On their own, unprocessed, these data are not very useful!

We have accumulated (prior) biological knowledge, including

- reference genome sequences
- genome annotations (gene structures, binding motifs, etc)

We must label (or **map**) reads to relate them to existing knowledge

- We wish to measure quantities pertaining to features (transcripts, binding sites)
- Hence we **map reads** → **features**

Mapping by alignment

A common technique for mapping is *alignment*:

```
                             Read: AGTCGACTGATGAG
Reference: ...GCAGCAGCGATCGAGTCAGTCAGTCGACTGACGAGCGCGCATACGACT...
```

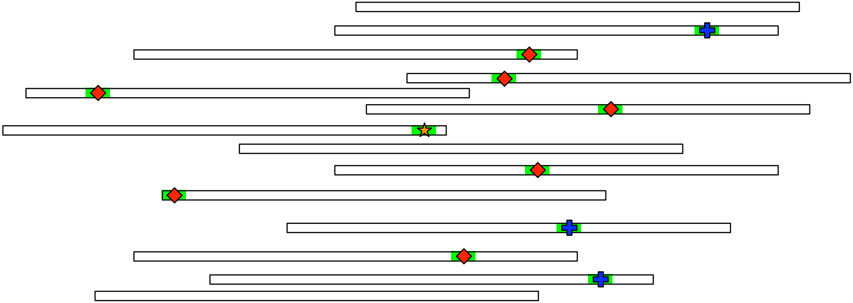
Not always easy:

- Reads are ∼100 bp long
- Genome is ∼3,000,000,000 bp long and rather repetitive
- Reference genome ≠ sample genome (SNPs, indels, structural variants)
- Reads prone to errors (if lucky 1/1000 base calls are wrong)

Mapping ChIP-seq reads

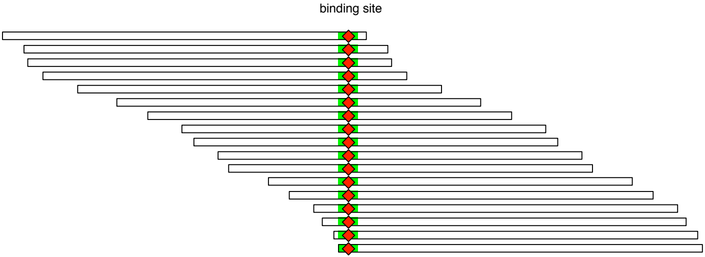
ChIP-seq protocol

Crosslink and shear.



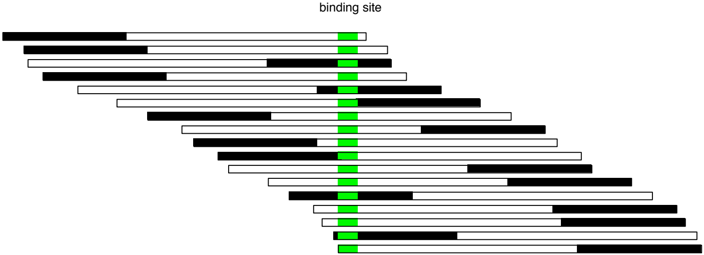
ChIP-seq read mapping

Add protein-specific (◆) antibody and immunoprecipitate.



ChIP-seq read mapping

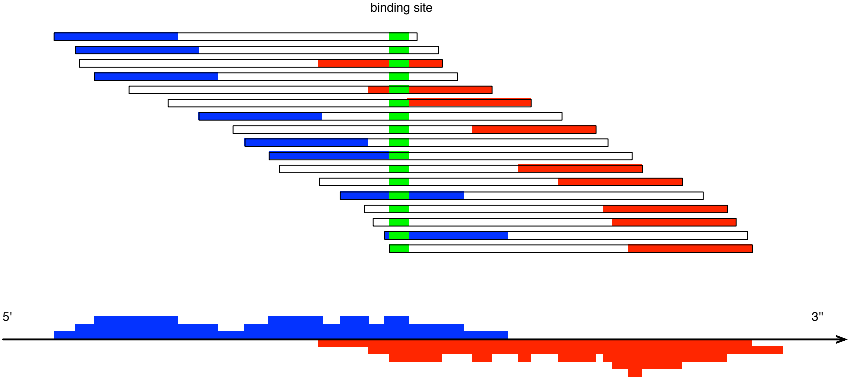
Sequence one end of each fragment.



ChIP-seq read mapping

Genome alignment: read → binding site (or thereabouts)

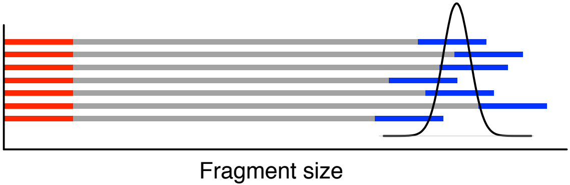
- 🟦 aligns directly
- 🟥 reverse complement aligns



Mapping RNA-seq reads

RNA-seq typical protocol

- Select RNAs of interest (e.g. mRNAs (polyadenylated))
- Fragment and reverse-transcribe to ds-cDNA
- Size-select, denature to ss-cDNA
- Sequence *n* bases from one/both ends of fragments (typically $n \in (50, 100)$ for Illumina)



read 1

ATCACTCTACTACGCGC
TACTATCGACTACTCTAC
TACTATCGACTACTCTAC
...

read 2

ATCTACTATCACTATCAC
TTAACTCCTATGTATCTC
ACCCGATACTCGACTCT
...

Gene expression

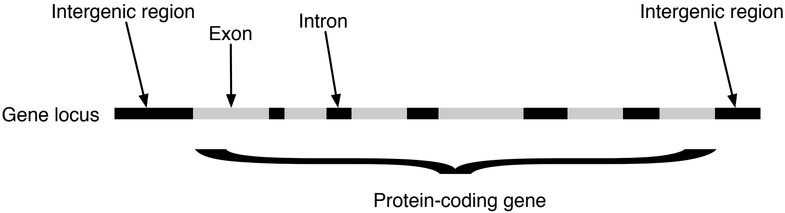
Different kinds of RNAs (tRNAs, rRNAs, mRNAs, other ncRNAs...).

Messenger RNAs of particular interest as they code for proteins.

Gene expression

Different kinds of RNAs (tRNAs, rRNAs, mRNAs, other ncRNAs...).

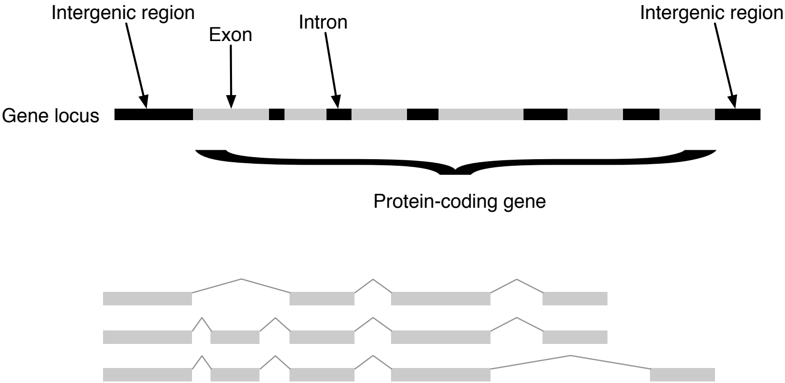
Messenger RNAs of particular interest as they code for proteins.



Gene expression

Different kinds of RNAs (tRNAs, rRNAs, mRNAs, other ncRNAs...).

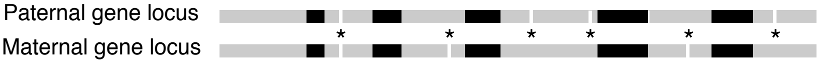
Messenger RNAs of particular interest as they code for proteins.



Gene expression

Different kinds of RNAs (tRNAs, rRNAs, mRNAs, other ncRNAs...).

Messenger RNAs of particular interest as they code for proteins.



Paternal gene locus

Maternal gene locus

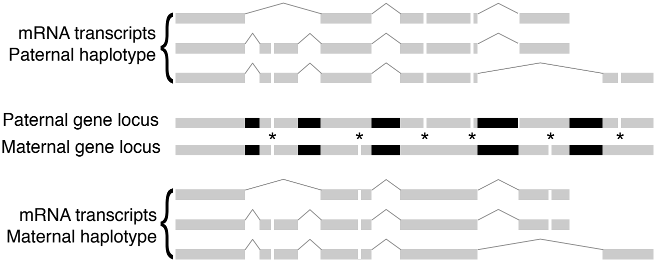
Gene expression

Different kinds of RNAs (tRNAs, rRNAs, mRNAs, other ncRNAs...).

Messenger RNAs of particular interest as they code for proteins.

No one-to-one gene→mRNA mapping:

1. Alternative isoforms have distinct sequences
2. Two versions of each isoform sequence in diploid organisms



RNA-seq mapping strategies

Where did the reads come from?

RNA-seq mapping strategies

Where did the reads come from?

We need to map reads → transcripts.

Three strategies:

1. *De novo* assembly
    - Genome unknown or of poor quality
2. Genome alignment + gene model assembly
    - Genome available
    - Gene models ("transcriptome") unknown or of poor quality
3. Transcriptome alignment
    - Genome available
    - Comprehensive gene models ("transcriptome") available

De novo assembly

- "*De novo* assembly" almost always involves constructing some form of "de Bruijn graph"
- De Bruijn graphs (and variations thereof) help assemble reads into sequences ("contigs") without a reference

De novo assembly

- "*De novo* assembly" almost always involves constructing some form of "de Bruijn graph"
- De Bruijn graphs (and variations thereof) help assemble reads into sequences ("contigs") without a reference

Example:

Say we sequence `ATGGCGTGCA` in three (stranded) reads:

- `ATGGC`
- `   GCGTG`
- `     GTGCA`

De Bruijn graphs

ATGGCGTGCA

ATGGC GCGTG GTGCA

List all distinct *k*-mers (substrings) of the reads:

ATGG TGGC GCGT CGTG GTGC TGCA

De Bruijn graphs

`ATGGCGTGCA`

`ATGGC` `GCGTG` `GTGCA`

List all distinct *k*-mers (substrings) of the reads:

`ATGG TGGC GCGT CGTG GTGC TGCA`

List all distinct $k - 1$-mers from the reads:

`ATG TGG GGC GCG CGT GTG TGC GCA`

De Bruijn graphs

`ATGGCGTGCA`

`ATGGC    GCGTG       GTGCA`

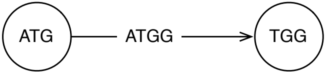
List all distinct k-mers (substrings) of the reads:

`ATGG TGGC GCGT CGTG GTGC TGCA`

List all distinct $k-1$-mers from the reads:

`ATG TGG GGC GCG CGT GTG TGC GCA`

Connect $k-1$-mers $A \rightarrow B$ (nodes) with a k-mer E (edge) if prefix(E) $= A$ and suffix(E) $= B$. E.g.:



De Bruijn graphs

ATGGCGTGCA

ATGGC GCGTG GTGCA

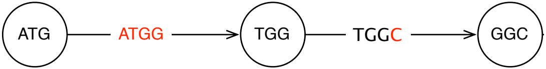
List all distinct k-mers (substrings) of the reads:

ATGG TGGC GCGT CGTG GTGC TGCA

List all distinct $k-1$-mers from the reads:

ATG TGG GGC GCG CGT GTG TGC GCA

Connect $k-1$-mers $A \to B$ (nodes) with a k-mer E (edge) if prefix(E) $=A$ and suffix(E) $=B$. E.g.:



De Bruijn graphs

ATGGCGTGCA

ATGGC GCGTG GTGCA

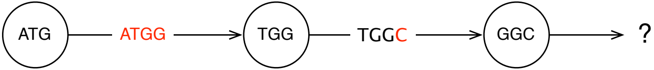
List all distinct k-mers (substrings) of the reads:

ATGG TGGC GCGT CGTG GTGC TGCA

List all distinct $k-1$-mers from the reads:

ATG TGG GGC GCG CGT GTG TGC GCA

Connect $k-1$-mers $A \rightarrow B$ (nodes) with a k-mer E (edge) if prefix$(E) = A$ and suffix$(E) = B$. E.g.:



We're stuck!

De Bruijn graphs

ATGGCGTGCA

ATGGC   GCGTG   GTGCA

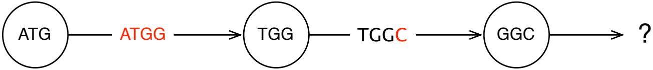
List all distinct k-mers (substrings) of the reads:

ATGG TGGC GCGT CGTG GTGC TGCA

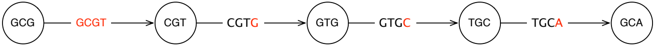
List all distinct $k-1$-mers from the reads:

ATG TGG GGC GCG CGT GTG TGC GCA

Connect $k-1$-mers $A \to B$ (nodes) with a k-mer E (edge) if prefix$(E) = A$ and suffix$(E) = B$. E.g.:

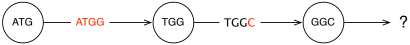


We're stuck! Create two contigs... ATGGC, GCGTGCA



De Bruijn graphs

Why was the transcript broken into two contigs?



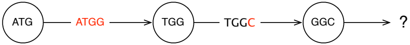
Original sequence: `ATGGCGTGCA`

- `ATGGC`
- `  GCGTG`
- `    GTGCA`

Minimum overlap is only 2, so our choice of *k* (4) is too high.

De Bruijn graphs

Why was the transcript broken into two contigs?



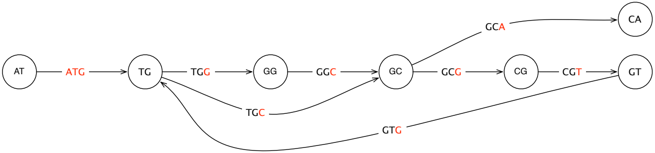
Original sequence: `ATGGCGTGCA`

- `ATGGC`
- `    GCGTG`
- `      GTGCA`

Minimum overlap is only 2, so our choice of *k* (4) is too high.
Try $k = 3$ (more edges, fewer nodes):

Edges: `ATG TGG GGC GCG CGT GTG GTG TGC GCA`

Nodes: `AT TG GG GC CG GT CA`



Choosing *k*

Optimal *k* depends on coverage

Higher expressed genes (higher coverage):

- produce more reads per kb
- more overlap between reads
- optimal *k* is larger (more specific)
- simpler graphs (fewer candidates sequences)

Lowly expressed genes (lower coverage):

- produce fewer reads per kb
- less overlap between reads
- optimal *k* is smaller (more sensitive)
- complex graphs (many candidate sequences)

→ use a range of *k* and merge contigs (cf. genome assembly)

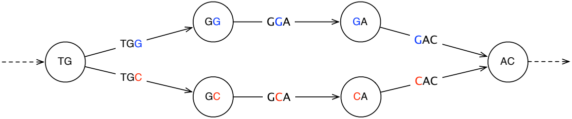
Robertson et al. 2010

Forks due to SNVs, alternative exons

SNPs/errors complicate the graphs (bubbles, which you can pop)

`..TGGAC..`

`..TGCAC..`

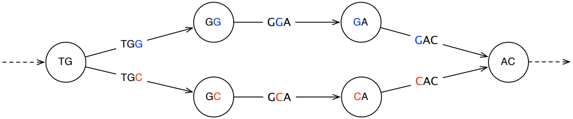


Forks due to SNVs, alternative exons

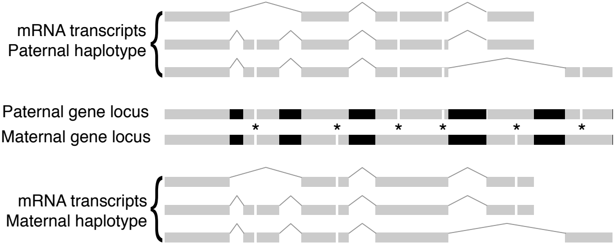
SNPs/errors complicate the graphs (bubbles, which you can pop)

`..TGGAC..`

`..TGCAC..`



Alternative splicing complicate graphs even more.

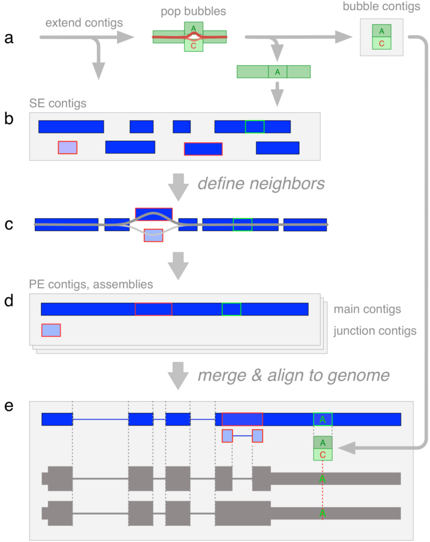


Processing contigs

- Myriad ways in which contigs can be processed
- Usually classifying (e.g. main, junction, bubble), merging and discarding contigs
- Paired-end information can be used to connect contigs
- Alignment to the genome and comparison to annotations

Processing contigs

- Myriad ways in which contigs can be processed
- Usually classifying (e.g. main, junction, bubble), merging and discarding contigs
- Paired-end information can be used to connect contigs
- Alignment to the genome and comparison to annotations

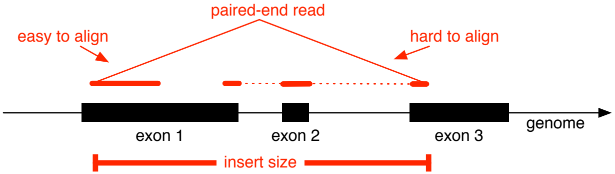


Robertson et al. 2010

RNA-seq alignment strategies

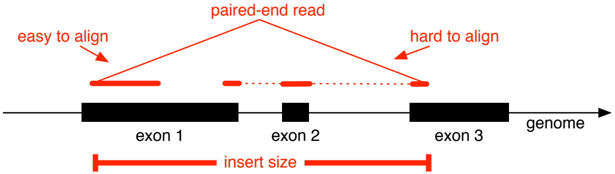
RNA-seq alignment strategies

Genome alignment (e.g. align to 23 chromosomes):

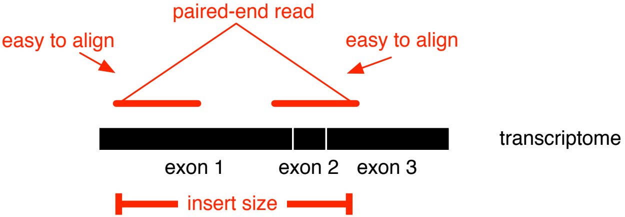


RNA-seq alignment strategies

Genome alignment (e.g. align to 23 chromosomes):



Transcriptome alignment (e.g. align to 150,000 *known* transcripts):



RNA-seq alignment strategies

Genome alignment

Pros:

- Detection of novel genes and isoforms

Cons:

- Spliced alignment is tough
- Requires mapping from genome coordinates to transcripts
- Insert sizes hard to interpret due to introns

RNA-seq alignment strategies

Genome alignment

Pros:

- Detection of novel genes and isoforms

Cons:

- Spliced alignment is tough
- Requires mapping from genome coordinates to transcripts
- Insert sizes hard to interpret due to introns

Transcriptome alignment

Pros:

- No need for spliced alignment
- Simplifies read counting for each isoform
- Simplifies discrimination between mappings using insert sizes

Cons:

- Potential confounding if gene model is wrong
- Novel genes go undetected

RNA-seq alignment strategies

Genome alignment

Pros:

- Detection of novel genes and isoforms

Cons:

- Spliced alignment is tough
- Requires mapping from genome coordinates to transcripts
- Insert sizes hard to interpret due to introns

Transcriptome alignment

Pros:

- No need for spliced alignment
- Simplifies read counting for each isoform
- Simplifies discrimination between mappings using insert sizes

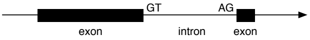
Cons:

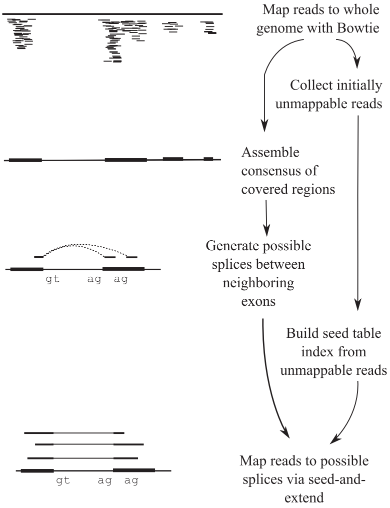
- Potential confounding if gene model is wrong
- Novel genes go undetected

TopHat spliced aligner

1. Align to genome
2. Assemble aligned reads into putative exons
3. Map remaining reads to putative canonical splice junctions

99% of splice junctions involve canonical splice sites:




Map reads to whole genome with Bowtie

Collect initially unmappable reads

Assemble consensus of covered regions

Generate possible splices between neighboring exons

Build seed table index from unmappable reads

Map reads to possible splices via seed-and-extend

Trapnell et al. 2009

Gene models

We now have aligned reads to the genome

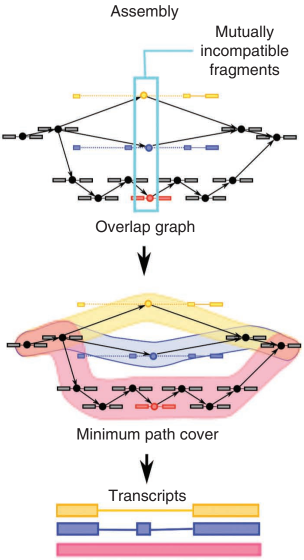
We would like to know which “features” (genes, isoforms, etc) produced the reads.

Two options:

- Use annotations
- Try to infer the gene structures from the data

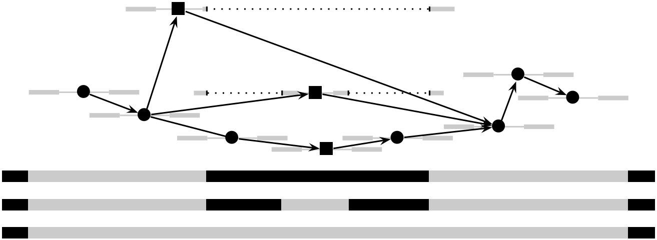
Cufflinks gene model assembler

1. Order spliced alignment pairs by start coordinate
2. Connect compatible read pairs in an overlap graph from left to right
3. Compatibility: same implied splices if they overlap
4. no. of transcripts = max. no. of mutually incompatible fragments = min. no of transcripts required to cover all nodes (max. parsimony)



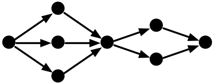
Trapnell et al. 2010

Cufflinks gene model assembler



Cufflinks gene model assembler

There may be several forks and joins in the graph:



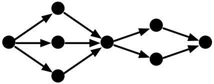
Above, there are 3*x*2 possible exhaustive paths.

Max. parsimony → keep only 3 transcripts

How to ‘phase’ distant exons?

Cufflinks gene model assembler

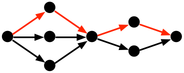
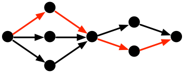
There may be several forks and joins in the graph:



Above, there are 3*x*2 possible exhaustive paths.

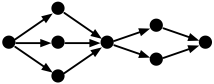
Max. parsimony → keep only 3 transcripts

How to 'phase' distant exons? E.g.

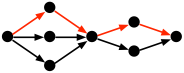
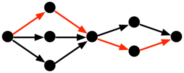
 or  ?

Cufflinks gene model assembler

There may be several forks and joins in the graph:



Above, there are 3*x*2 possible exhaustive paths.
Max. parsimony → keep only 3 transcripts
How to ‘phase’ distant exons? E.g.

 or  ?

Minimise total cost using cost function based on “percent-splice-in” (Wang et al. 2008): $C(y, z) = -\log(1 - |\phi_y - \phi_z|)$.

Trapnell et al. 2010

Cufflinks gene model assembler

Caveats:

- Assembles contiguous overlapping reads so may break up low expressed transcripts into pieces
- Paths maximally extended, so cannot find alternate transcript start or end sites within exons
- Maximum parsimony does not necessarily correspond to biological reality
- Heuristics (simple rules) used to filter out reads and transcripts

Transcriptome pseudoalignment using hash tables

Recent developments in “alignment-free” methods for RNA-seq using a pre-specified transcriptome reference:

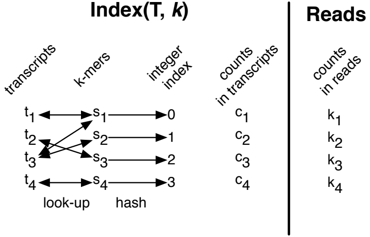
- Sailfish (2014, Nature Biotech.)
- RNA-Skim (2014, Bioinformatics)
- kallisto (2016, Nature Biotech.)

A hash table maps keys (e.g. a k-mer from a read or a transcript) to values (e.g. an integer identifier). Hash tables are not tolerant to mismatches.

Primary purpose is computational speed-up (e.g. compared to Bowtie1), as perfect hash functions allow fast, constant-time look-ups. However, index construction may be time-consuming.

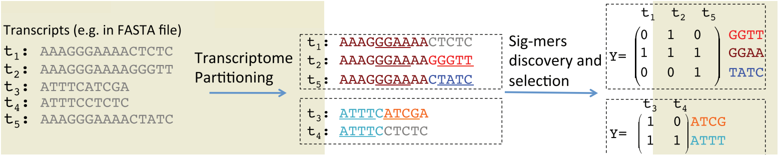
Unlike aligners, they also implement expression quantification using standard algorithms (see Li & Dewey 2011, Turro et al. 2011)

Sailfish



- Index construction depends only on transcriptome T and k
- A look-up table maps each k-mer (s_i) to a transcript set. The number of observations in the transcripts is also available (c_i)
- $k-$mers in the reads also in T are assigned integer indexes using the hash function and counted (k_i; others discarded)

RNA-Skim



Transcripts (e.g. in FASTA file)

t1: AAAGGGAAAACTCTC
t2: AAAGGGAAAAGGGTT
t3: ATTTCATCGA
t4: ATTTCCTCTC
t5: AAAGGGAAAACTATC

Transcriptome Partitioning

t1: AAAGGGAAAACTCTC
t2: AAAGGGAAAAGGGTT
t5: AAAGGGAAAACTATC

t3: ATTTCATCGA
t4: ATTTCCTCTC

Sig-mers discovery and selection

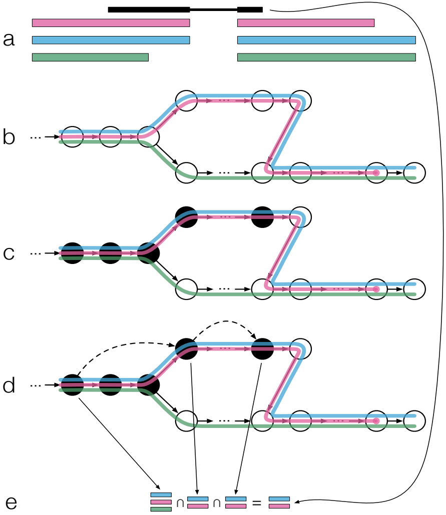
$$Y=\begin{array}{cccc} t_1 & t_2 & t_5 & \\ 0 & 1 & 0 & \text{GGTT} \\ 1 & 1 & 1 & \text{GGAA} \\ 0 & 0 & 1 & \text{TATC} \end{array}$$

$$Y=\begin{array}{ccc} t_3 & t_4 & \\ 1 & 0 & \text{ATCG} \\ 1 & 1 & \text{ATTT} \end{array}$$

- Partition transcripts into clusters
- Identify & select “sig-mers” (*k*-mers specific to one cluster)
- Run Sailfish-like algorithm independently on each cluster using subset of sig-mers (if all transcripts are in one cluster, then Sailfish ≡ RNA-Skim)

kallisto

- Generate a coloured transcriptome de Bruijn graph (each colour represents a transcript)
- *k*-compatibility class of a *k*-mer is the transcripts it is present in
- Identify *k*-compatibility class of a *read* as the intersection of the *k*-compatibility classes of its constituent *k*-mers



Filtering alignments

How to pick subset among competing alignments?

Number of mismatches (different genomic positions):

```
    genome  GCCCGACTCTAGCTAC........ATATTATCTCGAGTCCGA
candidates        CTCTAG                   CTCTAG
```

Filtering alignments

How to pick subset among competing alignments?

Number of mismatches (different genomic positions):

```
    genome  GCCCGACTCTAGCTAC........ATATTATCTCGAGTCCGA
candidates        CTCTAG                   CTCTAG
```

Number of mismatches (different alleles):

```
    haplotype1  GCACCCGACTCTAGCTAC
    haplotype2  GCACCCGACTCGAGCTAC
          read          CTCTAG
```

Filtering alignments

How to pick subset among competing alignments?

Number of mismatches (different genomic positions):

```
    genome  GCCCGACTCTAGCTAC........ATATTATCTCGAGTCCGA
candidates        CTCTAG                   CTCTAG
```

Number of mismatches (different alleles):

```
    haplotype1  GCACCCGACTCTAGCTAC
    haplotype2  GCACCCGACTCGAGCTAC
          read          CTCTAG
```

→ keep alignments within best “mismatch stratum”:

alignment	**A**	**B**	C	**D**
# mismatches	1	1	2	1

Filtering alignments

How to pick subset among competing alignments?

Multiple matches to same transcript (different positions):

```
transcript  TCCCGACTCTAGCTACGCCCGACGGTC
candidates   CCCGAC           CCCGAC
```

Filtering alignments

How to pick subset among competing alignments?

Multiple matches to same transcript (different positions):

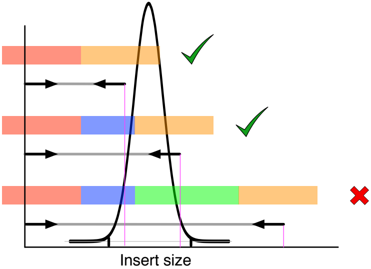
```
transcript  TCCCGACTCTAGCTACGCCCGACGGTC
candidates   CCCGAC           CCCGAC
```

- This fragment produced at ∼ twice the rate as other fragments
- We observe only one fragment, do not double count
- → This fragment should map only once to this transcript
- → Keep one alignment at random?

Filtering alignments

How to pick subset among competing alignments?

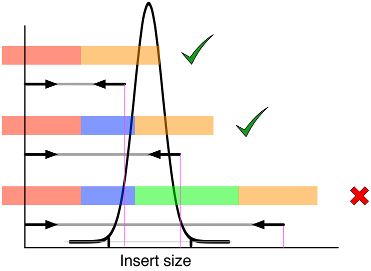
Multiple matches with different insert sizes:



Filtering alignments

How to pick subset among competing alignments?

Multiple matches with different insert sizes:



Or perhaps filter alignment i if $\frac{p(s_i|\mu,\sigma^2)}{\arg\max_j p(s_j|\mu,\sigma^2)} < k$,

s_i: insert size of candidate alignment i
μ, σ^2: mean and variance of insert size

Summary of mapping strategies

Reads can be...

- Assembled from scratch into features
- Aligned to the genome (using unspliced alignment for ChIP-seq or spliced alignment for RNA-seq and mapped to transcripts using reference or gene model assembly)
- Aligned to the transcriptome, thus mapped directly to transcripts

Summary of mapping strategies

Reads can be...

- Assembled from scratch into features
- Aligned to the genome (using unspliced alignment for ChIP-seq or spliced alignment for RNA-seq and mapped to transcripts using reference or gene model assembly)
- Aligned to the transcriptome, thus mapped directly to transcripts

The processed data comprise a table of *counts* for each feature (or set of features)

	sample 1	sample 2	sample 3	sample 4
feature (set) 1	24	14	33	15
feature (set) 2	29	11	76	91
feature (set) 3	0	2	1	4

. . .

Further reading

Turro E, Lewin A. **Statistical analysis of mapped reads from mRNA-seq data**. In: Do K-A, Qin ZS, Vannucci M, eds. *Advances in Statistical Bioinformatics: Models and Integrative Inference for High-Throughput Data*. Cambridge, England: Cambridge University Press; 2013:77-104.

Janes J*, Hu F*, Lewin AM, Turro E. **A comparative study of RNA-seq analysis strategies**. *Briefings in Bioinformatics*, 2015 Mar; 1–9.