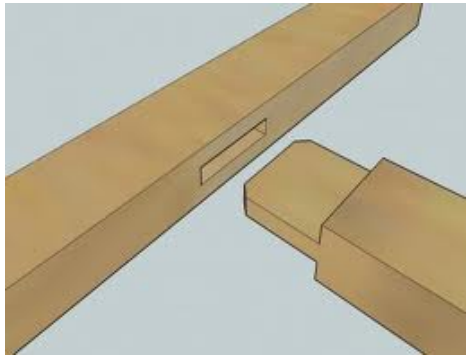
11/06/2014

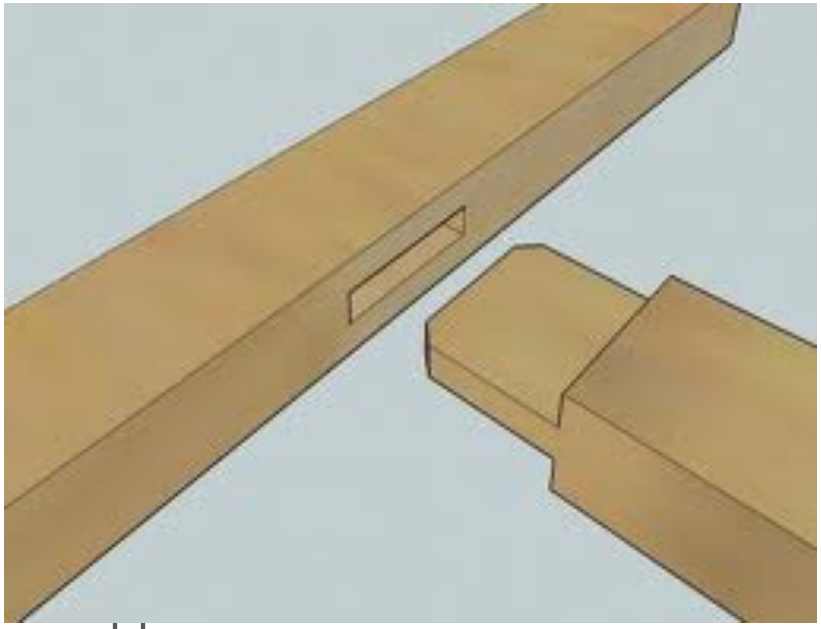
RNA Seq analysis

Transcriptome Assembly

Plateforme ABiMS

RNA Seq analysis





Transcriptome assembly

ASSEMBLY ALGORITHMS

Assembly algorithms

- Data model
  - Overlap-Layout-Consensus (OLC)
  - Eulerian / de Bruijn Graph (DBG)
- Search method
  - Greedy
  - Non-greedy
- Parallelizability
  - Multithreaded
  - Distributable

What is a “k-mer” ?

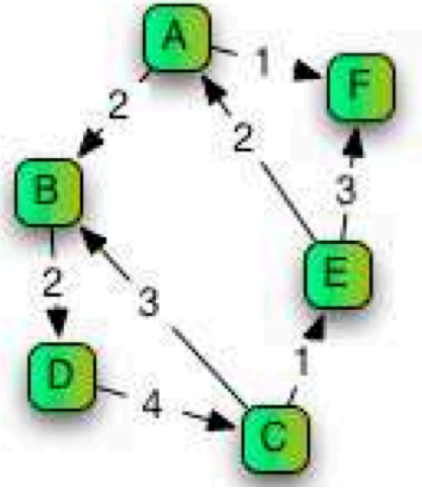
- A k-mer is a sub-string of length k
- A string of length L has (L-k+1) k-mers
- Example read L=8 has 5 k-mers when k=4

AGATCCGT

```
AGAT
  GATC
    ATCC
      TCCG
        CCGT
```

What is a graph ?

- Not an Excel chart!
- Nodes/Vertices
  - A,B,E,G,H,K,M
- Edges/Arcs
  - (lines between nodes)
- Directed graph
  - Arrow head on edge
- Weighted graph
  - Numerals on edges



Overlap - Layout – Consensus

- Overlap
  - All against all pair-wise comparison
  - Build graph: nodes=reads, edges=overlaps
- Layout
  - Analyse/simplify/clean the overlap graph
  - Determine Hamiltonian path (NP-hard)
- Consensus
  - Align reads along assembly path
  - Call bases using weighted voting

OLC : Pairwise Overlap

- All against all pair-wise comparison
  - 1⁄2 N(N-1) alignments to perform [N=no. reads]
- In practice, use smarter heuristics
  - Index all k-mers from all reads
  - Only check pairs that share enough k-mers
  - Similar approach to BLAST algorithm
- Both approaches parallelizable
  - Each comparison is independent

OLC: Overlap Example

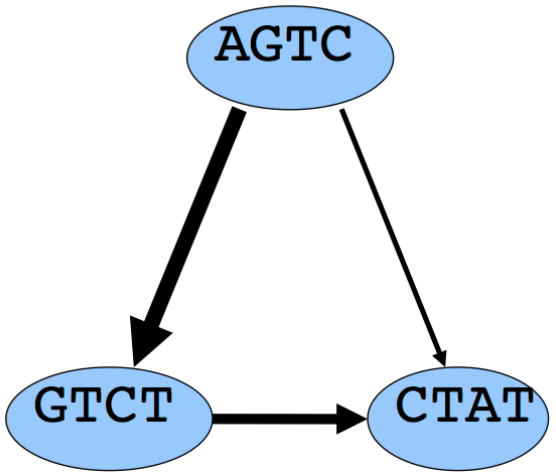
- True sequence (7bp) : AGTCTAT
- Reads (3 x 4bp) : AGTC, GTCT, CTAT
- Pairs to align (3)
  – AGTC+GTCT, AGTC+CTAT, GTCT+CTAT
- Best overlaps

```
AGTC-        AGTC---      GTCT—
-GTCT        ---CTAT      --CTAT
```

(good) *(poor)* *(ok)*

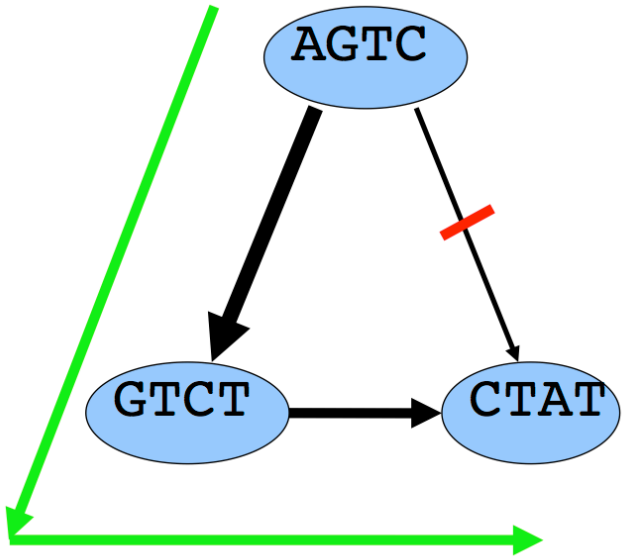
OLC: Overlap Graph

- Nodes are the 3 reads sequences
- Edges are the overlap alignment with orientation
- Edge thickness represents score of overlap



OLC: Layout – Consensus

- Optimal path shown in green
- Un-traversed weak overlap in red
- Consensus is read by outputting the overlapped nodes along the path
- `aGTCTCTat`



OLC : Softwares

- Phrap, PCAP, CAP3
  - Smaller scale assemblers
- Celera Assembler
  - Sanger-era assembler for large genomes
- Arachne, Edena, CABOG, Mira3
  - Modern Sanger/hybrid assemblers

- **Newbler (gsAssembler)**
  - Used for 454 NGS “long” reads
  - Can be used for IonTorrent flowgrams too

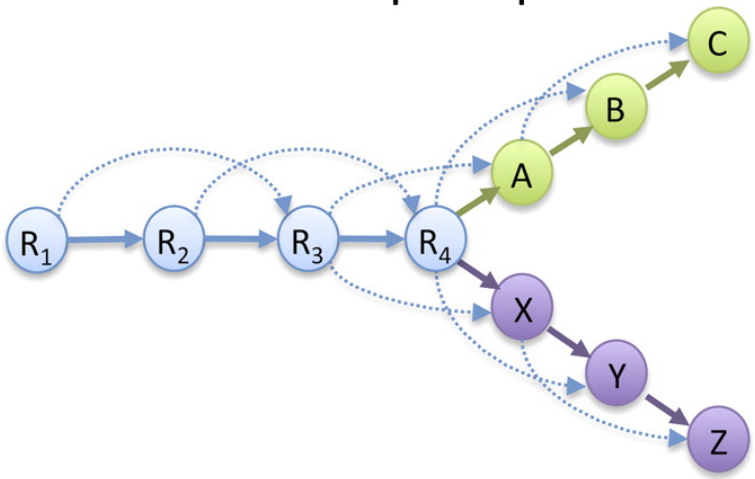
Eulerian approach

- Break all reads (length L) into (L-k+1) k-mers
  - L=36, k=31 gives 6 k-mers per read
- Construct a *de Bruijn* graph (DBG)
  - Nodes = one for each unique k-mer
  - Edges = k-1 exact overlap between two nodes
- Graph simplification
  - Merge chains, remove bubbles and tips
- Find a Eulerian path through the graph
  - Linear time algorithm, unlike Hamiltonian

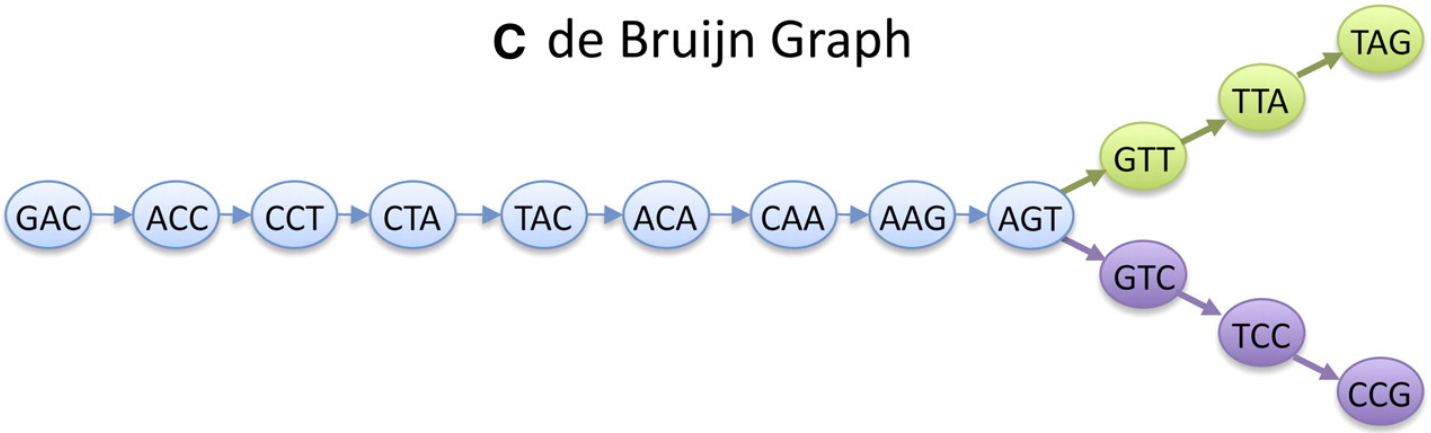
A Read Layout

```
R1: GACCTACA
R2:   ACCTACAA
R3:    CCTACAAG
R4:     CTACAAGT
A:       TACAAGTT
B:        ACAAGTTA
C:         CAAGTTAG
X:       TACAAGTC
Y:        ACAAGTCC
Z:         CAAGTCCG
```

B Overlap Graph



C de Bruijn Graph



DBG : simple

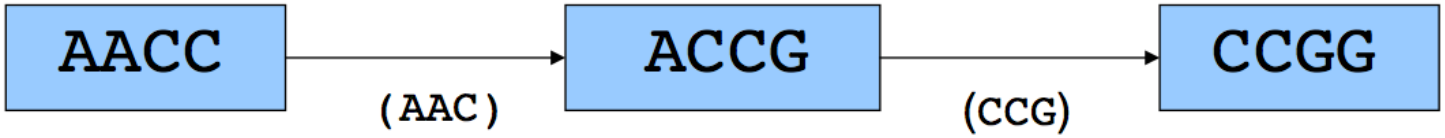
- Sequence

  `AACCGG`

- K-mers (k=4)

  `AACC  ACCG  CCGG`

- Graph



DBG : repeated k-mer

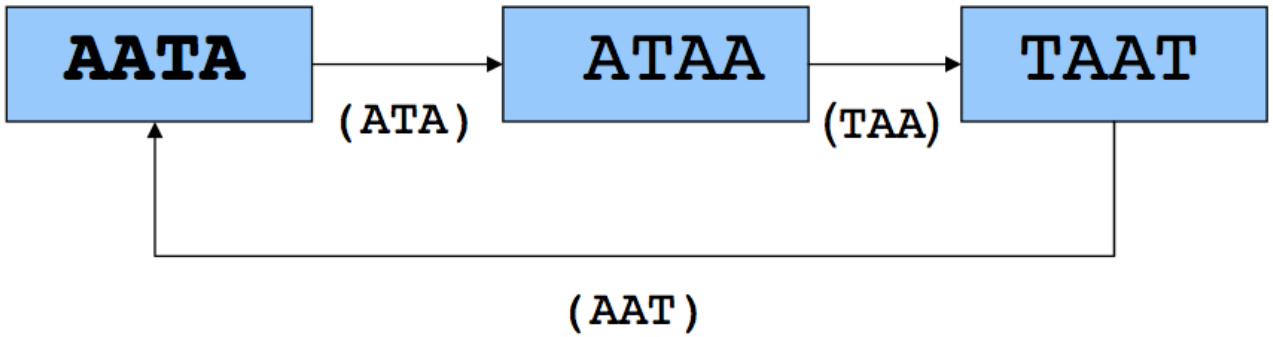
- Sequence

  `AATAATA`

- K-mers (k=4)

  `AATA ATAA TAAT AATA (repeat)`

- Graph



DBG: alternate paths

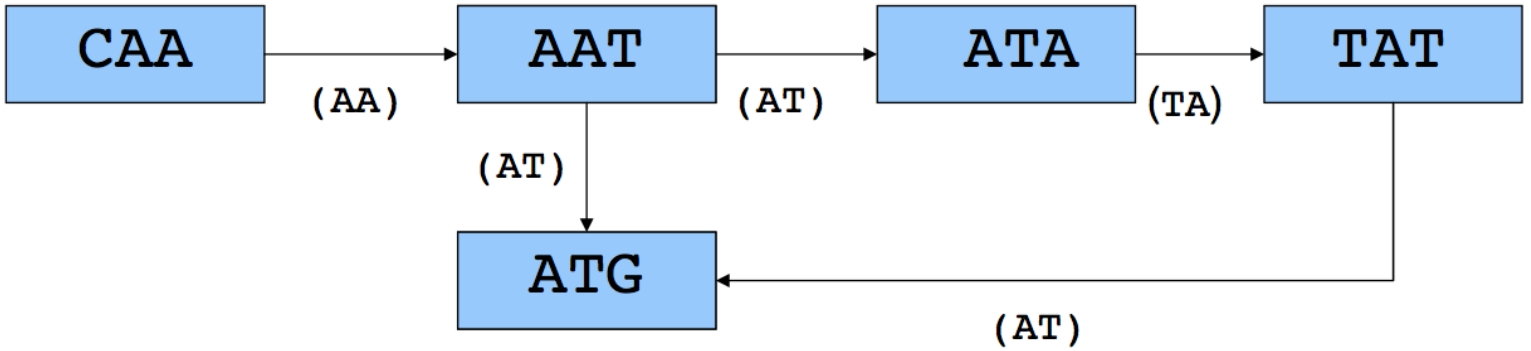
- Sequence

  `CAATATG`

- K-mers (k=3)

  `CAA AAT ATA TAT ATG`

- Graph



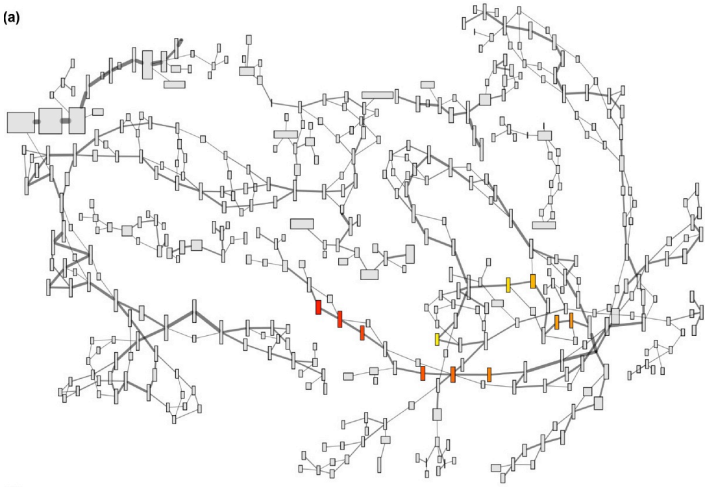
DBG: graph simplification

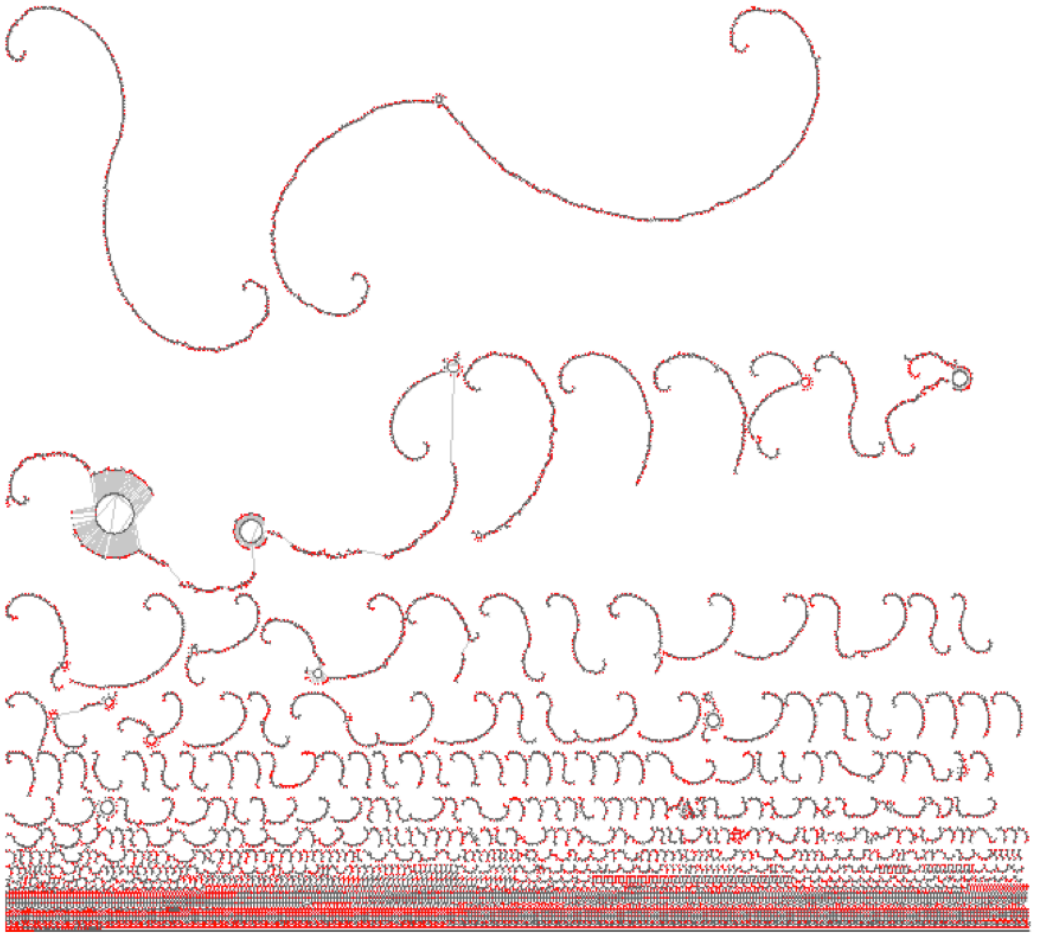
- This problem is known to be NP-complete
- In practice, heuristics are used which consist in simplifying the graph to « make it linear »
- However, the structures that are removed may correspond to relevant biological structures (SNPs, alternative splicing)

DBG: graph simplification

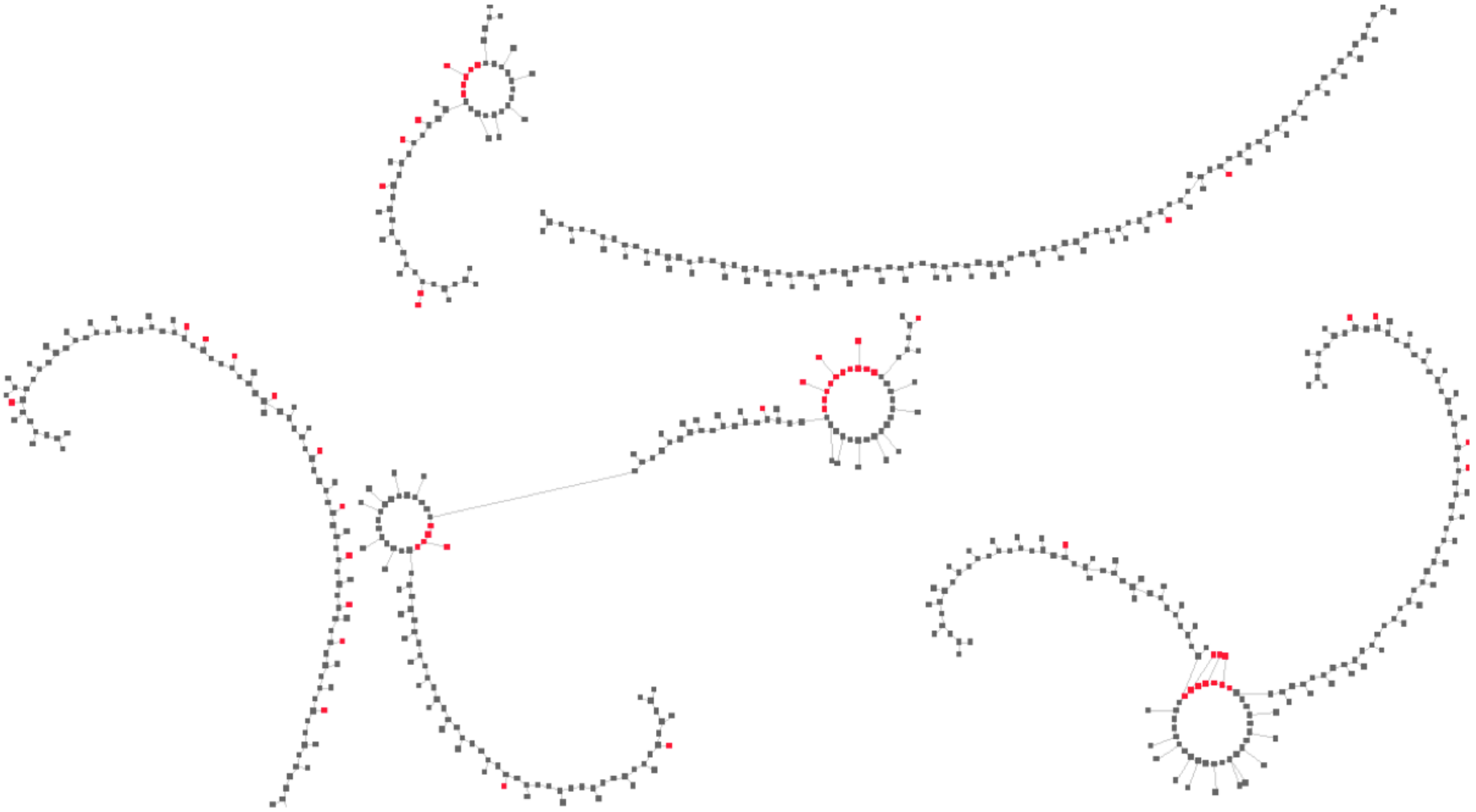
- Remove tips or spurs
  - Dead ends in graph due to errors at read end
- Collapse bubbles
  - Errors in middle of reads
  - But could be true SNPs or diploidity
- Remove low coverage paths
  - Possible contamination
- Makes final Eulerian path easier
  - And hopefully more accurate contigs

Example of DBG built from Genome data - RNA-seq data





Polymorphism in RNA-seq data

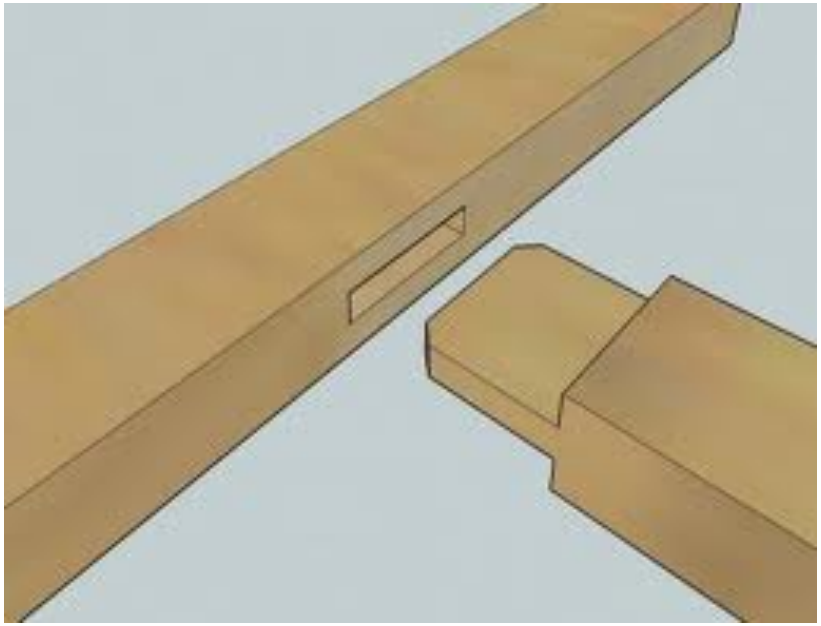


DBG: softwares

- **Velvet/Oases**
  - Velvet (Zerbino, Birney 2008) is a sophisticated set of algorithms that constructs de Bruijn graphs, simplifies the graphs, and corrects the graphs for errors and repeats.
  - Oases (Schulz et al. 2012) post-processes Velvet assemblies (minus the repeat correction) with different k-mer sizes.
- **Trans-ABySS**
  - Trans-ABySS (Robertson et al. 2010) takes multiple ABySS assemblies (Simpson et al. 2009)
- **CLC bio Genomics Workstation**
- **Trinity**

OLC vs DBG

- DBG
  - More sensitive to repeats and read errors
  - Graph converges at repeats of length k
  - One read error introduces k false nodes
  - Parameters: kmer_size cov_cutoff ...
- OLC
  - Less sensitive to repeats and read errors
  - Graph construction more demanding
  - Doesn't scale to voluminous short reads
  - Parameters: minOverlapLen %id ...
  - OLC assembly is best suited to lower coverage, longer read data such as Sanger, 454, or PacBio.

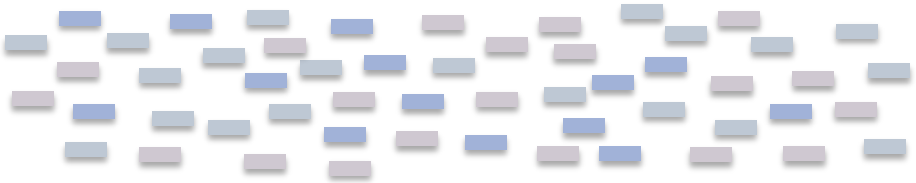


Transcriptome Assembly

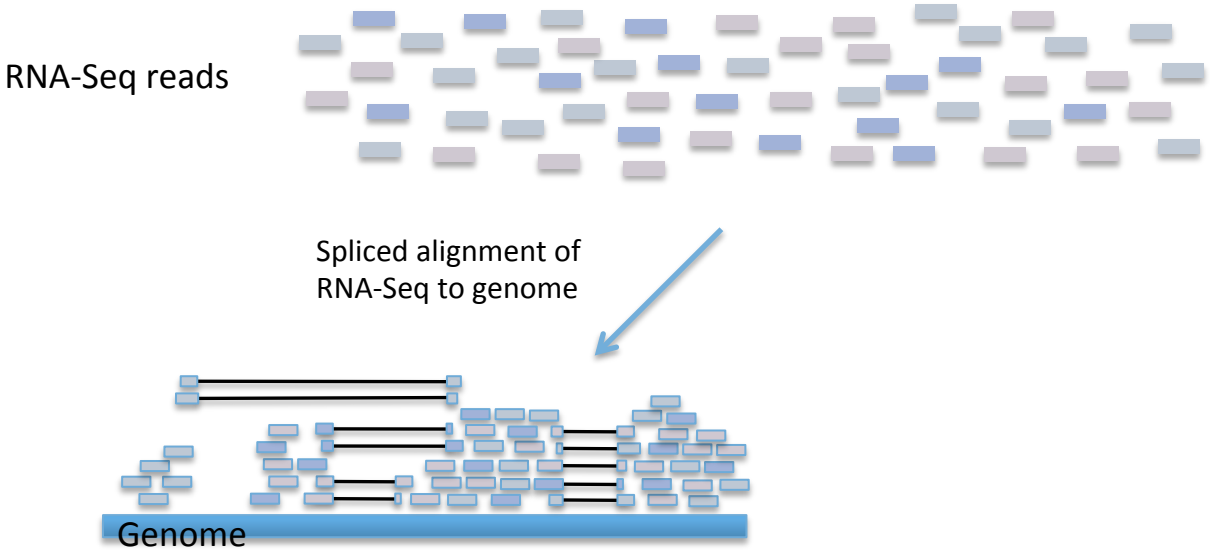
RNASEQ ASSEMBLY STRATEGIES

Contemporary strategies for transcript reconstruction from RNA-Seq

RNA-Seq reads

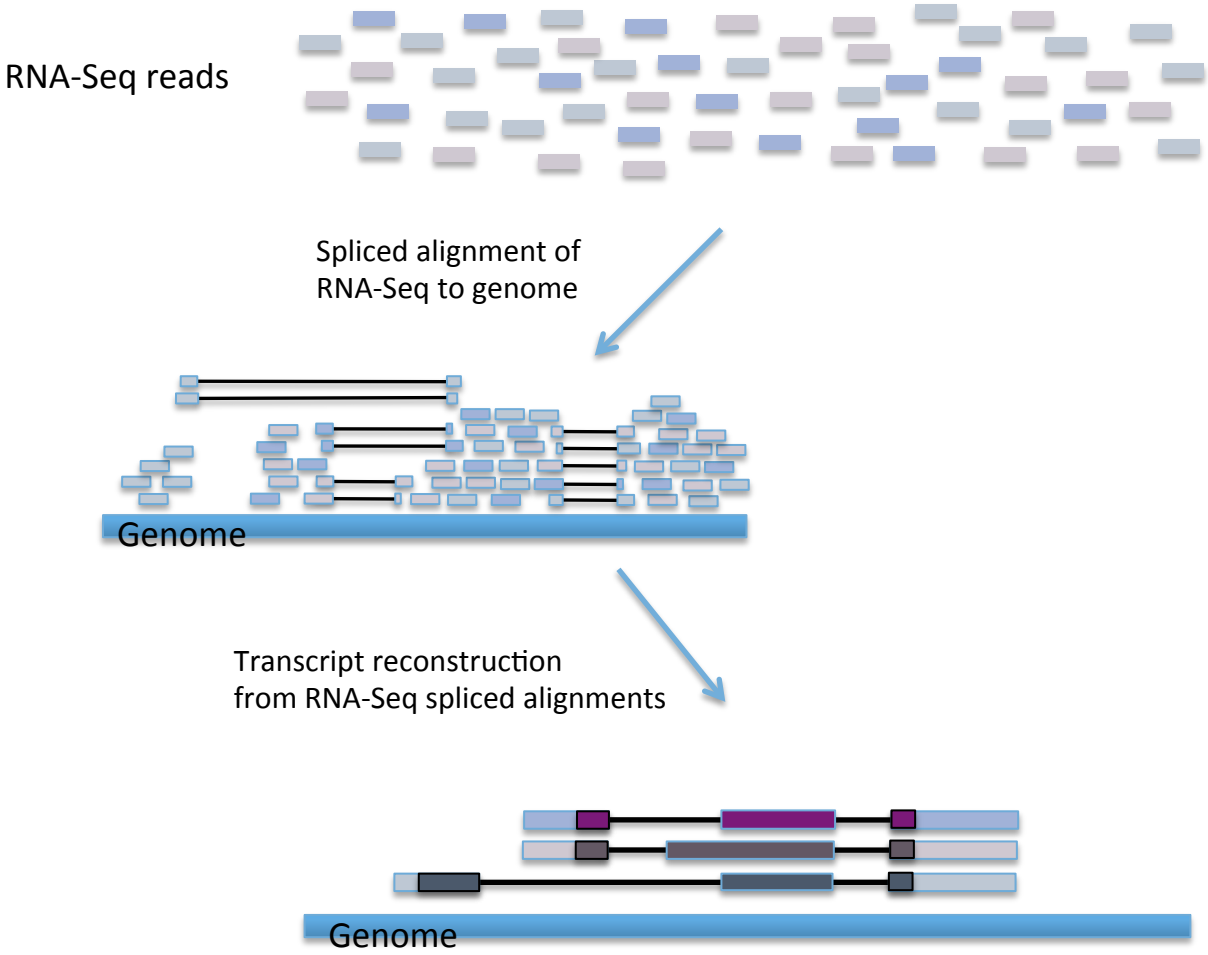


Contemporary strategies for transcript reconstruction from RNA-Seq



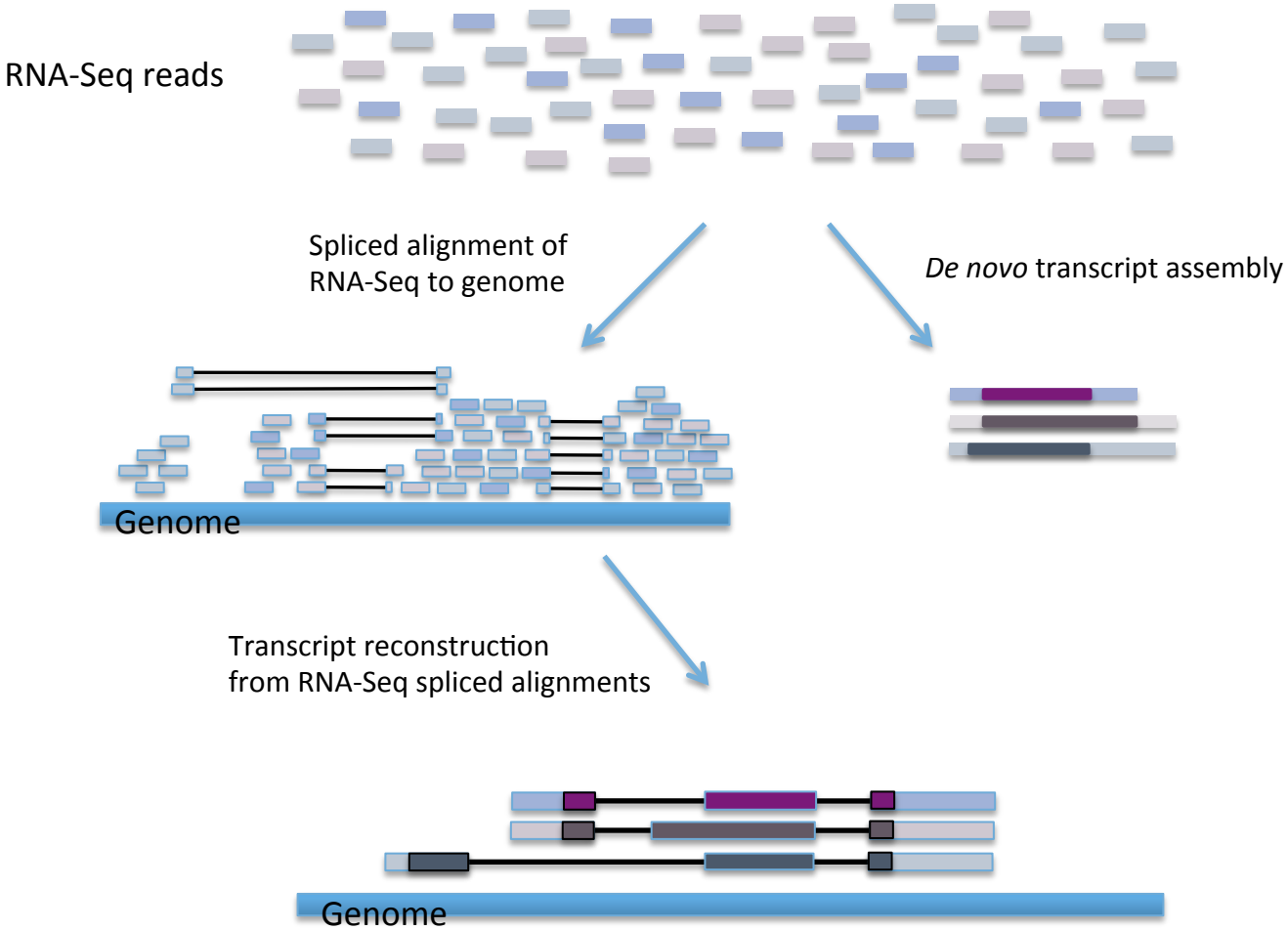
Reads are individually aligned to a reference genome sequence, using a short read spliced alignment tool that can map reads across introns.

Contemporary strategies for transcript reconstruction from RNA-Seq

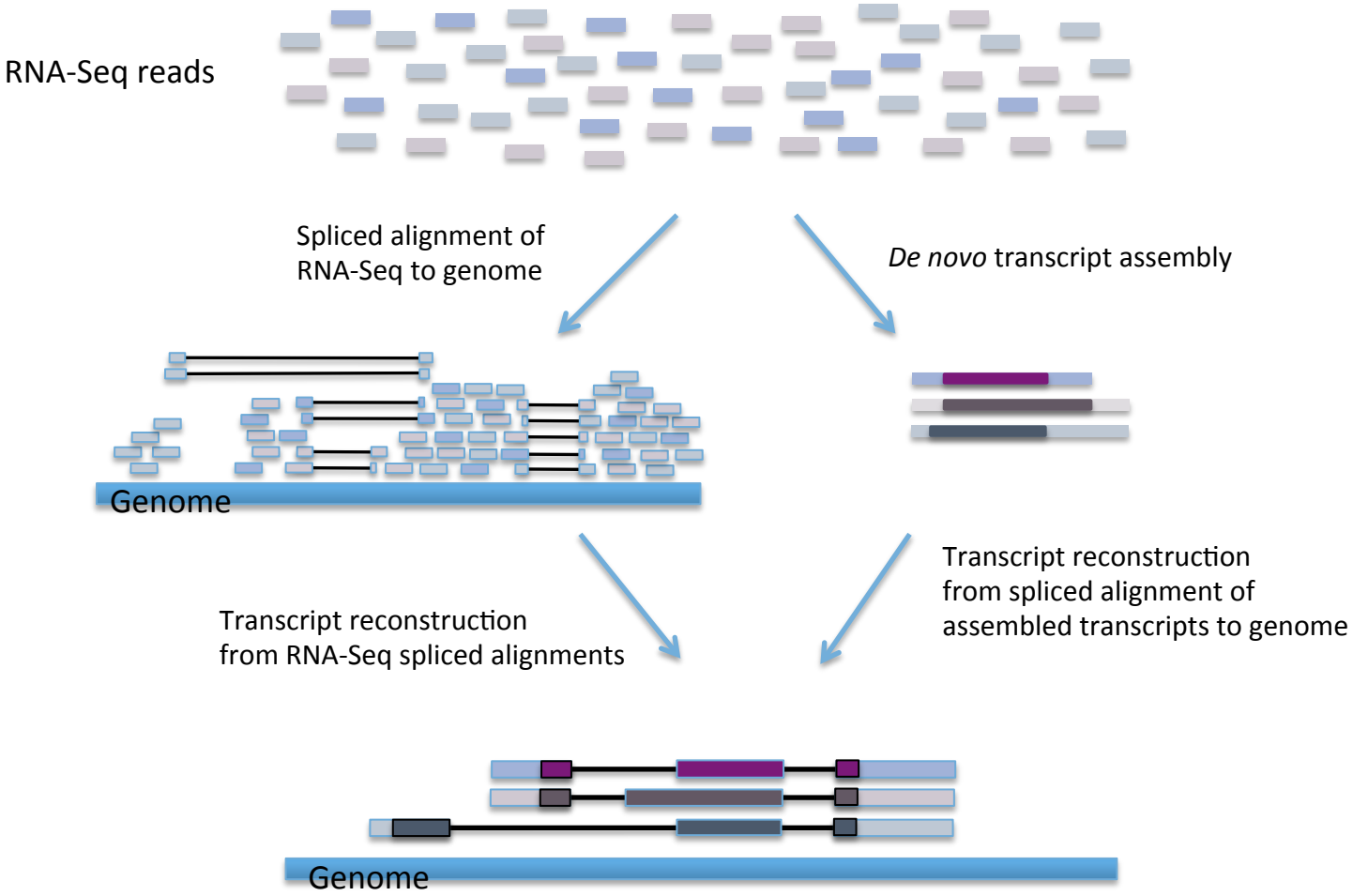


The short read alignments, instead of the reads themselves, are assembled into gene structures

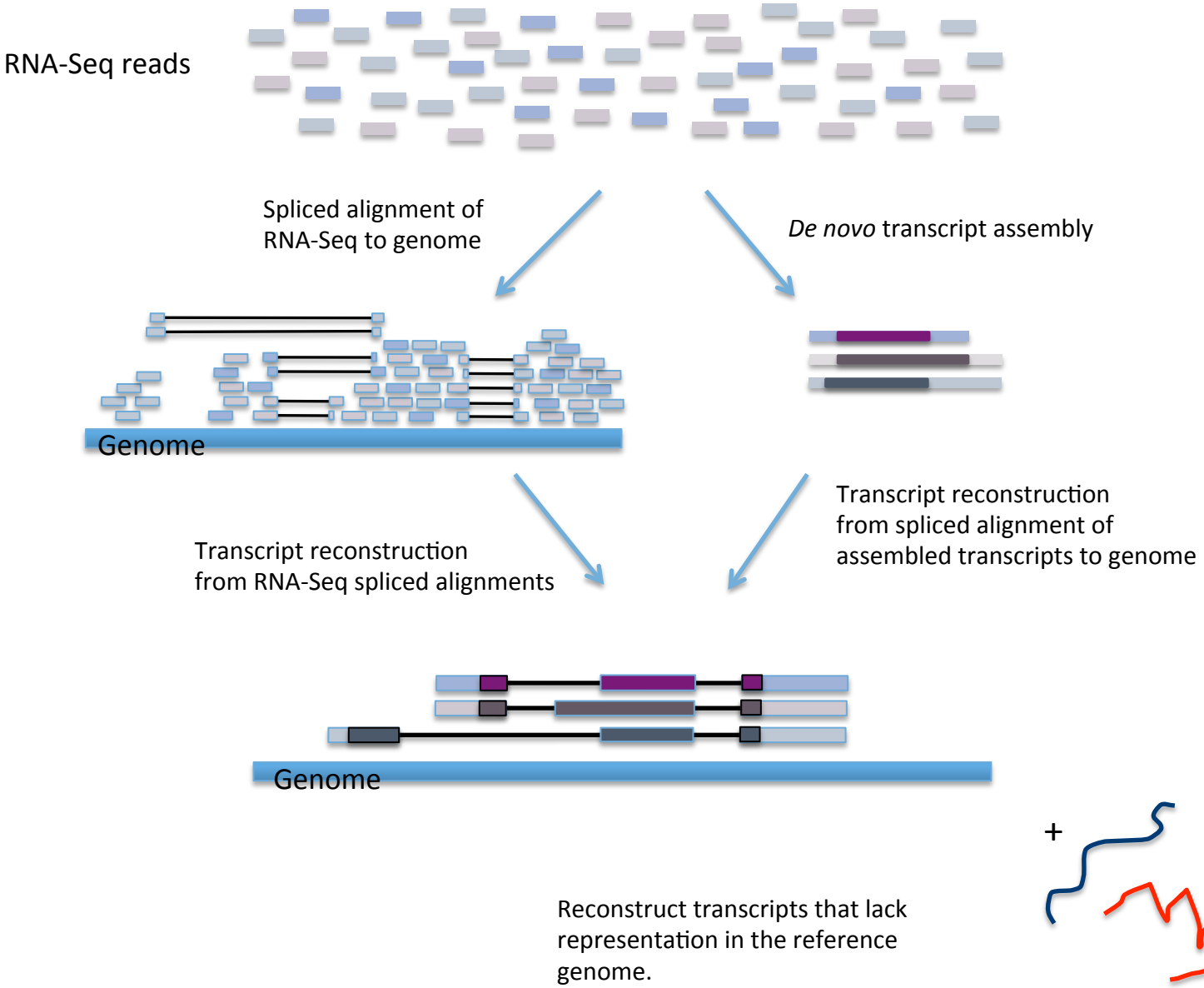
Contemporary strategies for transcript reconstruction from RNA-Seq



Contemporary strategies for transcript reconstruction from RNA-Seq



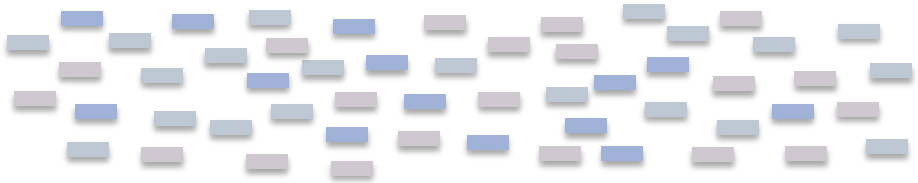
Contemporary strategies for transcript reconstruction from RNA-Seq



Reconstruct transcripts that lack representation in the reference genome.

Contemporary strategies for transcript reconstruction from RNA-Seq

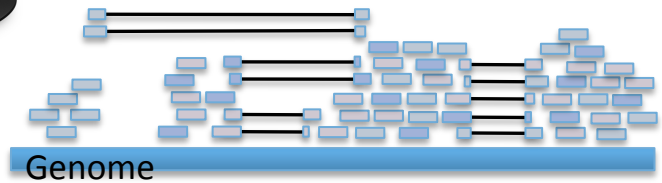
RNA-Seq reads



Spliced alignment of RNA-Seq to genome

De novo transcript assembly

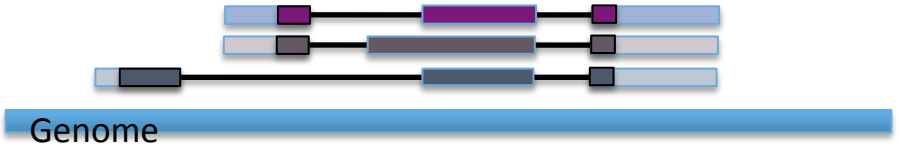
Tophat



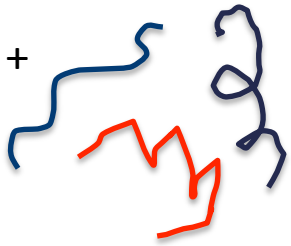
Transcript reconstruction from RNA-Seq spliced alignments

Transcript reconstruction from spliced alignment of assembled transcripts to genome

Cufflinks



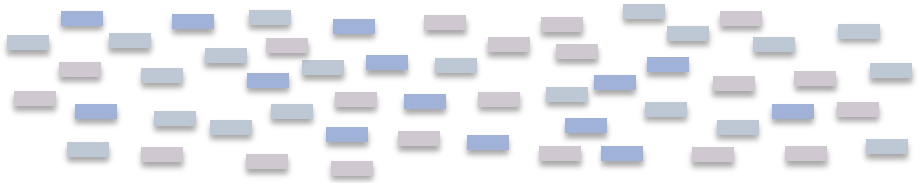
+



The Tuxedo software suite

Contemporary strategies for transcript reconstruction from RNA-Seq

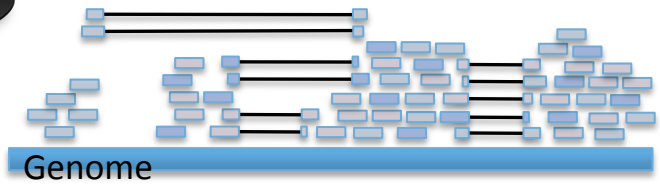
RNA-Seq reads



Spliced alignment of RNA-Seq to genome

De novo transcript assembly

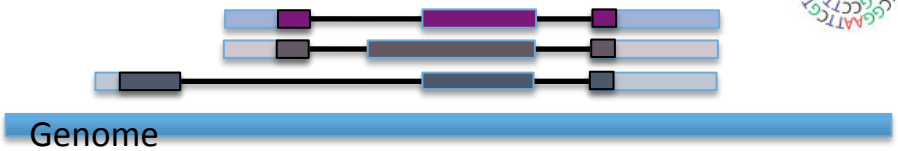
Tophat

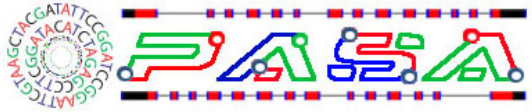


Transcript reconstruction from RNA-Seq spliced alignments

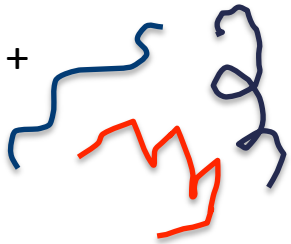
Transcript reconstruction from spliced alignment of assembled transcripts to genome

Cufflinks

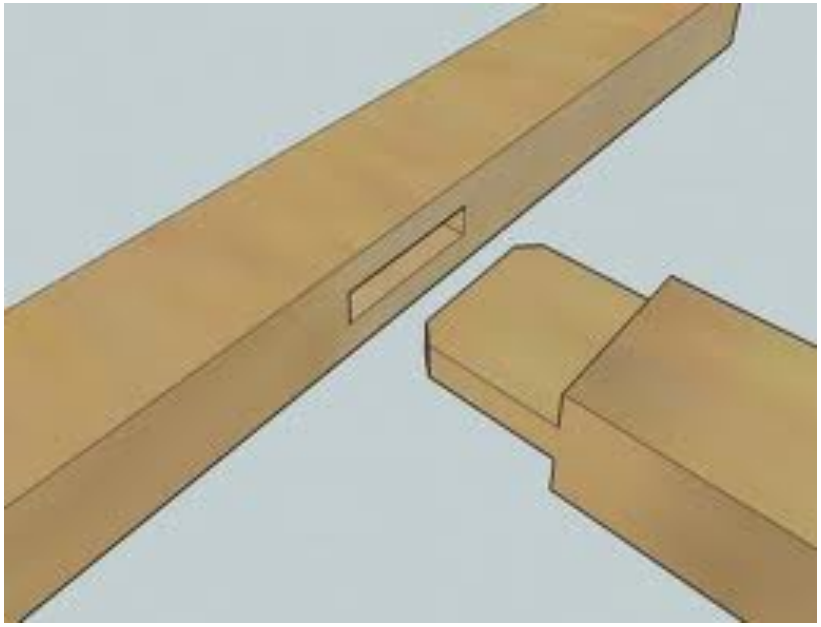




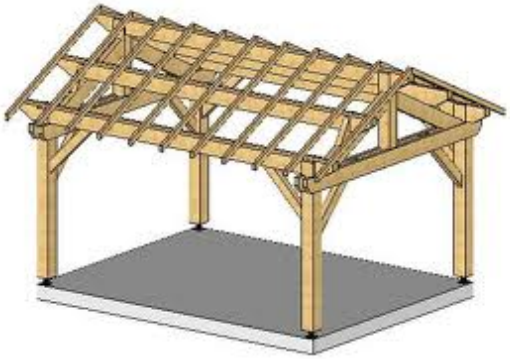
+



Denovo transcriptome assembly



?



Transcriptome assembly with Trinity: How it works

Brian Haas
Moran Yassour
Kerstin Lindblad-Toh
Aviv Regev
Nir Friedman
David Eccles
Alexie Papanicolaou
Michael Ott
…

developed at the
Broad Institute and the Hebrew University of Jerusalem

Trinity Team

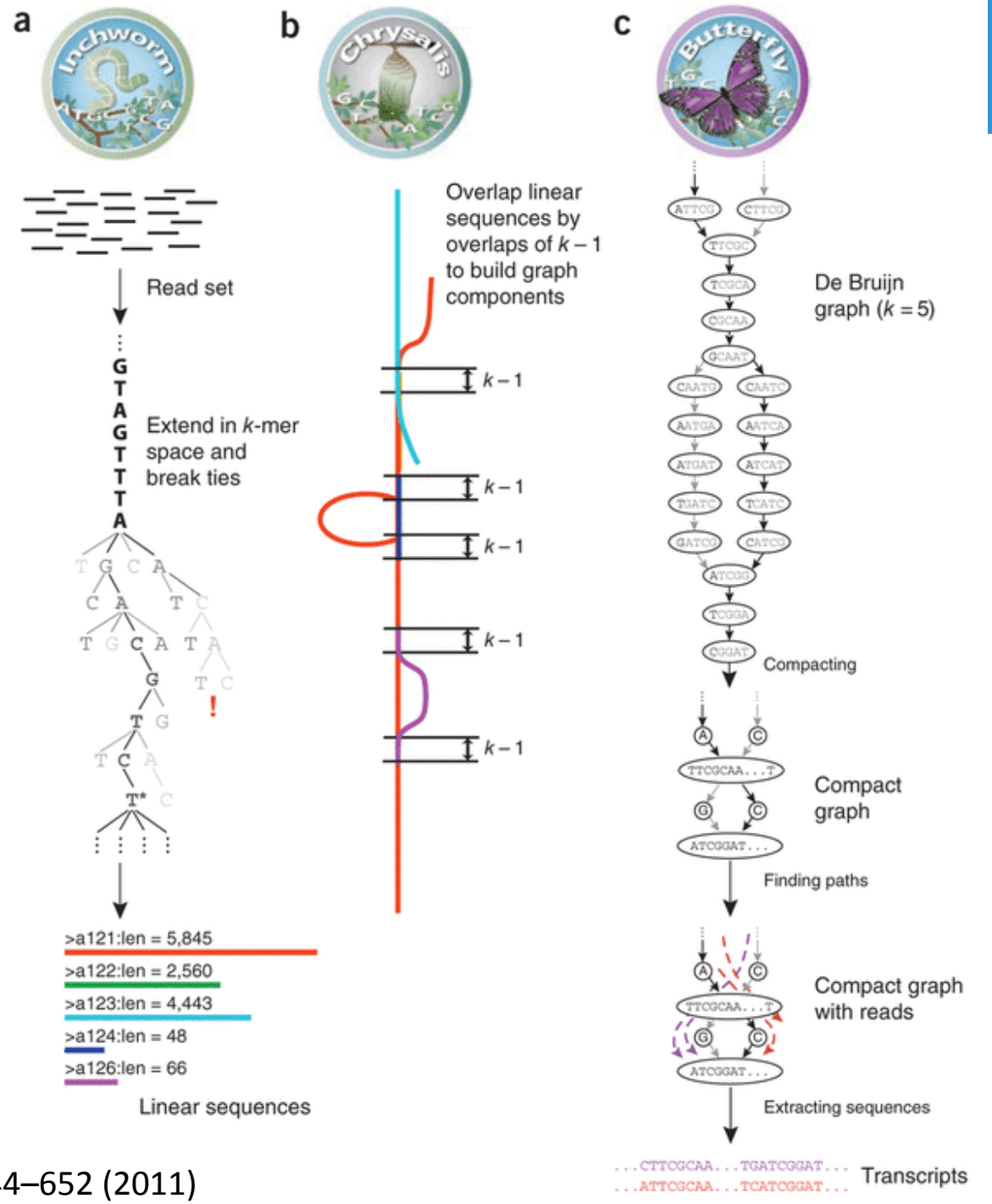
- Josh Bowden, CSIRO
- Brian Couger, Oklahoma State University
- David Eccles, Max Planck Institute for Molecular Biomedicine, Münster
- Nir Friedman, Hebrew University (PI)
- Manfred Grabherr, Biomedical Centre in Uppsala, Broad Institute
- Brian Haas, Broad Institute
- Robert Henschel, Indiana University
- Matthias Lieber, Technische Universitat Dresden
- Matthew MacManes, Berkeley
- Joshua Orvis, Institute for Genome Sciences, Broad Institute
- Michael Ott, CSIRO
- Alexie Papanicolaou, CSIRO
- Nathalie Pochet, Broad Institute
- Aviv Regev, Broad Institute (PI)
- Moran Yassour, Hebrew University, Broad Institute
- Nathan Weeks, USDA-ARS
- Rick Westerman, Purdue University

Transcript reconstruction is the 'end of the beginning'

RNA-Seq →

- Functional annotation
- Differential Expression
- Alternative Splicing
- SNPs & Alleles
- Coding regions
- Comparative Transcriptomics
- Fusion transcripts
- Support proteomic studies

Additional tools, plug-ins, and documentation continually added to the Trinity Suite



Nature Biotechnology **29**, 644–652 (2011)

Trinity workflow

- Compress data (inchworm):
  - Cut reads into k-mers (k consecutive nucleotides)
  - Overlap and extend (greedy)
  - Report all sequences (“contigs”)

- Build de Bruijn graph (chrysalis):
  - Collect all contigs that share k-1-mers
  - Build graph (disjoint “components”)
  - Map reads to components

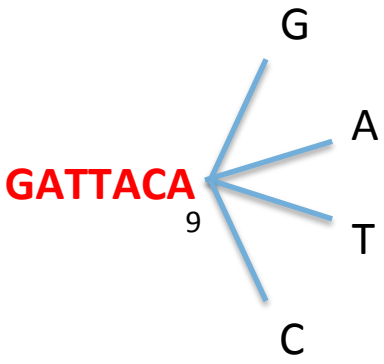
- Enumerate all consistent possibilities (butterfly):
  - Unwrap graph into linear sequences
  - Use reads and pairs to eliminate false sequences
  - Use dynamic programming to limit compute time (SNPs!!)

Inchworm Algorithm

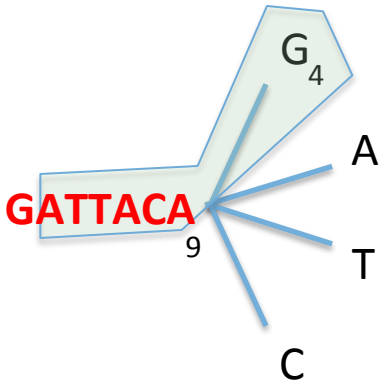
Decompose all reads into overlapping Kmers (25-mers) and count them : Jellyfish

Identify seed kmer as most abundant Kmer, ignoring low-complexity kmers.

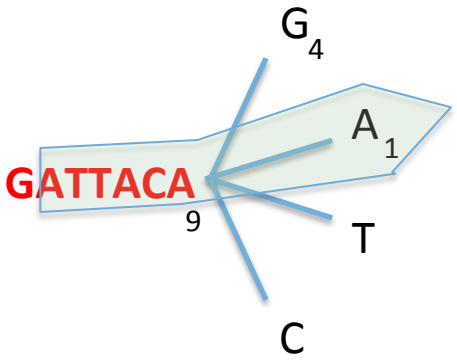
Extend kmer at 3' end, guided by coverage.



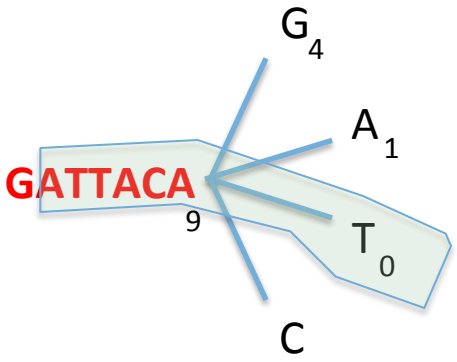
Inchworm Algorithm



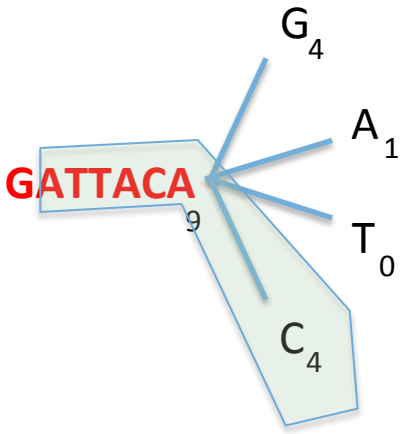
Inchworm Algorithm



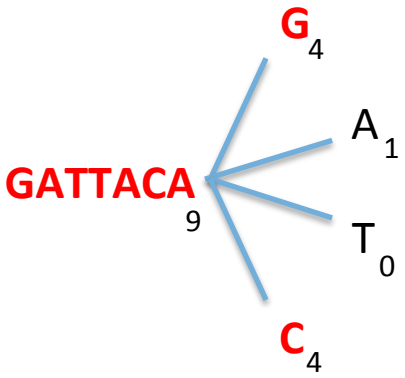
Inchworm Algorithm



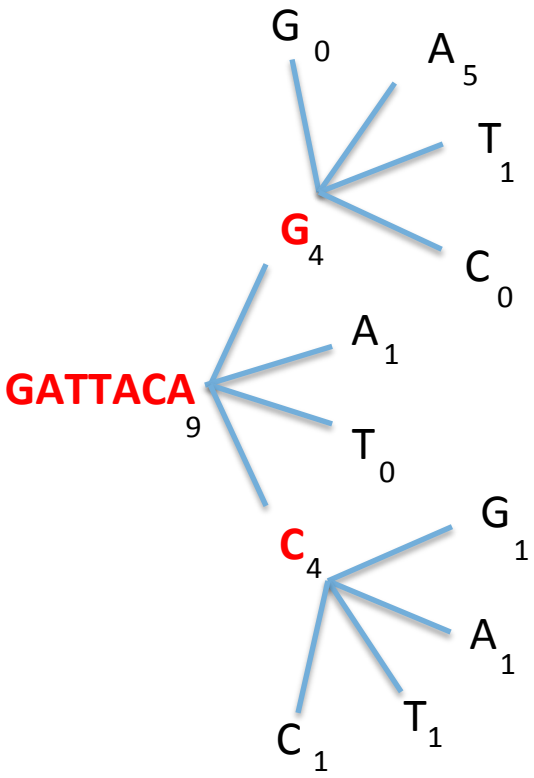
Inchworm Algorithm



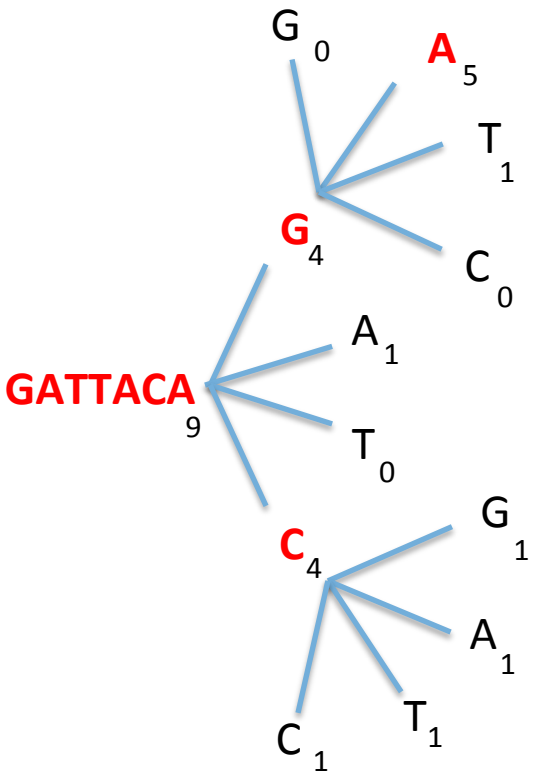
Inchworm Algorithm



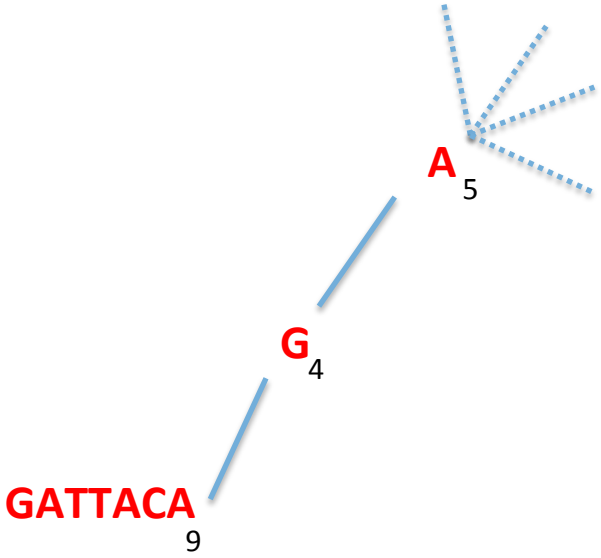
Inchworm Algorithm



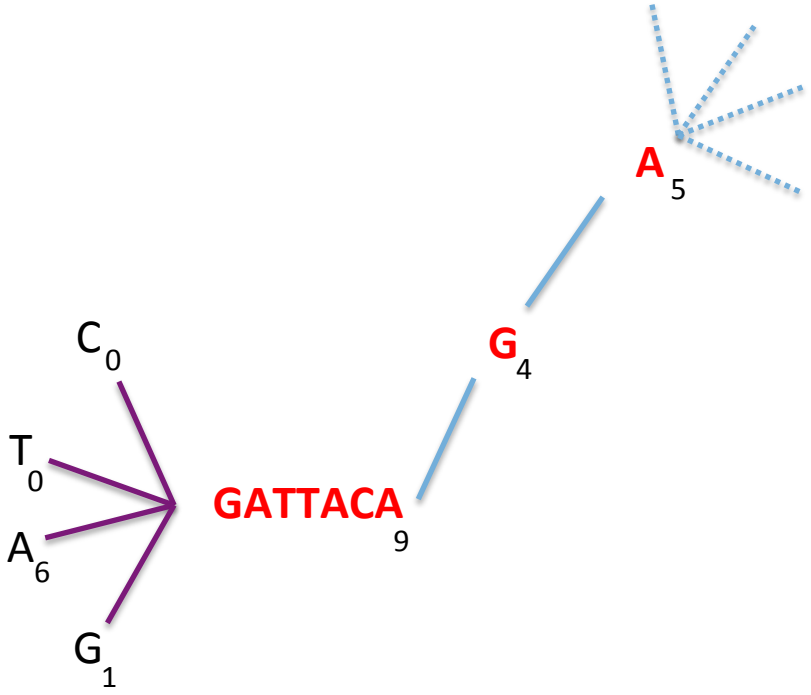
Inchworm Algorithm



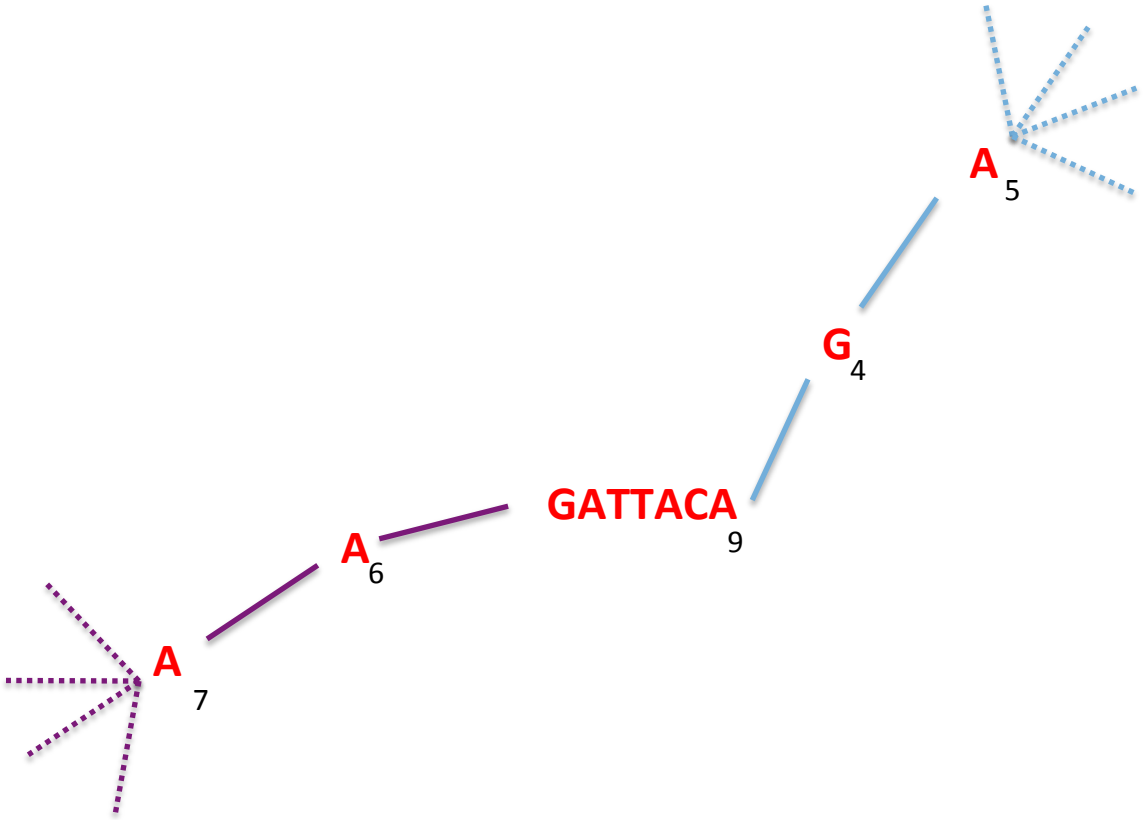
Inchworm Algorithm



Inchworm Algorithm



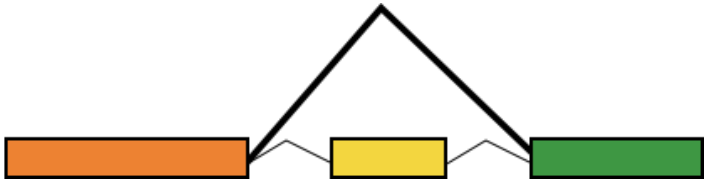
Inchworm Algorithm



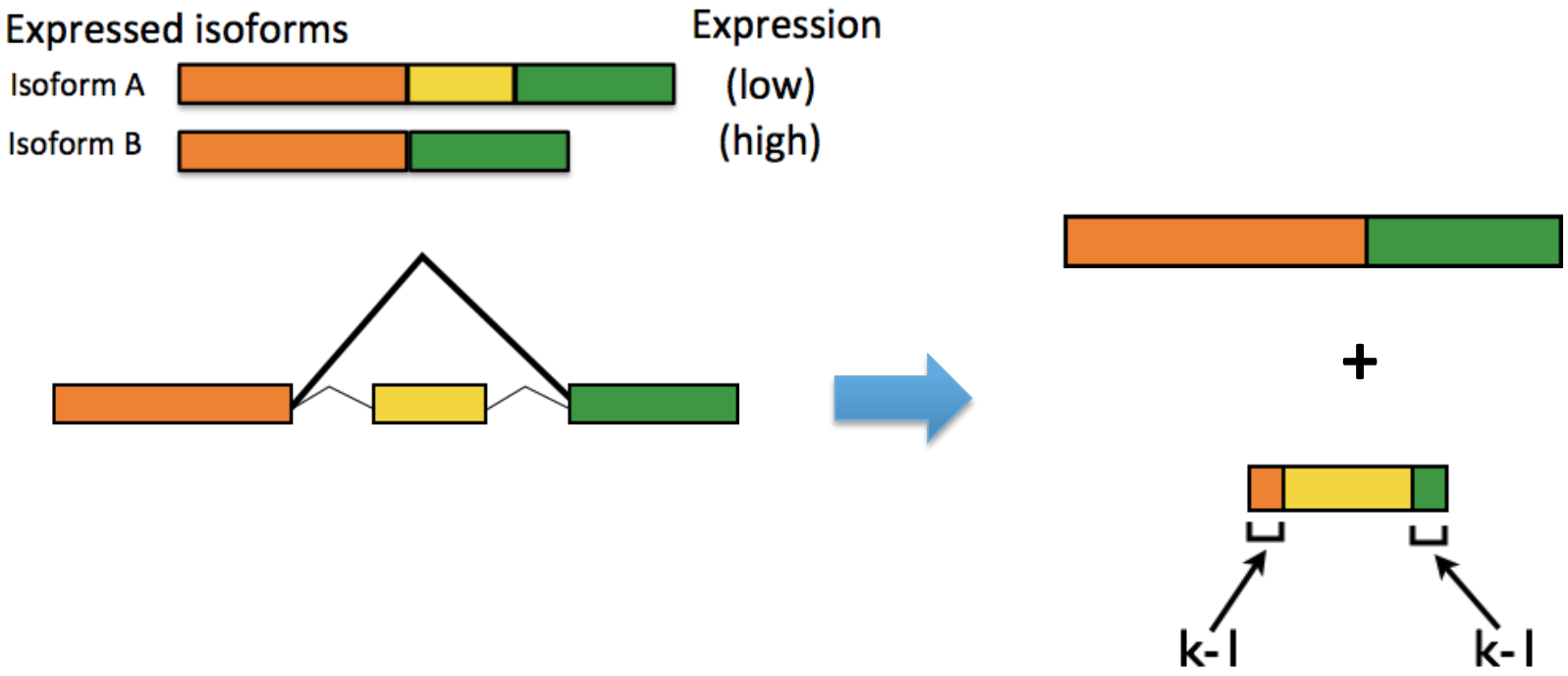
Report contig: ....AAGATTACAGA....

Remove assembled kmers from catalog, then repeat the entire process.

Inchworm Contigs from Alt-Spliced Transcripts



Inchworm Contigs from Alt-Spliced Transcripts

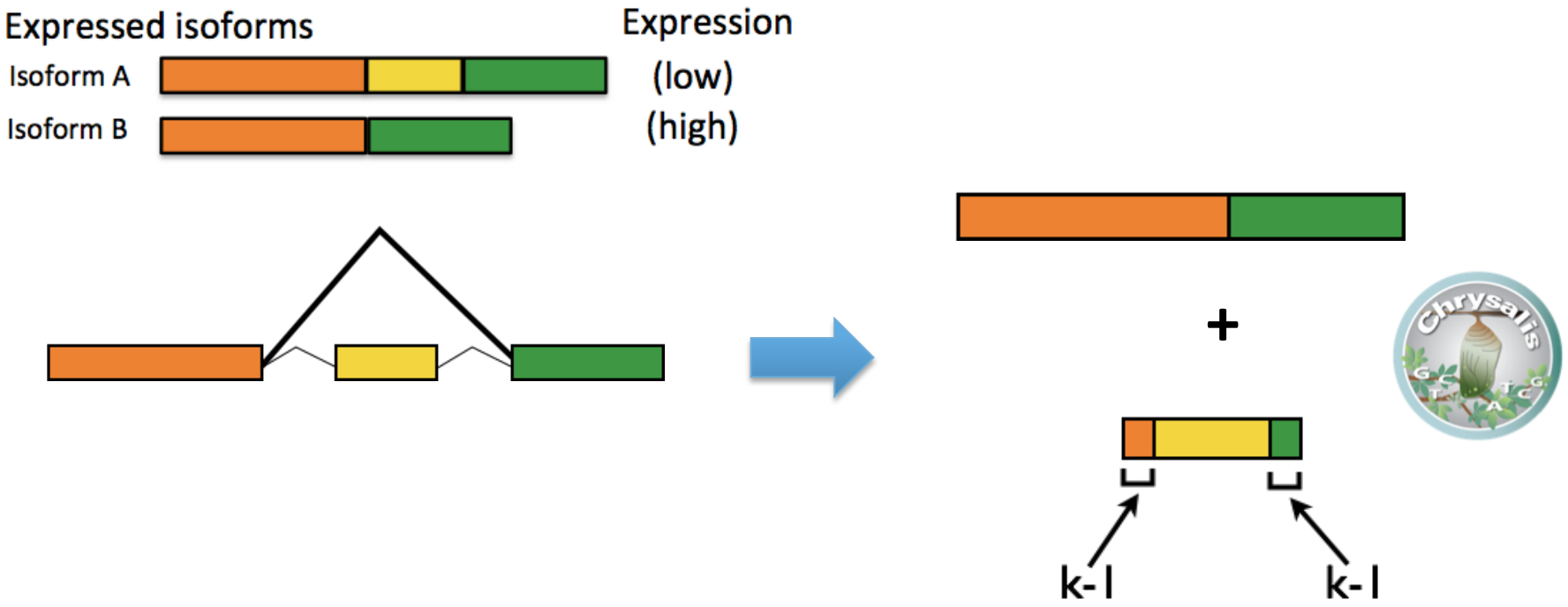


Inchworm can only report contigs derived from unique kmers.

Alternatively spliced transcripts :

- the more highly expressed transcript may be reported as a single contig,
- the parts that are different in the alternative isoform are reported separately.

Inchworm Contigs from Alt-Spliced Transcripts



Inchworm can only report contigs derived from unique kmers.

Alternatively spliced transcripts :

- the more highly expressed transcript may be reported as a single contig,
- the parts that are different in the alternative isoform are reported separately.

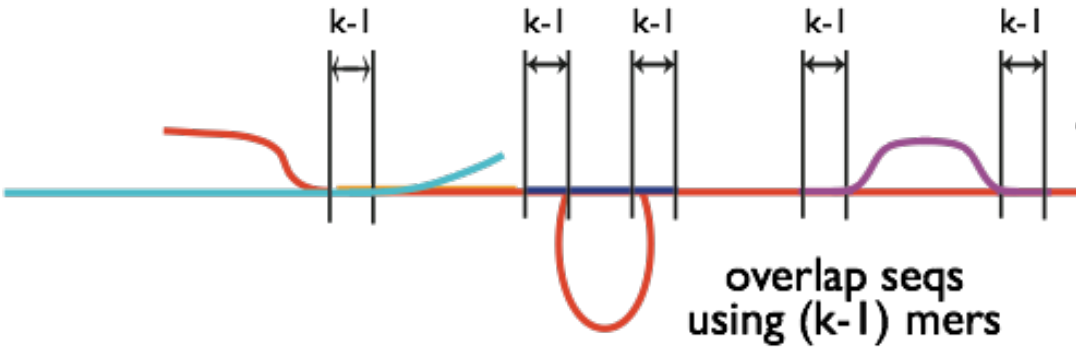
Chrysalis

Integrate (clustering)
Isoforms via k-1 overlaps

Chrysalis

>a121:len=5845
>a122:len=2560
>a123:len=4443
>a124:len=48
>a125:len=8876
>a126:len=66

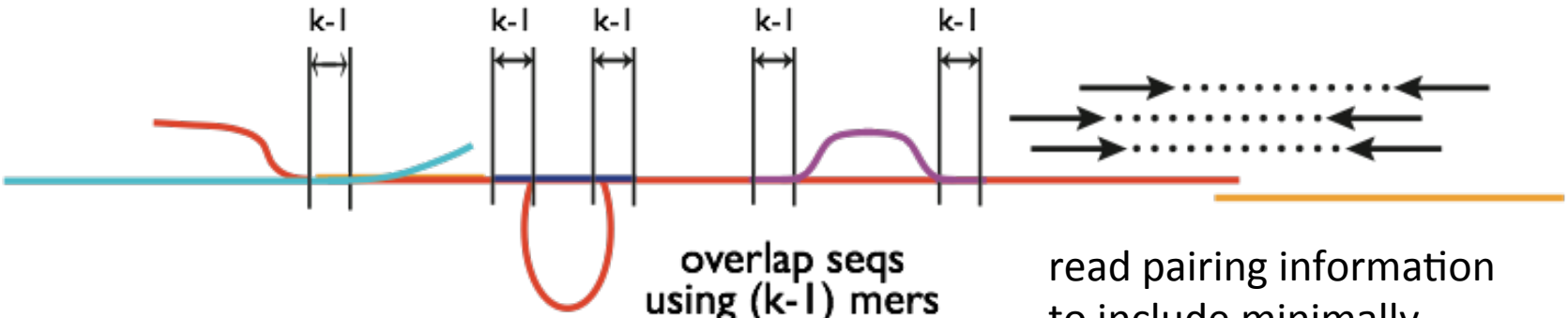
Integrate (clustering)
Isoforms via k-1 overlaps



Chrysalis

>a121:len=5845
>a122:len=2560
>a123:len=4443
>a124:len=48
>a125:len=8876
>a126:len=66

Integrate (clustering)
Isoforms via k-1 overlaps

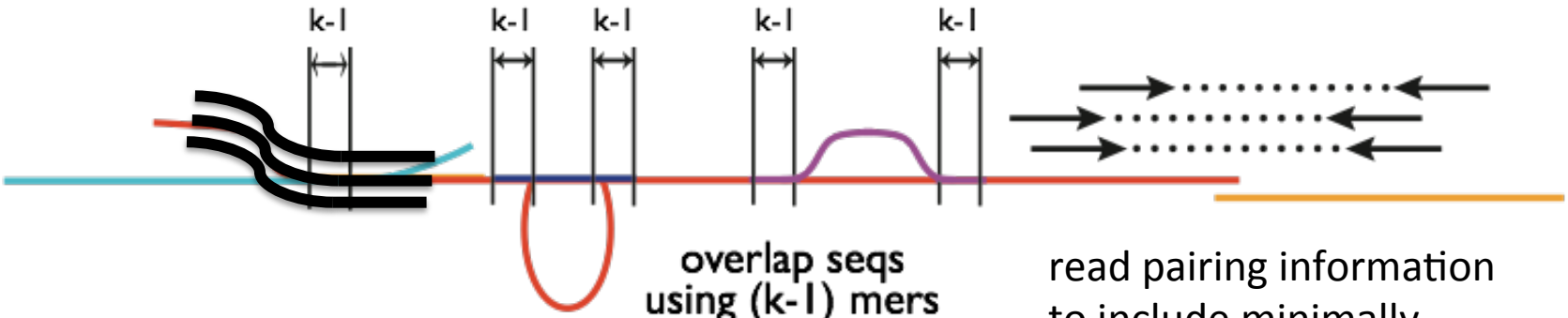


read pairing information
to include minimally
overlapping contigs

>a121:len=5845
>a122:len=2560
>a123:len=4443
>a124:len=48
>a125:len=8876
>a126:len=66

Integrate (clustering)
Isoforms via k-1 overlaps

Verify via “welds”



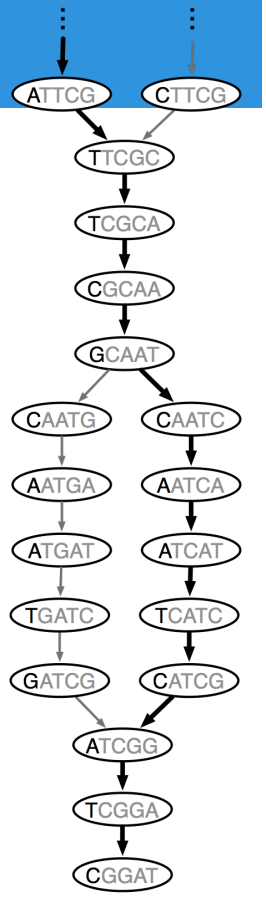
read pairing information
to include minimally
overlapping contigs

Chrysalis

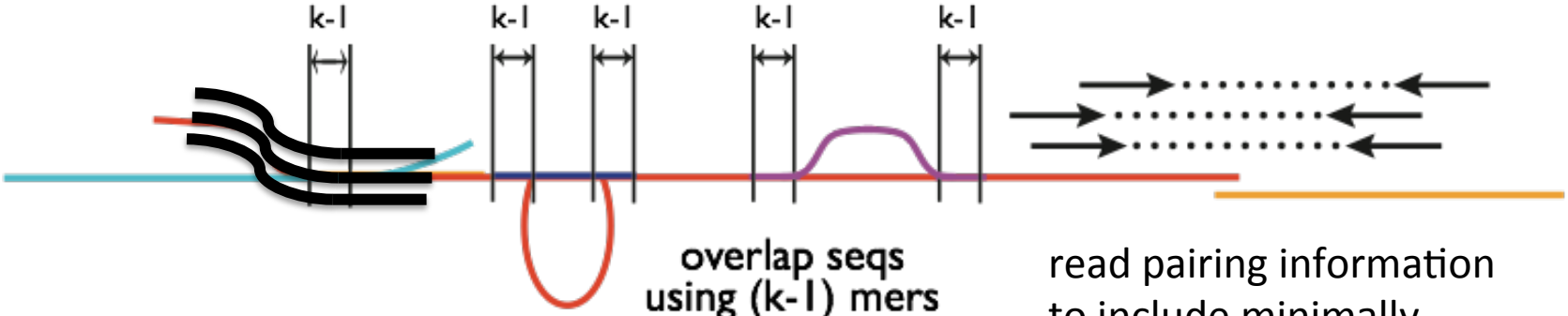
>a121:len=5845
>a122:len=2560
>a123:len=4443
>a124:len=48
>a125:len=8876
>a126:len=66

Integrate (clustering)
Isoforms via k-1 overlaps

Verify via “welds”

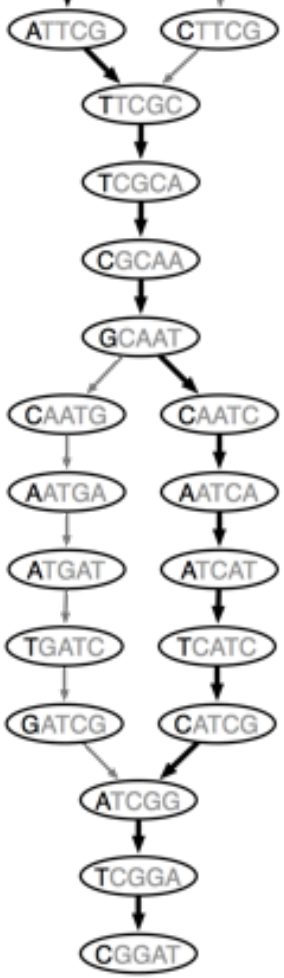


Build de Bruijn Graphs
(ideally, one per gene)



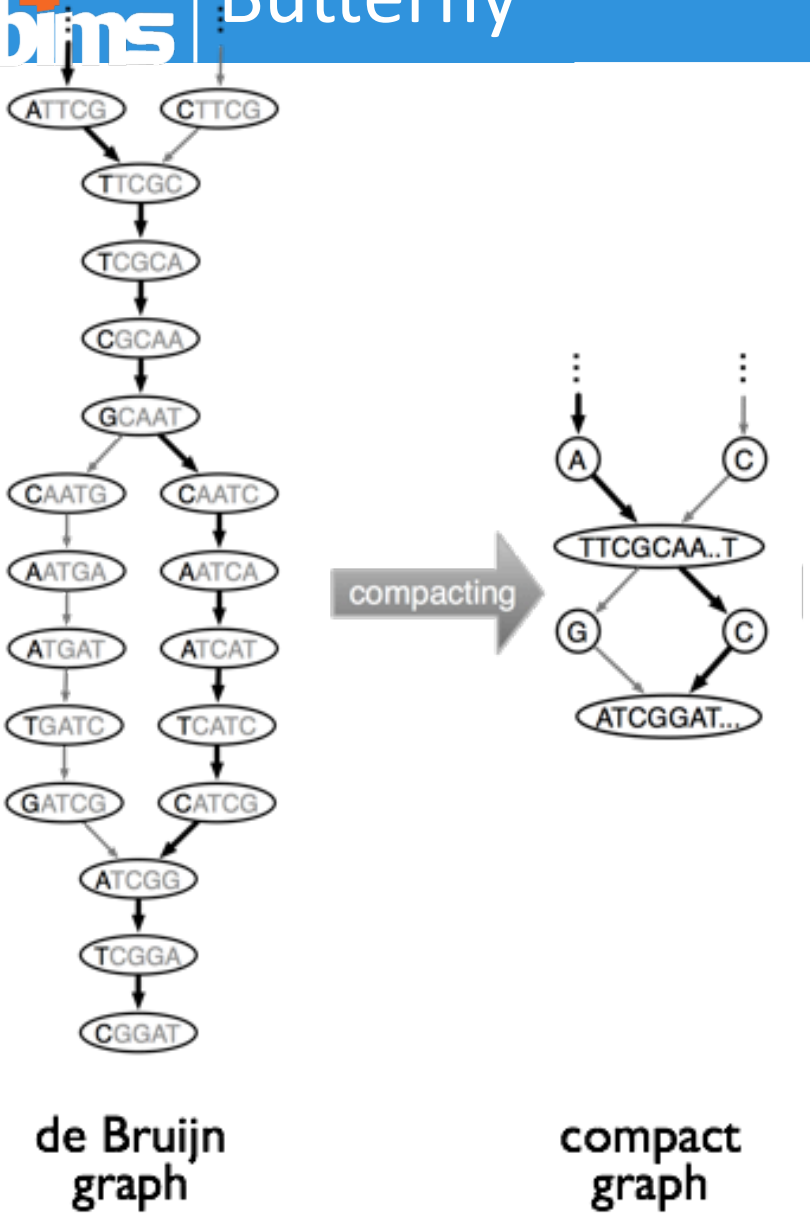
read pairing information
to include minimally
overlapping contigs

Butterfly

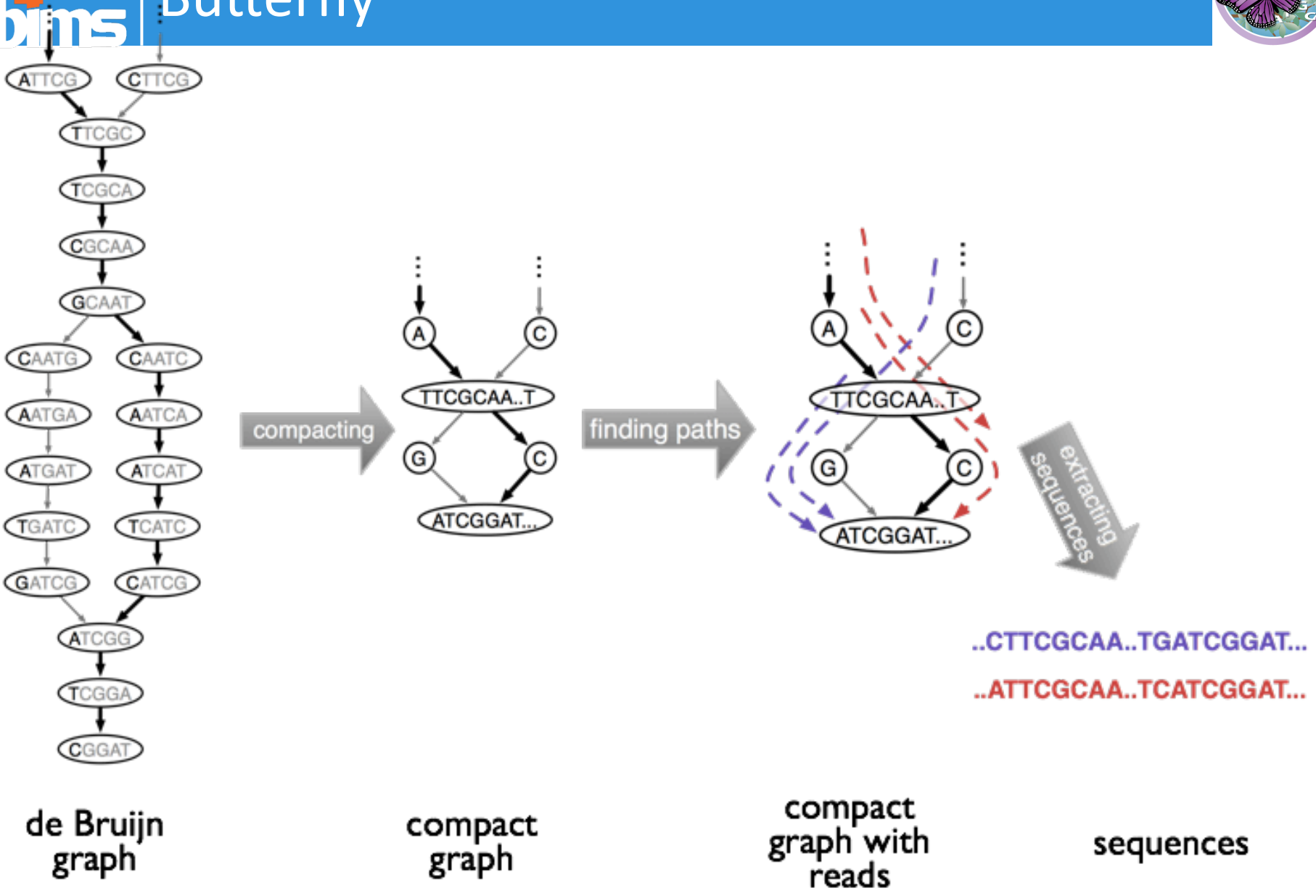


de Bruijn graph

Butterfly


ATTCG
CTTCG
TTCGC
TCGCA
CGCAA
GCAAT
CAATG
CAATC
AATGA
AATCA
ATGAT
ATCAT
TGATC
TCATC
GATCG
CATCG
ATCGG
TCGGA
CGGAT
compacting
A
C
TTCGCAA..T
G
C
ATCGGAT...
de Bruijn graph
compact graph

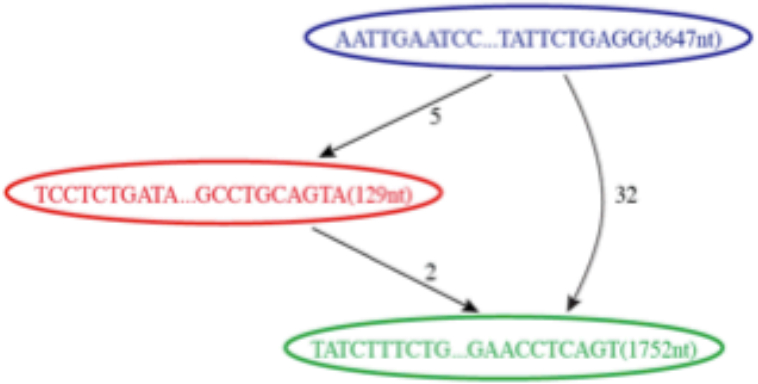
Butterfly


ATTCG
CTTCG
TTCGC
TCGCA
CGCAA
GCAAT
CAATG
CAATC
AATGA
AATCA
ATGAT
ATCAT
TGATC
TCATC
GATCG
CATCG
ATCGG
TCGGA
CGGAT
compacting
A
C
TTCGCAA..T
G
C
ATCGGAT...
finding paths
A
C
TTCGCAA..T
G
C
ATCGGAT...
extracting sequences
..CTTCGCAA..TGATCGGAT...
..ATTCGCAA..TCATCGGAT...
de Bruijn graph
compact graph
compact graph with reads
sequences

Alternatively spliced transcripts

Butterfly Example 1:
Reconstruction of Alternatively Spliced Transcripts

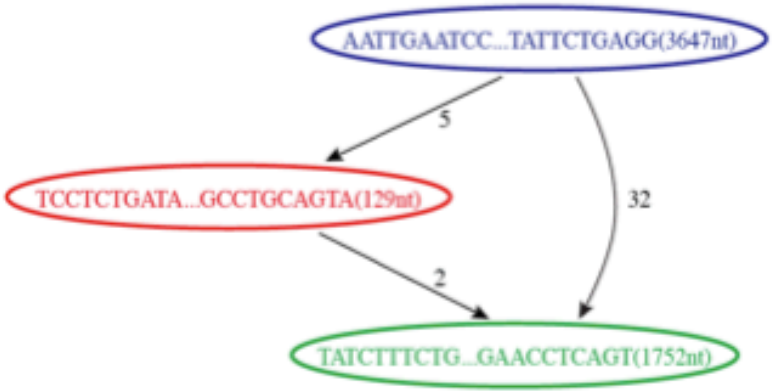
Butterfly's Compacted Sequence Graph



Alternatively spliced transcripts

Reconstruction of Alternatively Spliced Transcripts

Butterfly's Compacted Sequence Graph

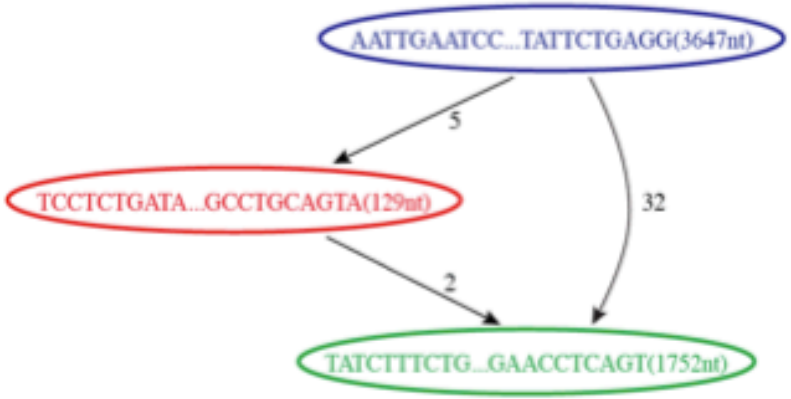


Reconstructed Transcripts

Alternatively spliced transcripts

Reconstruction of Alternatively Spliced Transcripts

Butterfly's Compacted Sequence Graph

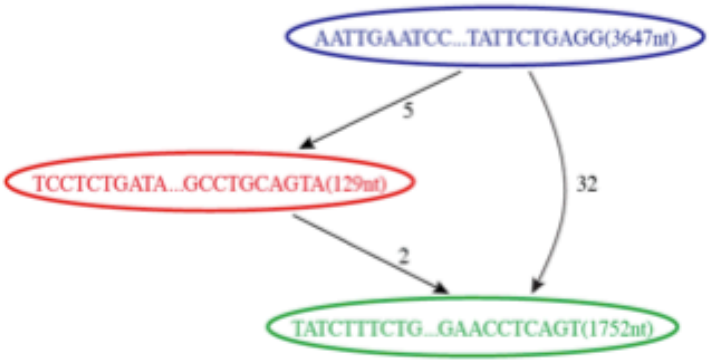


Reconstructed Transcripts

Alternatively spliced transcripts

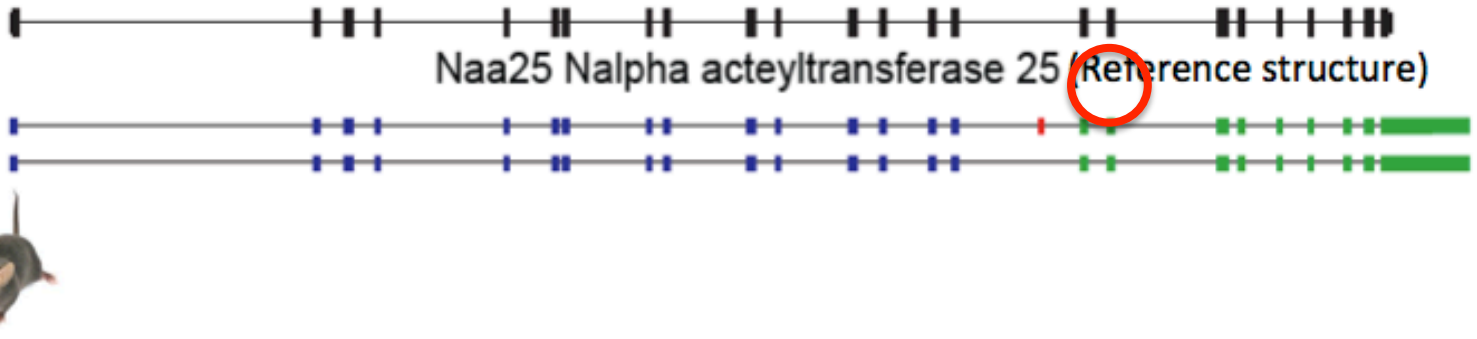
Reconstruction of Alternatively Spliced Transcripts

Butterfly's Compacted Sequence Graph



Reconstructed Transcripts

Aligned to Mouse Genome



Results

Result: linear sequences grouped in *components*, *contigs* and *sequences*

>comp1017_c1_seq1 len:373 path:[2317,2791,353]
GTTCGAGGACCTGAATAAGCGCAAGGACACCAAGGAGATCTACACGCACTTCACGTGCGCCACCGACACCAAGAACGTGC
AGTTTGTGTTTGATGCCGTCACCGACGTCATCATCAAGAACAACCTGAAGGACTGCGGCCTCTTCTGAGGGGCAGCGGGG
CCTGGCAGGATGGGCCACCGCCGACTCTGCTTCCCCCAGTTCCTGAGGAAGATGGGGGCAAGAGGACCACGCTCTCTGCC
TGTCCGTACCCCCGCCCTGGCTGCTTTTCCCCTTTTCTTTGTTCTTGGCTCCCCTGTTCCCTCCCTCAGTTCCAGAGACT
CGTGGGAGGAGCTGCCACAGGCCTCCCTGTTTGAAGCCGGCCCTTGTCC

>comp1017_c1_seq2 len:890 path:[2317,2791,5739,5784,5857,5863,353]
GTTCGAGGACCTGAATAAGCGCAAGGACACCAAGGAGATCTACACGCACTTCACGTGCGCCACCGACACCAAGAACGTGC
AGTTTGTGTTTGATGCCGTCACCGACGTCATCATCAAGAACAACCTGAAGGACTGCGGCCTCTTCTGAGGGGCAGCGGGG
CCTGGCAGGATGGTGAGCCCGGGGTGGAGCGGAGCAGAGCTGTGGAGCCCAGAGAAGGGAGCGGTGGGGGCTGGGGTGGG
CCGTGGTGGGGGTATGGTGGTAGAGTGGTAGGTCGGTAGGACGACCTGAGGGGCATGGGCACACGGATAGGCCGGGCCGG
GGCCCAGATGGCAGAAGCATCCGGCCGTGCGCCGGGAGACAACGGAATGGCTGTCCTGACCACCCTTGGAGAAAGCTTAC
CGGCTCTGTGCTCAGCCCTGCAGTCTTTCCCTCAGACCTATCTGAGGGTTCTGGGCTGACACTGGCCTCACTGGCCGTGG
GGGAGATGGGCACGGTTCTGCCAGTACTGTAGATCCCCCTCCCTCACGTAACCCAGCAACACACACACTGGCTCTGGGGC
AGCCACTGGGTCCCTCATAACAGGTGGAGGAGAAAAAGGAGAGAGTCCTTGTCTAGGGAGGGGGGAGGAGAGACACACCC
TGGCCACCTCCCGACCCATGCCCTGACTGTCCCCCACCTCCAGGGCCACCGCCGACTCTGCTTCCCCCAGTTCCTGAGGA
AGATGGGGGCAAGAGGACCACGCTCTCTGCCTGTCCGTACCCCCGCCCTGGCTGCTTTTCCCCTTTTCTTTGTTCTTGGC
TCCCCTGTTCCCTCCCTCAGTTCCAGAGACTCGTGGGAGGAGCTGCCACAGGCCTCCCTGTTTGAAGCCGGCCCTTGTCC

compX_cY_seqZ (since release2014 cX_gY_iZ)

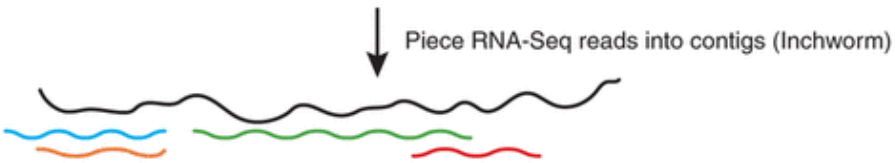
compX defines the graphical component generated by Chrysalis (from clustering inchworm contigs).
Butterfly might tease subgraphs apart from each other within a single component, based on the read support data . This gives rise to subgraphs (**cY**).
Each subgraph then gives rise to path sequences (**seqZ**).

Results

```
GTTCGAGGACCTGAATAAGCGCAAGGACACCAAGGAGATCTACACGCACTTCACGTGCGCCACCGACACCAAGAACGTGC
GTTCGAGGACCTGAATAAGCGCAAGGACACCAAGGAGATCTACACGCACTTCACGTGCGCCACCGACACCAAGAACGTGC

AGTTTGTGTTTGATGCCGTCACCGACGTCATCATCAAGAACAACCTGAAGGACTGCGGCCTCTTCTGAGGGGCAGCGGGG
AGTTTGTGTTTGATGCCGTCACCGACGTCATCATCAAGAACAACCTGAAGGACTGCGGCCTCTTCTGAGGGGCAGCGGGG

CCTGGCAGGATGG----------------------------------------------------------------
CCTGGCAGGATGGTGAGCCCGGGGTGGAGCGGAGCAGAGCTGTGGAGCCCAGAGAAGGGAGCGGTGGGGGCTGGGGTGGG

----------------------------------------------------------------------------
CCGTGGTGGGGGTATGGTGGTAGAGTGGTAGGTCGGTAGGACGACCTGAGGGGCATGGGCACACGGATAGGCCGGGCCGG

----------------------------------------------------------------------------
GGCCCAGATGGCAGAAGCATCCGGCCGTGCGCCGGGAGACAACGGAATGGCTGTCCTGACCACCCTTGGAGAAAGCTTAC

----------------------------------------------------------------------------
CGGCTCTGTGCTCAGCCCTGCAGTCTTTCCCTCAGACCTATCTGAGGGTTCTGGGCTGACACTGGCCTCACTGGCCGTGG

----------------------------------------------------------------------------
GGGAGATGGGCACGGTTCTGCCAGTACTGTAGATCCCCCTCCCTCACGTAACCCAGCAACACACACACTGGCTCTGGGGC

----------------------------------------------------------------------------
AGCCACTGGGTCCCTCATAACAGGTGGAGGAGAAAAAGGAGAGAGTCCTTGTCTAGGGAGGGGGGAGGAGAGACACACCC

-------------------------------------------GCCACCGCCGACTCTGCTTCCCCCAGTTCCTGAGGA
TGGCCACCTCCCGACCCATGCCCTGACTGTCCCCCACCTCCAGGGCCACCGCCGACTCTGCTTCCCCCAGTTCCTGAGGA

AGATGGGGGCAAGAGGACCACGCTCTCTGCCTGTCCGTACCCCCGCCCTGGCTGCTTTTCCCCTTTTCTTTGTTCTTGGC
AGATGGGGGCAAGAGGACCACGCTCTCTGCCTGTCCGTACCCCCGCCCTGGCTGCTTTTCCCCTTTTCTTTGTTCTTGGC

TCCCCTGTTCCCTCCCTCAGTTCCAGAGACTCGTGGGAGGAGCTGCCACAGGCCTCCCTGTTTGAAGCCGGCCCTTGTCC
TCCCCTGTTCCCTCCCTCAGTTCCAGAGACTCGTGGGAGGAGCTGCCACAGGCCTCCCTGTTTGAAGCCGGCCCTTGTCC
```

Results

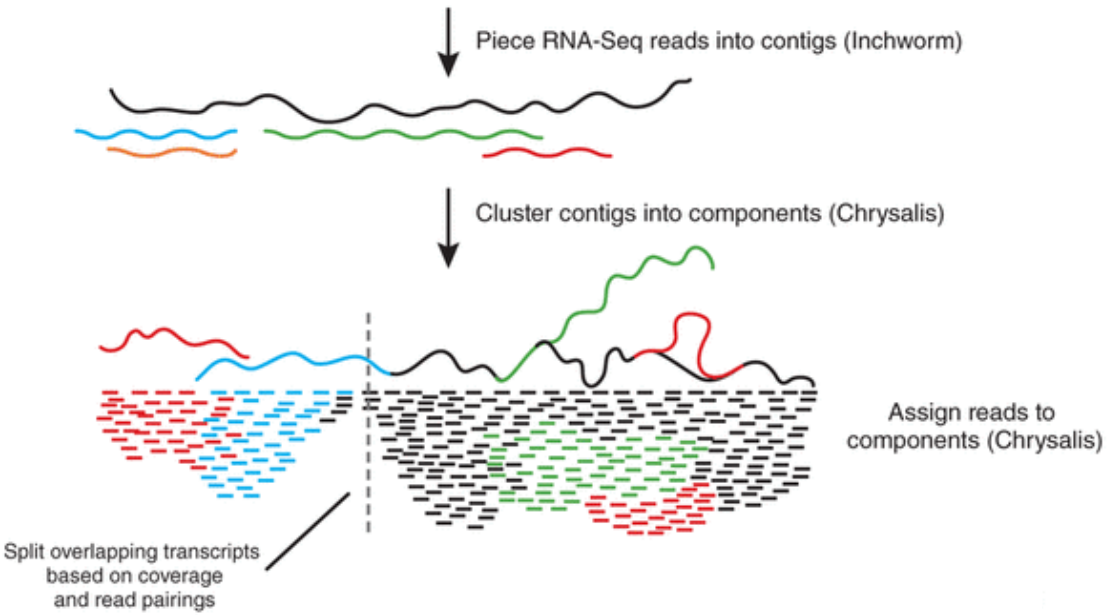
```
GTTCGAGGACCTGAATAAGCGCAAGGACACCAAGGAGATCTACACGCACTTCACGTGCGCCACCGACACCAAGAACGTGC
GTTCGAGGACCTGAATAAGCGCAAGGACACCAAGGAGATCTACACGCACTTCACGTGCGCCACCGACACCAAGAACGTGC

AGTTTGTGTTTGATGCCGTCACCGACGTCATCATCAAGAACAACCTGAAGGACTGCGGCCTCTTCTGAGGGGCAGCGGGG
AGTTTGTGTTTGATGCCGTCACCGACGTCATCATCAAGAACAACCTGAAGGACTGCGGCCTCTTCTGAGGGGCAGCGGGG

CCTGGCAGGATGG----------------------------------------------------------------
CCTGGCAGGATGGTGAGCCCGGGGTGGAGCGGAGCAGAGCTGTGGAGCCCAGAGAAGGGAGCGGTGGGGGCTGGGGTGGG

----------------------------------------------------------------------------
CCGTGGTGGGGGTATGGTGGTAGAGTGGTAGGTCGGTAGGACGACCTGAGGGGCATGGGCACACGGATAGGCCGGGCCGG

----------------------------------------------------------------------------
GGCCCAGATGGCAGAAGCATCCGGCCGTGCGCCGGGAGACAACGGAATGGCTGTCCTGACCACCCTTGGAGAAAGCTTAC

----------------------------------------------------------------------------
CGGCTCTGTGCTCAGCCCTGCAGTCTTTCCCTCAGACCTATCTGAGGGTTCTGGGCTGACACTGGCCTCACTGGCCGTGG

----------------------------------------------------------------------------
GGGAGATGGGCACGGTTCTGCCAGTACTGTAGATCCCCCTCCCTCACGTAACCCAGCAACACACACACTGGCTCTGGGGC

----------------------------------------------------------------------------
AGCCACTGGGTCCCTCATAACAGGTGGAGGAGAAAAAGGAGAGAGTCCTTGTCTAGGGAGGGGGGAGGAGAGACACACCC

------------------------------------------GCCACCGCCGACTCTGCTTCCCCCAGTTCCTGAGGA
TGGCCACCTCCCGACCCATGCCCTGACTGTCCCCCACCTCCAGGGCCACCGCCGACTCTGCTTCCCCCAGTTCCTGAGGA

AGATGGGGGCAAGAGGACCACGCTCTCTGCCTGTCCGTACCCCCGCCCTGGCTGCTTTTCCCCTTTTCTTTGTTCTTGGC
AGATGGGGGCAAGAGGACCACGCTCTCTGCCTGTCCGTACCCCCGCCCTGGCTGCTTTTCCCCTTTTCTTTGTTCTTGGC

TCCCCTGTTCCCTCCCTCAGTTCCAGAGACTCGTGGGAGGAGCTGCCACAGGCCTCCCTGTTTGAAGCCGGCCCTTGTCC
TCCCCTGTTCCCTCCCTCAGTTCCAGAGACTCGTGGGAGGAGCTGCCACAGGCCTCCCTGTTTGAAGCCGGCCCTTGTCC
```



Summary

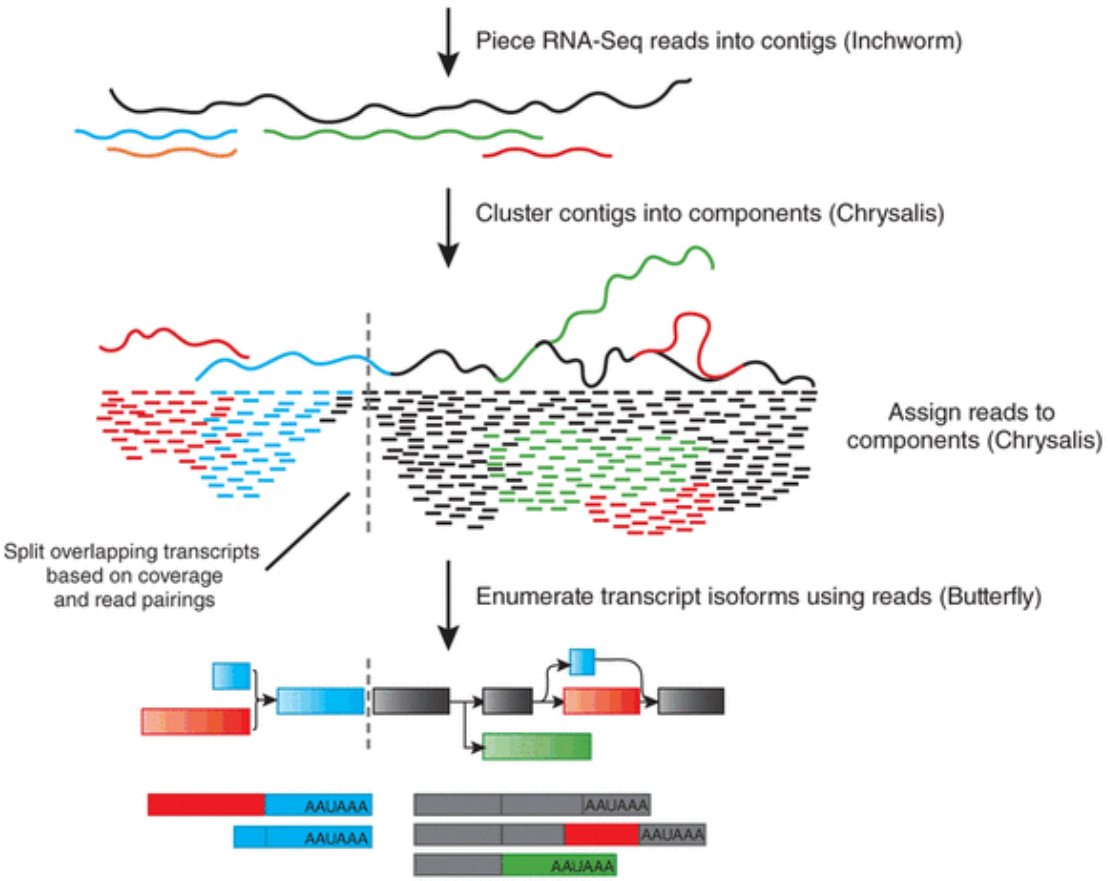
Iyer MK, Chinnaiyan AM (2011)
Nature Biotechnology **29**, 599–600

Summary



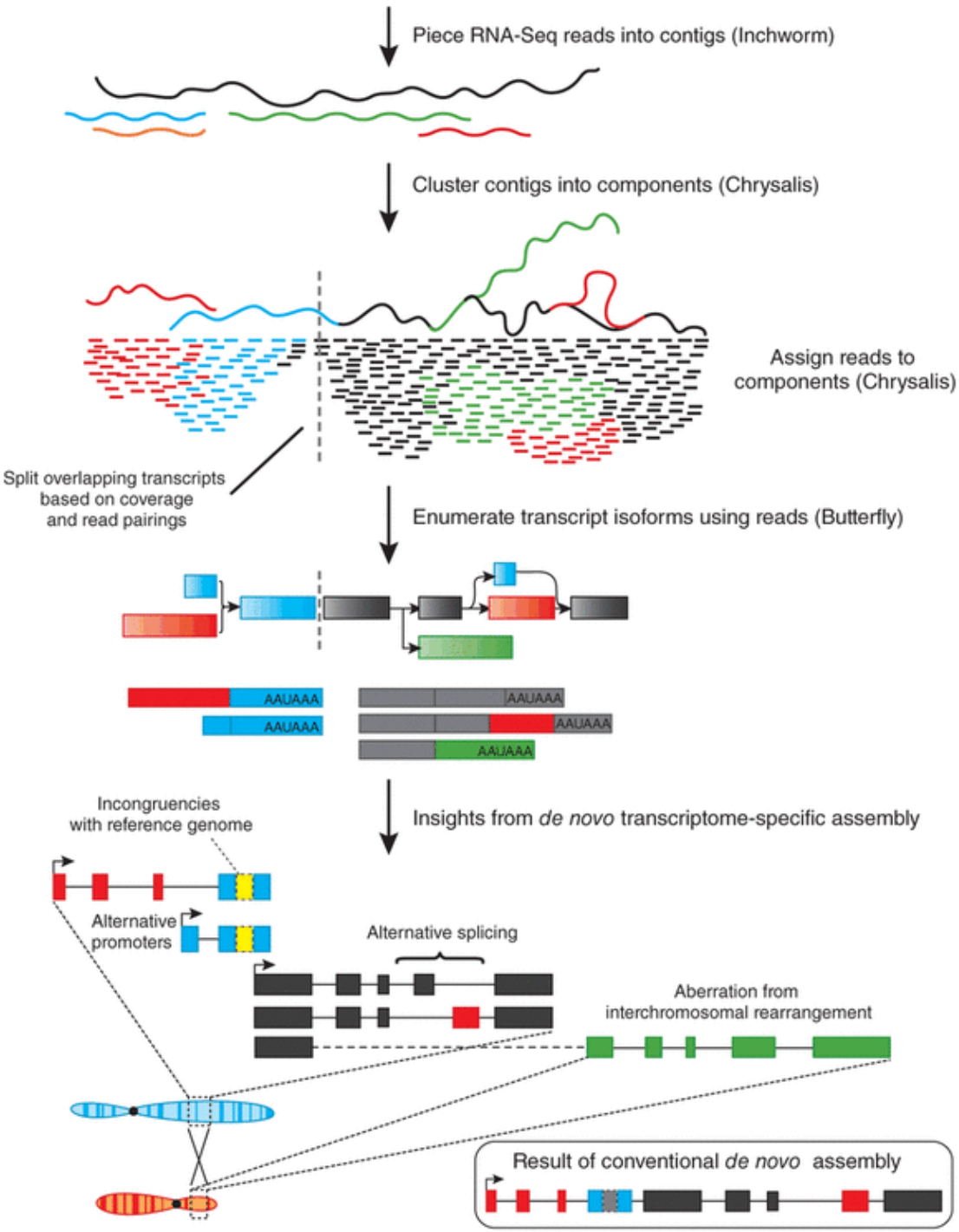
Iyer MK, Chinnaiyan AM (2011)
Nature Biotechnology **29**, 599–600

Summary



Iyer MK, Chinnaiyan AM (2011)
Nature Biotechnology **29**, 599–600

Summary



Iyer MK, Chinnaiyan AM (2011)
Nature Biotechnology **29**, 599–600

Trinity cons

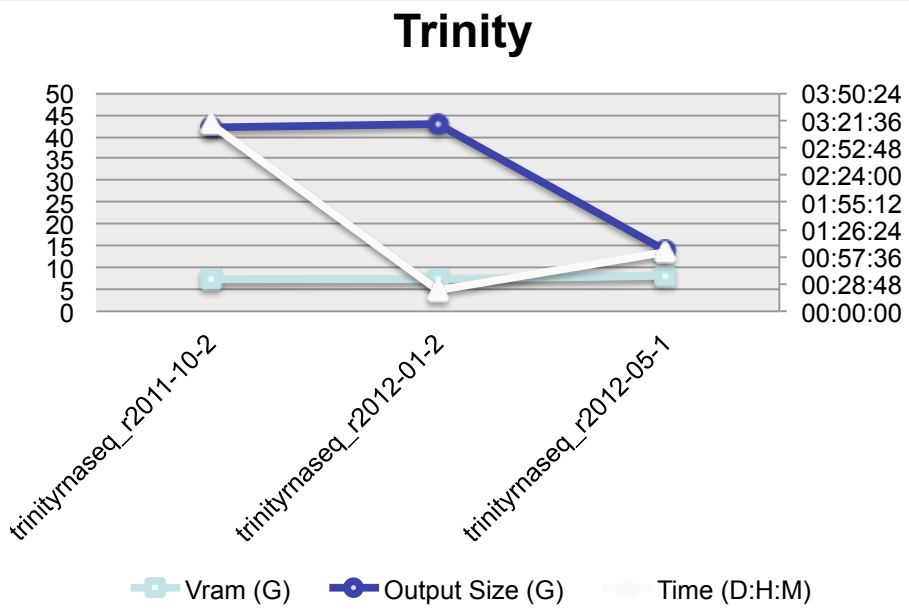
- Heuristic : re-running the assembly step lead to a different assembly
- Many transcripts (up to 300 000 for 30 000 genes)
  - Add a clustering step (uclust, cap 3)
  - Trinity «REDUCE» option : `Trinity.pl --bfly_opts " --REDUCE "`
  - Exclude the low coverage contigs (FPKM < 1)
- Frequent Version release : 18 versions in 1 ½ year. (4 v. 2013 and 2 v. 2014)

Cufflinks		TopHat		Trinity	
		02/11/12	2.06		
08/07/12	2.02	18/09/12	2.05		
15/06/12	2.0.1	21/06/12	2.04	05/10/12	trinityrnaseq_r2012-10-05
04/05/12	2.0.0	26/05/12	2.0.3	09/06/12	trinityrnaseq_r2012-06-08
02/01/12	1.3.0	23/05/12	2.0.2	19/05/12	trinityrnaseq_r2012-05-18
23/11/11	1.2.0	17/05/12	2.0.1	28/04/12	trinityrnaseq_r2012-04-27
08/09/11	1.1.0	09/04/12	2.0.0	23/04/12	trinityrnaseq_r2012-04-22-beta
01/06/11	1.0.3	02/02/12	1.4.1	20/03/12	trinityrnaseq_r2012-03-17
22/05/11	1.0.2	05/01/12	1.4.0	01/03/12	trinityrnaseq_r2012-01-25p1
06/05/11	1.0.1	16/10/11	1.3.3	27/01/12	trinityrnaseq_r2012-01-25
05/05/11	1.0.0	05/09/11	1.3.2	28/11/11	trinityrnaseq_r2011-11-26
30/11/10	0.9.3	23/06/11	1.3.1	29/10/11	trinityrnaseq_r2011-10-29
26/10/10	0.9.2	02/06/11	1.3.0	20/08/11	trinityrnaseq_r2011-08-20
03/10/10	0.9.1	18/01/11	1.2.0	17/08/11	trinityrnaseq_r2011-08-15-p1
27/09/10	0.9.0	16/11/10	1.1.4	16/08/11	trinityrnaseq_r2011-08-15
30/06/10	0.8.3	13/11/10	1.1.3	13/07/11	trinityrnaseq_r2011-07-13
26/03/10	0.8.2	26/10/10	1.1.2	20/05/11	trinityrnaseq-r20110519
13/02/10	0.8.1	11/10/10	1.1.1	13/05/11	trinityrnaseq_r2011-05-13
05/02/10	0.8.0	03/10/10	1.1.0	25/04/11	trinityrnaseq-2011_04_24
26/09/09	0.7.0	27/10/08	0.7.0	13/03/11	trinityrnaseq-03122011

Trinity pro

- Very active community (optimized very often) last release : 2014-04-23

2011	Jan	Feb	Mar (10)	Apr (41)	May (35)	Jun (56)	Jul (65)	Aug (125)	Sep (72)	Oct (82)	Nov (93)	Dec (164)
2012	Jan (252)	Feb (255)	Mar (172)	Apr (161)	May (192)	Jun (167)	Jul (86)	Aug (160)	Sep (220)	Oct (152)	Nov (161)	Dec (109)
2013	Jan (166)	Feb (201)	Mar (325)	Apr (213)	May (149)	Jun (117)	Jul (165)	Aug (133)	Sep (178)	Oct (164)	Nov (106)	Dec (71)
2014	Jan (211)	Feb (138)	Mar (193)	Apr (267)	May (274)	Jun (51)	Jul	Aug	Sep	Oct	Nov	Dec



Trinity pro

- Integration cleanning step (trimomatic)

- Integration of normalization steps
  - Accept reads when its average kmer coverage does not exceed a defined threshold
  - Removes reads with too much variability in kmer coverage

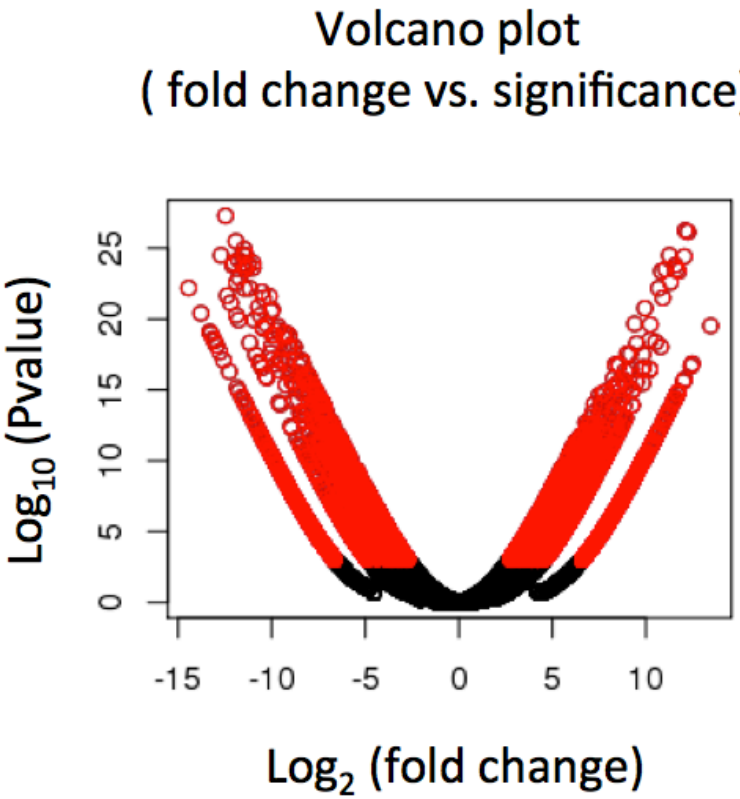
- Integration of DE step

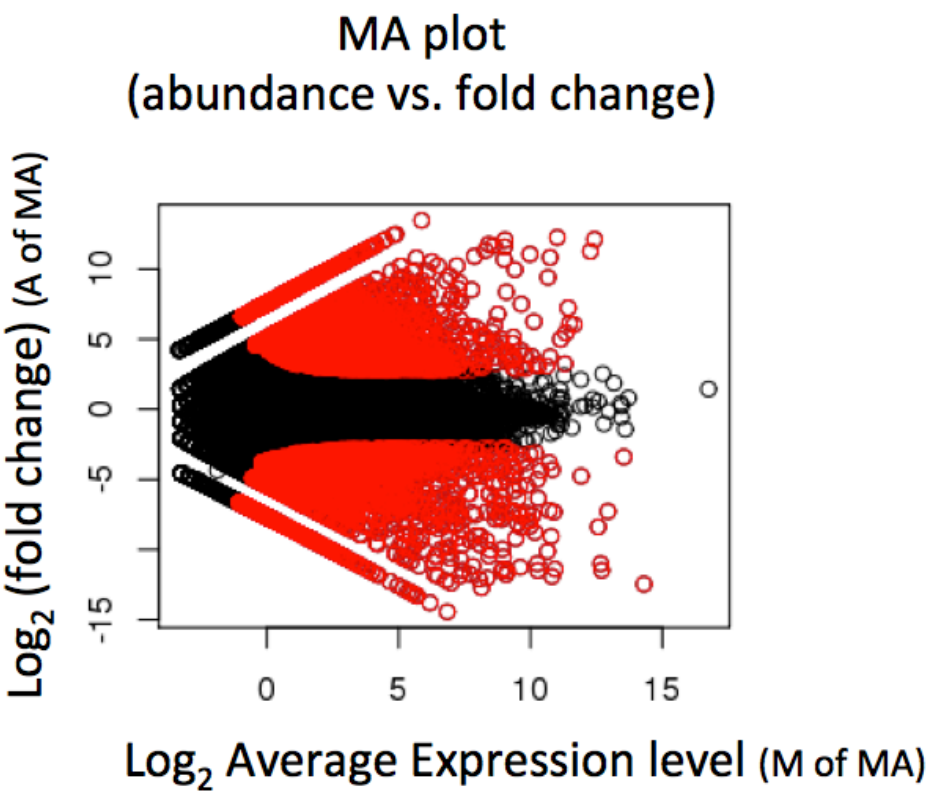
- Integration of annotation step (trinotate)

- Using the reference genome (PASA)

- Multiple kmer choice (in progress)

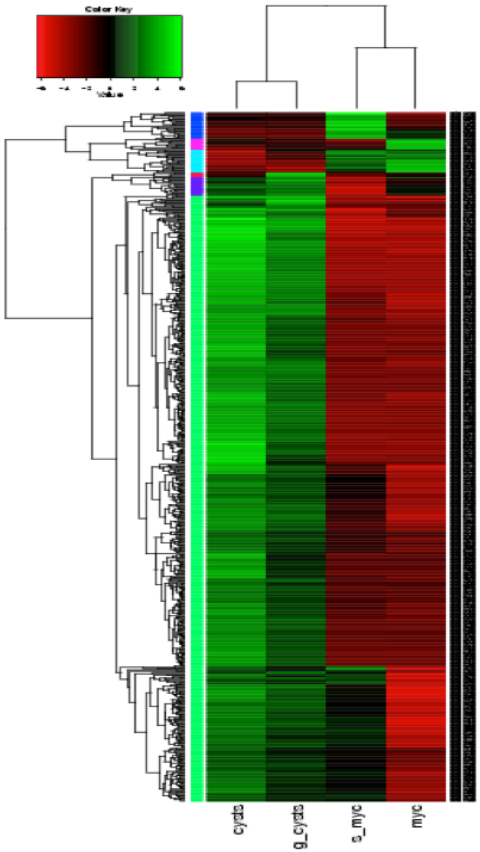
Additional Trinity scripts : Plotting Pairwise Differential Expression Data





Significantly differently expressed transcripts have FDR <= 0.001 (shown in red)

Additional Trinity scripts : Comparing Multiple Samples



Heatmaps provide an effective tool for navigating differential expression across multiple samples.

Clustering can be performed across both axes:

-cluster transcripts with similar expression patterns.

-cluster samples according to similar expression values among transcripts.

Trinity …. and friends

Other *de novo* assemblers

Single K-mer: SOAPdenovo, ABySS, Oases and Trinity

Multiple K-mer: SOAPdenovo-MK, trans-ABySS and Oases-MK, clc-denovo-assembly.

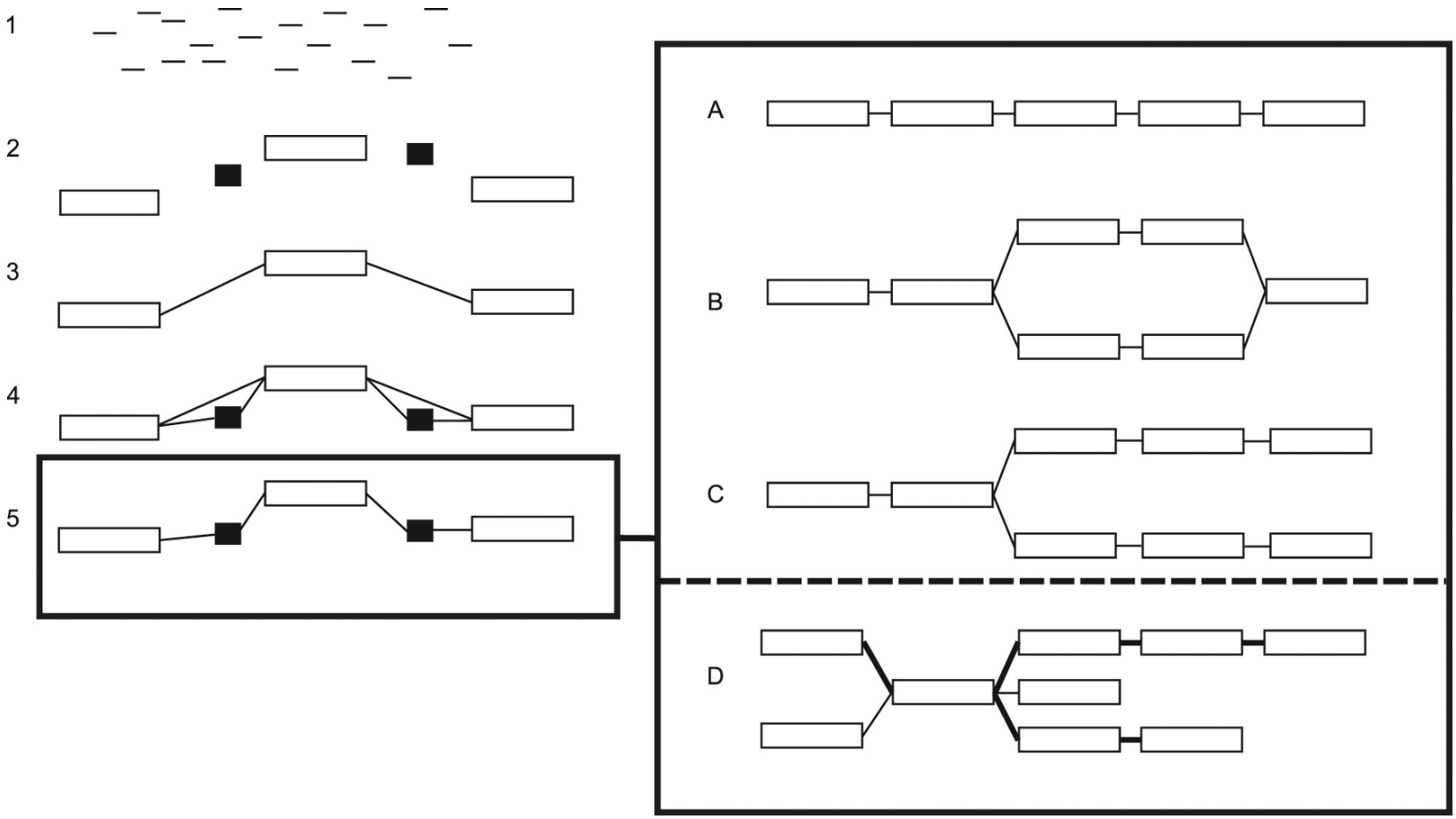
Qiong-Yi Zhao et al., Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study. BMC Bioinformatics 2011, 12(Suppl 14):S2

Clarke, K., Yang, Y., Marsh, R., Xie, L., & Zhang, K. K. (2013). Comparative analysis of de novo transcriptome assembly. Science China Life Sciences, 56(2), 156–162. doi:10.1007/s11427-013-4444-x

Chen, G., Yin, K., Wang, C., & Shi, T. (n.d.). De novo transcriptome assembly of RNA-Seq reads with different strategies. Science China Life Sciences, 54(12), 1129–1133. doi:10.1007/s11427-011-4256-9

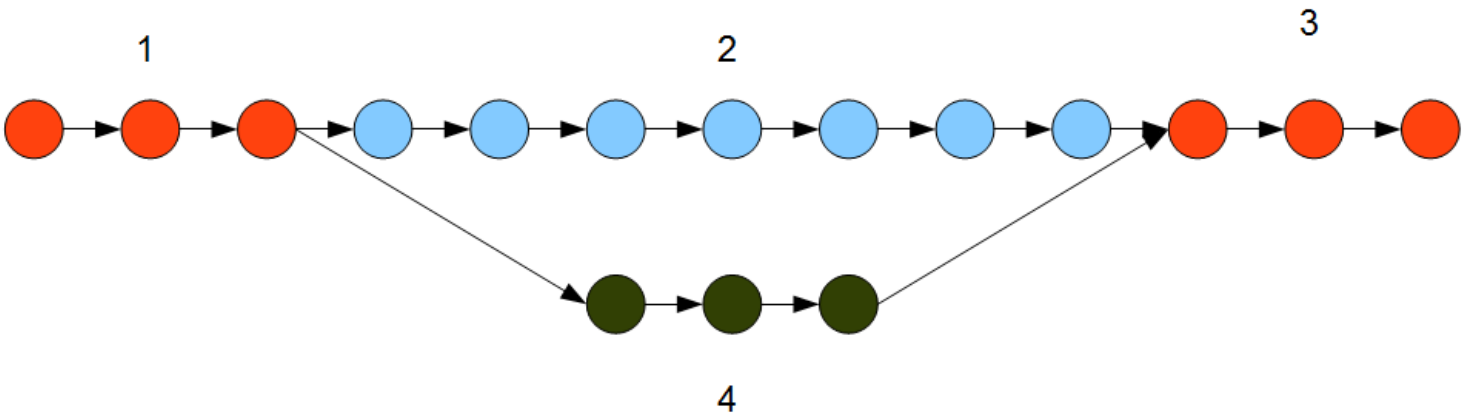
Velvet/Oases

Schematic overview of the Oases pipeline: (1) Individual reads are sequenced from an RNA sample; (2) Contigs are built from those reads, some of them are labeled as long (clear), others short (dark); (3) Long contigs, connected by single reads or read-pairs are grouped into connected components called loci; (4) Short contigs are attached to the loci; and (5) The loci are transitively reduced.



Schulz M H et al. Bioinformatics 2012;28:1086-1092

Velvet/Oases vs Trinity



Velvet searches for connectivity in a de Bruijn graph using a depth search module. The search for a contig stops, as soon as a junction is reached in the de Bruijn graph. So, in the example graph presented above, **Velvet will identify the branches 1, 2, 3 and 4 as separate contigs. The role of Oases is to connect all those separate contigs** to build the gene structures.

Inchworm is different, because it runs a greedy algorithm that connects as many k-mers as possible without placing any k-mer into two separate contigs. Inchworm does not stop at junctions, but continues forward with assembling contigs. For the graph presented in the above picture, **Inchworm will identify 1+2+3 as one contig and 4 as a different contig**.

Velvet/Oases : Kmer size effect

Kmer size effect

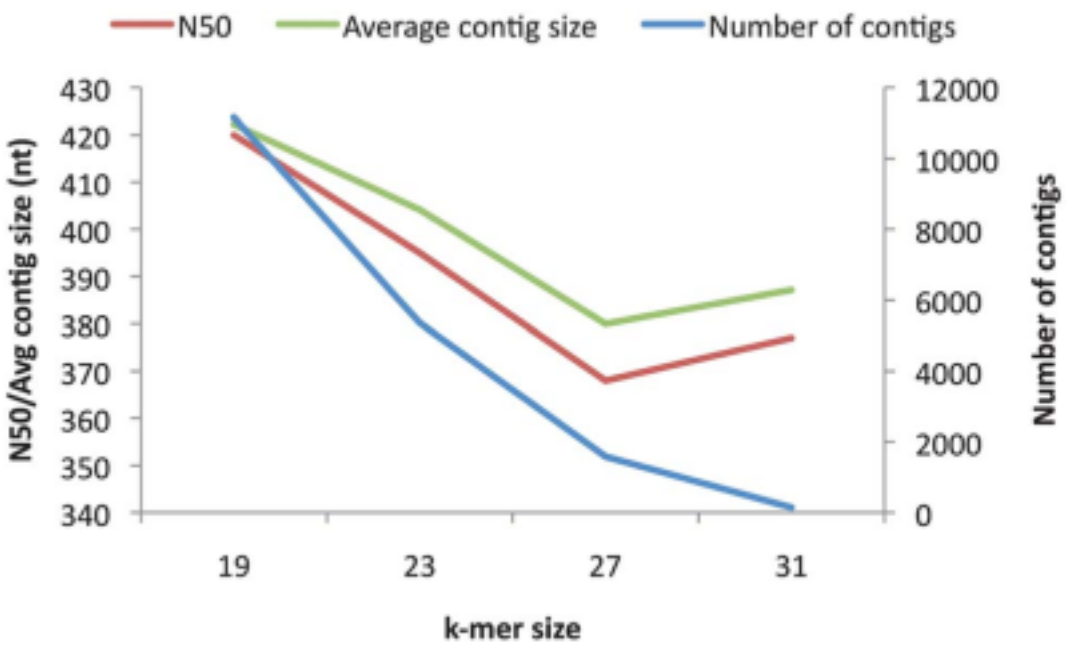


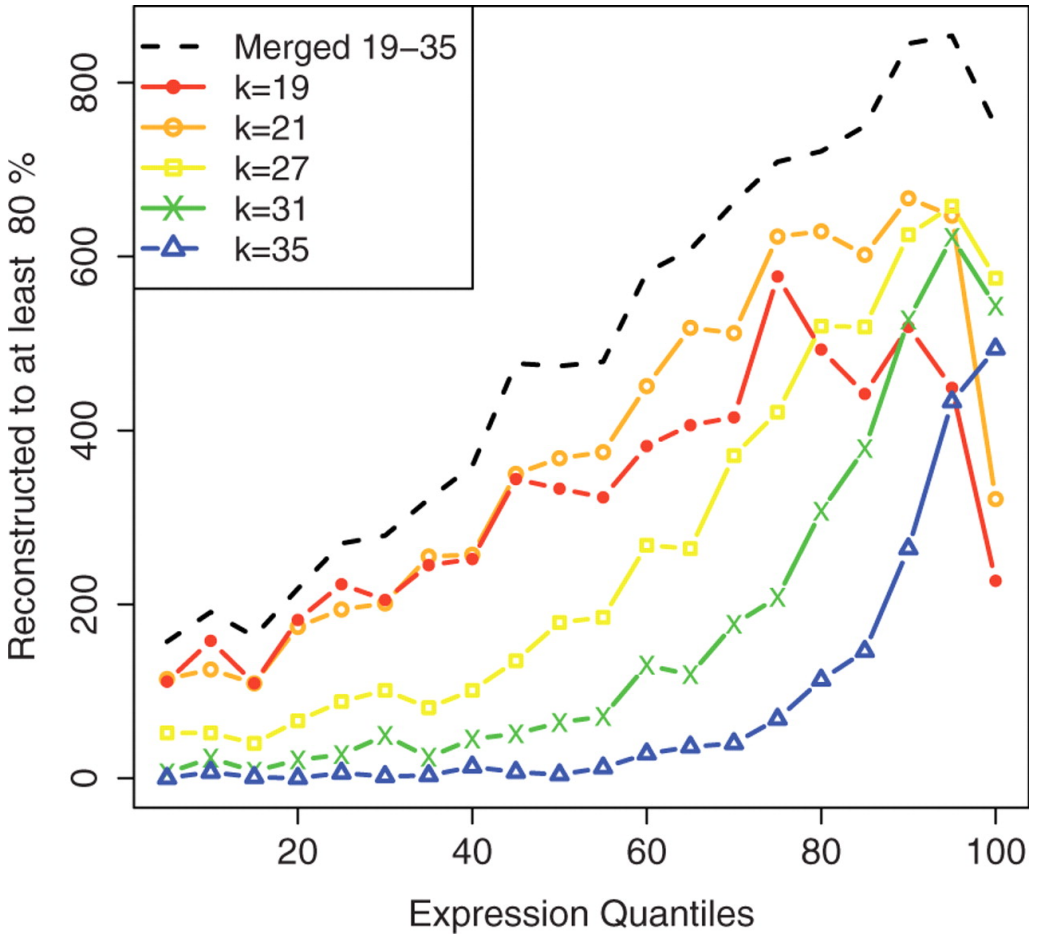
Fig. 2 Effect of the k-mer length on the assembly. A library of single end reads (read length = 36 bp) was assembled using an array of k-mers (19, 23, 27 and 31). The number of contigs, N50 contig size and average contig size decrease as the k-mer length begins to approximate the read length. The read length is broken into fewer sub-sequences as the k-mer length is increased, thereby resulting in fewer connections between the k-mers and creation of less contigs. For instance, the assembly using a k-mer length of 31 resulted in 140 total contigs.

Velvet/Oases : single kmer vs merged

Comparison of single k-mer Oases assemblies and the merged assembly from kMIN=19 to kMAX=35 by Oases-M, on the human dataset.

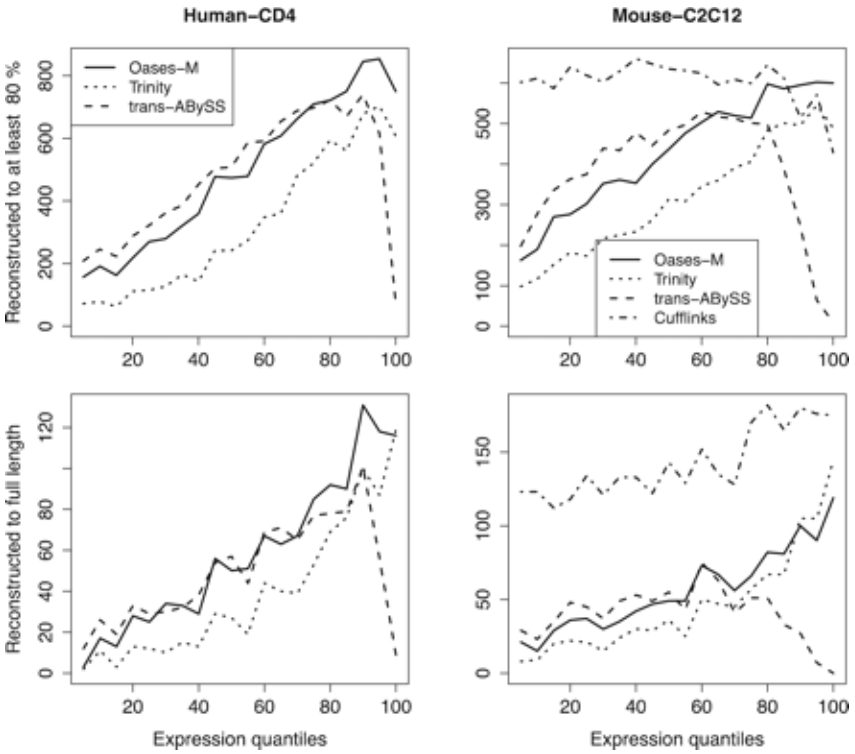
The total number of Ensembl transcripts assembled to 80 of their length is provided by RPKM gene expression quantiles of 1464 genes each.

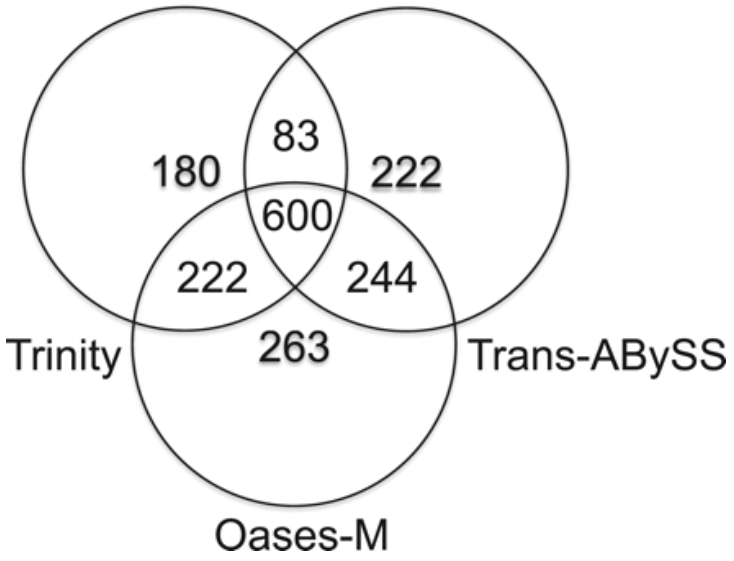
As expected, the assemblies with longer k-values perform best on high expression genes, but poorly on low expression genes. However, short k-mer assemblies have the disadvantage of introducing misassemblies



Schulz M H et al. Bioinformatics 2012;28:1086-1092

Assembler comparison





Schulz M H et al. Bioinformatics 2012;28:1086-1092

« In summary, no assembler had consistent good performance in all the statistics. For transcriptome assembly of prokaryotic cells that have simple gene structure, Trinity would be recommended. For eukaryotic genome, both Oases and Trinity gave acceptable performance. »

Clarke, K., Yang, Y., Marsh, R., Xie, L., & Zhang, K. K. (2013). Comparative analysis of de novo transcriptome assembly. Science China Life Sciences, 56(2), 156–162.

Transcriptome assembly

ASSEMBLY QUALITY ASSESSMENT

Assembly quality assessment

3 tracks :

- Assembly metrics
- Shape of contig length histogram
- Reads mapping back rate

Metrics

The possible metrics derived from genome assembly:

- Idea of global size (# bases)
- Idea of number of elements (# contigs/ scaffolds)
- Idea of compactness (N50):
  - much more difficult to predict with transcriptome data

Vocabulary

- **Contig**: a set of overlapping segments that together represent a consensus sequence

- **Scaffold**: a series of contigs that are in the right order but not necessarily connected in one continuous stretch of sequence

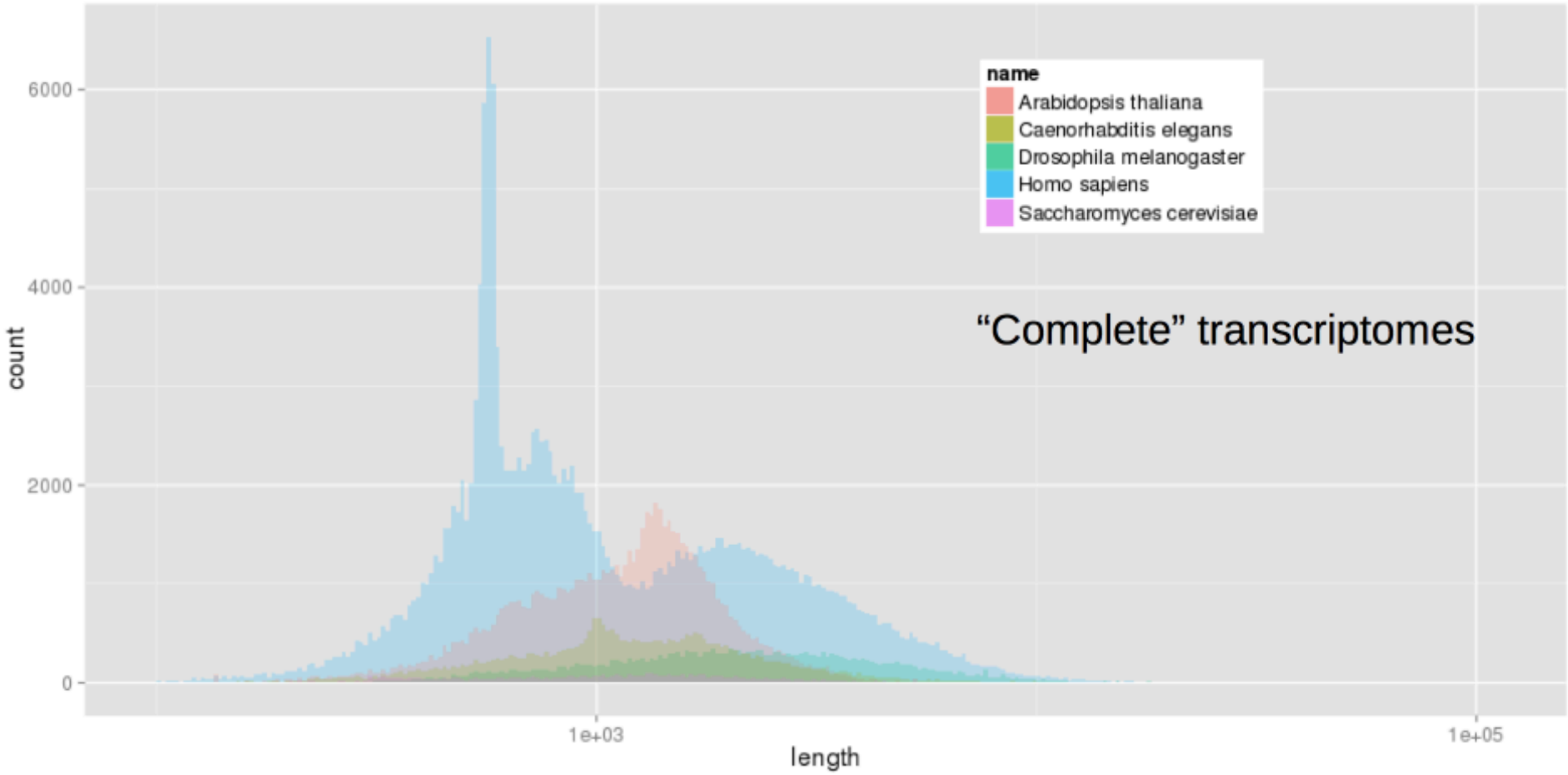
- **N50**: given a set of contigs of varying lengths, the N50 length is defined as the length N for which 50% of all bases in the contigs are in contigs of length L < N

contig size list L = (2, 2, 2, 3, 3, 4, 8, 8)
we have 50% of total length (16/32) above 4 **N50** is equal to 4+8/2 = **6**

- **L50**: number of contigs that are greater than, or equal to, the N50 length

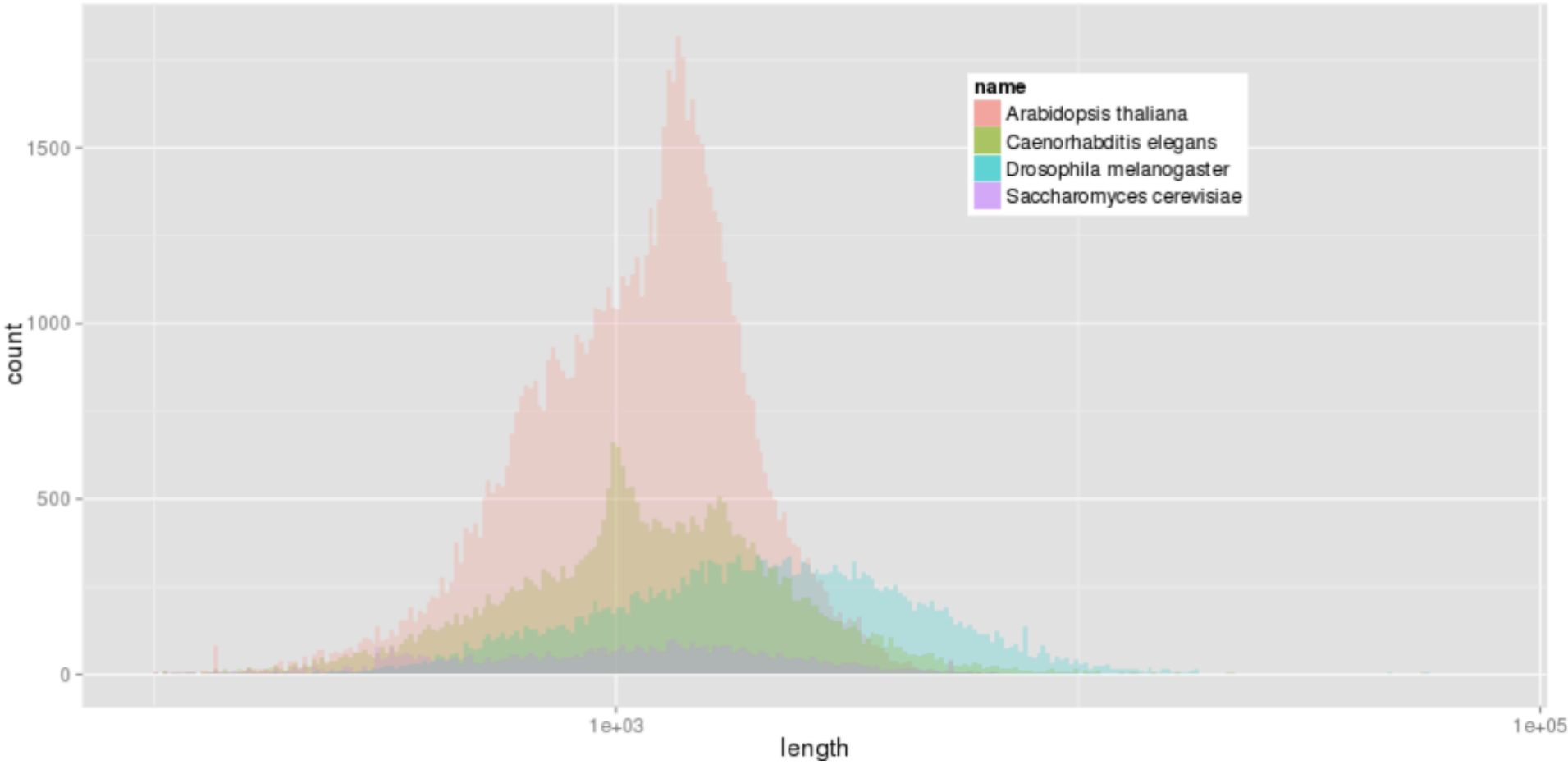
Transcripts lenght histogram

Transcript lengths are not randomly distribute
We should get a known distribution shape



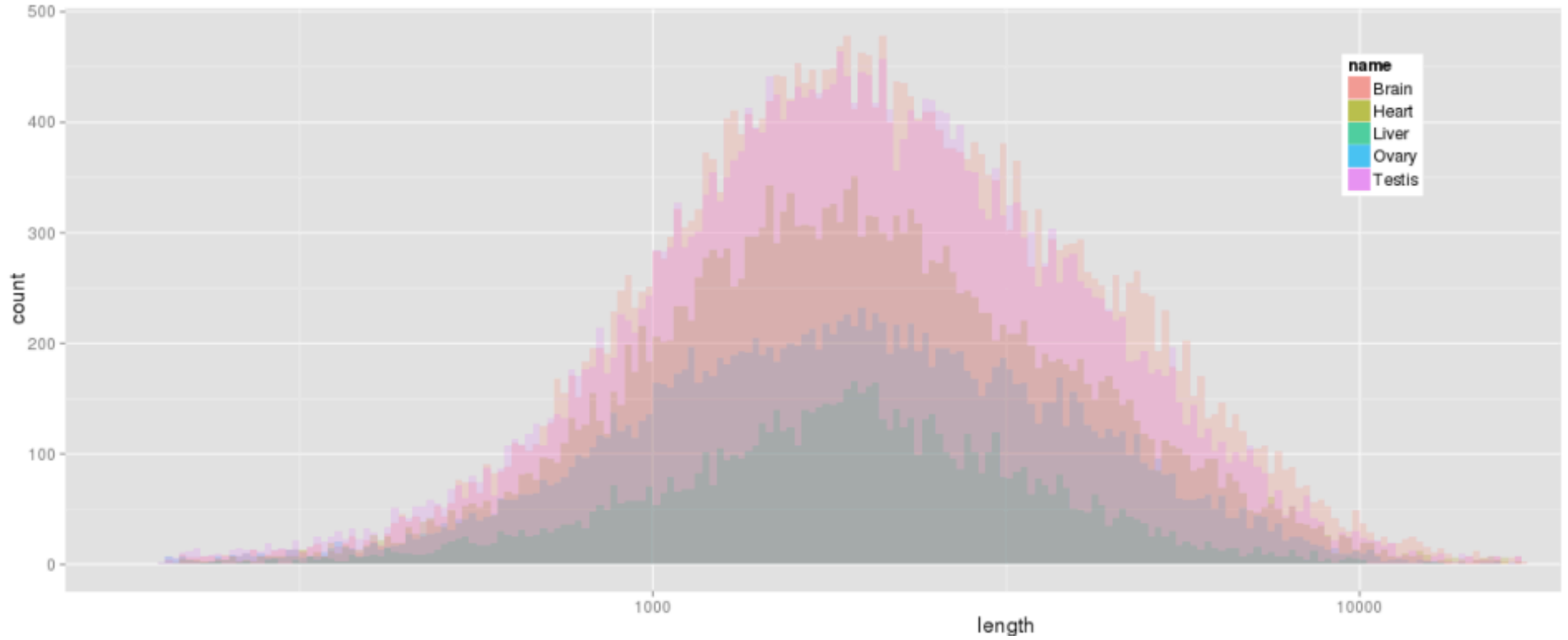
Transcripts lenght histogram

RNAseq data



Transcripts lenght histogram

Zebrafish tissue specific assembled transcriptomes : not so different



Realignment metrics

The assembly is a sum-up. The realignment rate gives how much of the initial information is inside the contigs.

Reads mapped back to transcripts (RMBT)

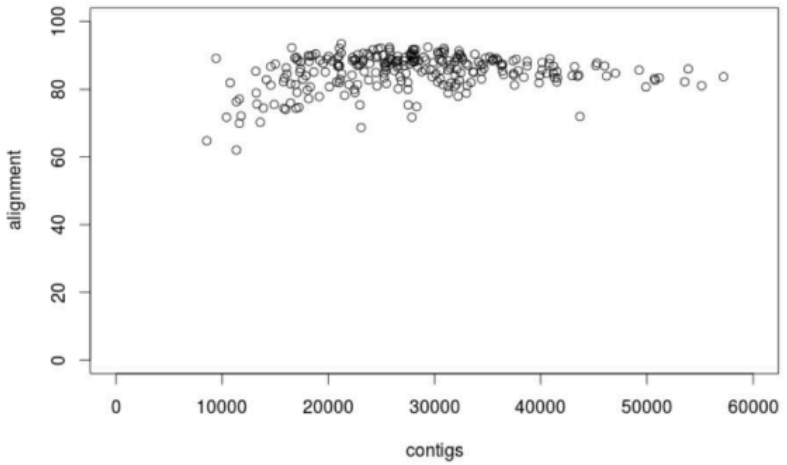
- align reads against assembly generated transcripts
- compute percentage of reads mapped

Realignment metrics

Factors affecting realignment rate:

- Presence of highly expressed genes
- Contamination by building blocks (adaptors)
- Reads quality

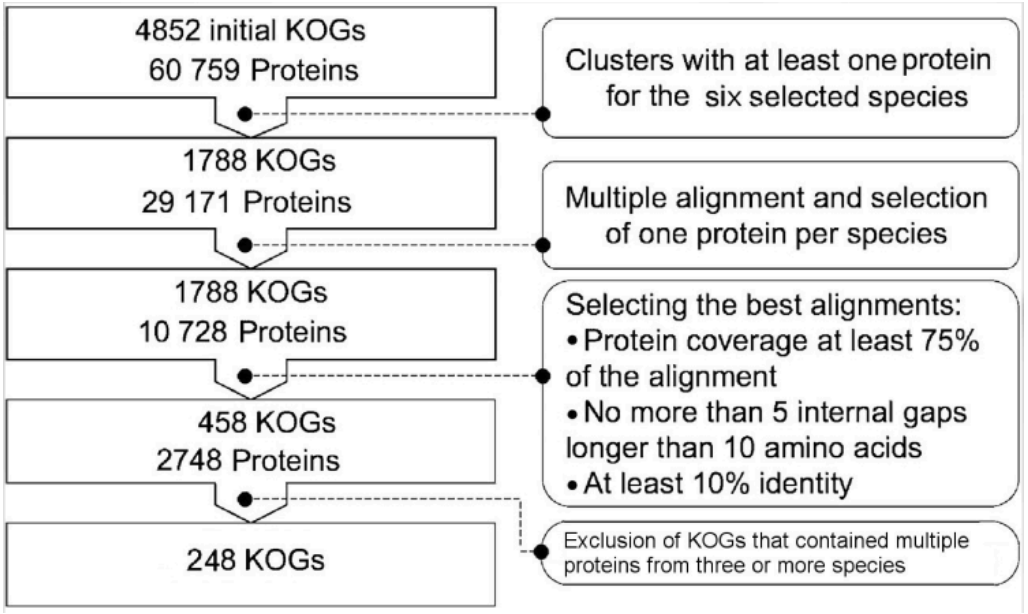
Should be higher or around 80% of mapped reads



CEGMA analysis

Core **E**ukaryotic **G**enes **M**apping **A**pproach

Mapping a set of conserved protein families that occur in a wide range of eukaryotes onto assembly to assess completeness .
A set of eukaryotic core proteins (KOG = euKaryotic Orthologous Groups) from 6 species: H. sapiens, D. melanogaster, C. elegans, A. thaliana, S. cerevisiae, S.pombe



First set of 458 core genes

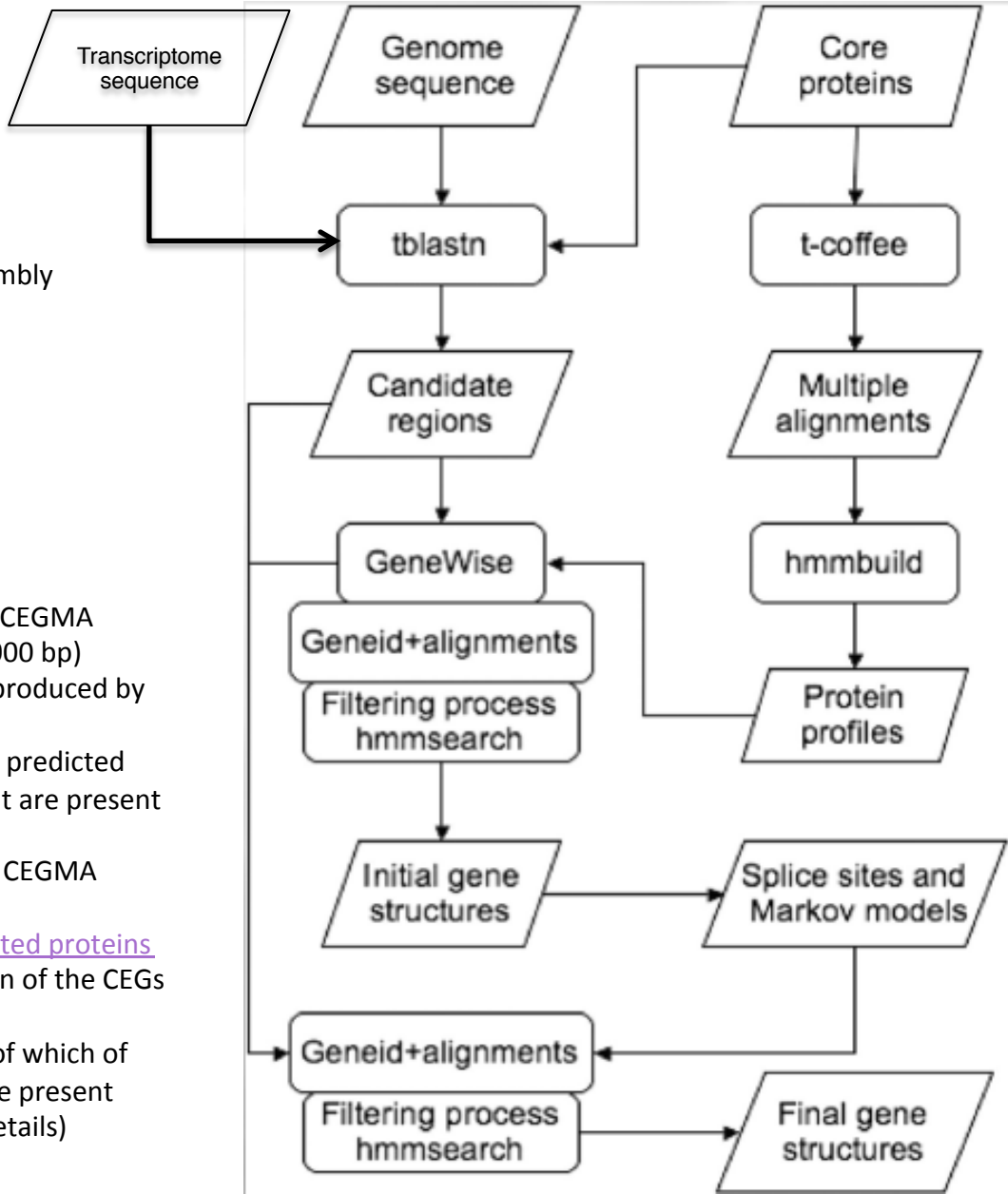
First set of 248 core genes with less paralogs

Genis Parra, Keith Bradnam and Ian Korf. CEGMA: a pipeline to accurately annotate core genes in eukaryotic genomes. » . Bioinformatics, 23: 1061-1067 (2007)
Genis Parra, Keith Bradnam, Zemin Ning, Thomas Keane, and Ian Korf. Assessing the gene space in draft genomes » . Nucleic Acids Research, 37(1): 298-297 (2009)

CEGMA analysis

Mapping on assembly

- protein profiles are built from set of core protein
- profiles are aligned on candidate regions from assembly
- the final structure of the gene is refined
- count of profiles which are found



7 output files :

- **output.cegma.dna** - contains DNA sequence of each CEGMA prediction along with flanking DNA (defaults to Â± 2000 bp)
- **output.cegma.errors** - contains any error messages produced by all of the CEGMA scripts
- **output.cegma.fa** - contains protein sequences of the predicted CEGs. One protein for each of the 458 core genes that are present in your genome
- **output.cegma.gff** - contains exon details of all of the CEGMA predicted genes
- **output.cegma.id** - contains the **KOG** IDs for the selected proteins
- **output.cegma.local.gff** - contains the GFF information of the CEGs using local coordiantes (relative to the dna file)
- **output.completeness_report** - contains a summary of which of the subset of the 248 most highly-conserved CEGs are present (either partially or completely, see below for more details)

CEGMA analysis : Output example (output.completeness_report)

- Complete (70% of the protein length)
- Partial (not matching “complete” criteria but exceed a pre-computed alignment score)

```
#      Statistics of the completeness of the genome based on 248 CEGs      #

              #Prots  %Completeness  -  #Total  Average  %Ortho

  Complete      245       98.79      -   593     2.42     64.90

   Group 1       66      100.00      -   146     2.21     60.61
   Group 2       56      100.00      -   129     2.30     60.71
   Group 3       58       95.08      -   140     2.41     67.24
   Group 4       65      100.00      -   178     2.74     70.77

   Partial      245       98.79      -   631     2.58     67.76

   Group 1       66      100.00      -   152     2.30     62.12
   Group 2       56      100.00      -   142     2.54     64.29
   Group 3       58       95.08      -   148     2.55     68.97
   Group 4       65      100.00      -   189     2.91     75.38

#    These results are based on the set of genes selected by Genis Parra   #

#    Key:                                                                  #
#    Prots = number of 248 ultra-conserved CEGs present in genome          #
#    %Completeness = percentage of 248 ultra-conserved CEGs present        #
#    Total = total number of CEGs present including putative orthologs     #
#    Average = average number of orthologs per CEG                         #
#    %Ortho = percentage of detected CEGS that have more than 1 ortholog   #
```

A Trinity alternative

BlastX of Trinity.fasta against uniprot
Script Trinity : *analyze_blastPlus_topHit_coverage.pl*

hit_pct_cov_bin	count_in_bin	>bin_below
100	3242	3242
90	268	3510
80	186	3696
70	202	3898
60	216	4114
50	204	4318
40	164	4482
30	135	4617
20	76	4693
10	0	4693
0	0	4693

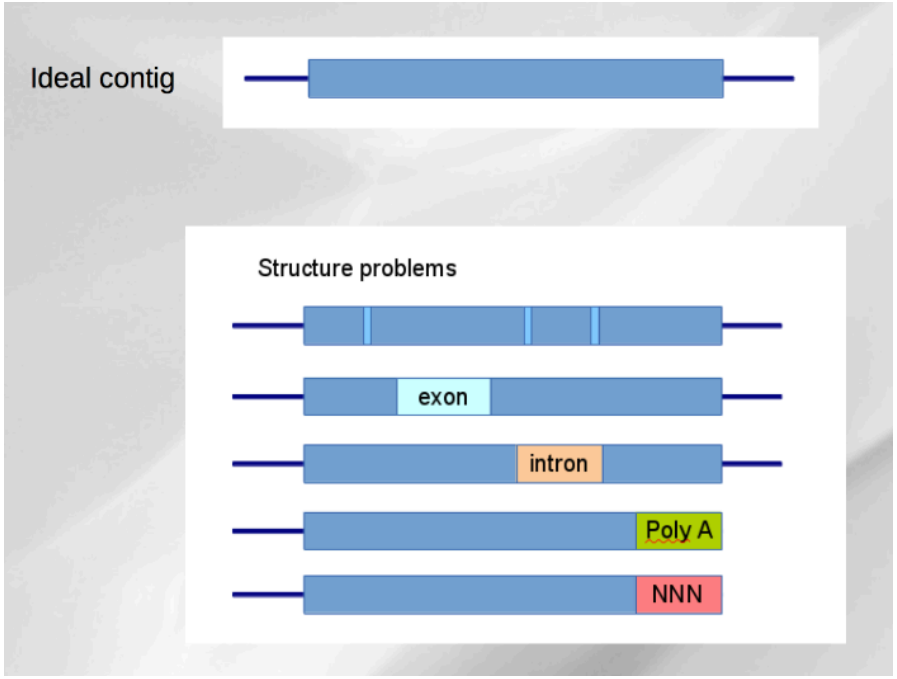
- There are 268 proteins that each match a Trinity transcript by >80% and ⇐ 90% of their protein lengths.
- There are 3510 proteins that are represented by nearly full-length transcripts, having >80% alignment coverage.
- There are 3242 proteins that are covered by more than 90% of their protein lengths.

Transcriptome assembly

CLEANING THE ASSEMBLY

Cleaning the assembly

Transcripts



Proteins

- cleaning polyA tails, terminal N blocks, low complexity areas
- insertion/deletion correction using the alignment
- cis or trans-chimera detection
- low fold coverage filtering (graph data)
- low expression filtering
- possible filtering of contigs which do not have a long enough ORF (phylogenomy)

Transcriptome cleaning

- Remove remaining polyA tails
- Remove blocks of Ns located at the extremities
- Remove low complexity areas

→ **Seqclean:** a script for automated trimming and validation of ESTs or other DNA sequences by screening for various contaminants, low quality and low-complexity sequences.

- Finding frame-shifts :
- Insertion/deletion correction

→ - Going back to alignment reads vs transcripts to find INDEL
- Using a proteic reference to find frame-shifts

- Detect splice form

→ - Going back to alignment reads vs transcripts to find splice
- Isoforms alignments + reads
- Alignment against « close » reference genome

Transcriptome cleaning : Chimera

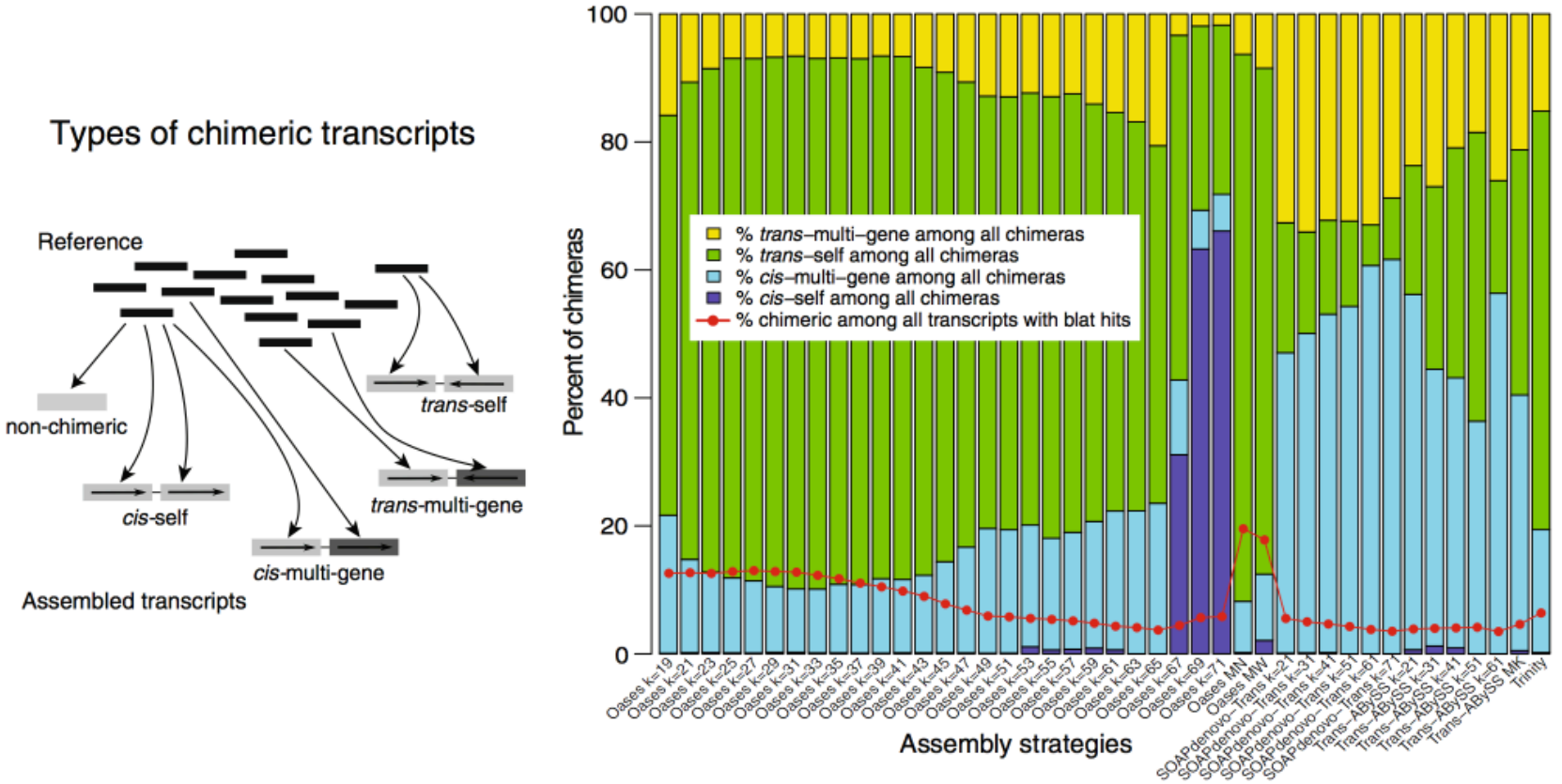


Figure 1 Chimera compositions among assembled transcripts before post-processing. Oases MN: Oases-M merging single *k*-mer assemblies of 21, 31, 41, 51 and 61; MW: Oases-M merging single *k*-mer assemblies of 19–71, with increment of 2; Trans-ABySS MK: Trans-ABySS merging single *k*-mer assemblies of 21, 31, 41, 51 and 61.

Majority of trans-self chimeras for small-middle k-mers
Majority of cis-self chimeras for large k-mers and oases merge
Chimeras increase with merging and small kmer

Without reference, cannot tackle multi-gene chimeras
Blast again itself

Yang, Y., & Smith, S. A. (2013). Optimizing de novo assembly of short-read RNA-seq data for phylogenomics. BMC Genomics, 14(1), 328.

Transcriptome cleaning : Redondancy

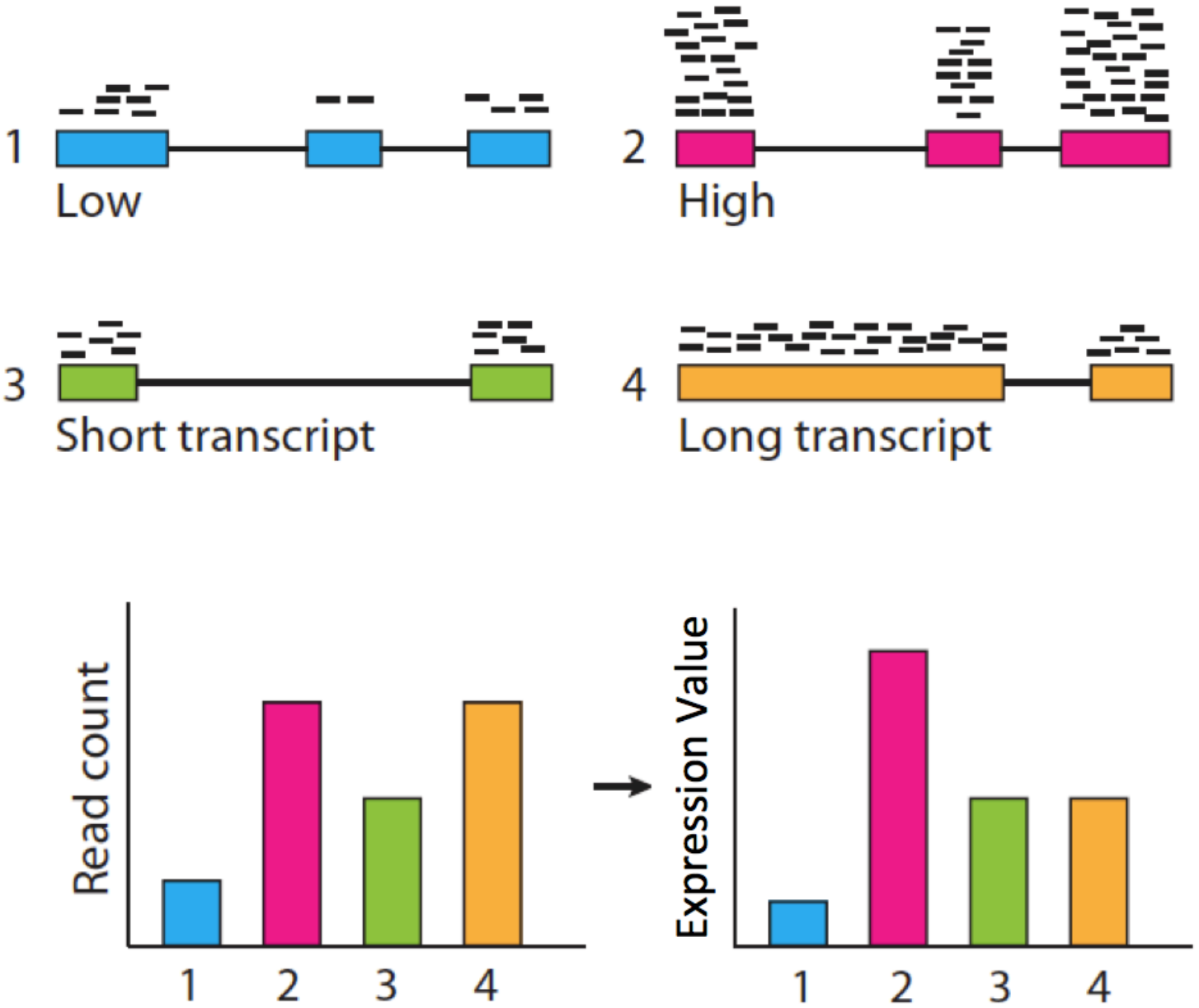
- Considere results at genes level

- Filtering base upon RPKM and % isoforms

« If you want to filter out the likely transcript artifacts and lowly expressed transcripts, you might consider retaining only those that represent at least 1% of the per-component (IsoPct) expression level. Because Trinity transcripts are not currently scaffolded across sequencing gaps, there will be cases where smaller transcript fragments may lack enough properly-paired read support to show up as *expressed*, but are still otherwise supported by the read data. Therefore, filter cautiously and we don't recommend discarding such lowly expressed (or seemingly unexpressed) transcripts, but rather putting them aside for further study »

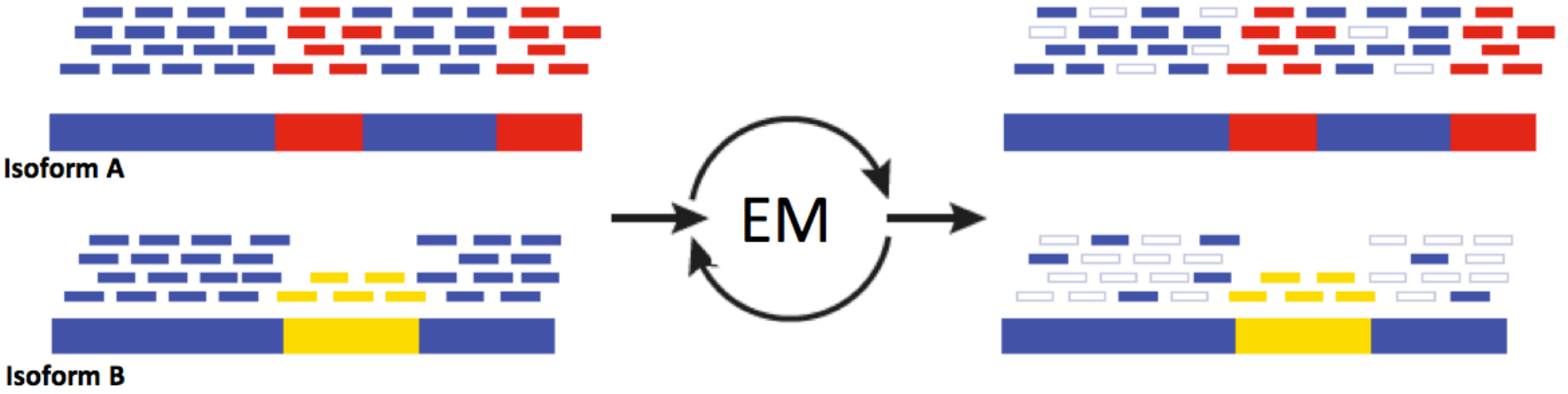
- CDHIT-EST + TGICL

Calulating Expression of genes and transcripts



Multiply-mapped Reads Confound Abundance Estimation

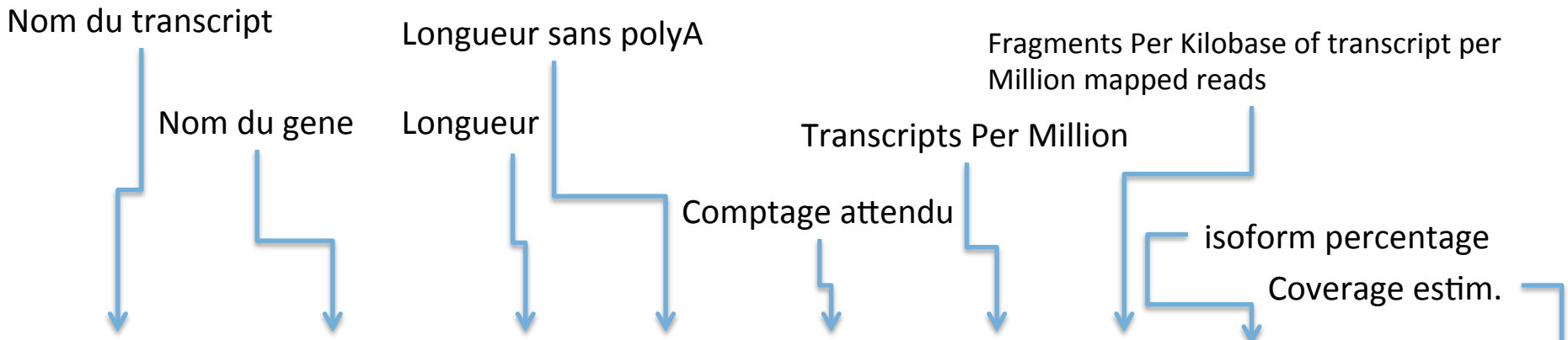
RSEM Count



Blue = multiply-mapped reads
Red, Yellow = uniquely-mapped reads

Use Expectation Maximization (EM) to find the most likely assignment of reads to transcripts.

RSEM.isoforms.results



transcript_id	gene_id	length	effective_length	expected_count	TPM	FPKM	IsoPct	cov
comp100000_c0_seq1	comp100000_c0	340	340	13.09	2.79	3.29	100	2.926
comp10000_c0_seq1	comp10000_c0	353	353	43.44	8.84	10.43	100	9.35252125
comp10001_c0_seq1	comp10001_c0	569	569	48.01	5.61	6.62	100	6.41258348
comp10002_c0_seq1	comp10002_c0	1563	1563	197.27	7.78	9.19	93.26	9.59214331
comp10002_c0_seq2	comp10002_c0	1563	1563	0	0	0	0	0
comp10002_c0_seq3	comp10002_c0	1087	1087	9.73	0.56	0.66	6.74	0.68029439
comp10002_c0_seq4	comp10002_c0	1087	1087	0	0	0	0	0
comp10004_c0_seq1	comp10004_c0	661	661	105.99	10.48	12.37	100	12.1864448
comp100058_c0_seq1	comp100058_c0	879	879	45	3.26	3.85	100	3.89078498
comp10005_c0_seq1	comp10005_c0	274	274	28	7.82	9.23	100	7.76642336
comp10006_c0_seq1	comp10006_c0	309	309	42	10.07	11.88	100	10.3300971
comp10007_c0_seq1	comp10007_c0	477	477	66	9.42	11.11	100	10.5157233
comp100094_c0_seq1	comp100094_c0	279	279	14	3.82	4.51	100	3.81362007
comp10009_c0_seq1	comp10009_c0	256	256	13.77	4.2	4.96	100	4.08796875
comp1000_c0_seq1	comp1000_c0	292	292	20	5.15	6.08	100	5.20547945

Because 1) each read aligning to this transcript has a probability of being generated from background noise; 2) RSEM may filter some alignable low quality reads, the sum of expected counts for all transcript are generally less than the total number of reads aligned.

transcript_id	gene_id	length	effective_length	expected_count	TPM	FPKM	IsoPct
c128_g0_i1	c128_g0	209	1.73	0.00	0.00	0.00	0.00
c13_g0_i1	c13_g0	235	7.16	1.00	12561.51	5282.75	100.00
c22_g0_i1	c22_g0	215	2.62	0.00	0.00	0.00	0.00
c28_g0_i1	c28_g0	329	54.60	4.00	6591.85	2772.21	100.00
c33_g0_i1	c33_g0	307	40.30	3.00	6697.56	2816.66	100.00
c35_g0_i1	c35_g0	219	3.33	0.00	0.00	0.00	0.00
c35_g1_i1	c35_g1	204	1.19	1.00	75295.99	31665.75	100.00
c39_g0_i1	c39_g0	348	68.20	1.00	1319.32	554.84	100.00
c39_g0_i2	c39_g0	255	13.97	0.00	0.00	0.00	0.00
c41_g0_i1	c41_g0	592	295.77	12.00	3650.37	1535.16	100.00
c44_g0_i1	c44_g0	361	78.10	1.00	1151.96	484.46	100.00
c44_g1_i1	c44_g1	280	25.22	1.00	3568.05	1500.54	100.00

Transcripts

gene_id	transcript_id(s)	length	effective_length	expected_count	TPM	FPKM
c128_g0	c128_g0_i1	0.00	0.00	0.00	0.00	0.00
c13_g0	c13_g0_i1	235.00	7.16	1.00	12561.51	5282.75
c22_g0	c22_g0_i1	0.00	0.00	0.00	0.00	0.00
c28_g0	c28_g0_i1	329.00	54.60	4.00	6591.85	2772.21
c33_g0	c33_g0_i1	307.00	40.30	3.00	6697.56	2816.66
c35_g0	c35_g0_i1	0.00	0.00	0.00	0.00	0.00
c35_g1	c35_g1_i1	204.00	1.19	1.00	75295.99	31665.75
c39_g0	c39_g0_i1,c39_g0_i2	348.00	68.20	1.00	1319.32	554.84
c41_g0	c41_g0_i1	592.00	295.77	12.00	3650.37	1535.16
c44_g0	c44_g0_i1	361.00	78.10	1.00	1151.96	484.46
c44_g1	c44_g1_i1	280.00	25.22	1.00	3568.05	1500.54

Genes

Meta assembly vs read normalisation

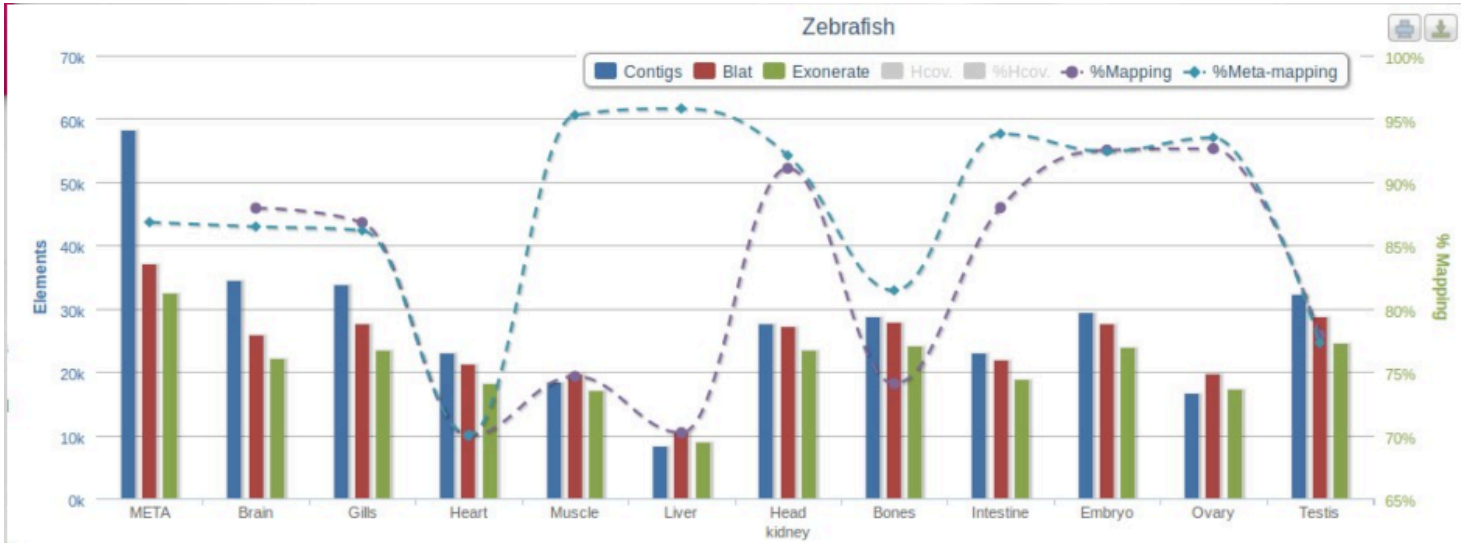
Metassembly
Produce a unique transcriptome from several samples assembled separately

Samples could be:
- from different organisms
- from different tissues
- from different experimental conditions

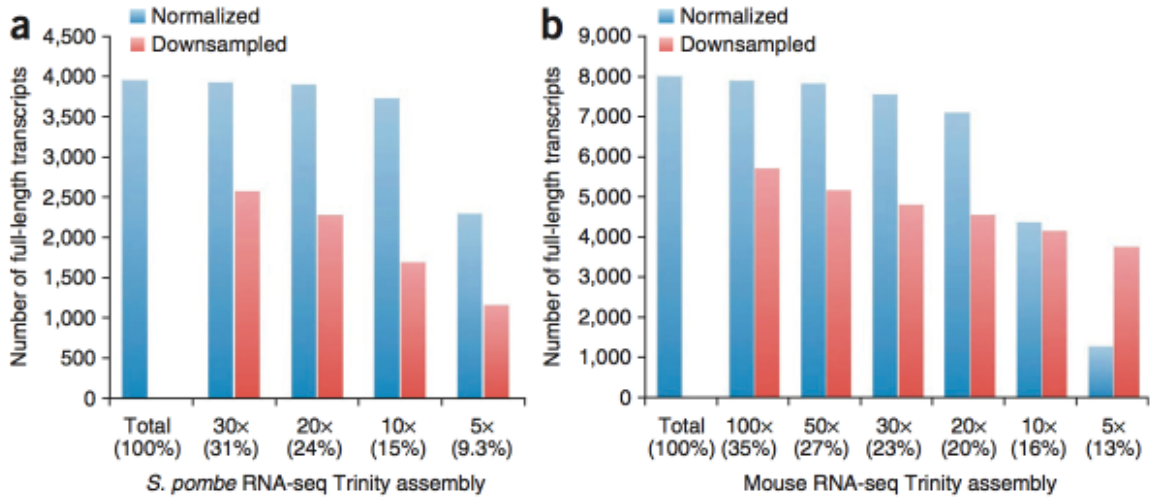
-> clusterize transcripts from same genes rebuilt in each sample
-> keep only one representative transcript per cluster

Six steps:
- merge assemblies (concatenate files)
- get the longest ORF for each transcript
- clusterize ORFs with CD-HIT
- get transcript with the longest ORF or the longest transcript for each CD-HIT cluster
- clusterize transcripts with CD-HIT-EST
- filter low coverage transcripts (RMBT, at least 1/1M mapped reads)

Meta assembly vs read normalisation

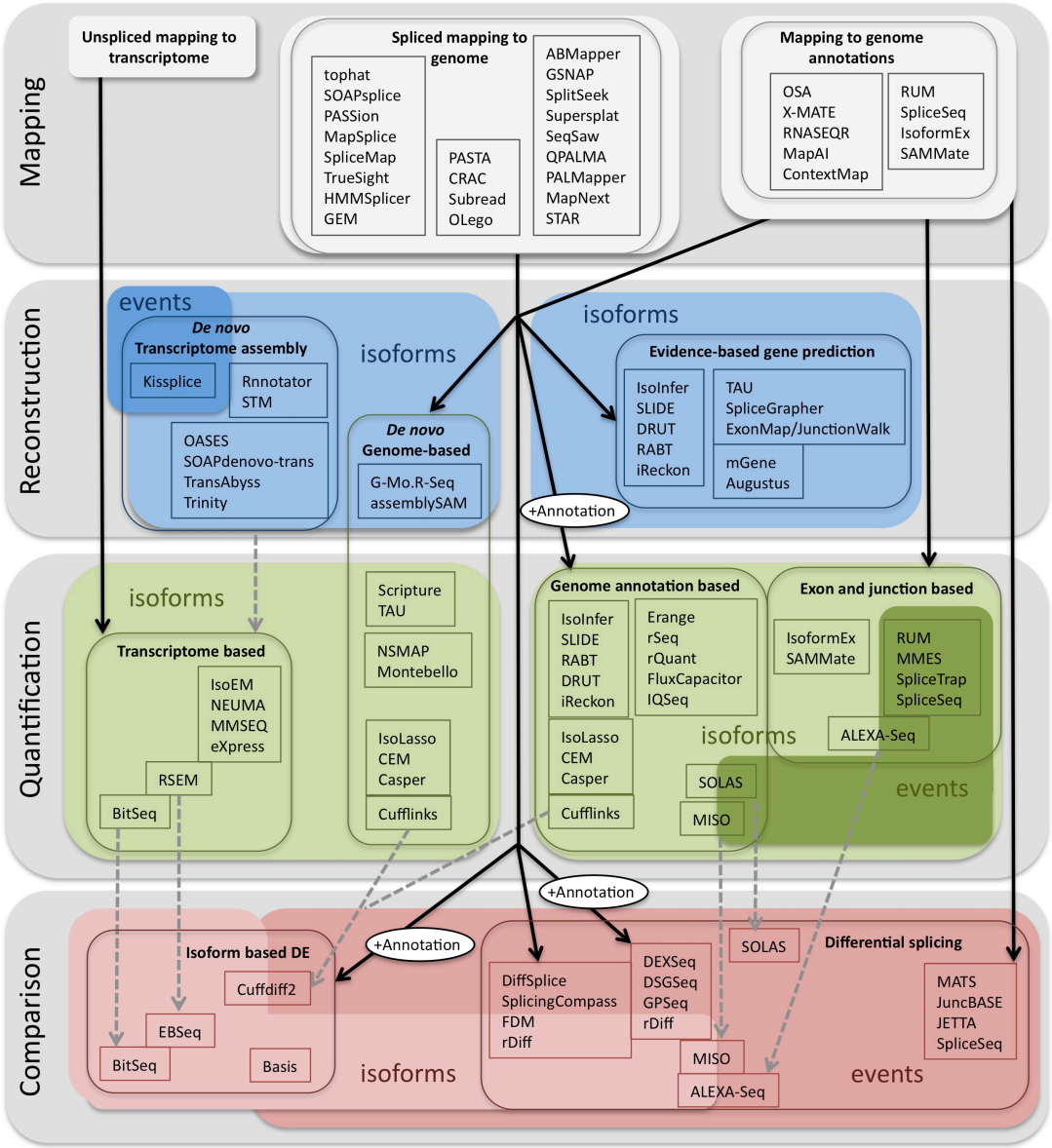


Sigenae data



Haas, B. J., Papanicolaou, A., Yassour, M., Grabherr, M., Blood, P. D., Bowden, J., et al. (2013). De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nature Protocols, 8(8), 1494–1512

A special case : Analyse the splicing events



Alamancos, G. P., Agirre, E., & Eyras, E. (2014). Methods to study splicing from high-throughput RNA sequencing data. Methods in Molecular Biology (Clifton, N.J.), 1126, 357–397. doi:10.1007/978-1-62703-980-2_26

Options for handling splice variants

- Ignore them – analyse at gene level
  - Simple, powerful, inaccurate in some cases
  - DE-Seq, EdgeR, BaySeq
- Ignore them – analyse at exon level
  - Simple, some splicing detection, mixed signals
  - DEXSeq
- Assign ambiguous reads based on unique ones
  - Potentially cleaner more powerful signal
  - High degree of uncertainty – false confidence
  - Cufflinks etc.

S. Anders

With real data

Samples


Caiman crocodilus


Tarentola mauritanica


Pleurodeles waltl


Protopterus annectens


Protopterus aethiopicus


Lepidosiren paradoxa


Scyliorhinus canicula

Chondrichthyes

Sarcopterygii

Teleostomi

Actinopterygii

Gnathostomata

Jawed vertebrates

X 19


Pantodon buchholzi


Pantodon buchholzi


Osteoglossum bicirrhosum


Anguilla anguilla


Amia calva


Lepisosteus platyrhincus


Corydoras aeneus


Astyanax mexicanus

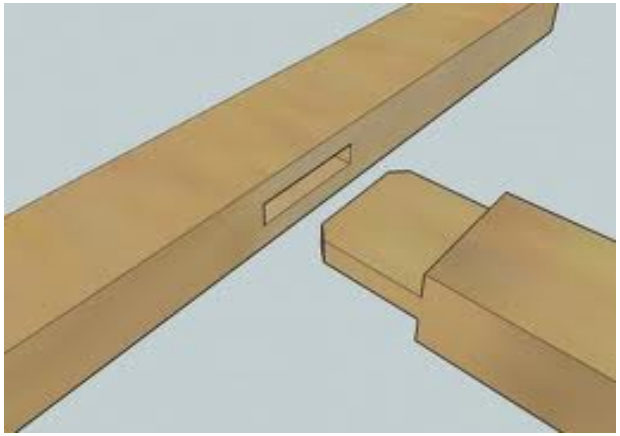

Alosa alosa


Salvelinus alpinus

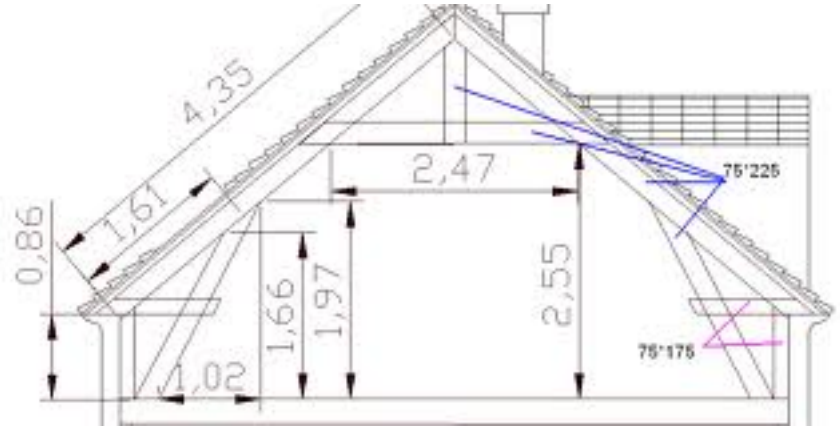

Oncorbynchus mykiss


Hemichromis letourneuxi

RNAseq with reference



+



Overview of the Tuxedo software suite

Bowtie (fast short-read alignment)

TopHat (spliced short-read alignment)

Cufflinks (transcript reconstruction from alignments)

Cuffdiff (differential expression analysis)

CummeRbund (visualization & analysis)

Bowtie

Extremely fast, general purpose short read aligner

TopHat

Aligns RNA-Seq reads to the genome using Bowtie
Discovers splice sites

Cufflinks package

Cufflinks
Assembles transcripts

Cuffcompare
Compares transcript assemblies to annotation

Cuffmerge
Merges two or more transcript assemblies

Cuffdiff
Finds differentially expressed genes and transcripts
Detects differential splicing and promoter use

CummeRbund

Plots abundance and differential
expression results from Cuffdiff

Tuxedo Team


Daehwan Kim
Johns Hopkins


Ben Langmead
Johns Hopkins


Ali Mortazavi
Caltech


Barbara Wold
Caltech


Geo Pertea
Johns Hopkins


Steven Salzberg
Johns Hopkins


Lior Pachter
UC Berkeley


Cole Trapnell
Harvard


Brian Williams
Caltech


Mihai Pop
U. Maryland


Jeltje van Baren
MBARI


Loyal Goff
Harvard

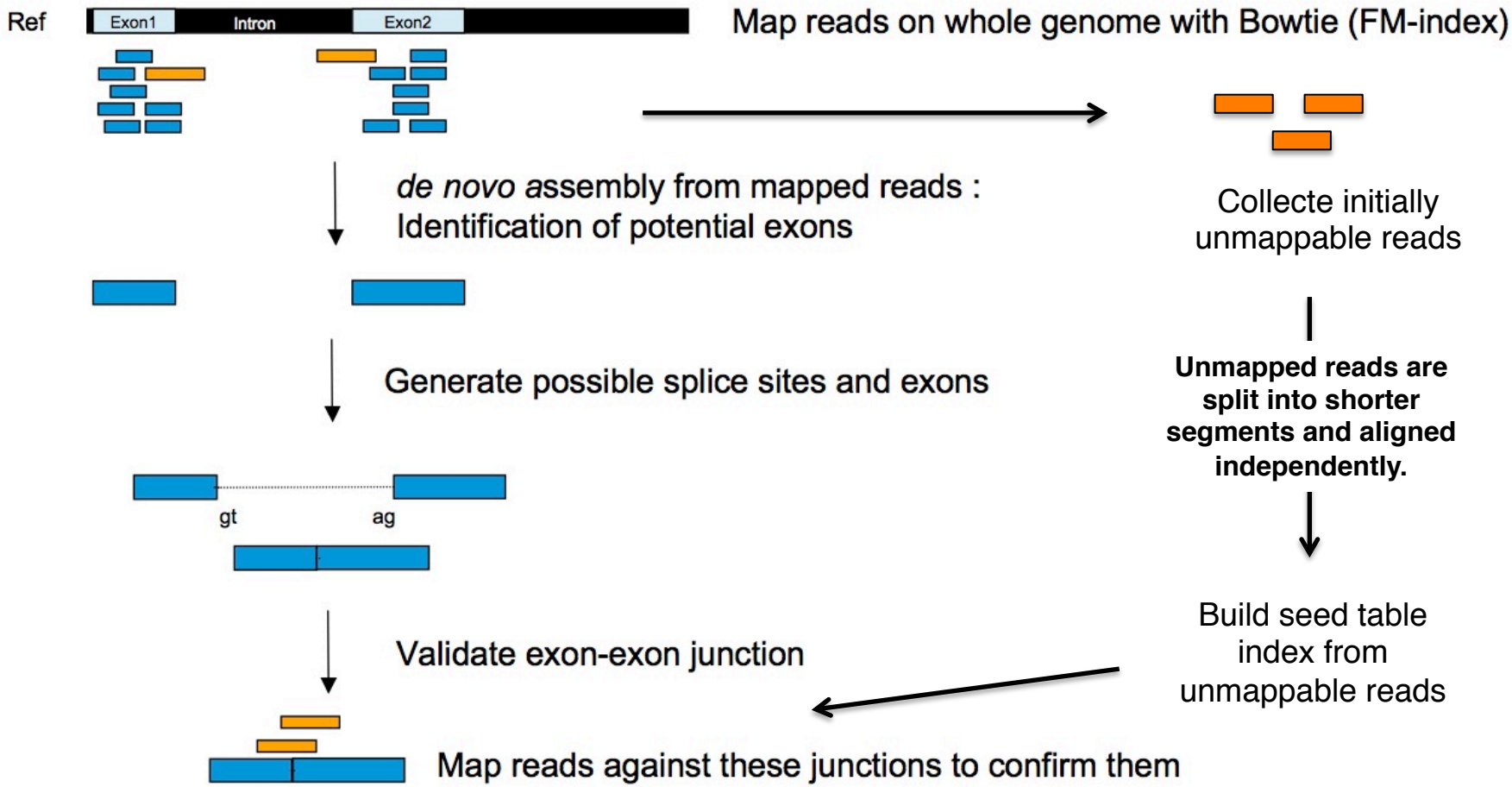

Adam Roberts
UC Berkeley


John Rinn
Harvard

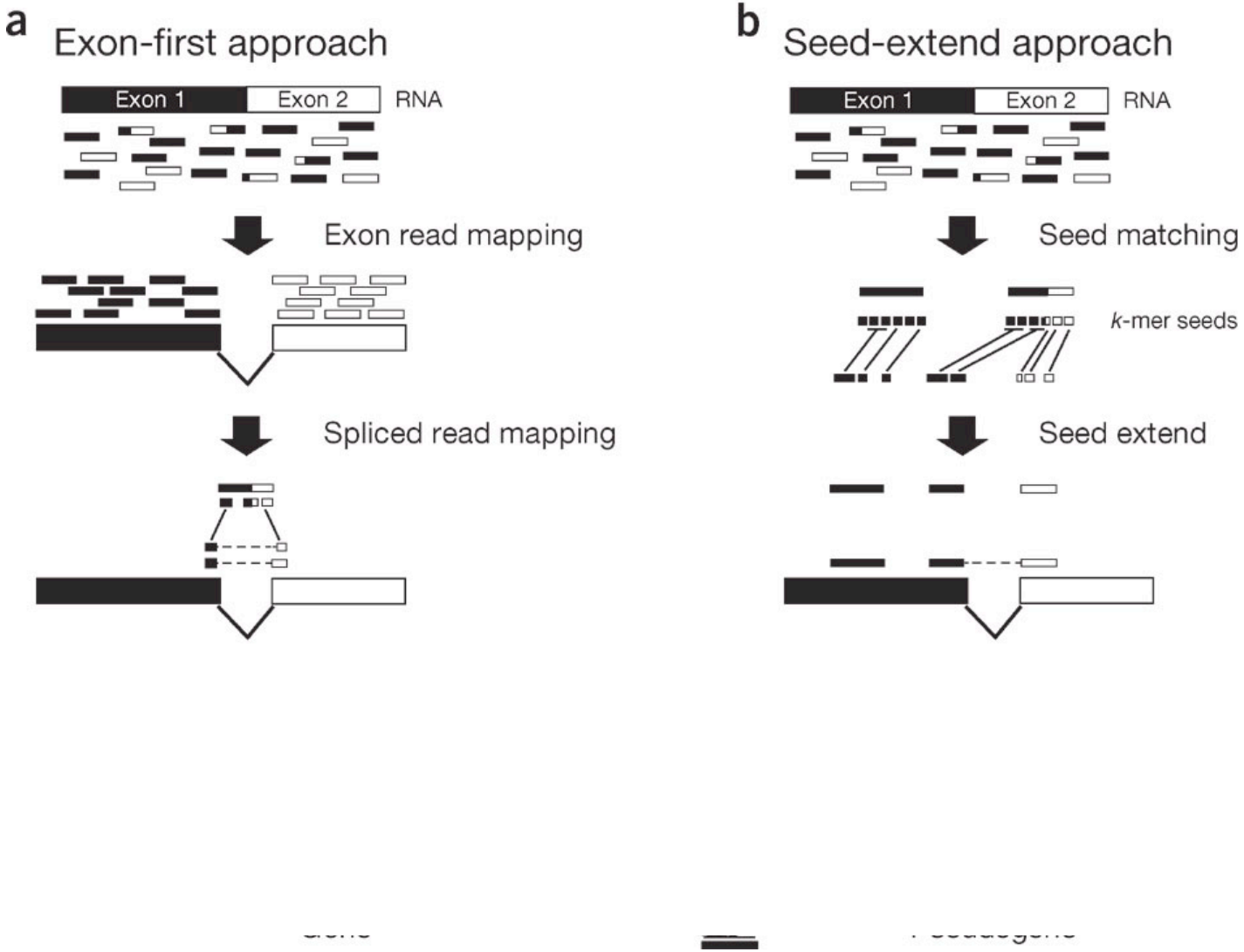

Dave
Hendrickson
Harvard


Manolis Kellis
MIT

Tophat pipeline



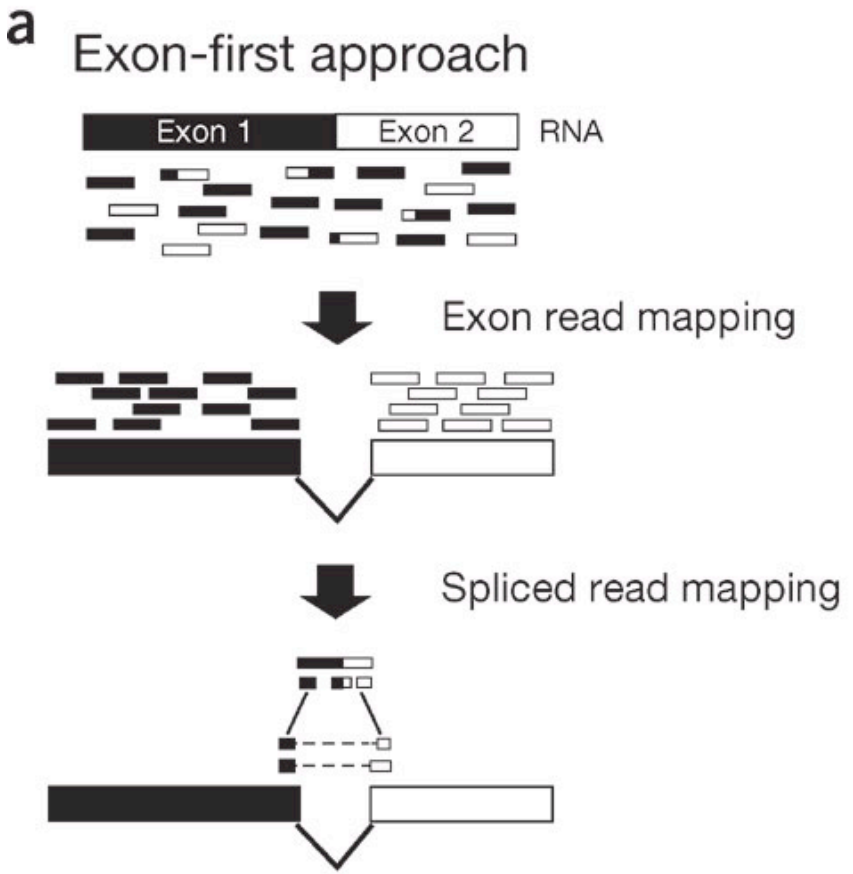
Read-to-reference alignment



Garber et al. Nature Methods 8, 469–477 (2011)

Exon-first approach : TopHat

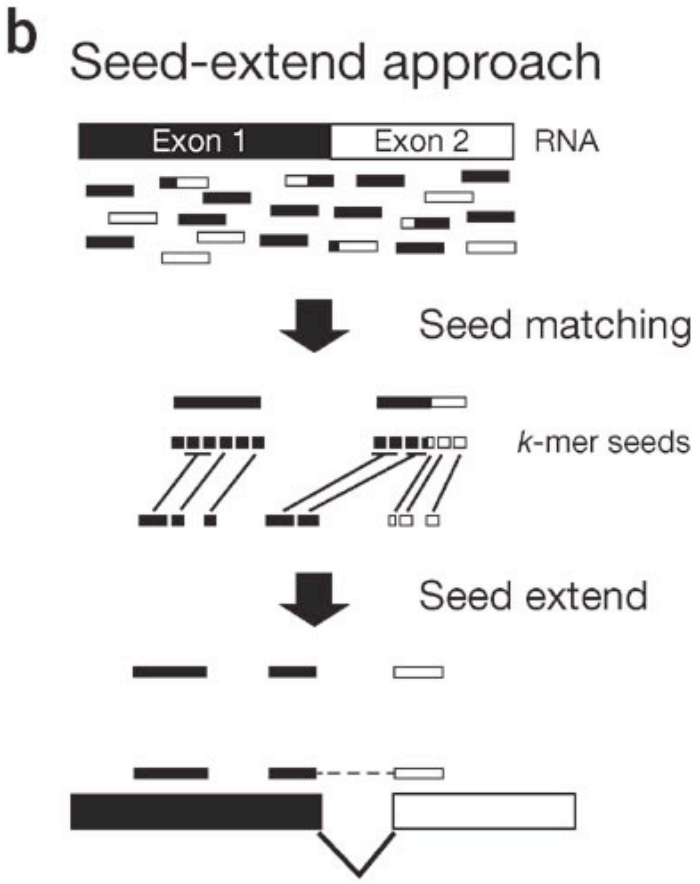
- Align reads to ref. genome
- Chop up unaligned reads and try to identify matching regions
- Find splice junctions around the matches



Garber M, Grabherr MG, Guttman M, Trapnell C. Computational methods for transcriptome annotation and quantification using RNA-seq. **Nat Methods**. 2011 PMID: 21623353.

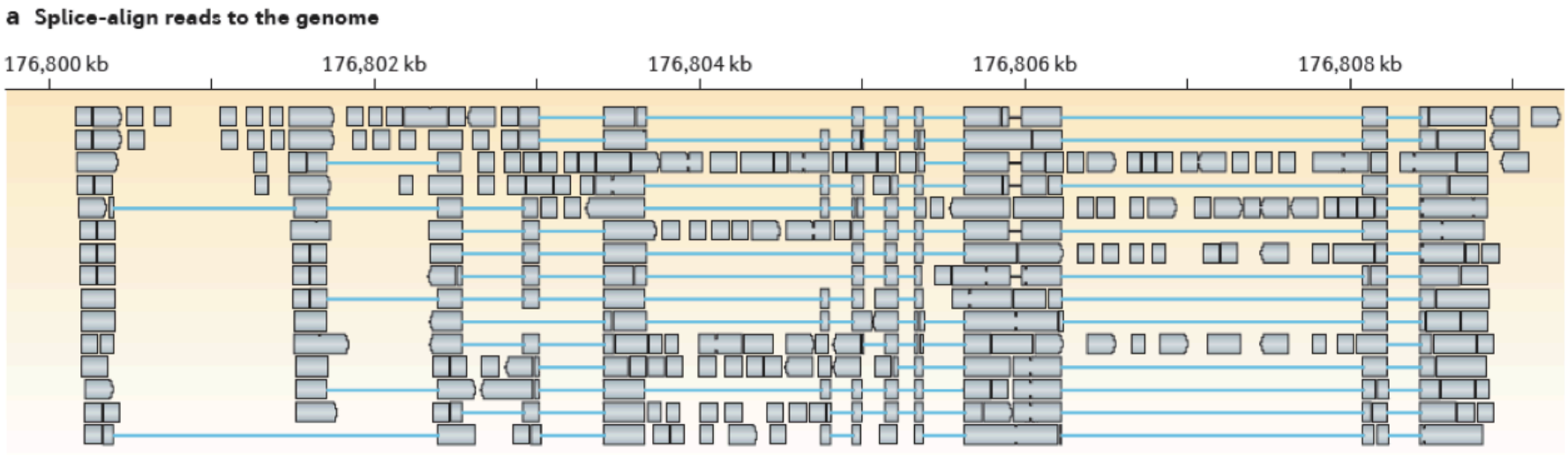
Seed-extend approach : GSNAP

- Break reads in smaller k-mers and find matches
- Iteratively extend k-mers to identify exact spliced alignment



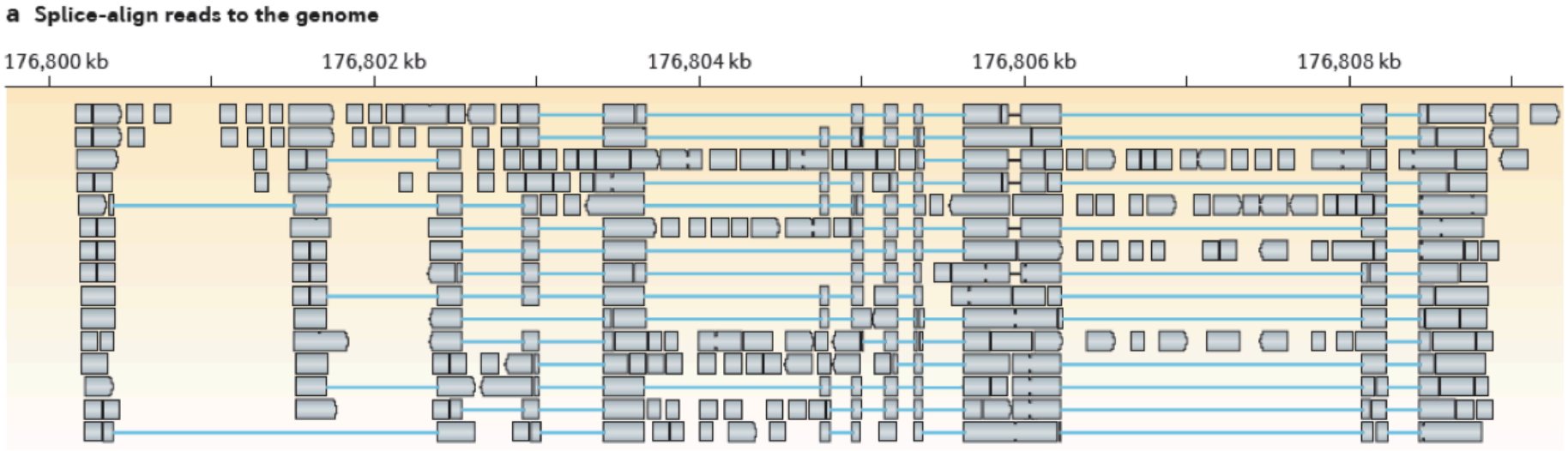
Garber M, Grabherr MG, Guttman M, Trapnell C. Computational methods for transcriptome annotation and quantification using RNA-seq. **Nat Methods**. 2011 PMID: 21623353.

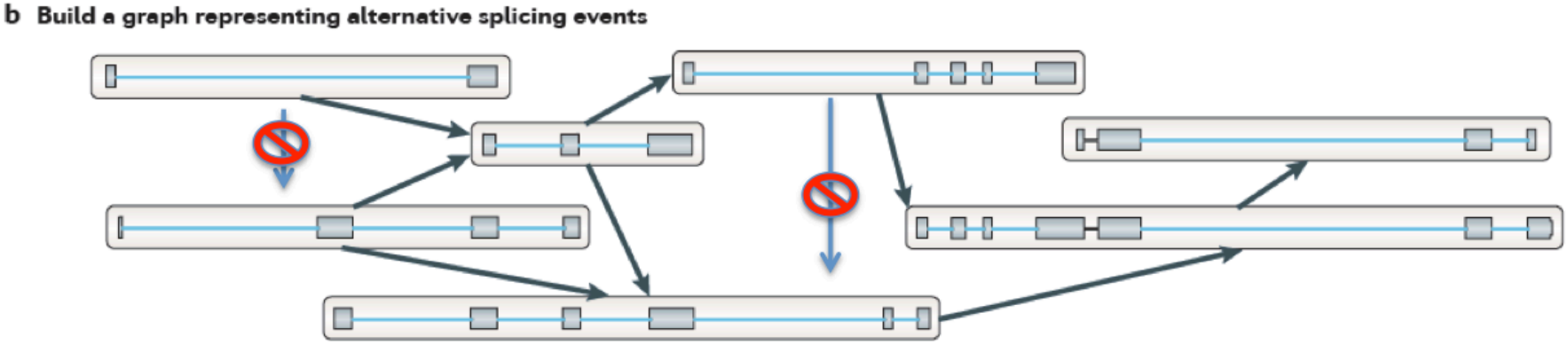
Transcripts reconstruction using cufflinks



From Martin & Wang. Nature Reviews in Genetics. 2011

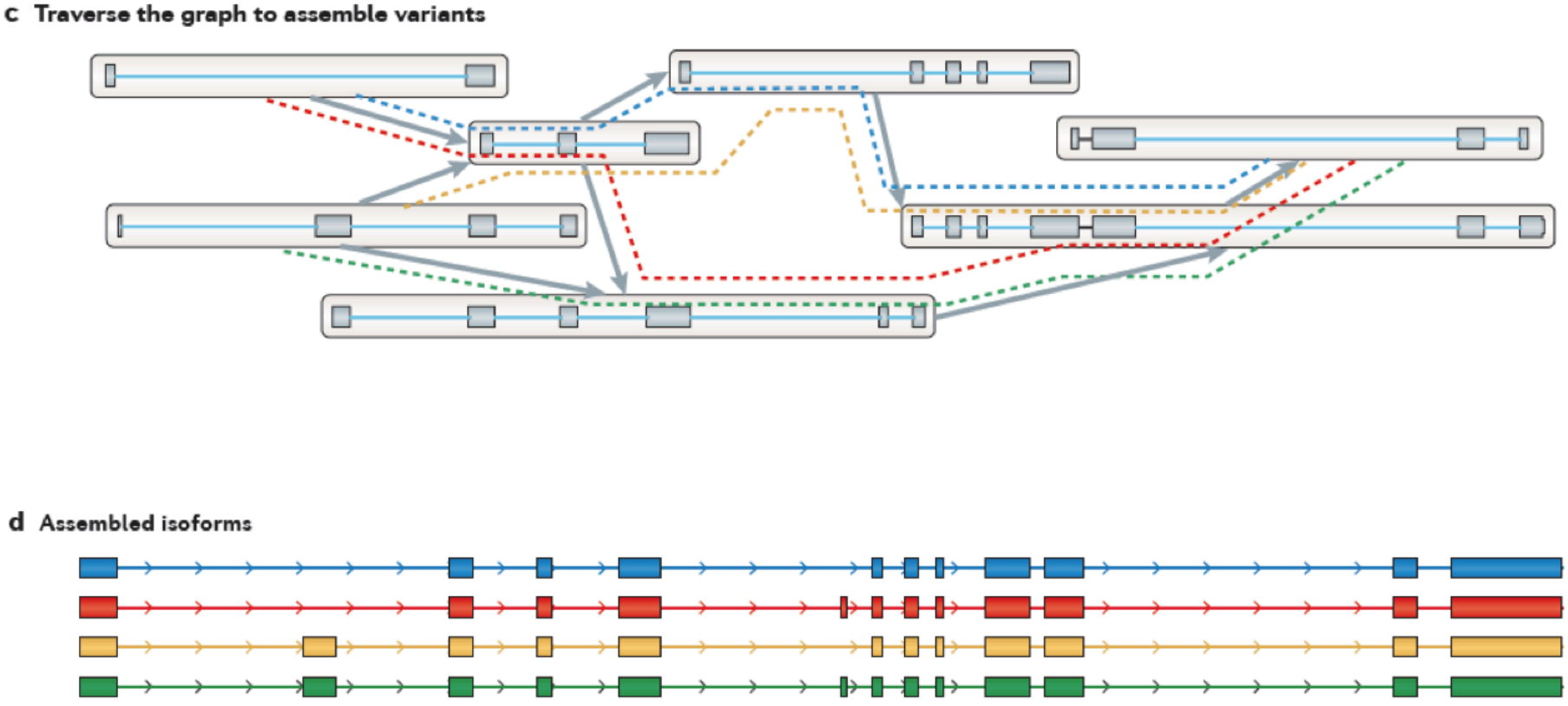
Transcripts reconstruction using cufflinks





From Martin & Wang. Nature Reviews in Genetics. 2011

Transcripts reconstruction using cufflinks



From Martin & Wang. Nature Reviews in Genetics. 2011

Differential Expression Analysis

Transcriptome De novo Assemby

PRACTICAL SESSION

The Data

Transcriptome sequencing and comparative expression profiling analysis of *Saccharina japonica* (Kombu).
It is one of the two most consumed species of kelp in China and Japan

Overall Design: mRNA expression of Saccharina japonica with 2 different treatment (sample exposed to Dark condition, and sample exposed to blue light respectively) was determined by method of RNA-Seq

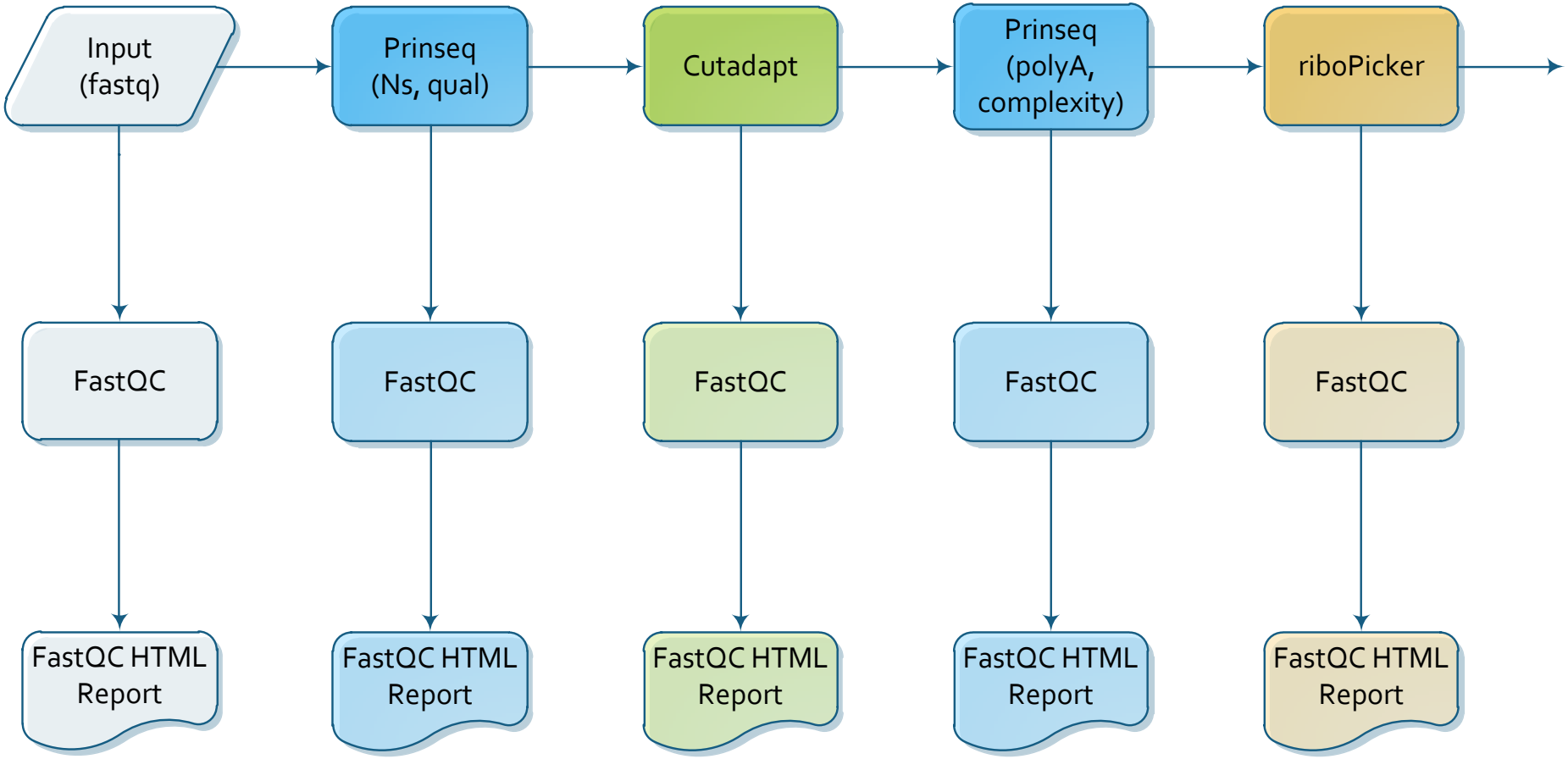
Citation: Deng Y, Yao J, Wang X, Guo H, Duan D (2012) Transcriptome Sequencing and Comparative Analysis of *Saccharina japonica* (Laminariales, Phaeophyceae) under Blue Light Induction. PLoS ONE 7(6): e39704. doi:10.1371/journal.pone.0039704

!! WARNING !!

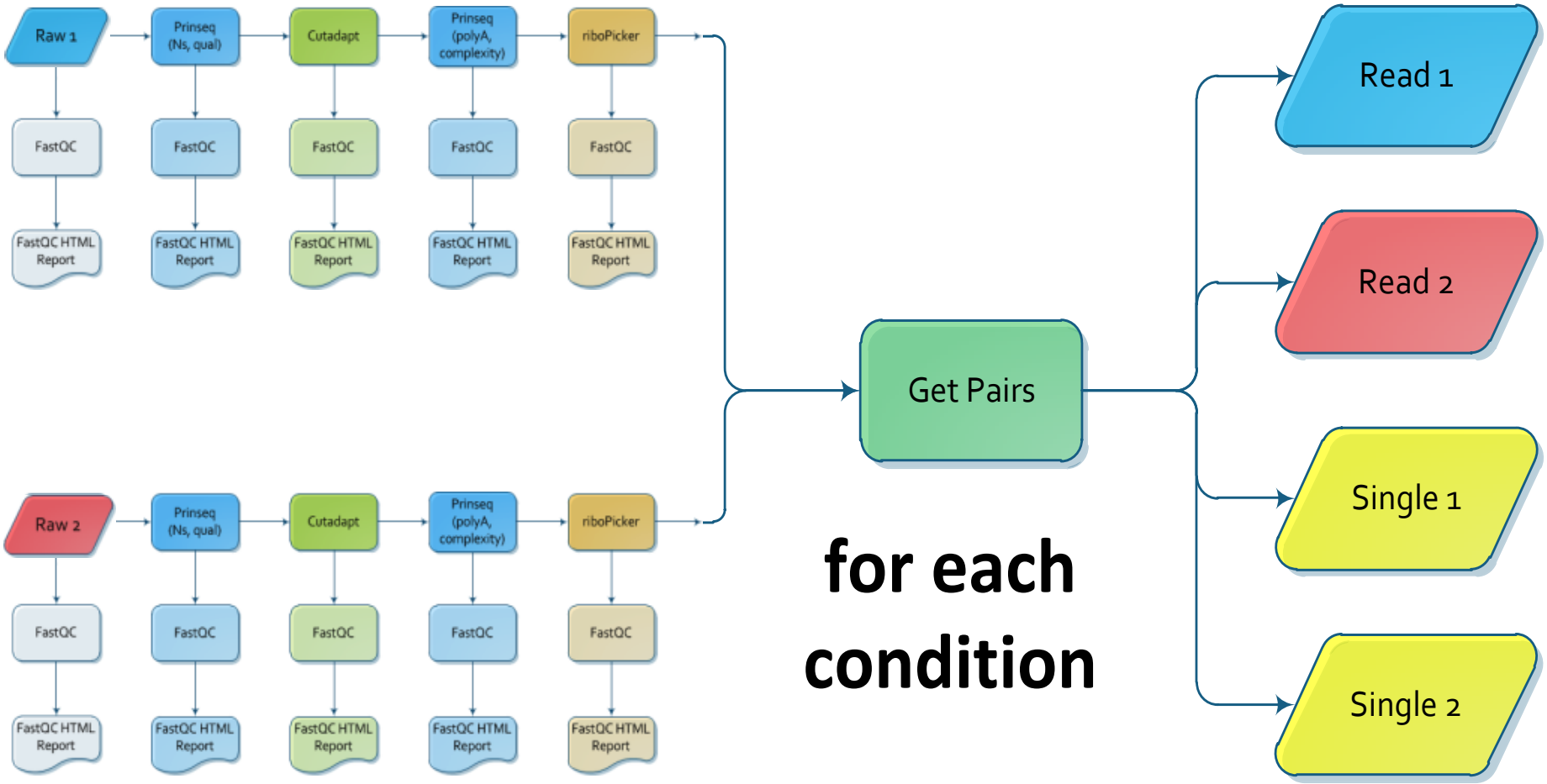
- « TRUE LIFE » data but « re-ingeenered » data
- Reverse analyse
- EdgeR + 1 biological replicate …

- Selection of 800 genes : 400 NDE– 400 DE vs 70500 unigenes
- 1 millions reads selected vs 24 millions reads sequenced

Data Cleaning



Pairs Retrieval



Pairs Retrieval

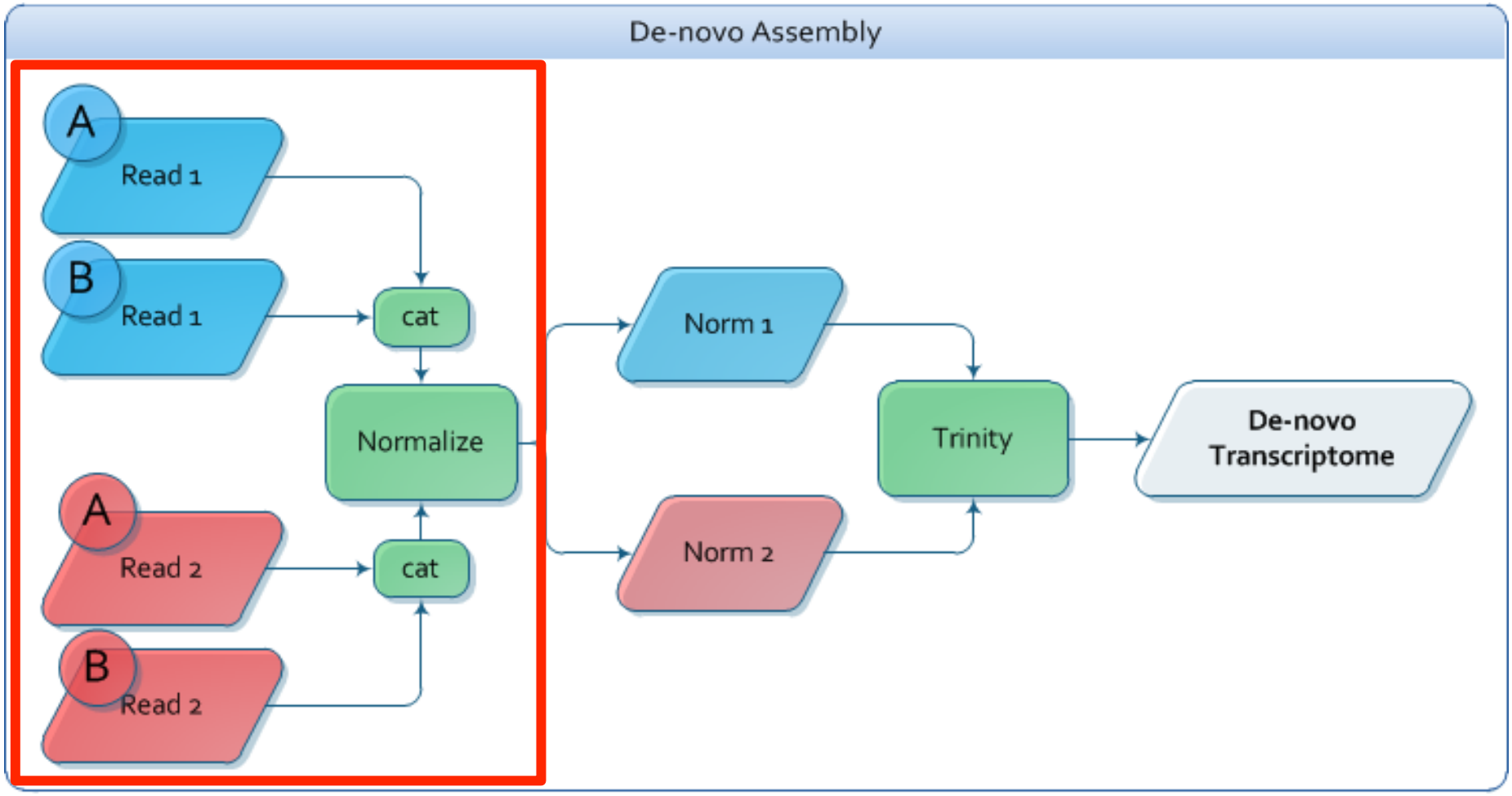
Step 1a: Get pairs

left reads fastq file	BlueLight.sample. read1. […] .nonrrna.fastq
right reads fastq file	BlueLight.sample. read2. […] .nonrrna.fastq

Step 1b: Get pairs

left reads fastq file	Dark.sample. read1. […] .nonrrna.fastq
right reads fastq file	Dark.sample. read2. […] .nonrrna.fastq

Concatenate dataset



Normalization step

Concatenate dataset

Step 2a: Concatenate datasets

Concatenate Dataset BlueLight.sample. read1. […] .paired.fastq

Add new Dataset → Dataset 1 Dark.sample. read1. […] .paired.fastq

Step 2b: Concatenate datasets

Concatenate Dataset BlueLight.sample. read2. […] .paired.fastq

Add new Dataset → Dataset 1 Dark.sample. read2. […] .paired.fastq

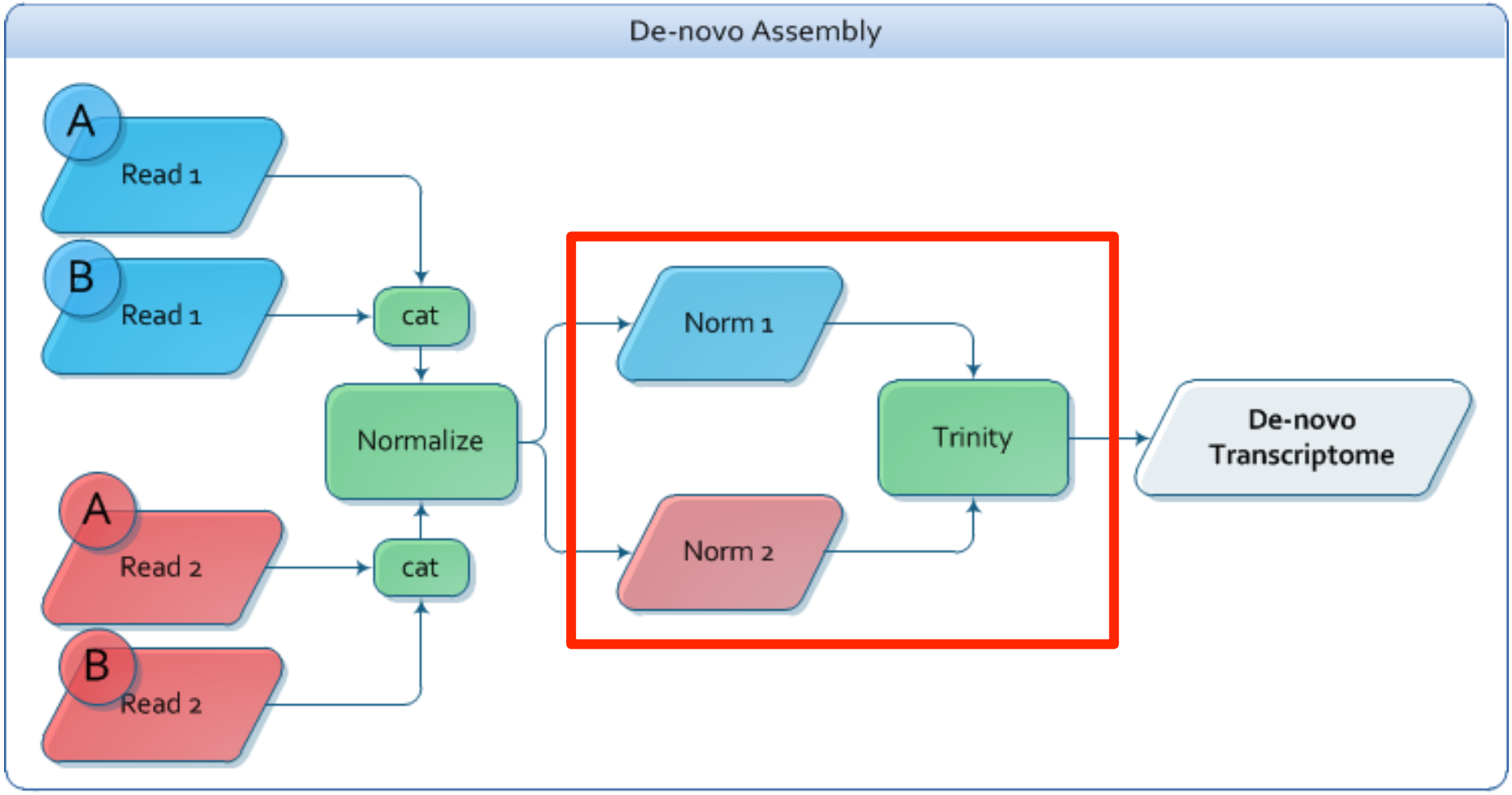
Step 3:

Rename your datasets. Ex:

all.read1.cleaned.paired.fastq

all.read2.cleaned.paired.fastq

Normalisation



Normalized assembly step

Normalisation ans Assembly

Step 4: normalize_by_kmer_coverage

single or paired reads	paired
left reads fastq file	all.read1.cleaned.paired.fastq
right reads fastq file	all.read2.cleaned.paired.fastq
pairs_together	True
max_cov	30
KMER_SIZE	25
min_kmer_cov	1
max_pct_stdev	100

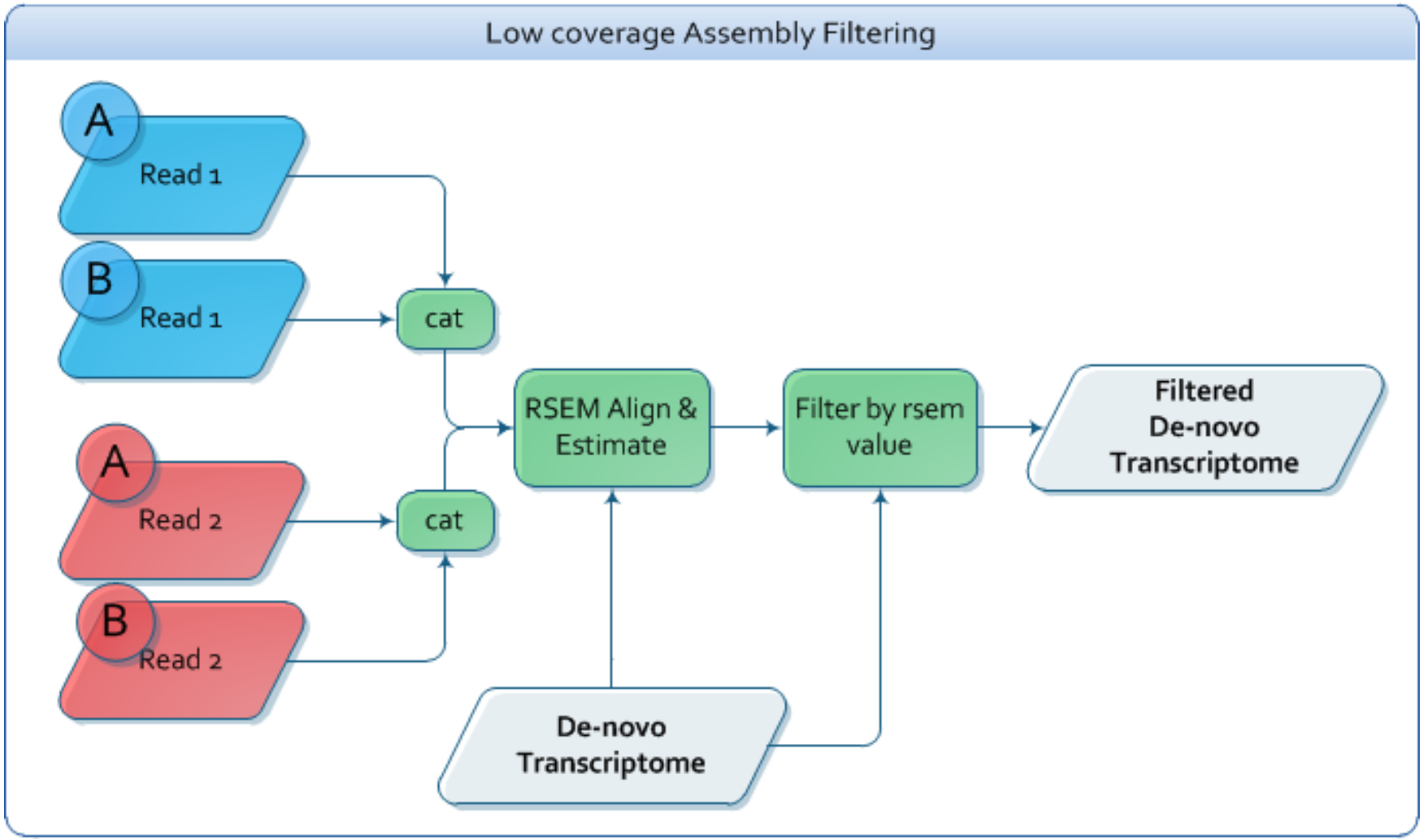
Step 5: Trinity

Left/Forward strand reads	all.read1. […] K25_C30_pctSD100.fastq
Right/Reverse strand reads	all.read2. […] K25_C30_pctSD100.fastq
Strand-specific Library type	None
Group pairs distance	500

Step 6:

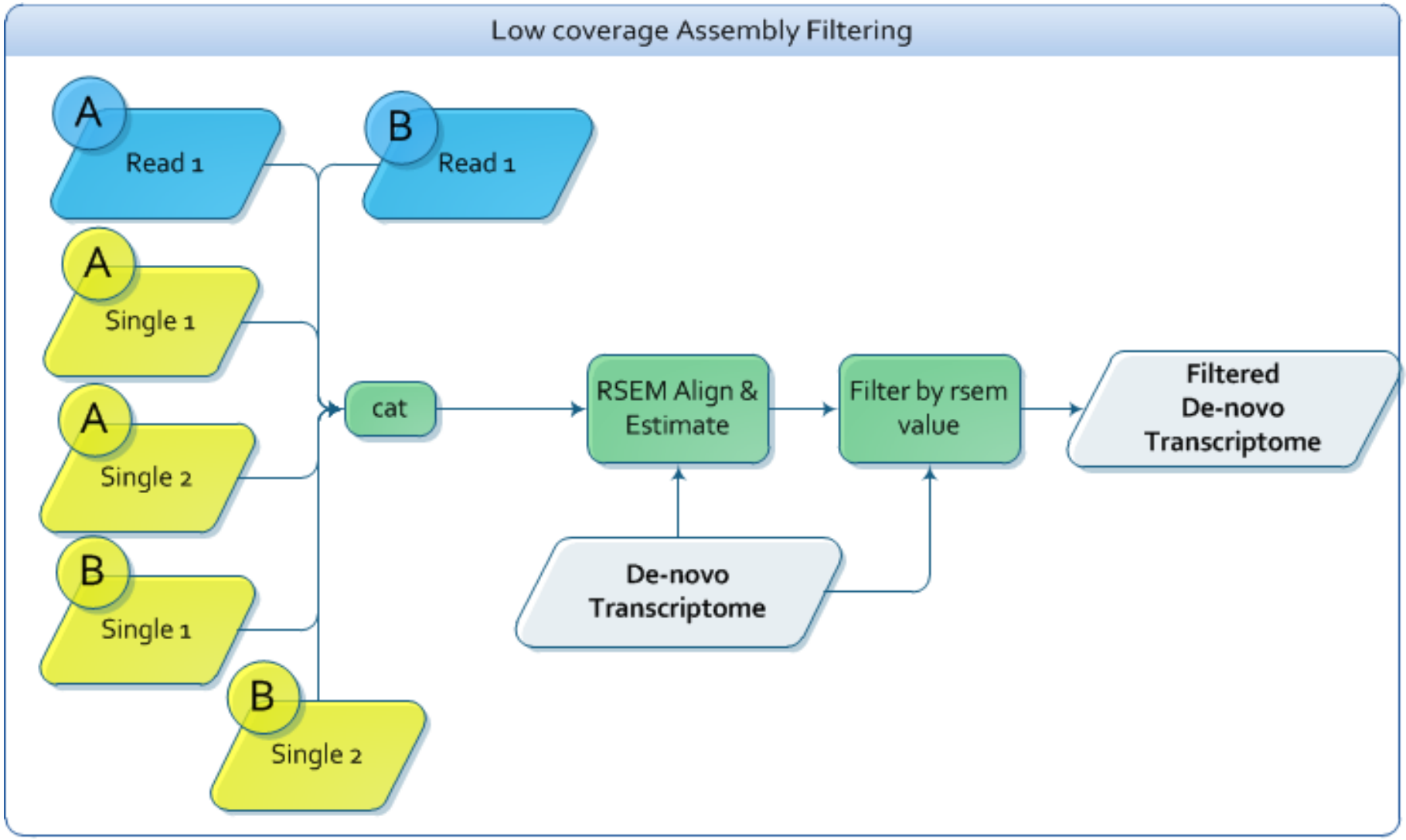
Rename your assembly file. Ex: Trinity_assembly.fasta

Low coverage filtering : normal dataset



« Normal » paired data set

Low coverage filtering : high singlet dataset



« high singleton number » data set

Low coverage filtering

Step 7: RSEM Align and Estimate

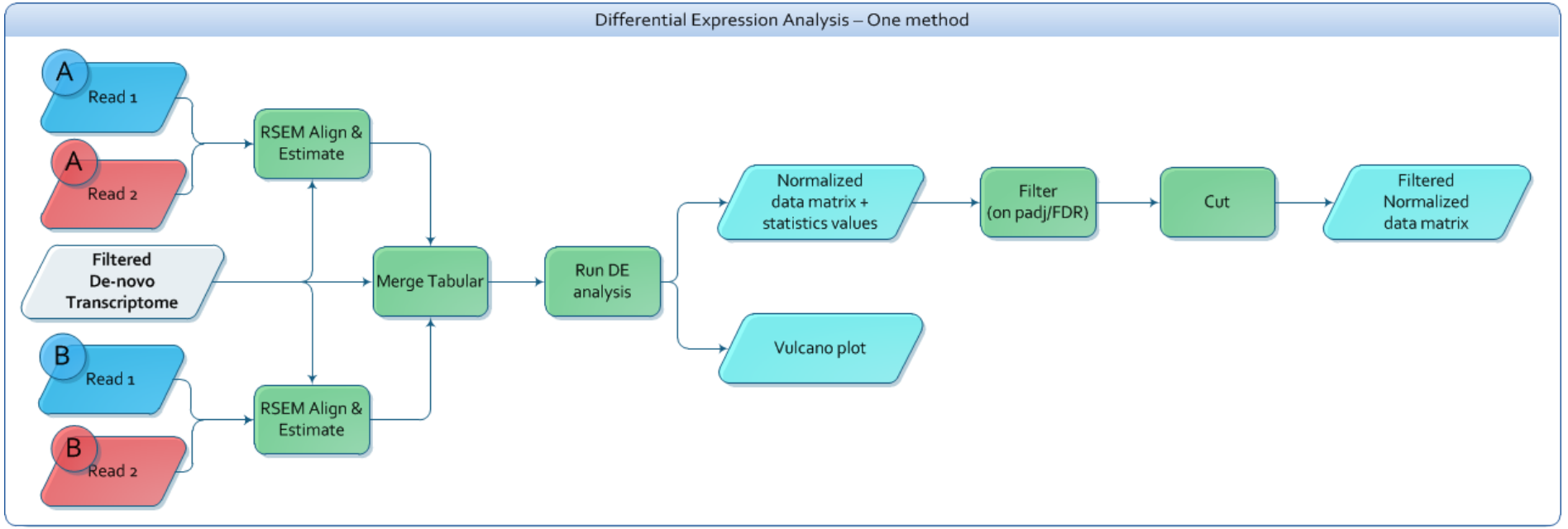
Trinity assembly	Trinity_assembly.fasta
Left/Forward strand reads	all.read1.cleaned.paired.fastq
Right/Reverse strand reads	all.read2.cleaned.paired.fastq

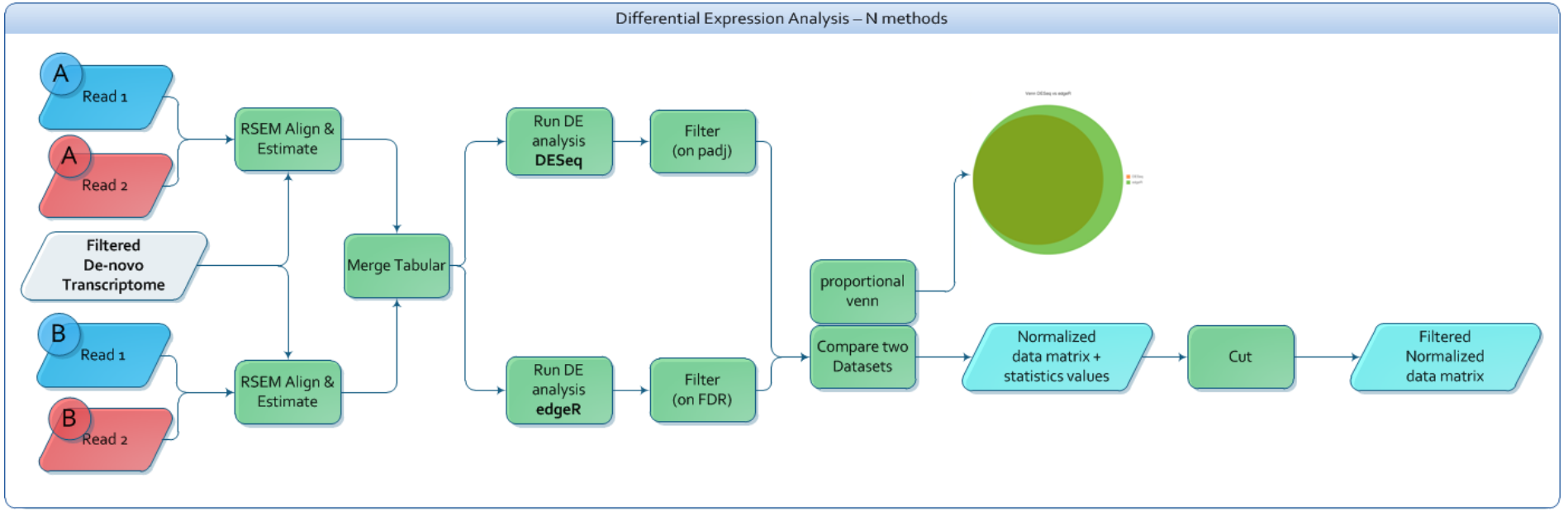
Step 8: Filter fasta by rsem values

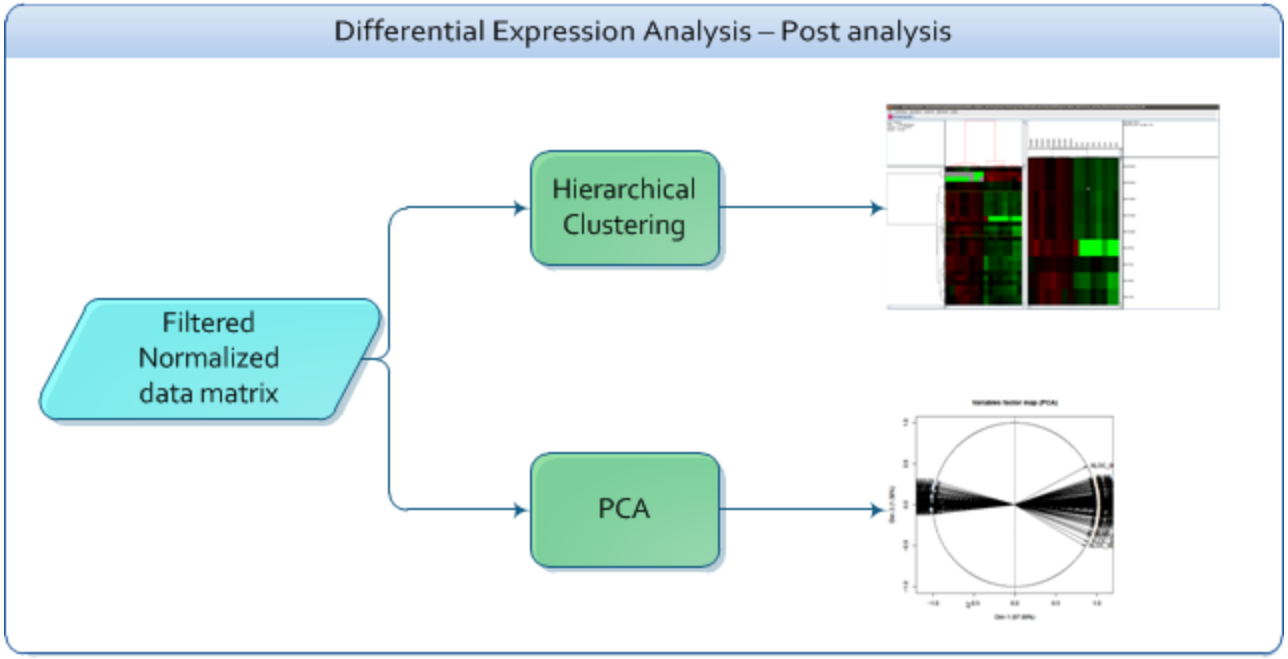
Trinity Fasta File	Trinity_assembly.fasta
RSEM output	RSEM.isoforms.results
FPKM cutoff	1
Isopct cutoff	1

Step 9:

Rename your filtered assembly file. Ex: Trinity_assembly.filtered.fasta


Differential Expression Analysis – One method
A
Read 1
A
Read 2
Filtered
De-novo
Transcriptome
B
Read 1
B
Read 2
RSEM Align &
Estimate
RSEM Align &
Estimate
Merge Tabular
Run DE
analysis
Normalized
data matrix +
statistics values
Vulcano plot
Filter
(on padj/FDR)
Cut
Filtered
Normalized
data matrix


Differential Expression Analysis – N methods
A
Read 1
A
Read 2
Filtered
De-novo
Transcriptome
B
Read 1
B
Read 2
RSEM Align &
Estimate
RSEM Align &
Estimate
Merge Tabular
Run DE
analysis
DESeq
Run DE
analysis
edgeR
Filter
(on padj)
Filter
(on FDR)
proportional
venn
Compare two
Datasets
Normalized
data matrix +
statistics values
Cut
Filtered
Normalized
data matrix


Differential Expression Analysis – Post analysis
Filtered Normalized data matrix
Hierarchical Clustering
PCA

Pipeline modification

Very few singletons :

- Assembly : (pairedR1 + pairedR2)_norm
- Remapping - filtering : pairedR1 + pairedR2
- Remapping - counting : pairedR1 + pairedR2

Few singletons :

- Assembly : (pairedR1 + pairedR2)_norm
- Remapping - filtering : pairedR1 + singletonR1+ singletonR2
- Remapping - counting : pairedR1 + singletonR1 + singletonR2

Lot of singletons :

- Assembly : (pairedR1 + pairedR2)_norm + (singletonR1 + singletonR2)_norm
- Remapping - filtering : pairedR1 + singletonR1 + singletonR2
- Remapping - counting : pairedR1 + singletonR1 + singletonR2