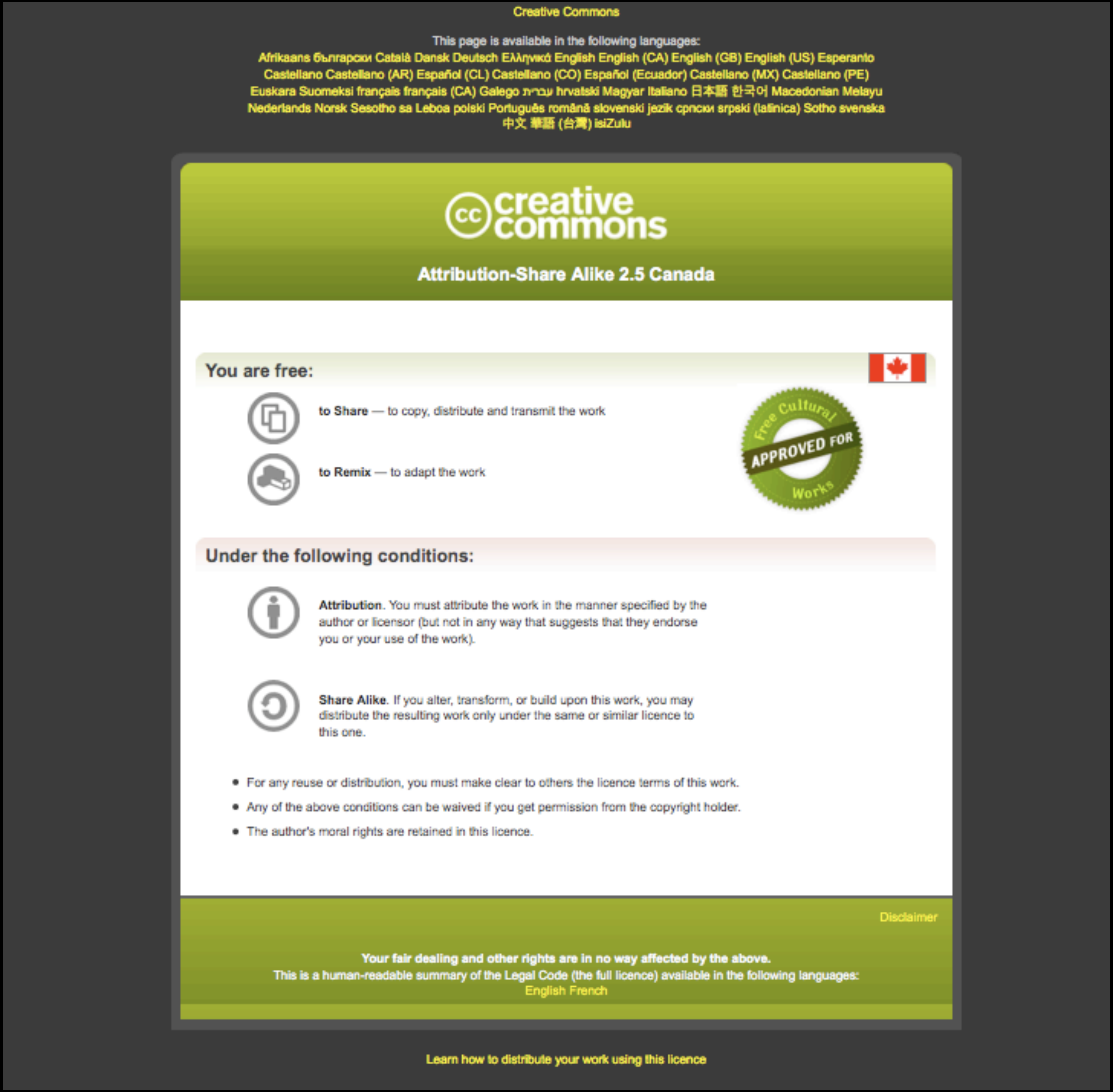
bioinformatics.ca

Canadian Bioinformatics Workshops

www.bioinformatics.ca



Creative Commons

This page is available in the following languages:
Afrikaans български Català Dansk Deutsch Ελληνικά English English (CA) English (GB) English (US) Esperanto Castellano Castellano (AR) Español (CL) Castellano (CO) Español (Ecuador) Castellano (MX) Castellano (PE) Euskara Suomeksi français français (CA) Galego עברית hrvatski Magyar Italiano 日本語 한국어 Macedonian Melayu Nederlands Norsk Sesotho sa Leboa polski Português română slovenski jezik српски srpski (latinica) Sotho svenska 中文 華語 (台灣) isiZulu

Attribution-Share Alike 2.5 Canada

You are free:

to Share — to copy, distribute and transmit the work

to Remix — to adapt the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

Share Alike. If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar licence to this one.

- For any reuse or distribution, you must make clear to others the licence terms of this work.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- The author's moral rights are retained in this licence.

Disclaimer

Your fair dealing and other rights are in no way affected by the above.
This is a human-readable summary of the Legal Code (the full licence) available in the following languages:
English French

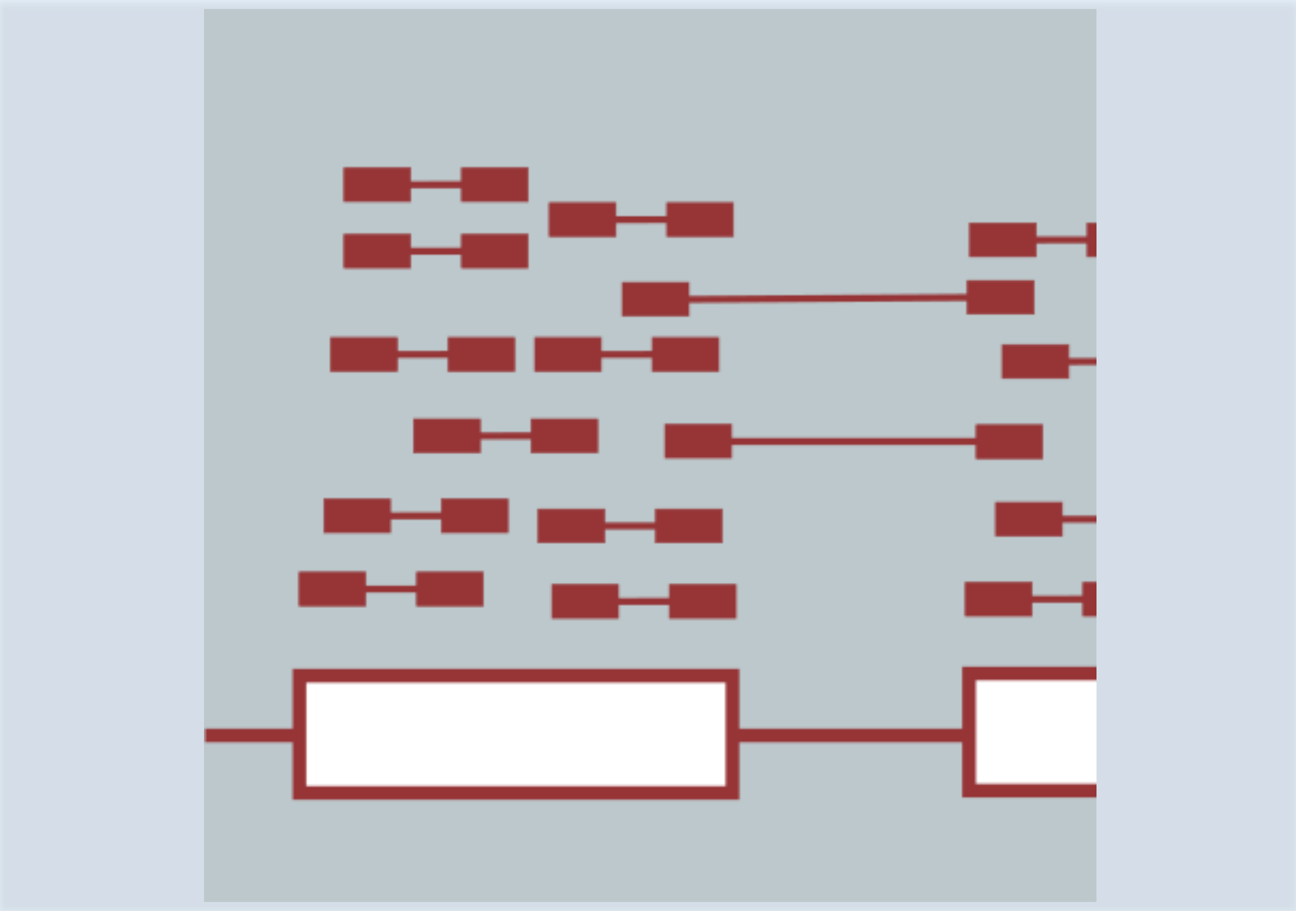
Learn how to distribute your work using this licence

Genome-Free De novo Transcriptome Assembly

Brian Haas

Informatics for RNA-Seq Analysis
July 10-12, 2017

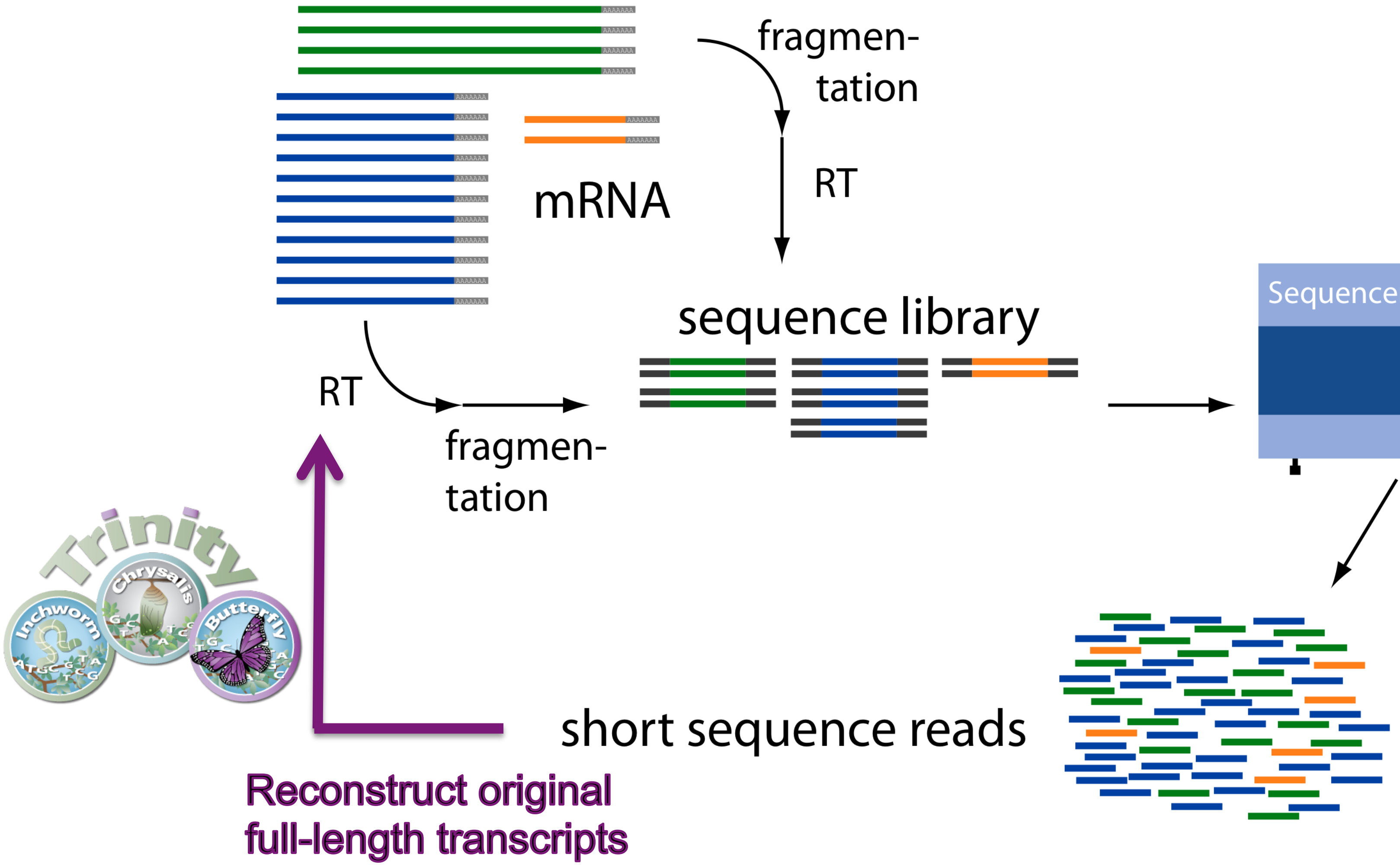
bioinformatics.ca



Learning Objectives of Module

- Understand the challenges involved in reconstructing transcripts from RNA-Seq data
- Become familiar with computational algorithms and data structures leveraged for transcript assembly
- Appreciate the importance of strand-specific RNA-Seq data.
- Learn various ways to assess the quality of an assembled transcriptome.

Assembly Required



mRNA

fragmen-
tation

RT

sequence library

Sequencer

RT

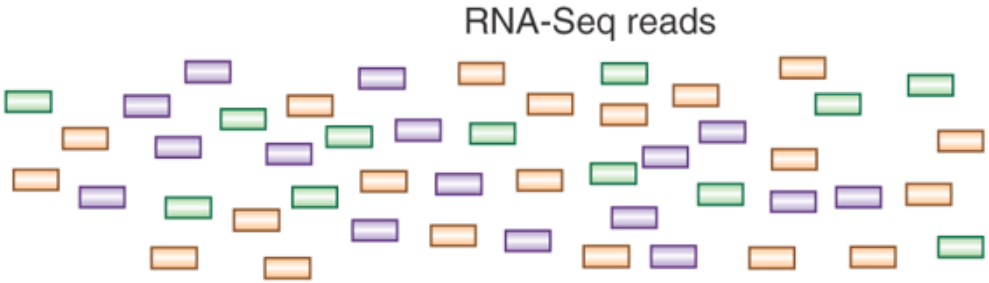
fragmen-
tation

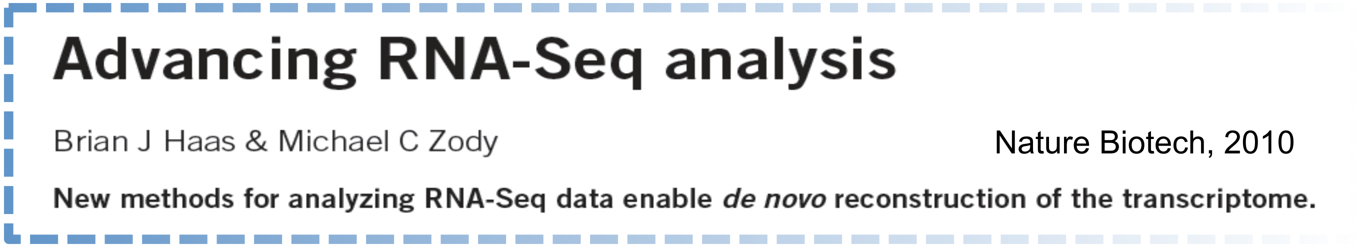
short sequence reads

Reconstruct original
full-length transcripts

Adapted from G. Raetsch

Transcript Reconstruction from RNA-Seq Reads





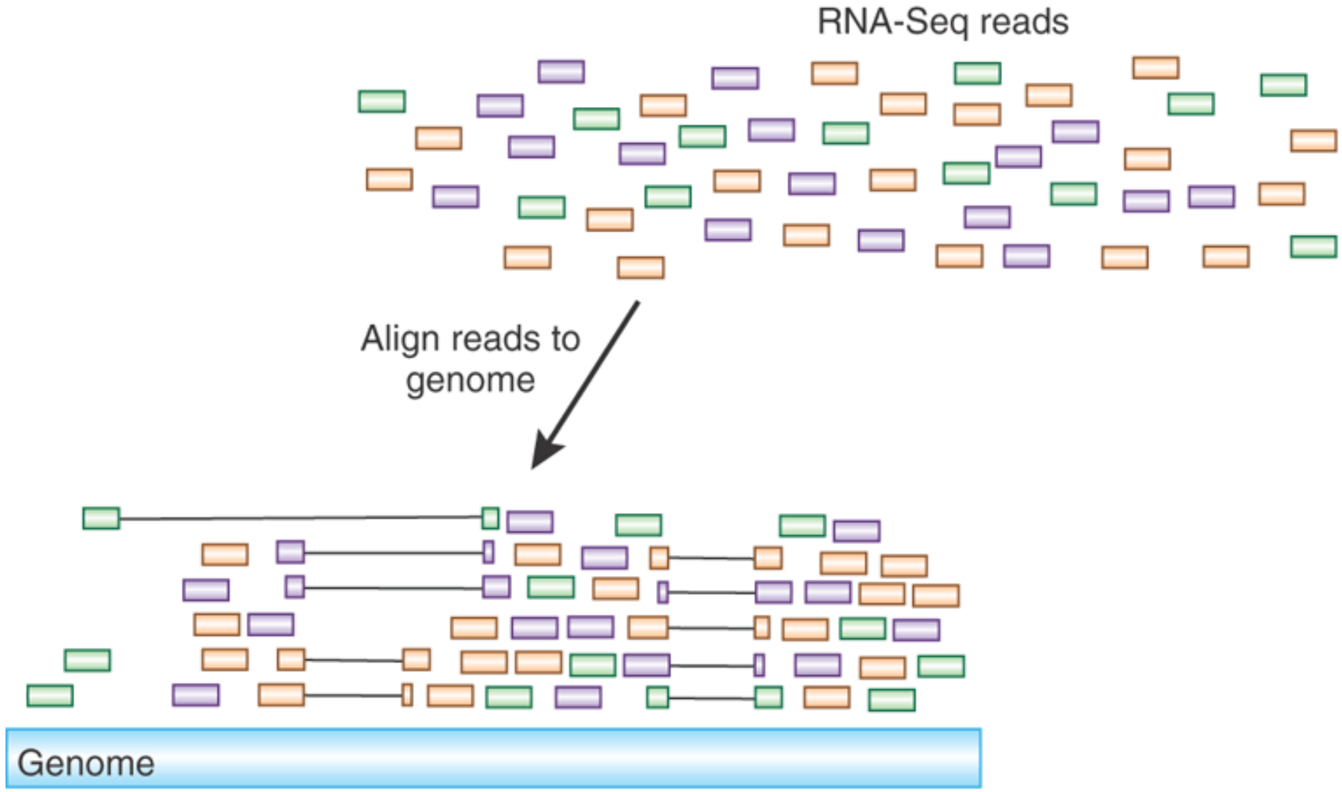
Advancing RNA-Seq analysis

Brian J Haas & Michael C Zody

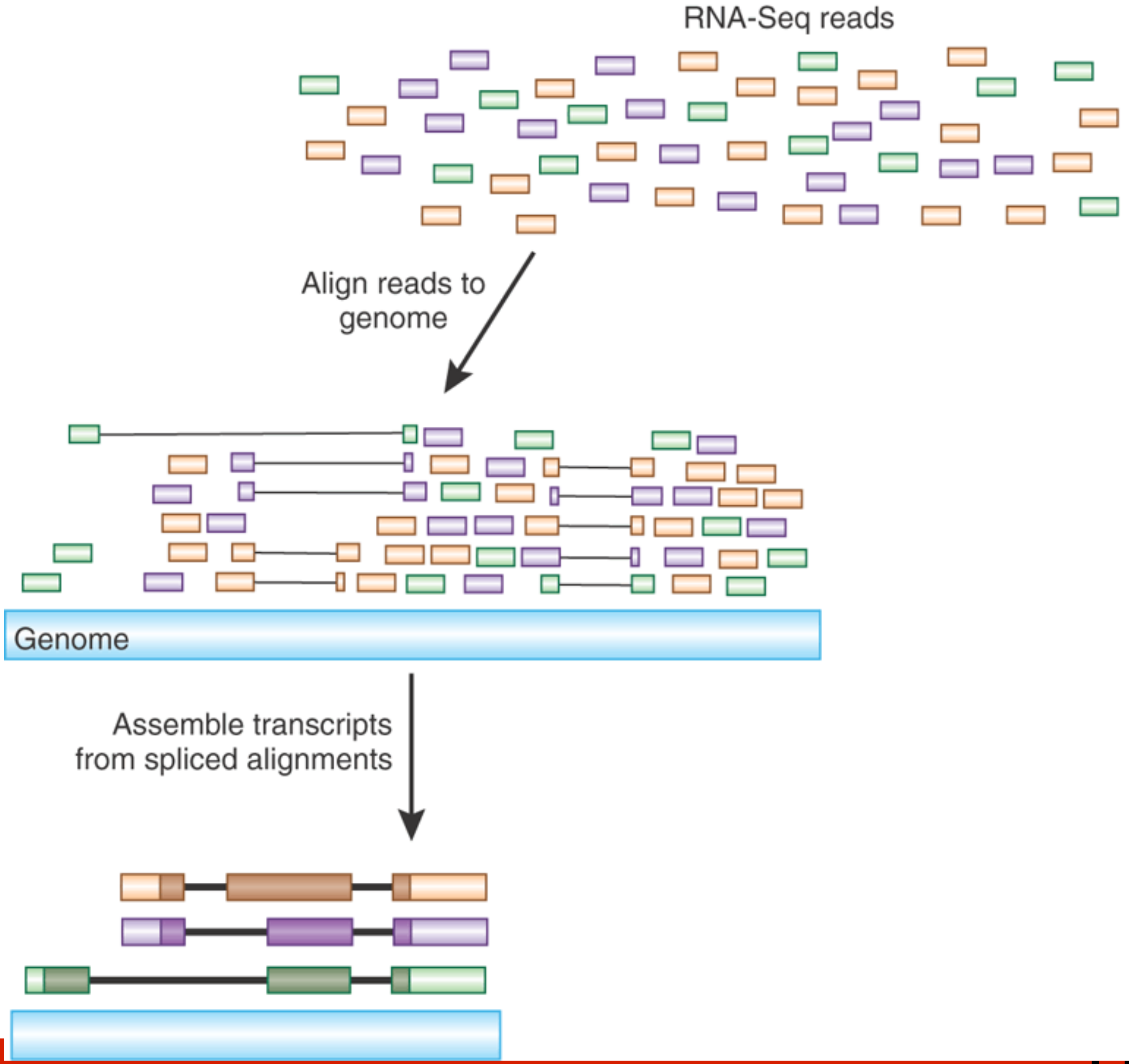
Nature Biotech, 2010

New methods for analyzing RNA-Seq data enable *de novo* reconstruction of the transcriptome.

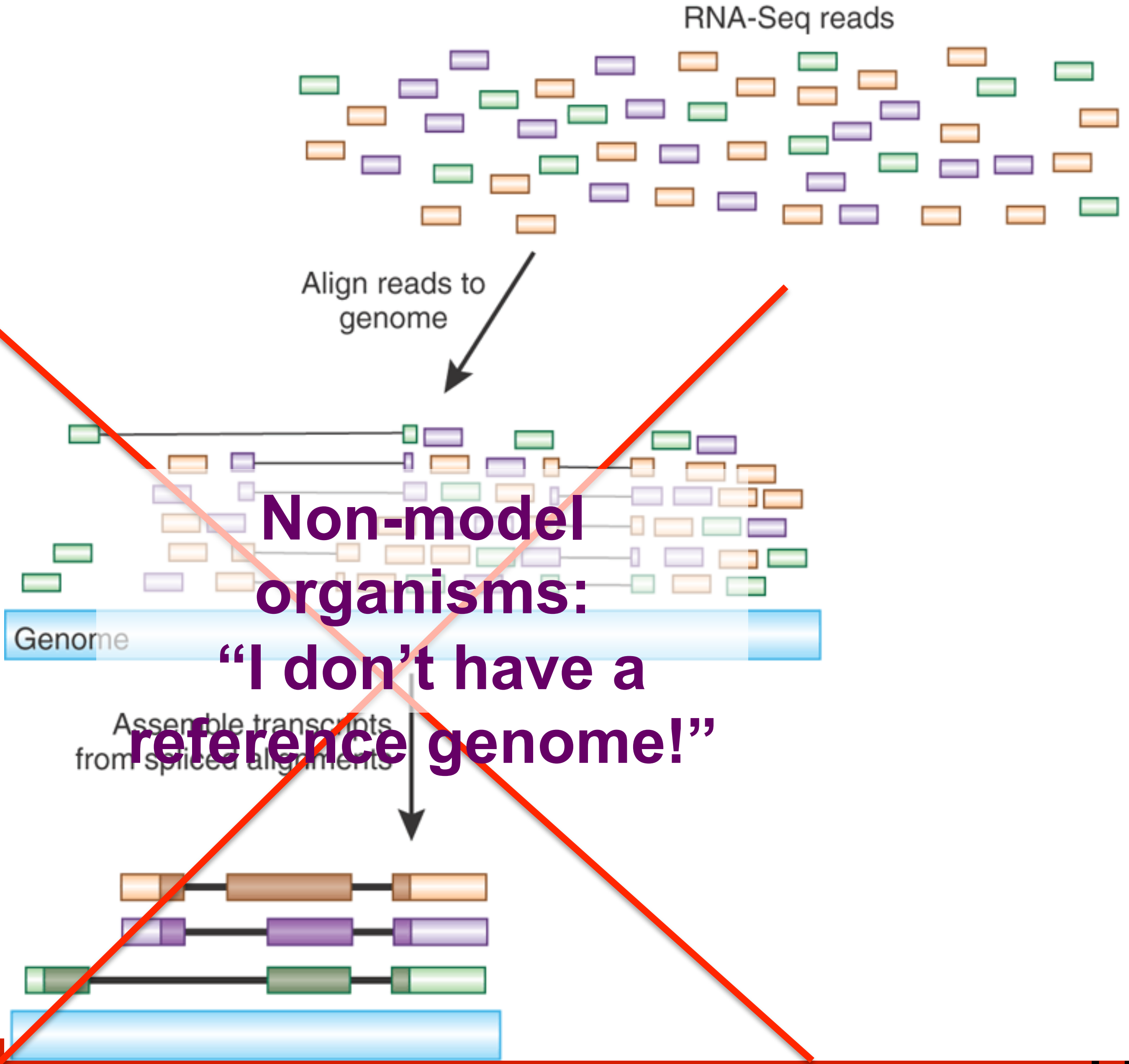
Transcript Reconstruction from RNA-Seq Reads



Transcript Reconstruction from RNA-Seq Reads

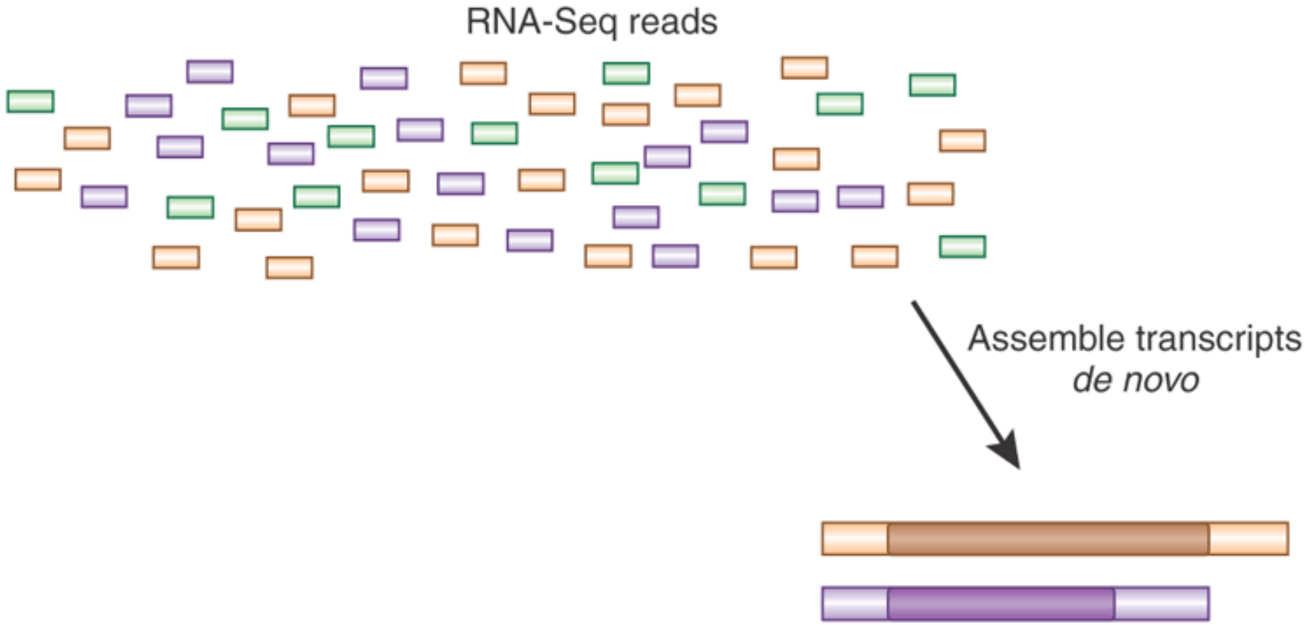


Transcript Reconstruction from RNA-Seq Reads

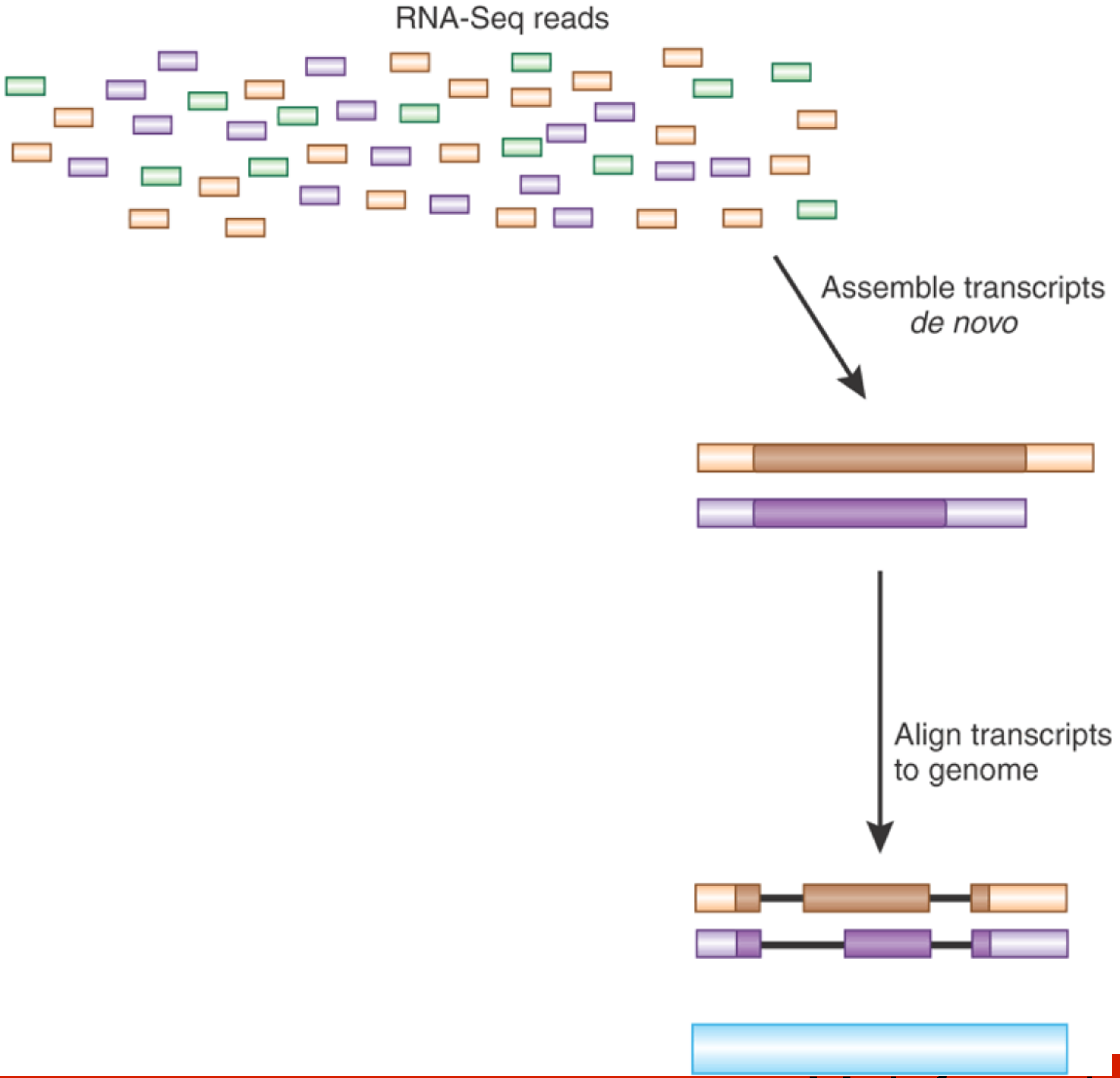


Non-model organisms: “I don’t have a reference genome!”

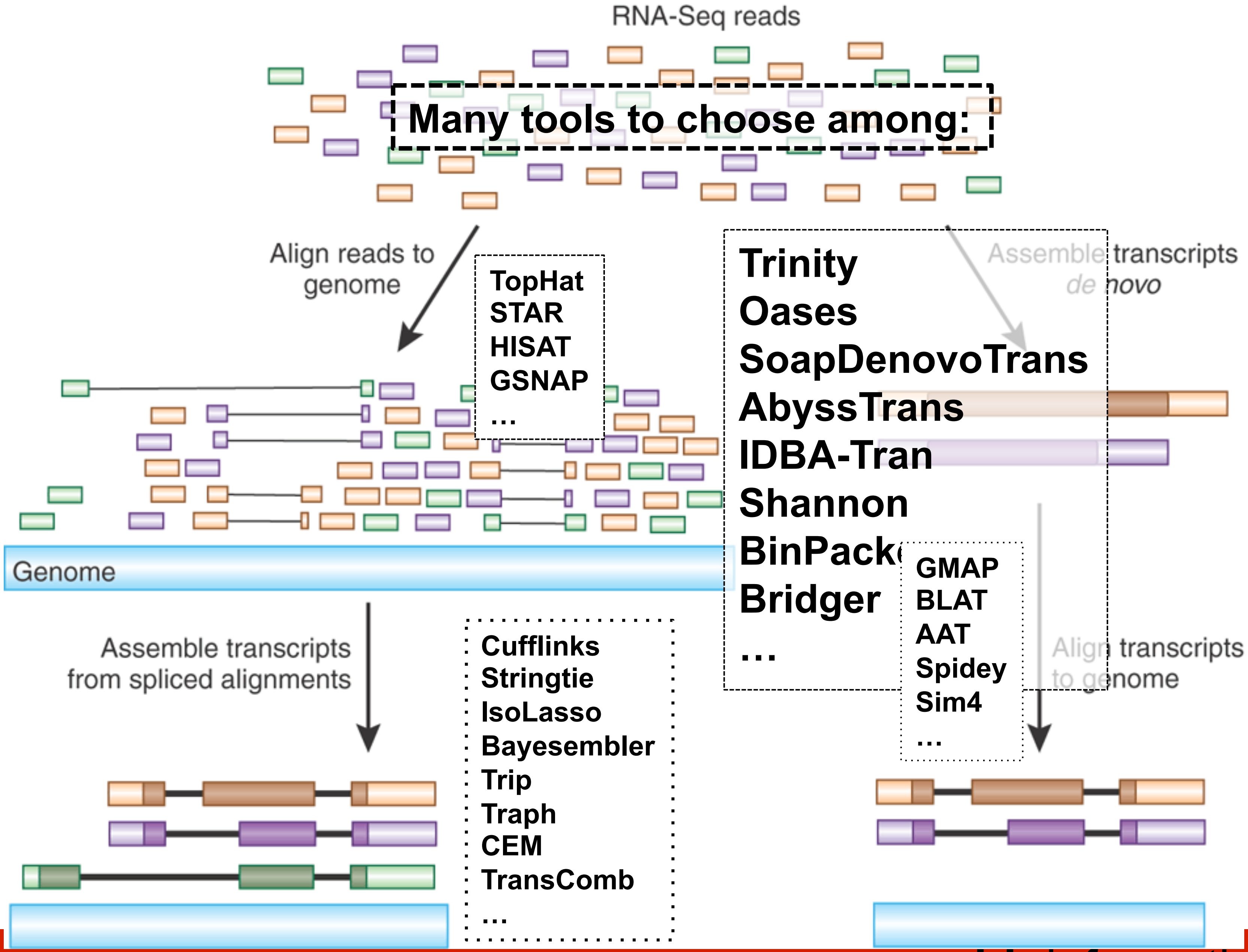
Transcript Reconstruction from RNA-Seq Reads



Transcript Reconstruction from RNA-Seq Reads



Transcript Reconstruction from RNA-Seq Reads


RNA-Seq reads
Align reads to genome
Assemble transcripts de novo
Genome
Assemble transcripts from spliced alignments
Align transcripts to genome

Many tools to choose among:

- TopHat
- STAR
- HISAT
- GSNAP
- …

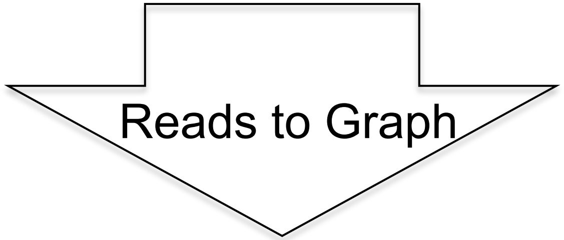
- Trinity
- Oases
- SoapDenovoTrans
- AbyssTrans
- IDBA-Tran
- Shannon
- BinPacker
- Bridger
- …

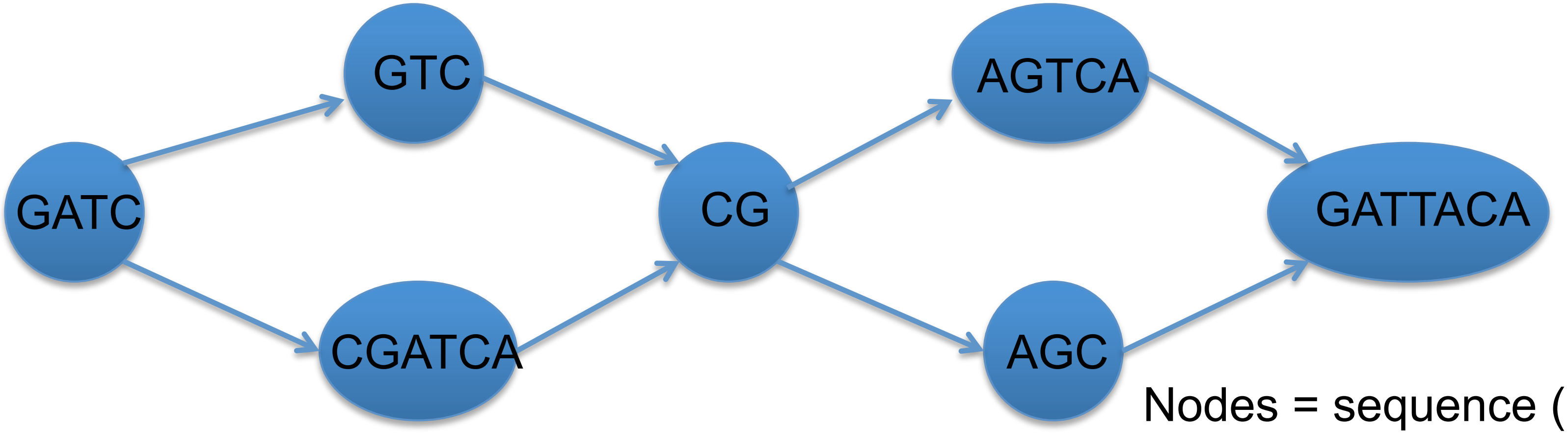
- GMAP
- BLAT
- AAT
- Spidey
- Sim4
- …

- Cufflinks
- Stringtie
- IsoLasso
- Bayesembler
- Trip
- Traph
- CEM
- TransComb
- …

Graph Data Structures Commonly Used For Assembly

- Sequence
- Order
- Orientation (+, -)
- Overlap





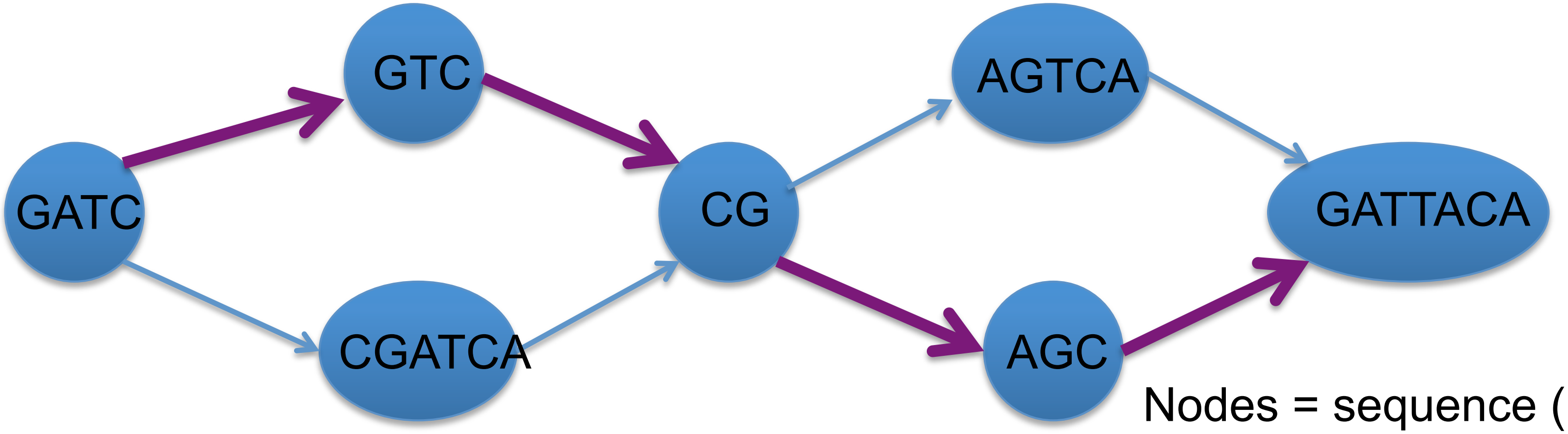
Nodes = sequence (+/-)
Edges = order, overlap

Graph Data Structures Commonly Used For Assembly

RNA-Seq reads

- Sequence
- Order
- Orientation (+, -)
- Overlap

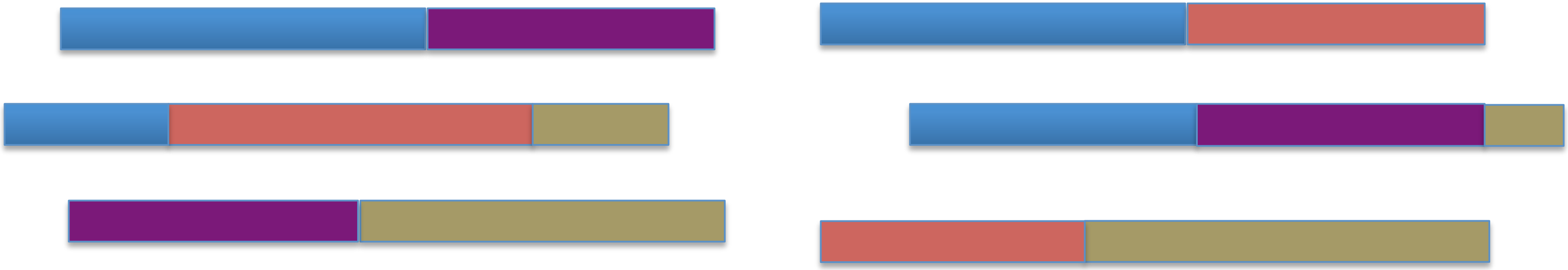
Reads to Graph



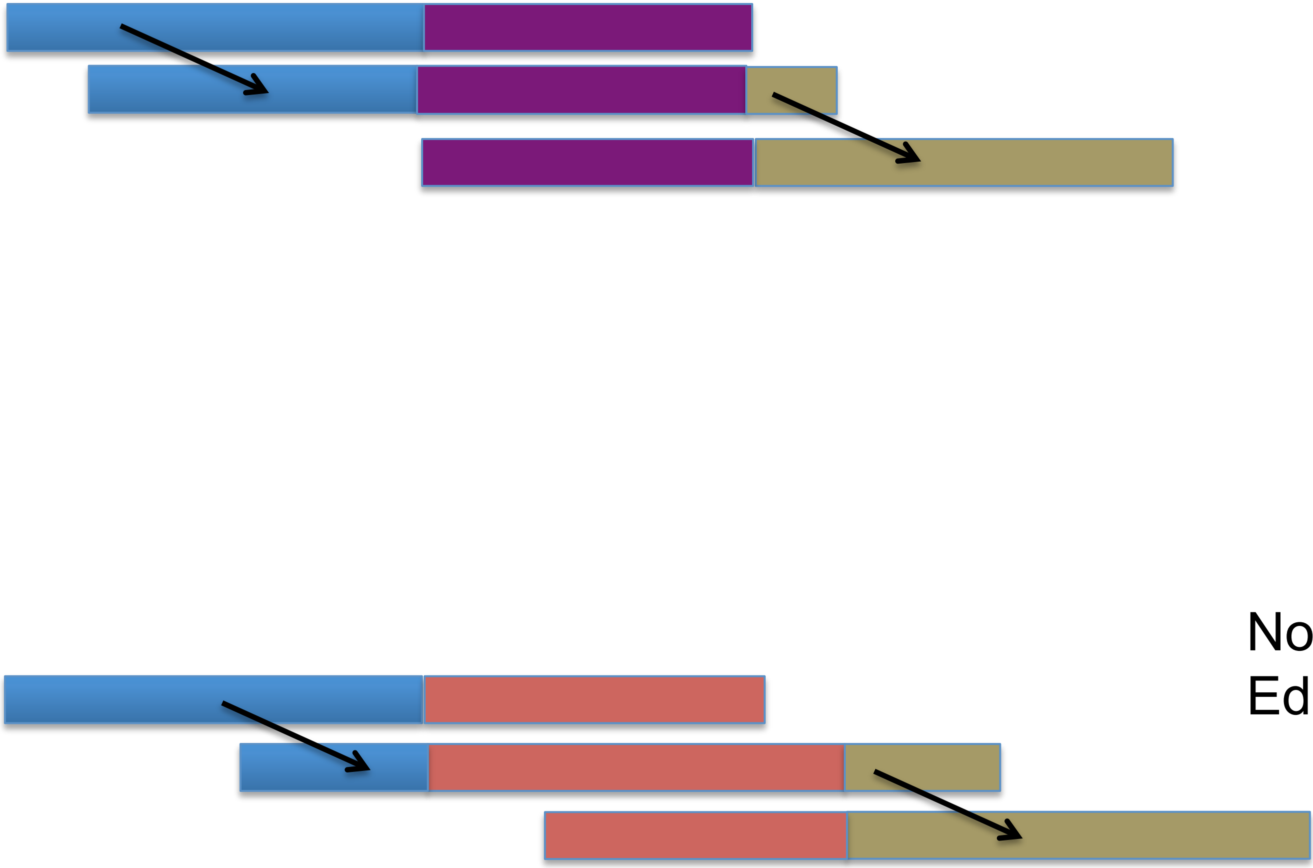
GATCGTCCGAGCGATTACA

Nodes = sequence (+/-)
Edges = order, overlap

Read Overlap Graph: Reads as nodes, overlaps as edges

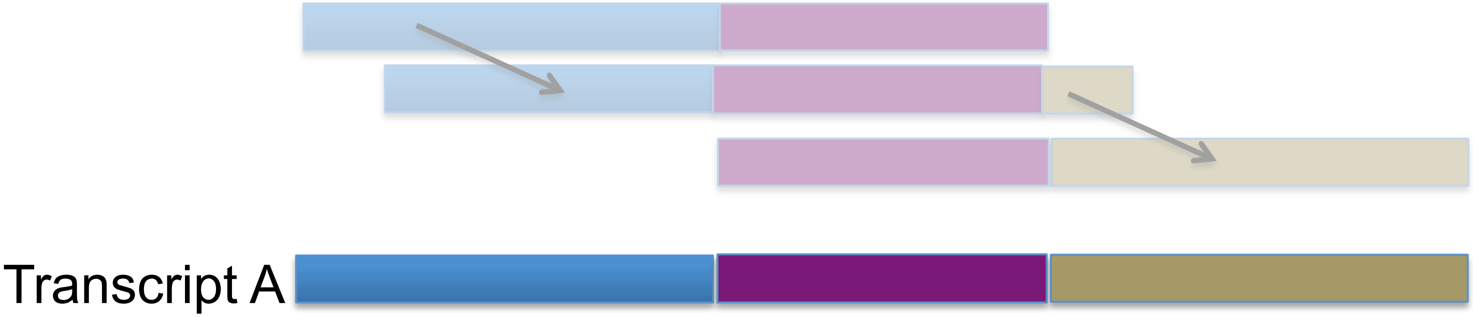


Read Overlap Graph: Reads as nodes, overlaps as edges



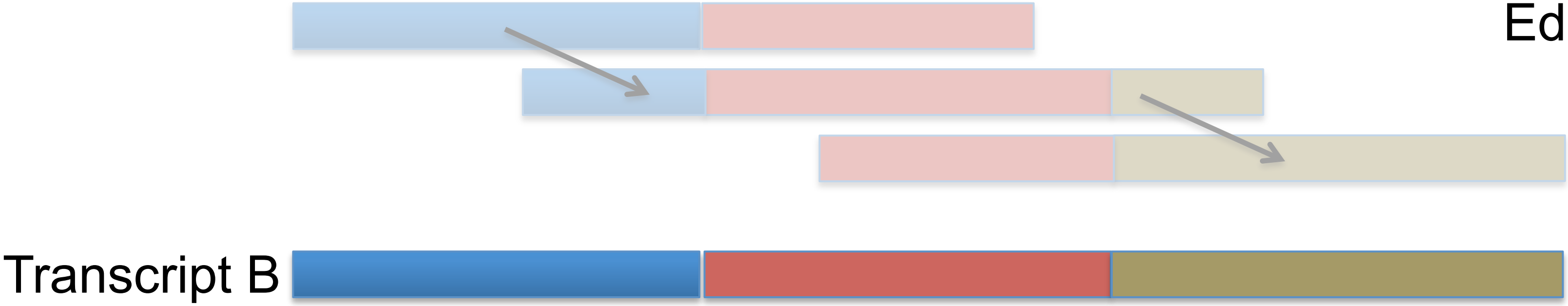
Node = read
Edge = overlap

Read Overlap Graph: Reads as nodes, overlaps as edges

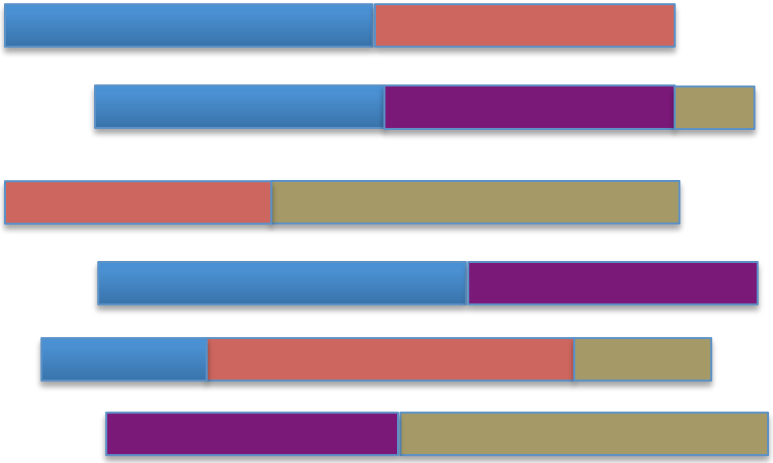


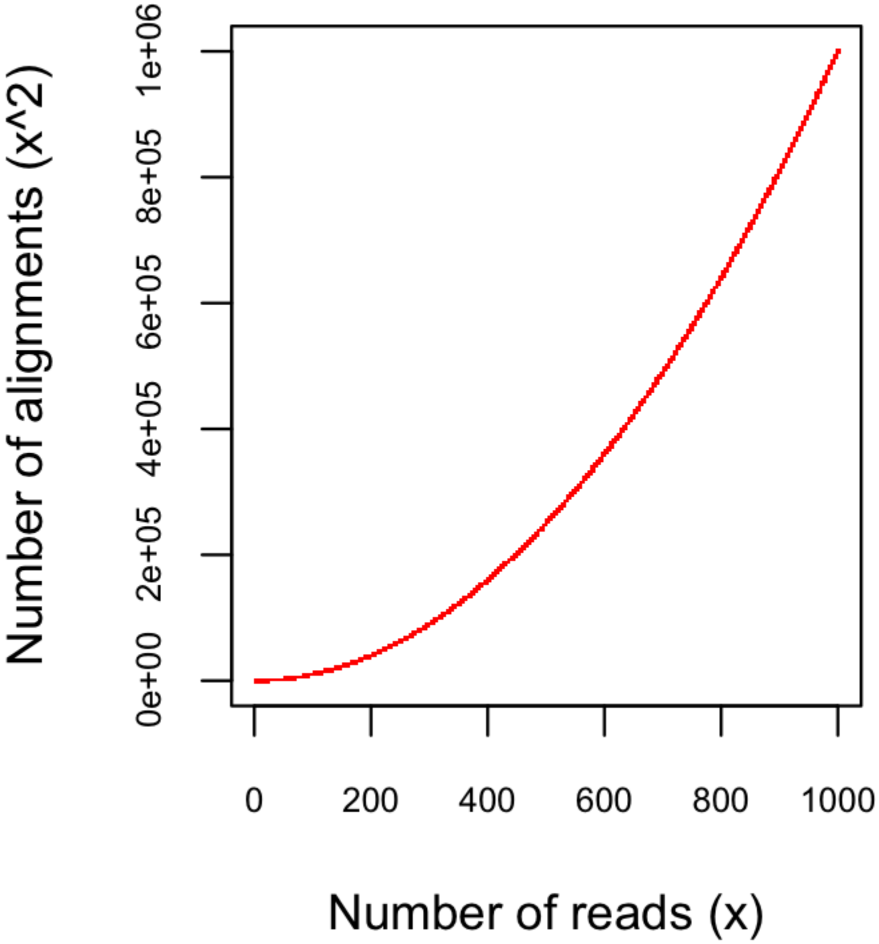
Generate consensus sequence where reads overlap

Node = read
Edge = overlap



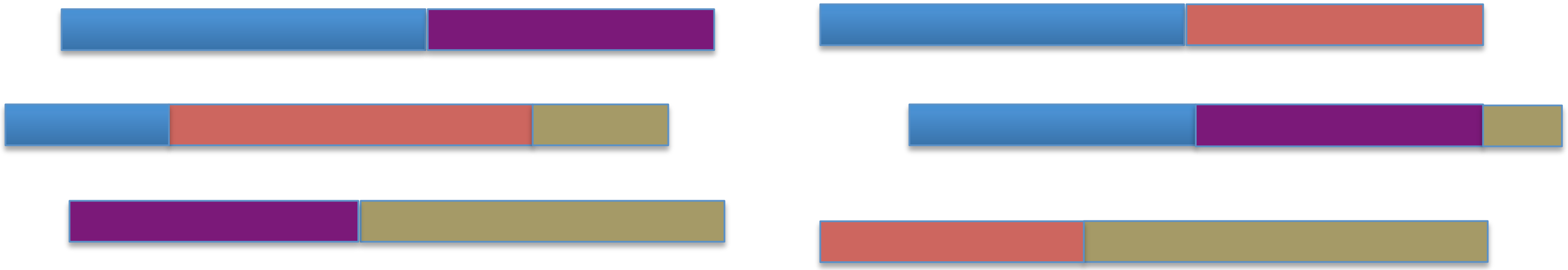
Finding pairwise overlaps between *n* reads involves ~ n^2 comparisons.





Impractical for typical RNA-Seq data (50M reads)

No genome to align to… De novo assembly required

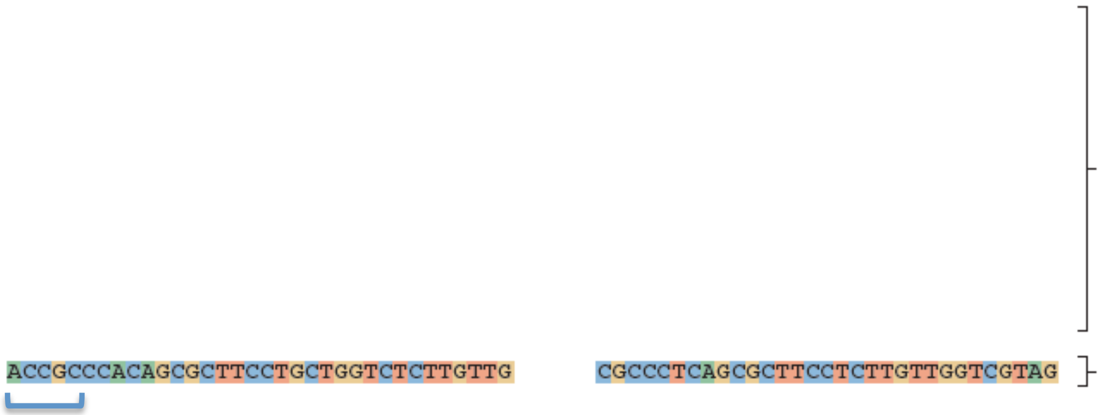


Want to avoid ***n***^{2} read alignments to define overlaps

Use a de Bruijn graph

Sequence Assembly via De Bruijn Graphs

Generate all substrings of length k from the reads


ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG
CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG

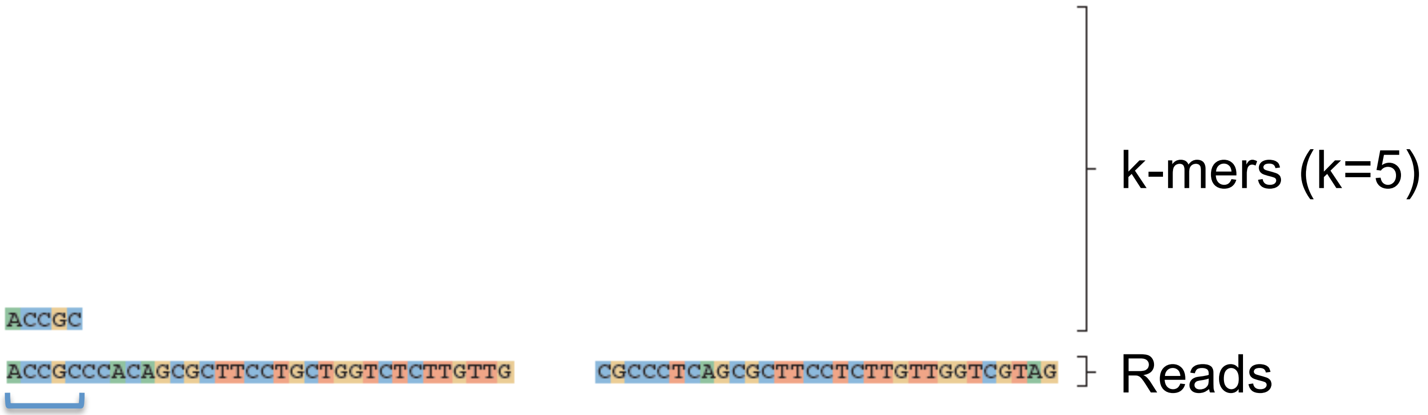
k-mers (k=5)

Reads

From Martin & Wang, Nat. Rev. Genet. 2011

Sequence Assembly via De Bruijn Graphs

Generate all substrings of length k from the reads



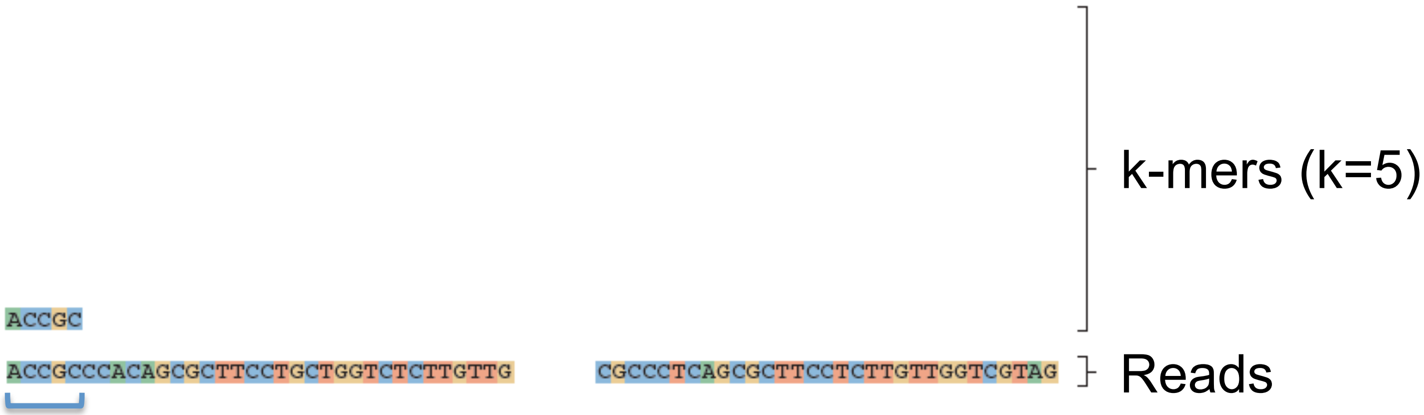
ACCGC

ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG   CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG

From Martin & Wang, Nat. Rev. Genet. 2011

Sequence Assembly via De Bruijn Graphs

Generate all substrings of length k from the reads


ACCGC
ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG
CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG
k-mers (k=5)
Reads

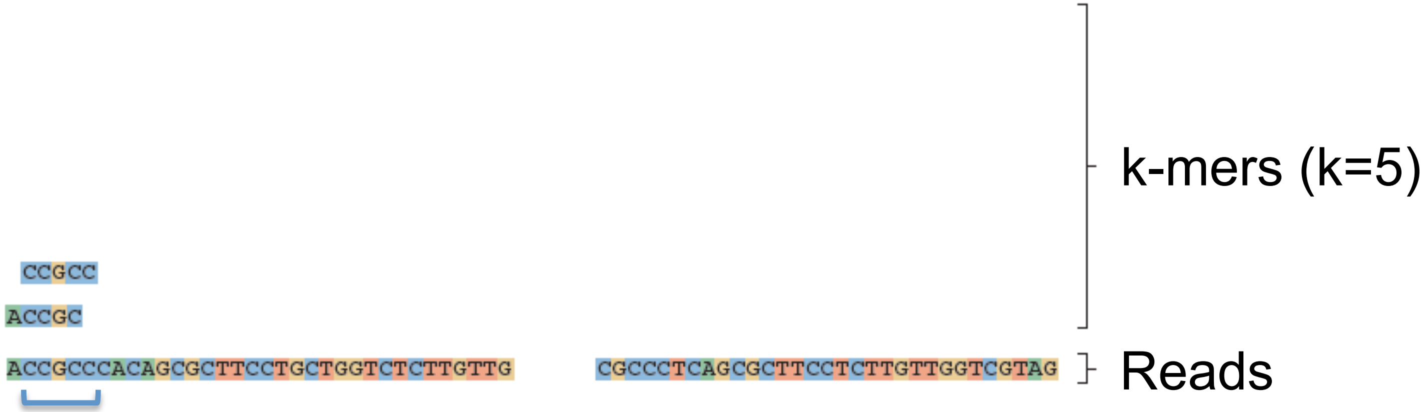
Construct the de Bruijn graph

From Martin & Wang, Nat. Rev. Genet. 2011

Nodes = unique k-mers, Edges = overlap by (k-1)

Sequence Assembly via De Bruijn Graphs

Generate all substrings of length k from the reads


CCGCC
ACCGC
ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG
CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG
k-mers (k=5)
Reads

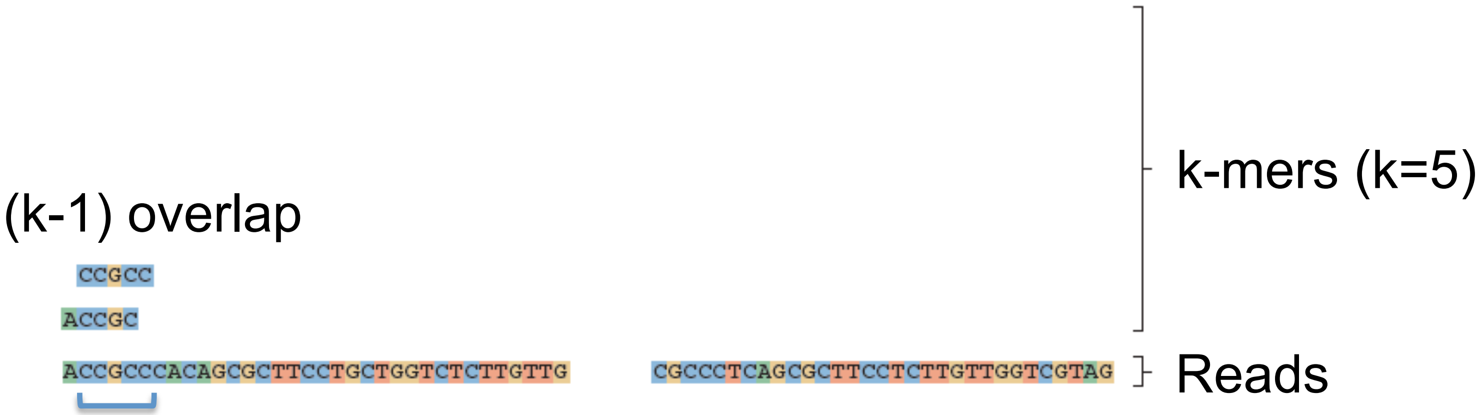
Construct the de Bruijn graph

From Martin & Wang, Nat. Rev. Genet. 2011

Nodes = unique k-mers, Edges = overlap by (k-1)

Sequence Assembly via De Bruijn Graphs

Generate all substrings of length k from the reads



k-mers (k=5)

(k-1) overlap

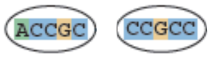
CCGCC

ACCGC

ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG   CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG

Reads

Construct the de Bruijn graph



ACCGC

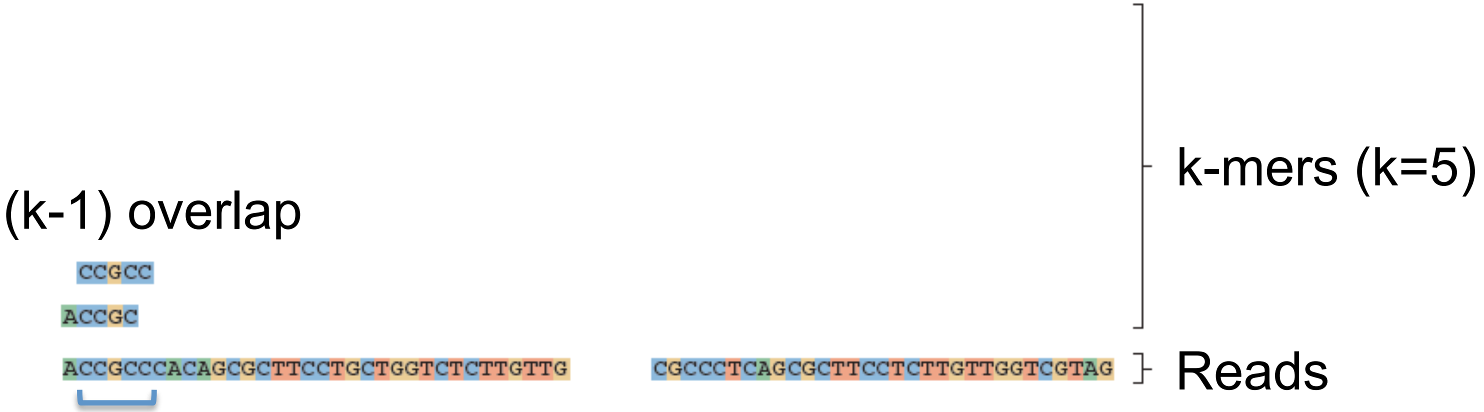
CCGCC

From Martin & Wang, Nat. Rev. Genet. 2011

Nodes = unique k-mers, Edges = overlap by (k-1)

Sequence Assembly via De Bruijn Graphs

Generate all substrings of length k from the reads



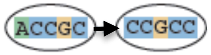
k-mers (k=5)

(k-1) overlap

CCGCC
ACCGC
ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTG
CGCCCTCAGCGCTTCCTCTTGTTGGTCGTAG

Reads

Construct the de Bruijn graph

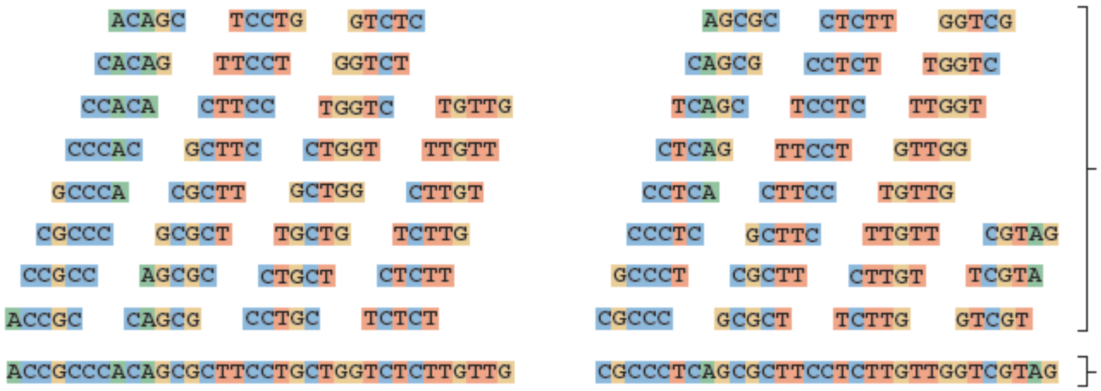


From Martin & Wang, Nat. Rev. Genet. 2011

Nodes = unique k-mers, Edges = overlap by (k-1)

Sequence Assembly via De Bruijn Graphs

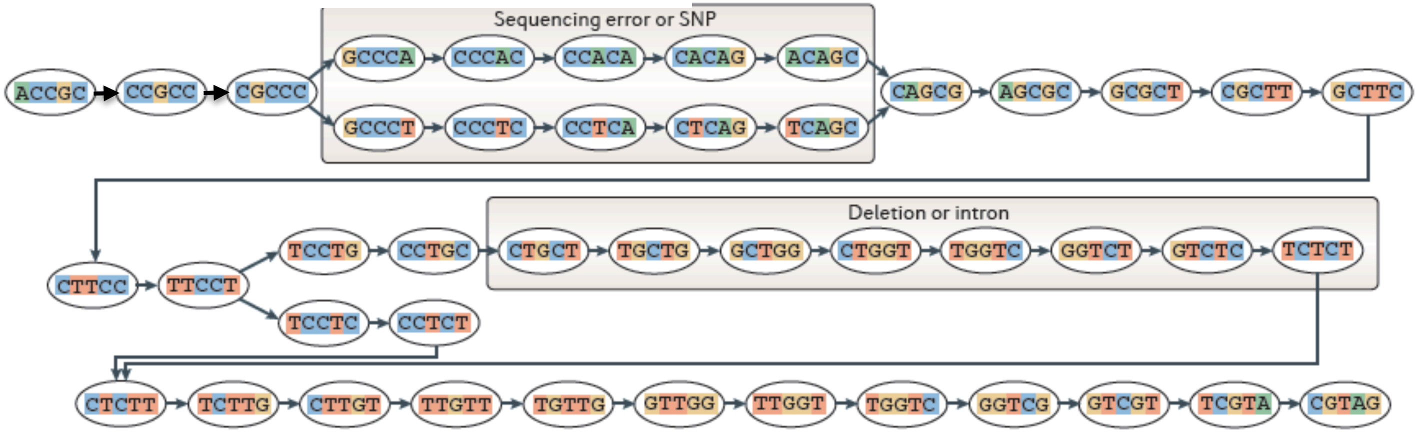
Generate all substrings of length k from the reads



k-mers (k=5)

Reads

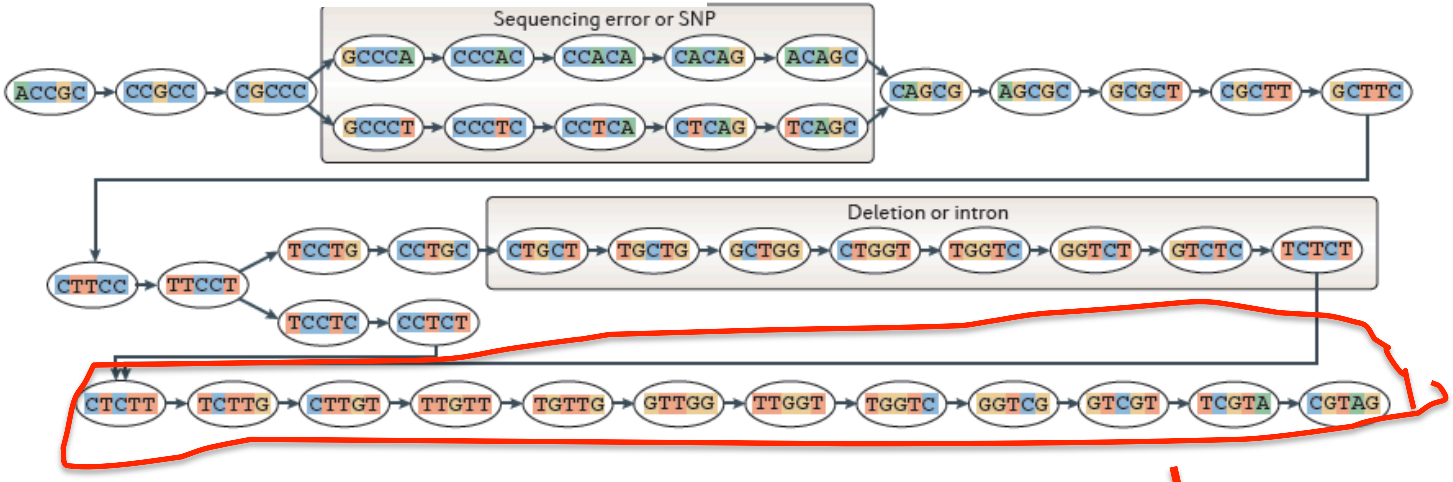
Construct the de Bruijn graph



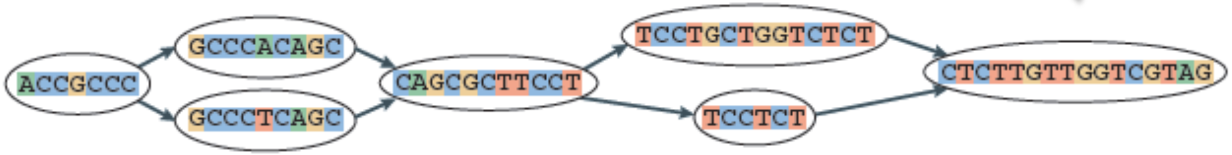
From Martin & Wang, Nat. Rev. Genet. 2011

Nodes = unique k-mers, Edges = overlap by (k-1)

Construct the de Bruijn graph

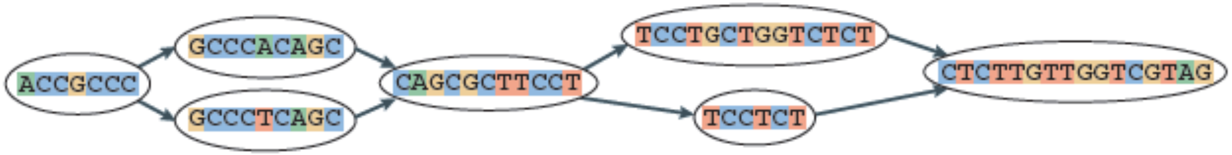


Collapse the de Bruijn graph

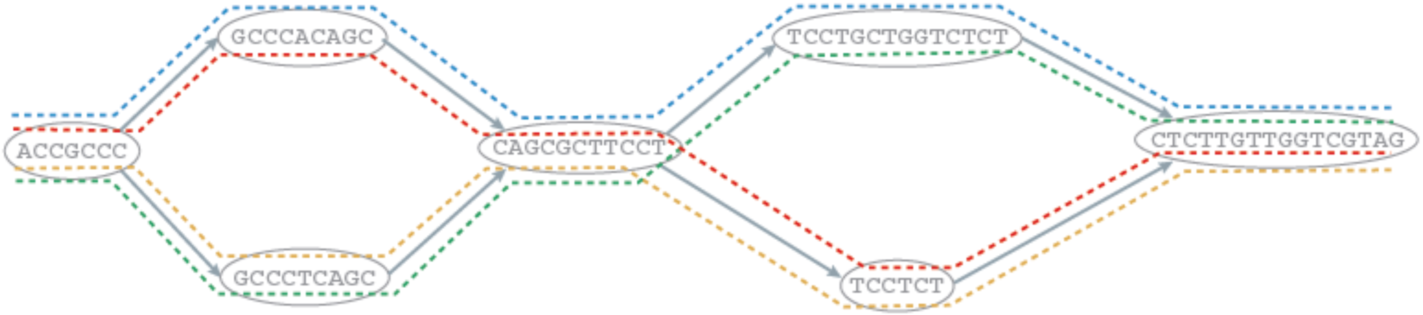


From Martin & Wang, Nat. Rev. Genet. 2011

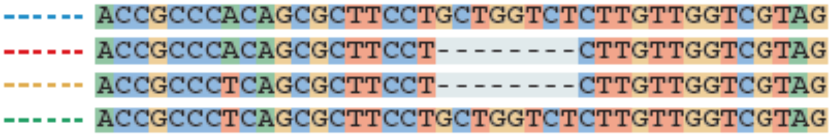
Collapse the de Bruijn graph



Traverse the graph



Assemble Transcript Isoforms


ACCGCCCACAGCGCTTCCTGCTGGTCTCTTGTTGGTCGTAG
ACCGCCCACAGCGCTTCCT--------CTTGTTGGTCGTAG
ACCGCCCTCAGCGCTTCCT--------CTTGTTGGTCGTAG
ACCGCCCTCAGCGCTTCCTGCTGGTCTCTTGTTGGTCGTAG

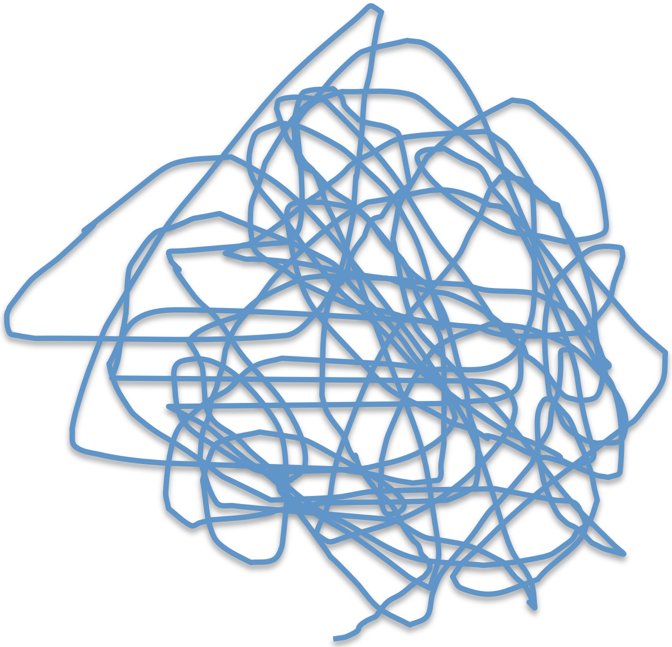
From Martin & Wang, Nat. Rev. Genet. 2011

Contrasting Genome and Transcriptome *De novo* Assembly

Genome Assembly	**Transcriptome Assembly**
Uniform coverage	Exponentially distributed coverage levels
Single contig per locus	Multiple contigs per locus (alt splicing)
Assemble small numbers of large Mb-length chromosomes	Assemble many thousands of Kb-length transcripts
Double-stranded data	Strand-specific data available

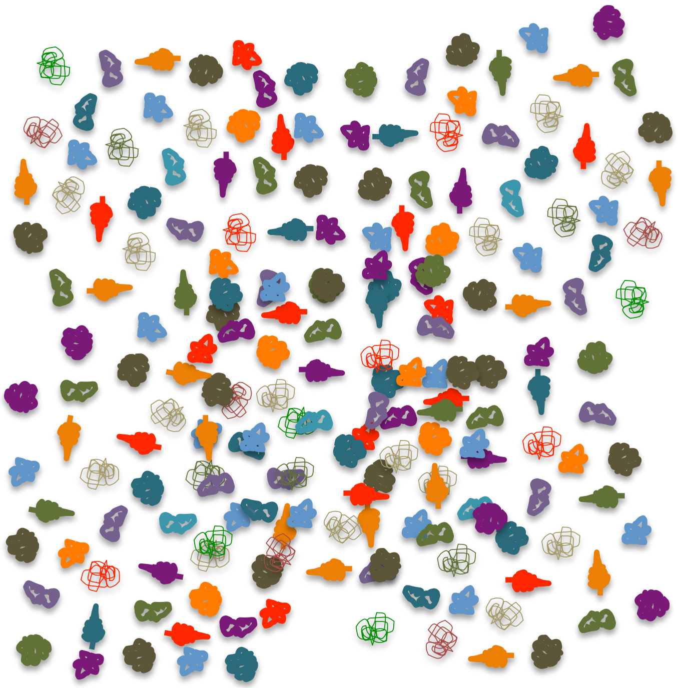
Trinity Aggregates Isolated Transcript Graphs

Genome Assembly
Single Massive Graph



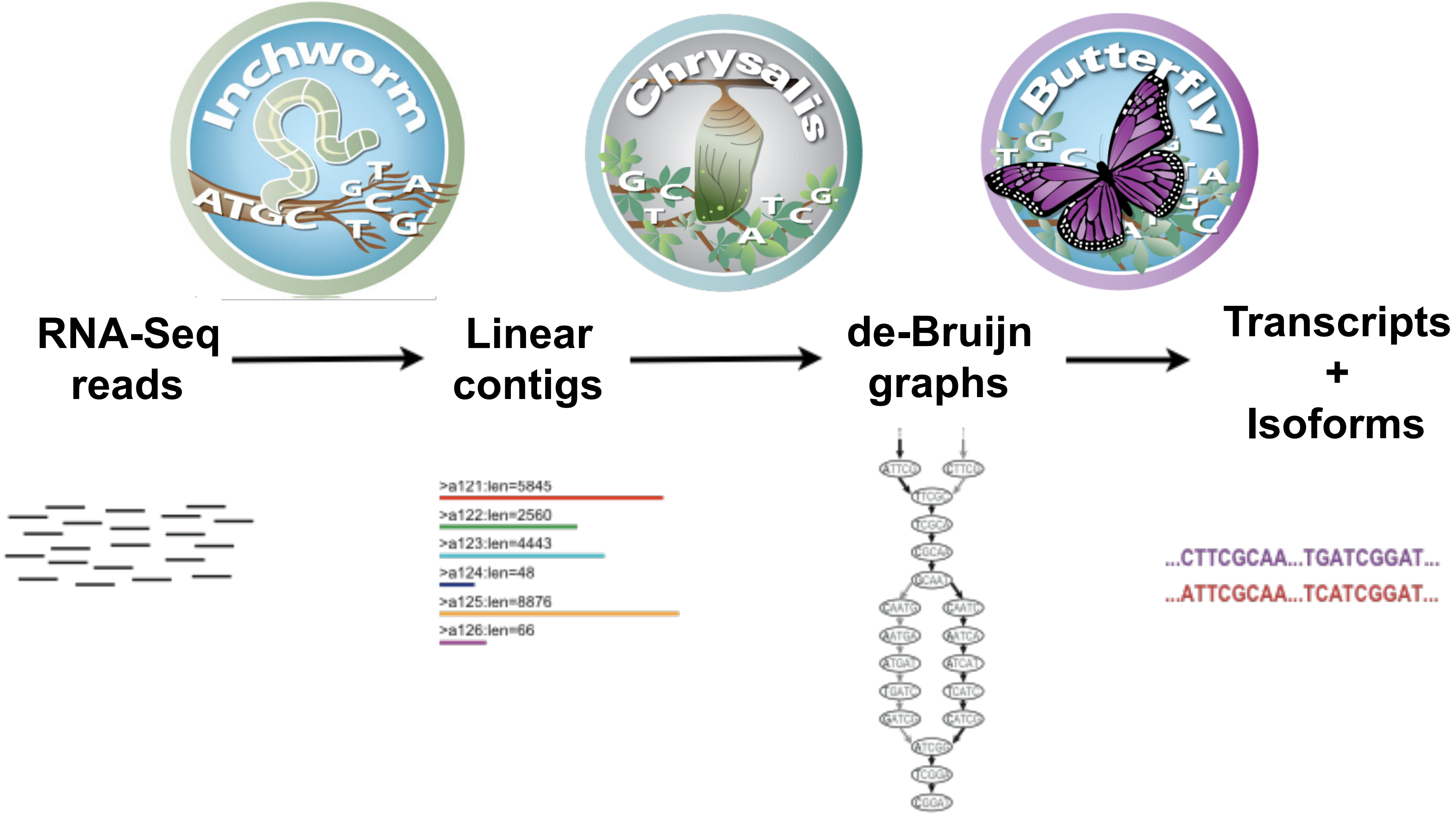
Entire chromosomes represented.

Trinity Transcriptome Assembly
Many Thousands of Small Graphs



Ideally, one graph per expressed gene.

Trinity – How it works:



RNA-Seq reads → Linear contigs → de-Bruijn graphs → Transcripts + Isoforms

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

...CTTCGCAA...TGATCGGAT...

...ATTCGCAA...TCATCGGAT...

Thousands of disjoint graphs

Inchworm Algorithm

- Decompose all reads into overlapping Kmers => hashtable(kmer, count)

Read: **AATGTGAAAACTGGATTACATGCTGGTATGTC…**

AATGTGA

ATGTGAA

TGTGAAA

…

Overlapping kmers of length (k)

Kmer Catalog (hashtable)

Kmer	Count among all reads
AATGTGA	4
ATGTGAA	2
TGTGAAA	1
GATTACA	9

Inchworm Algorithm

- Decompose all reads into overlapping Kmers => hashtable(kmer, count)
- Identify seed kmer as most abundant Kmer, ignoring low-complexity kmers.

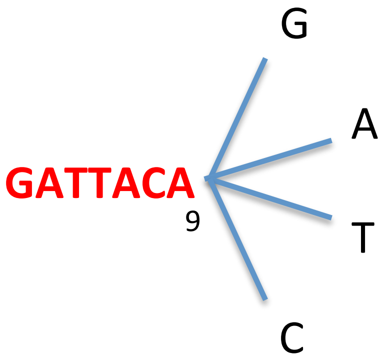
GATTACA
9

Kmer Catalog (hashtable)

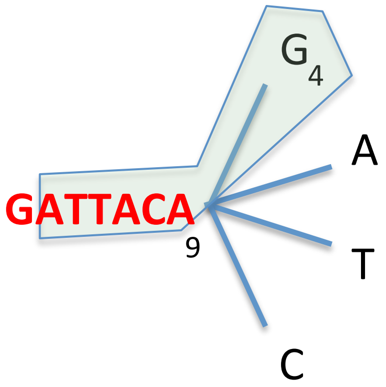
Kmer	Count among all reads
AATGTGA	4
ATGTGAA	2
TGTGAAA	1
GATTACA	9

Inchworm Algorithm

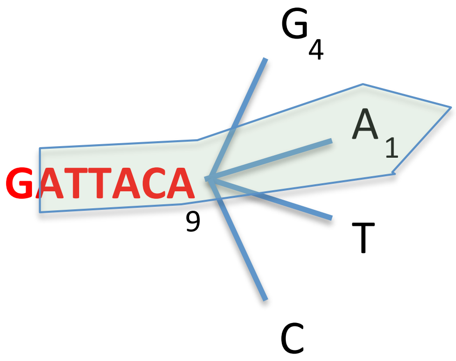
- Decompose all reads into overlapping Kmers => hashtable(kmer, count)
- Identify seed kmer as most abundant Kmer, ignoring low-complexity kmers.
- Extend kmer at 3' end, guided by coverage.



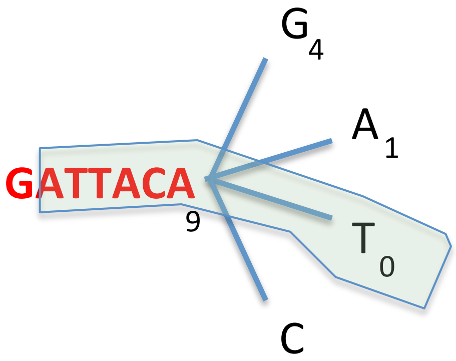
Inchworm Algorithm



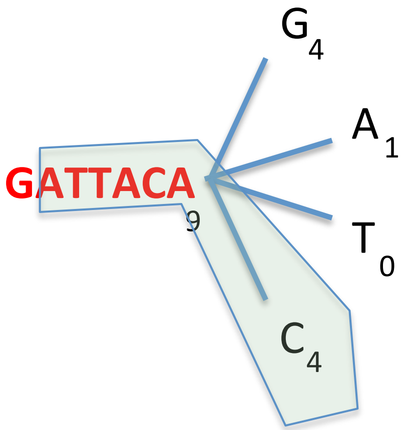
Inchworm Algorithm



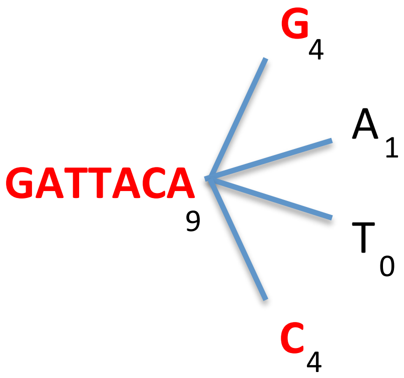
Inchworm Algorithm



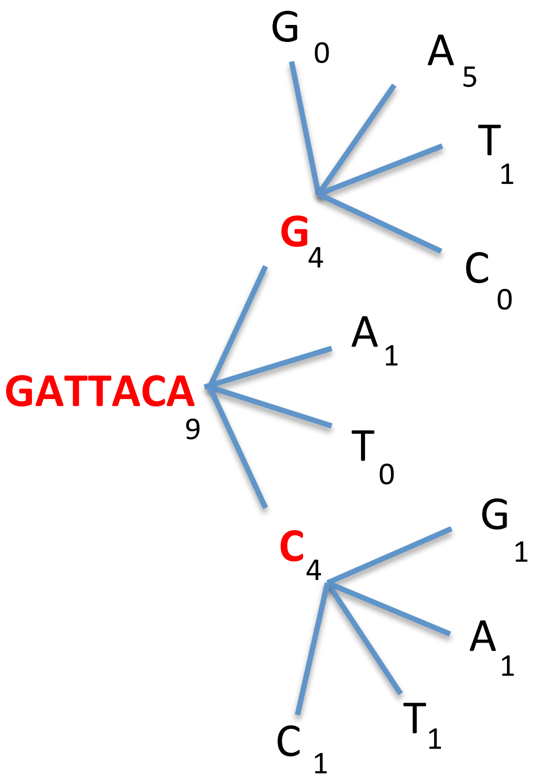
Inchworm Algorithm



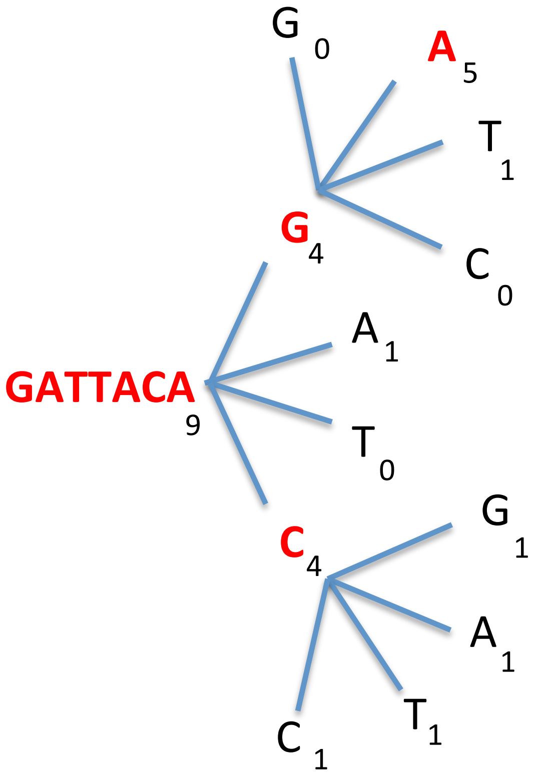
Inchworm Algorithm



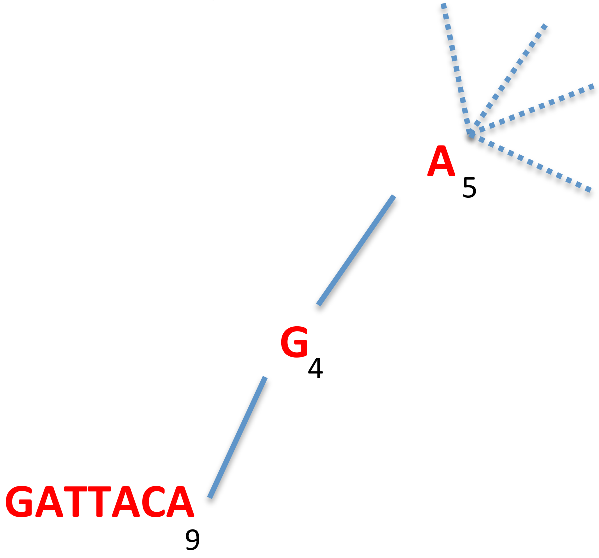
Inchworm Algorithm



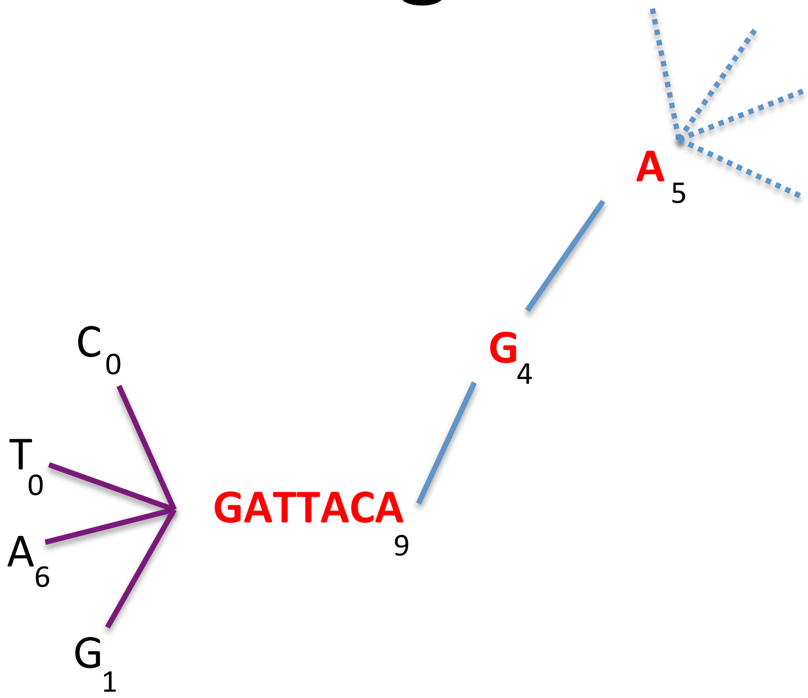
Inchworm Algorithm



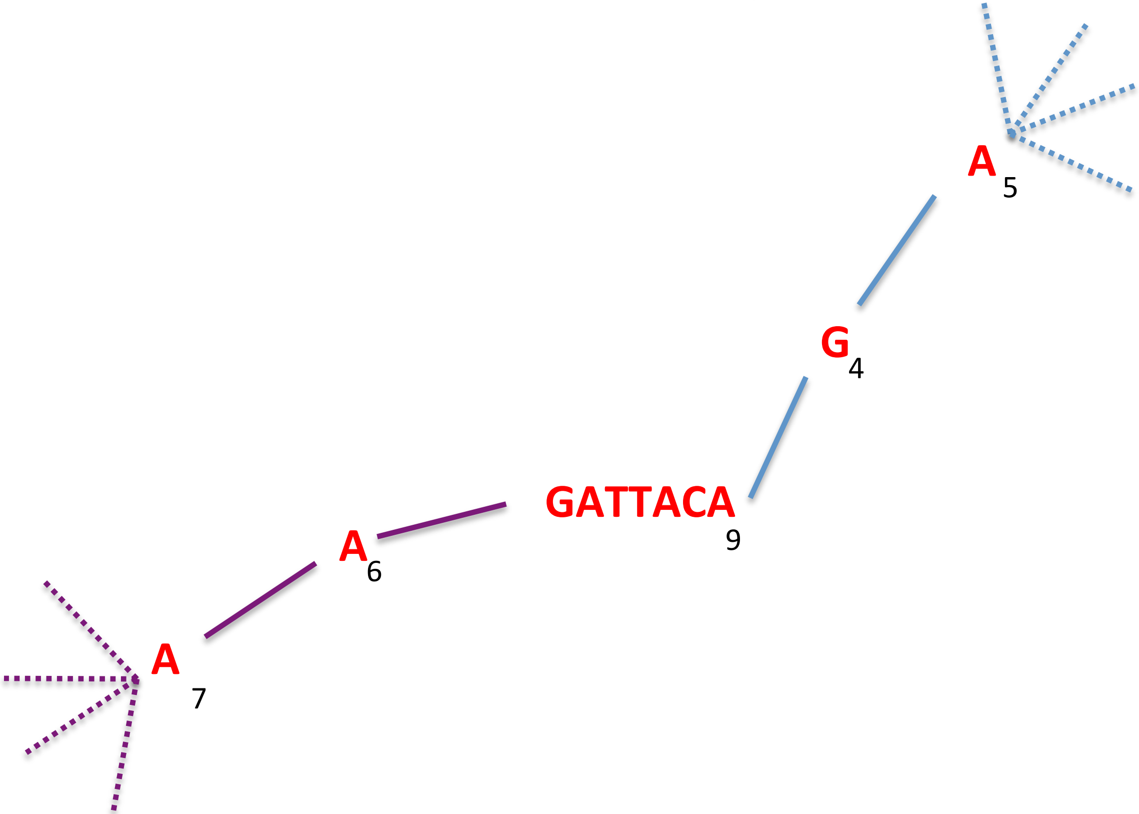
Inchworm Algorithm



Inchworm Algorithm



Inchworm Algorithm

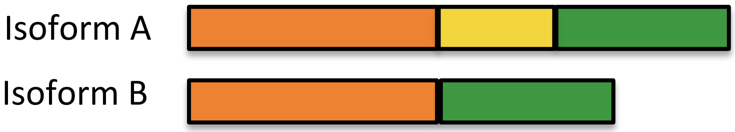


Report contig: ....AAGATTACAGA....

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

Inchworm Contigs from Alt-Spliced Transcripts

Expressed isoforms

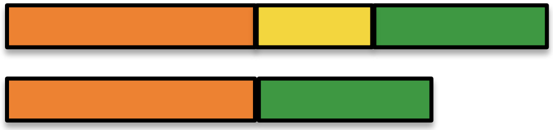


Inchworm Contigs from Alt-Spliced Transcripts

Expressed isoforms

Expression

Isoform A

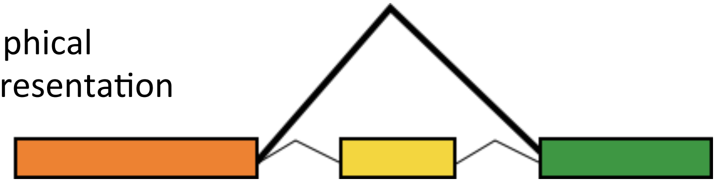


Isoform B

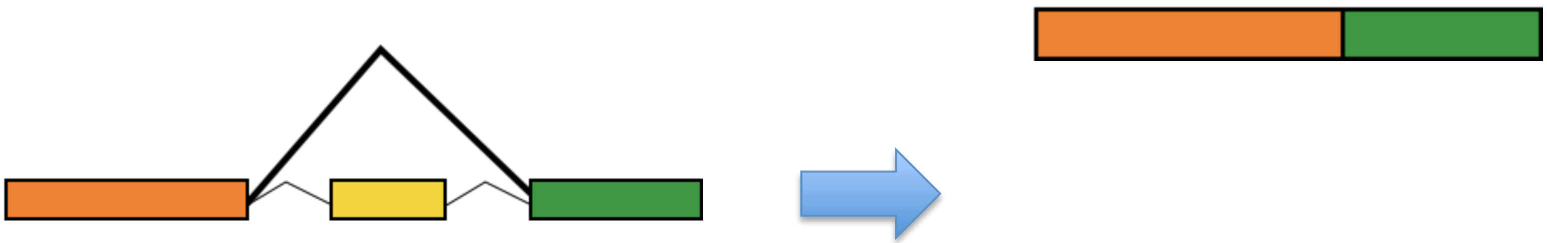
(low)

(high)

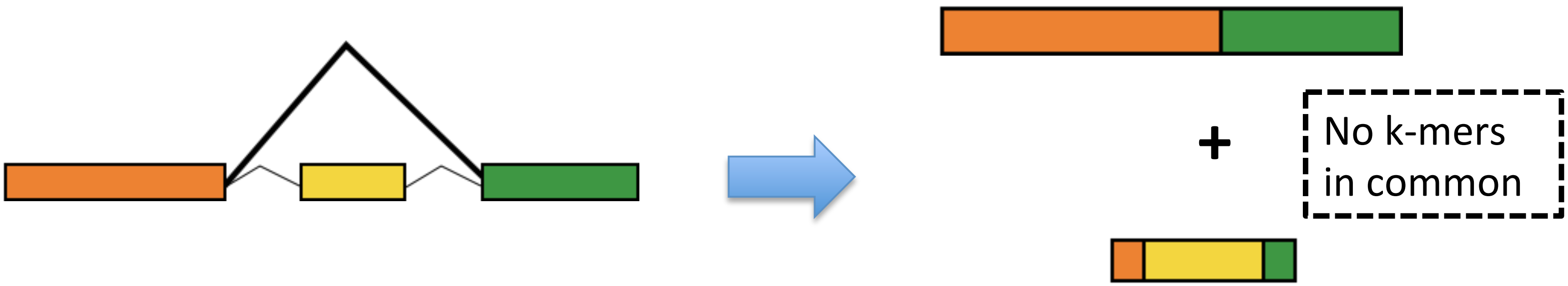
Graphical representation



Inchworm Contigs from Alt-Spliced Transcripts

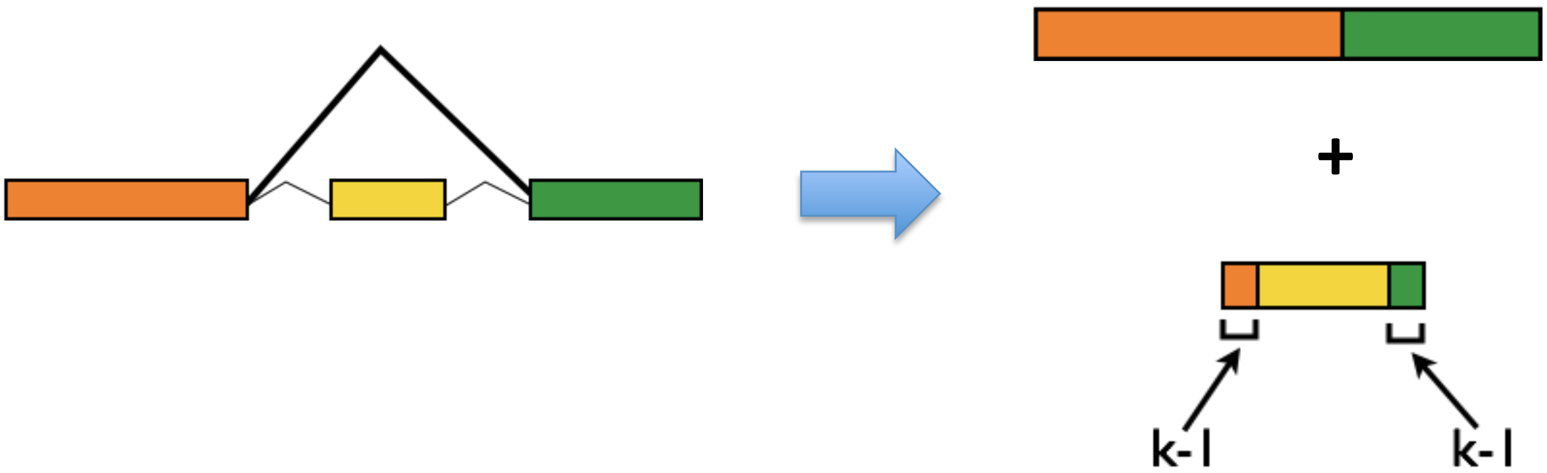


Inchworm Contigs from Alt-Spliced Transcripts

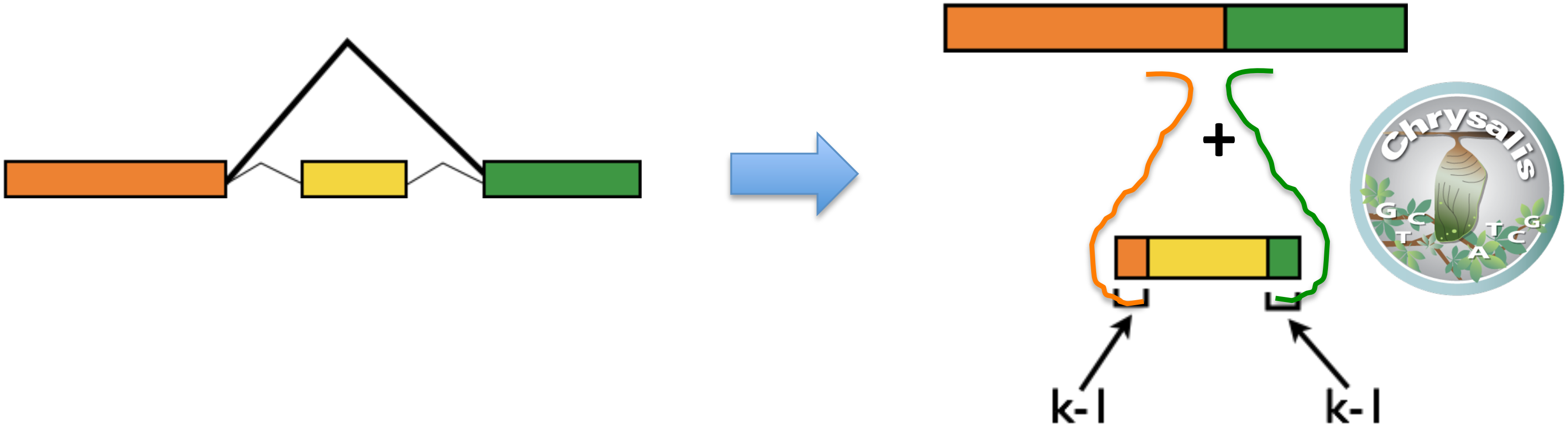


No k-mers in common

Inchworm Contigs from Alt-Spliced Transcripts

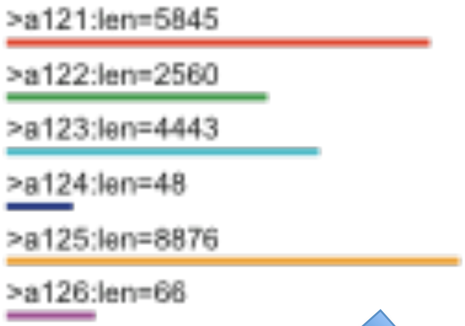


Chrysalis Re-groups Related Inchworm Contigs



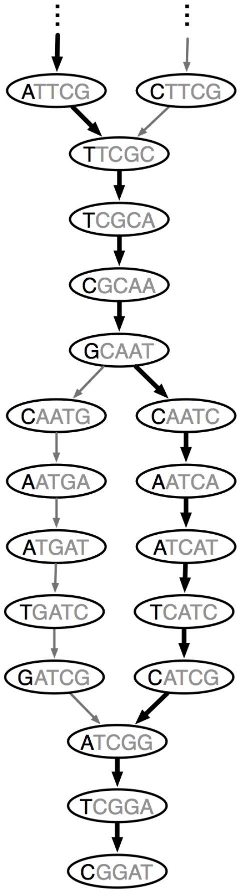
Chrysalis uses (k-1) overlaps and read support to link related Inchworm contigs

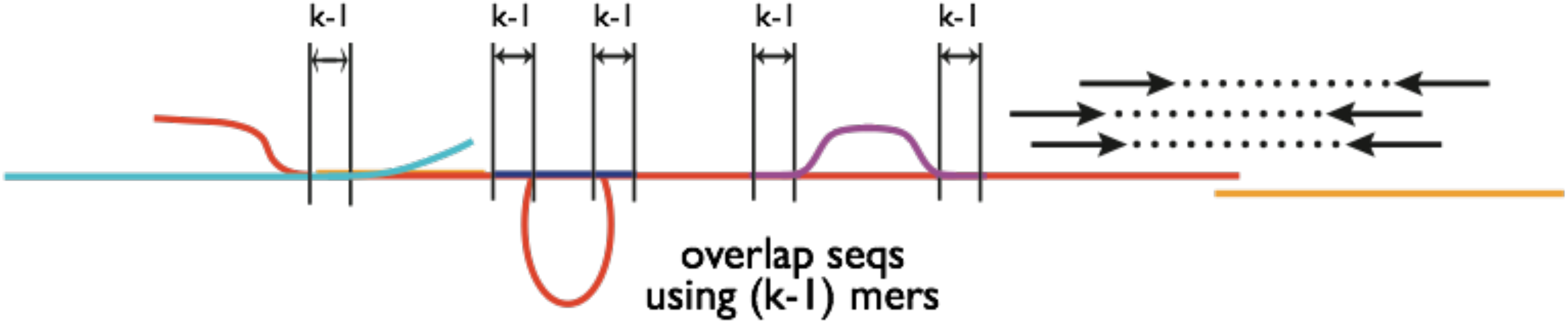
Chrysalis



Integrate isoforms via k-1 overlaps

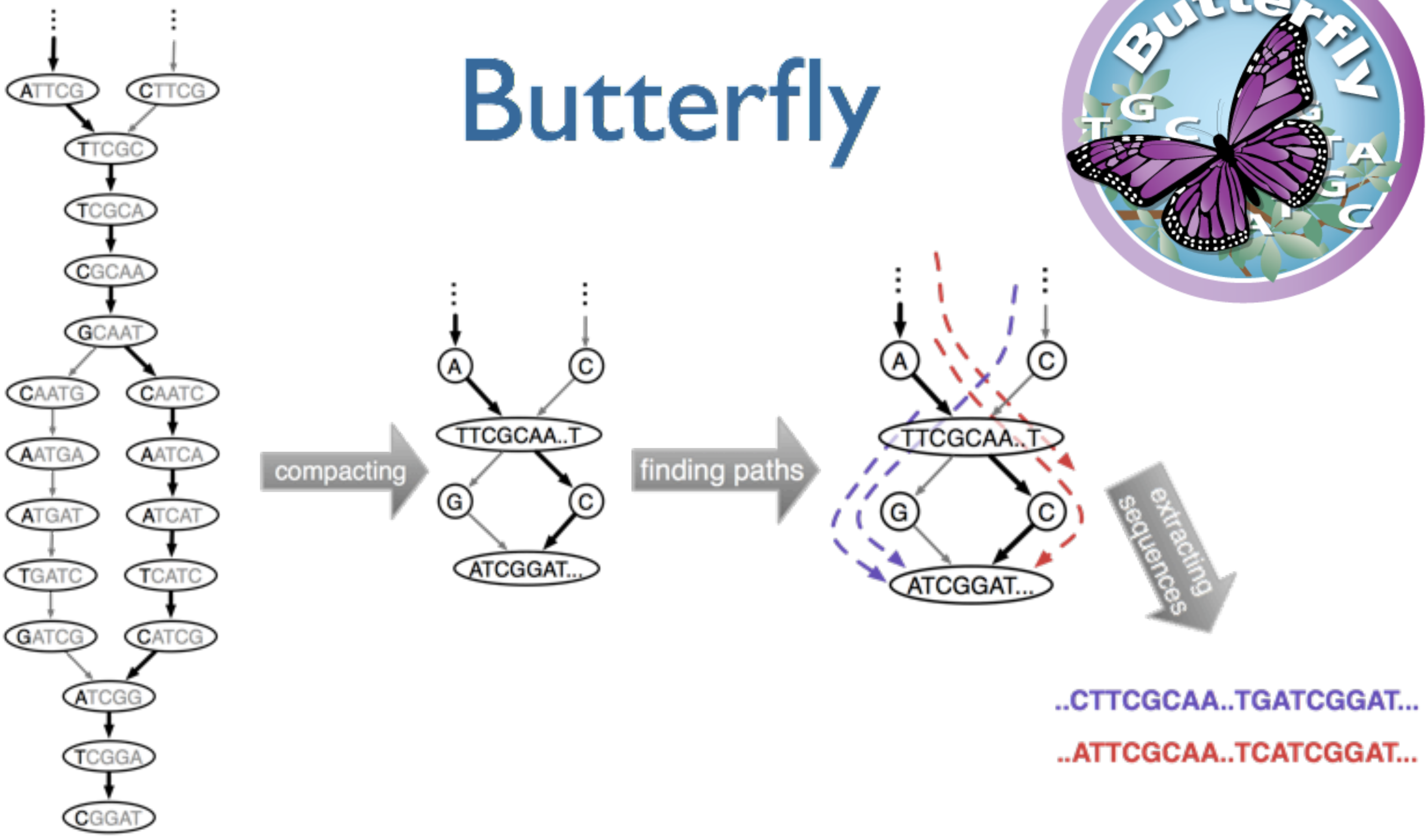
Build de Bruijn Graphs (ideally, one per gene)





Thousands of Chrysalis Clusters

Butterfly



..CTTCGCAA..TGATCGGAT...

..ATTCGCAA..TCATCGGAT...

de Bruijn graph

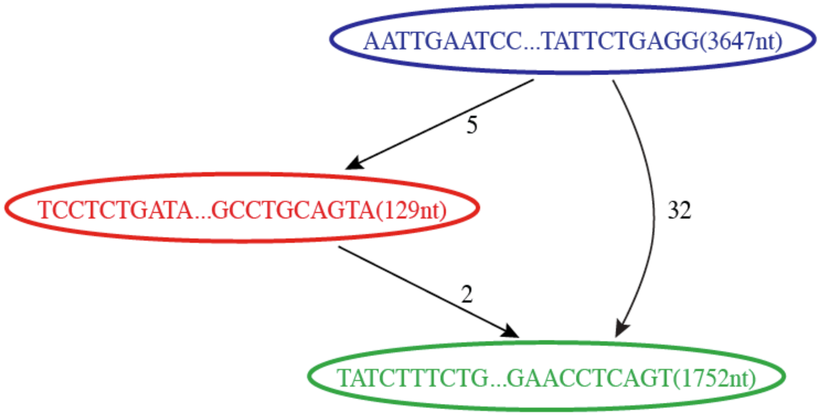
compact graph

compact graph with reads

sequences (isoforms and paralogs)

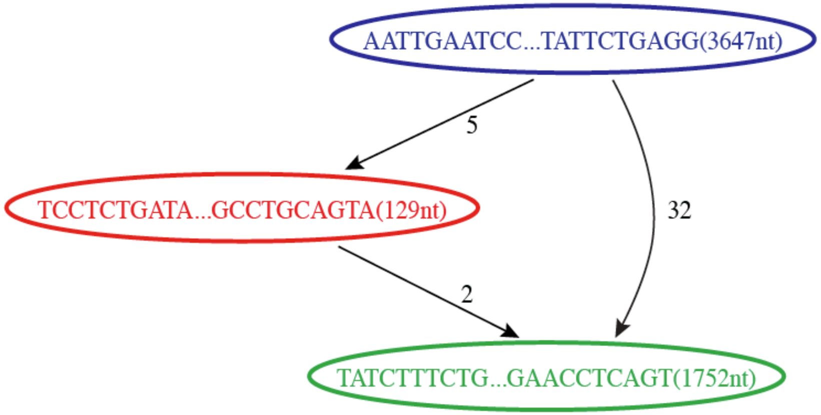
Butterfly Example 1:
Reconstruction of Alternatively Spliced Transcripts

Butterfly's Compacted Sequence Graph

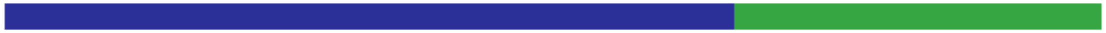

AATTGAATCC...TATTCTGAGG(3647nt)
5
32
TCCTCTGATA...GCCTGCAGTA(129nt)
2
TATCTTTCTG...GAACCTCAGT(1752nt)

Reconstruction of Alternatively Spliced Transcripts

Butterfly's Compacted Sequence Graph

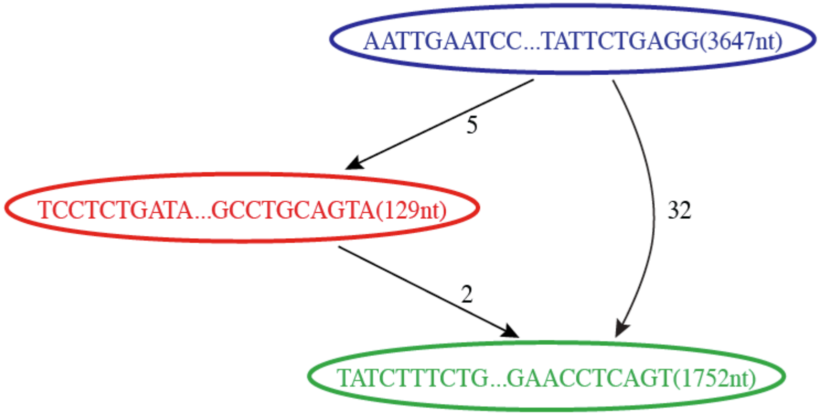

AATTGAATCC...TATTCTGAGG(3647nt)
5
32
TCCTCTGATA...GCCTGCAGTA(129nt)
2
TATCTTTCTG...GAACCTCAGT(1752nt)

Reconstructed Transcripts

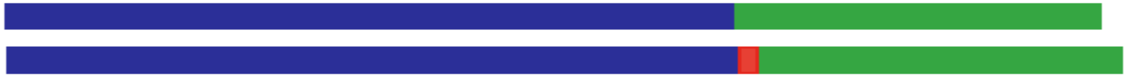


Reconstruction of Alternatively Spliced Transcripts

Butterfly’s Compacted Sequence Graph

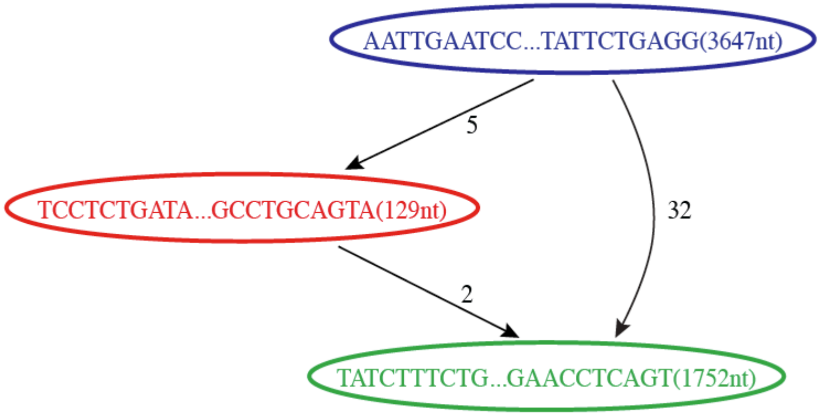

AATTGAATCC...TATTCTGAGG(3647nt)
5
32
TCCTCTGATA...GCCTGCAGTA(129nt)
2
TATCTTTCTG...GAACCTCAGT(1752nt)

Reconstructed Transcripts

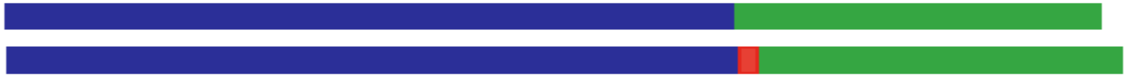


Reconstruction of Alternatively Spliced Transcripts

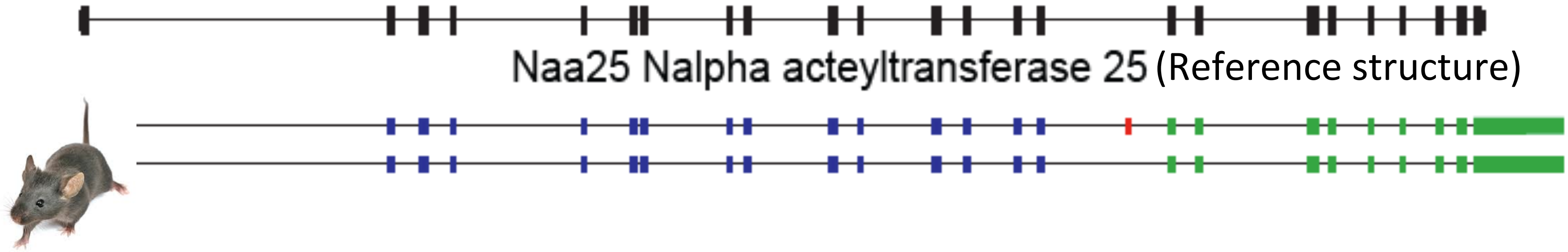
Butterfly's Compacted Sequence Graph


AATTGAATCC...TATTCTGAGG(3647nt)
5
32
TCCTCTGATA...GCCTGCAGTA(129nt)
2
TATCTTTCTG...GAACCTCAGT(1752nt)

Reconstructed Transcripts

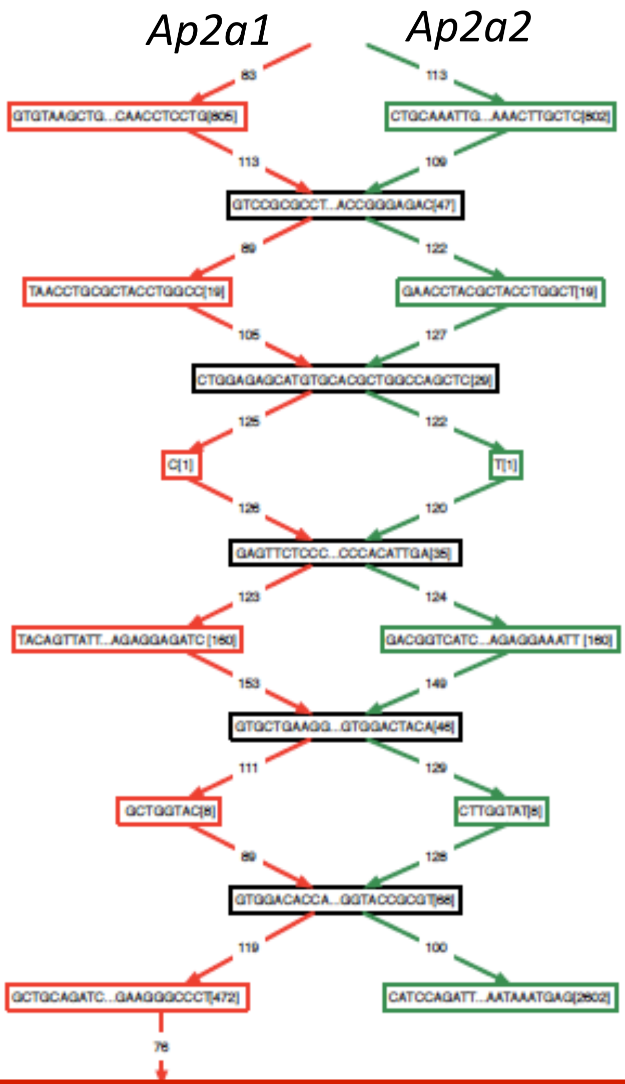


Aligned to Mouse Genome

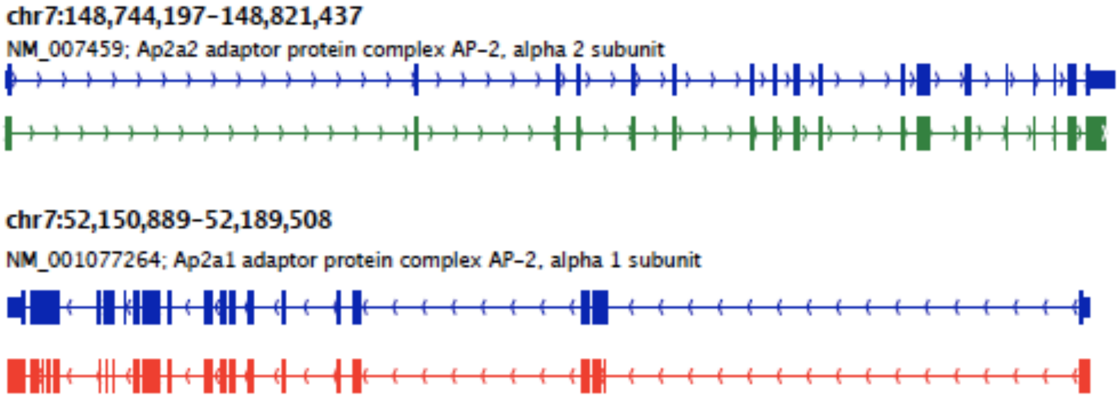


(Reference structure)

Butterfly Example 2:
Teasing Apart Transcripts of Paralogous Genes


Ap2a1
Ap2a2
83
113
GTGTAAGCTG...CAACCTCCTG[806]
CTGCAAATTG...AAACTTGCTC[802]
113
109
GTCCGCGCCT...ACCGGGAGAC[47]
89
122
TAACCTGCGCTACCTGGCC[19]
GAACCTACGCTACCTGGCT[19]
105
127
CTGGAGAGCATGTGCACGCTGGCCAGCTC[29]
125
122
C[1]
T[1]
128
120
GAGTTCTCCC...CCCACATTGA[35]
123
124
TACAGTTATT...AGAGGAGATC [160]
GACGGTCATC...AGAGGAAATT [160]
153
149
GTGCTGAAGG...GTGGACTACA[46]
111
129
GCTGGTAC[8]
CTTGGTAT[8]
89
128
GTGGACACCA...GGTACCGCGT[68]
119
100
GCTGCAGATC...GAAGGGCCCT[472]
CATCCAGATT...AATAAATGAG[2602]
78

Butterfly Example 2:
Teasing Apart Transcripts of Paralogous Genes

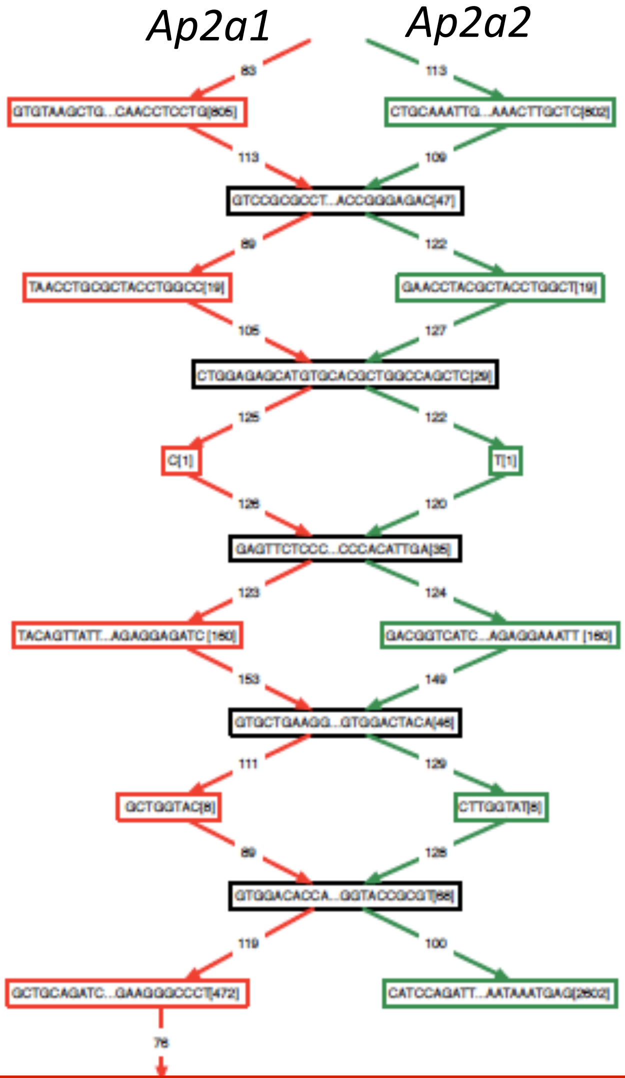


chr7:148,744,197-148,821,437

NM_007459; Ap2a2 adaptor protein complex AP-2, alpha 2 subunit

chr7:52,150,889-52,189,508

NM_001077264; Ap2a1 adaptor protein complex AP-2, alpha 1 subunit


Ap2a1
Ap2a2
83
113
GTGTAAGCTG...CAACCTCCTG[806]
CTGCAAATTG...AAACTTGCTC[802]
113
109
GTCCGCGCCT...ACCGGGAGAC[47]
89
122
TAACCTGCGCTACCTGGCC[19]
GAACCTACGCTACCTGGCT[19]
105
127
CTGGAGAGCATGTGCACGCTGGCCAGCTC[29]
125
122
C[1]
T[1]
128
120
GAGTTCTCCC...CCCACATTGA[35]
123
124
TACAGTTATT...AGAGGAGATC [160]
GACGGTCATC...AGAGGAAATT [160]
153
149
GTGCTGAAGG...GTGGACTACA[48]
111
129
GCTGGTAC[8]
CTTGGTAT[8]
89
128
GTGGACACCA...GGTACCGCGT[68]
119
100
GCTGCAGATC...GAAGGGCCCT[472]
CATCCAGATT...AATAAATGAG[2802]
78

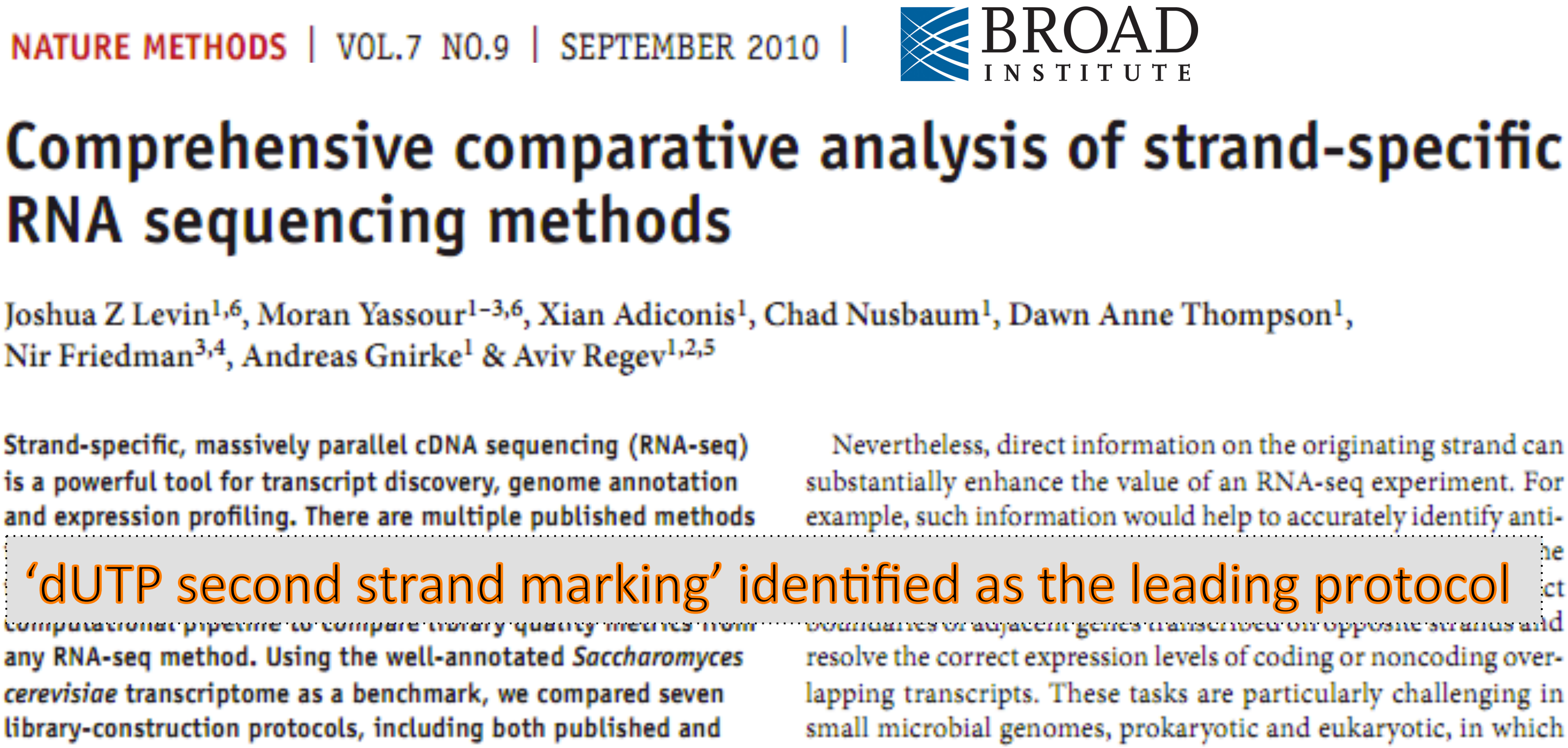
Strand-specific RNA-Seq is Preferred

Computationally: fewer confounding graph structures in de novo assembly:

ex. Forward != reverse complement

(GGAA != TTCC)

Biologically: separate sense vs. antisense transcription



NATURE METHODS | VOL.7 NO.9 | SEPTEMBER 2010 |

BROAD INSTITUTE

Comprehensive comparative analysis of strand-specific RNA sequencing methods

Joshua Z Levin[1,6], Moran Yassour[1–3,6], Xian Adiconis[1], Chad Nusbaum[1], Dawn Anne Thompson[1], Nir Friedman[3,4], Andreas Gnirke[1] & Aviv Regev[1,2,5]

Strand-specific, massively parallel cDNA sequencing (RNA-seq) is a powerful tool for transcript discovery, genome annotation and expression profiling. There are multiple published methods

computational pipeline to compare library quality metrics from any RNA-seq method. Using the well-annotated *Saccharomyces cerevisiae* transcriptome as a benchmark, we compared seven library-construction protocols, including both published and

Nevertheless, direct information on the originating strand can substantially enhance the value of an RNA-seq experiment. For example, such information would help to accurately identify anti-

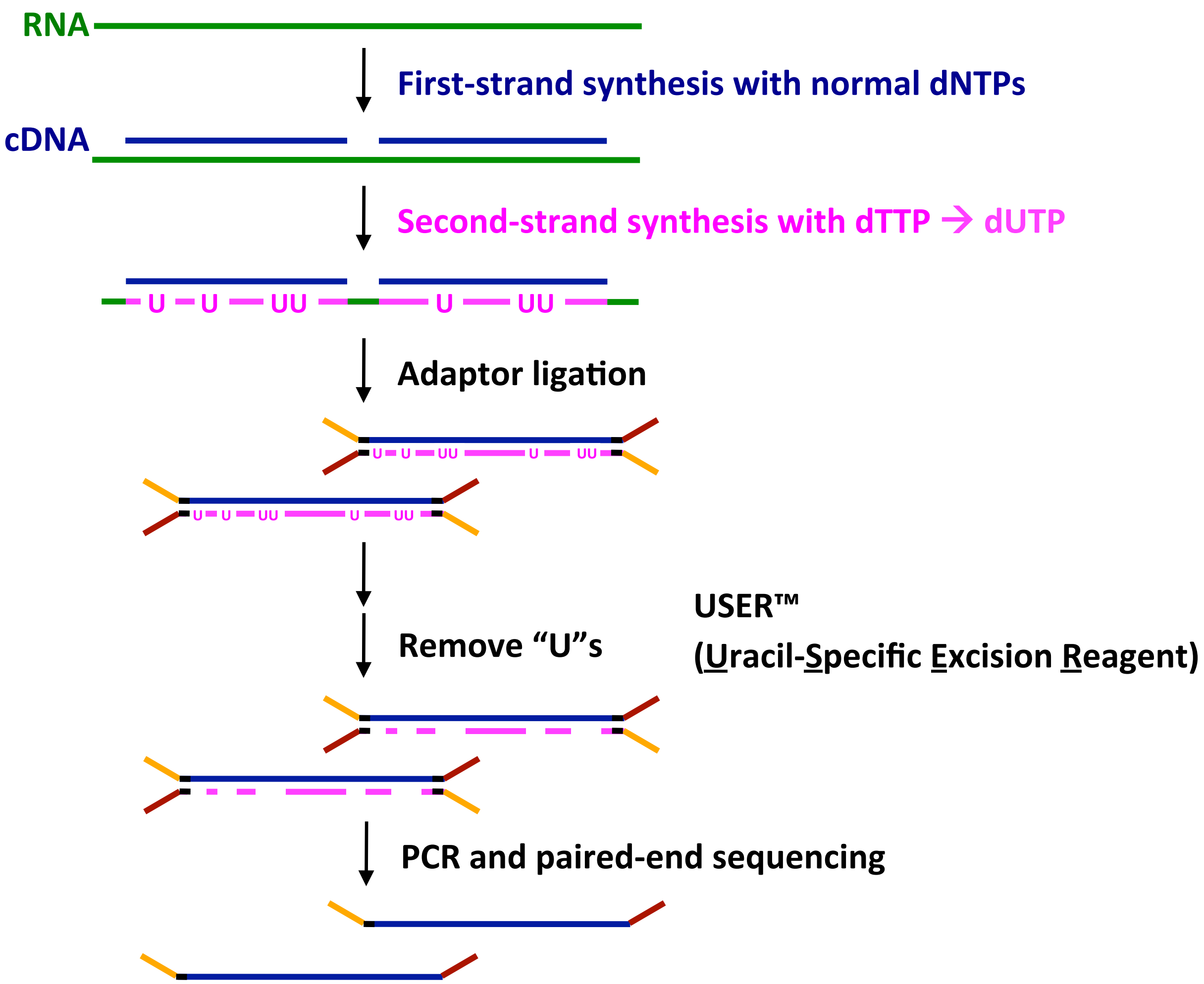
he

ct

boundaries of adjacent genes transcribed on opposite strands and resolve the correct expression levels of coding or noncoding overlapping transcripts. These tasks are particularly challenging in small microbial genomes, prokaryotic and eukaryotic, in which

'dUTP second strand marking' identified as the leading protocol

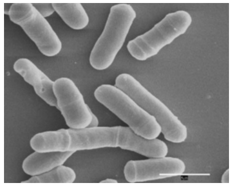
dUTP 2nd Strand Method: Our Favorite


RNA
First-strand synthesis with normal dNTPs
cDNA
Second-strand synthesis with dTTP → dUTP
U U UU U UU
Adaptor ligation
Remove "U"s
USER™
(Uracil-Specific Excision Reagent)
PCR and paired-end sequencing

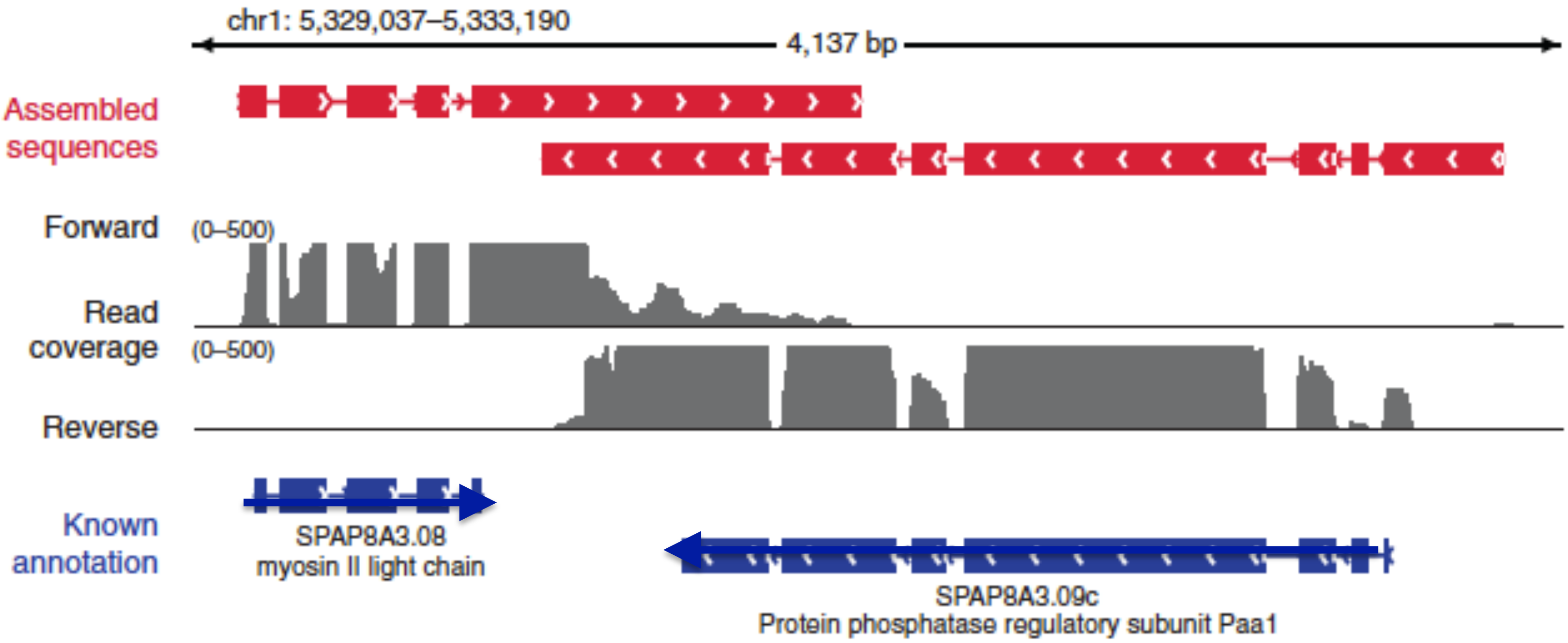
Modified from Parkhomchuk *et al.* (2009) *Nucleic Acids Res.* 37:e123

Slide from J. Levin

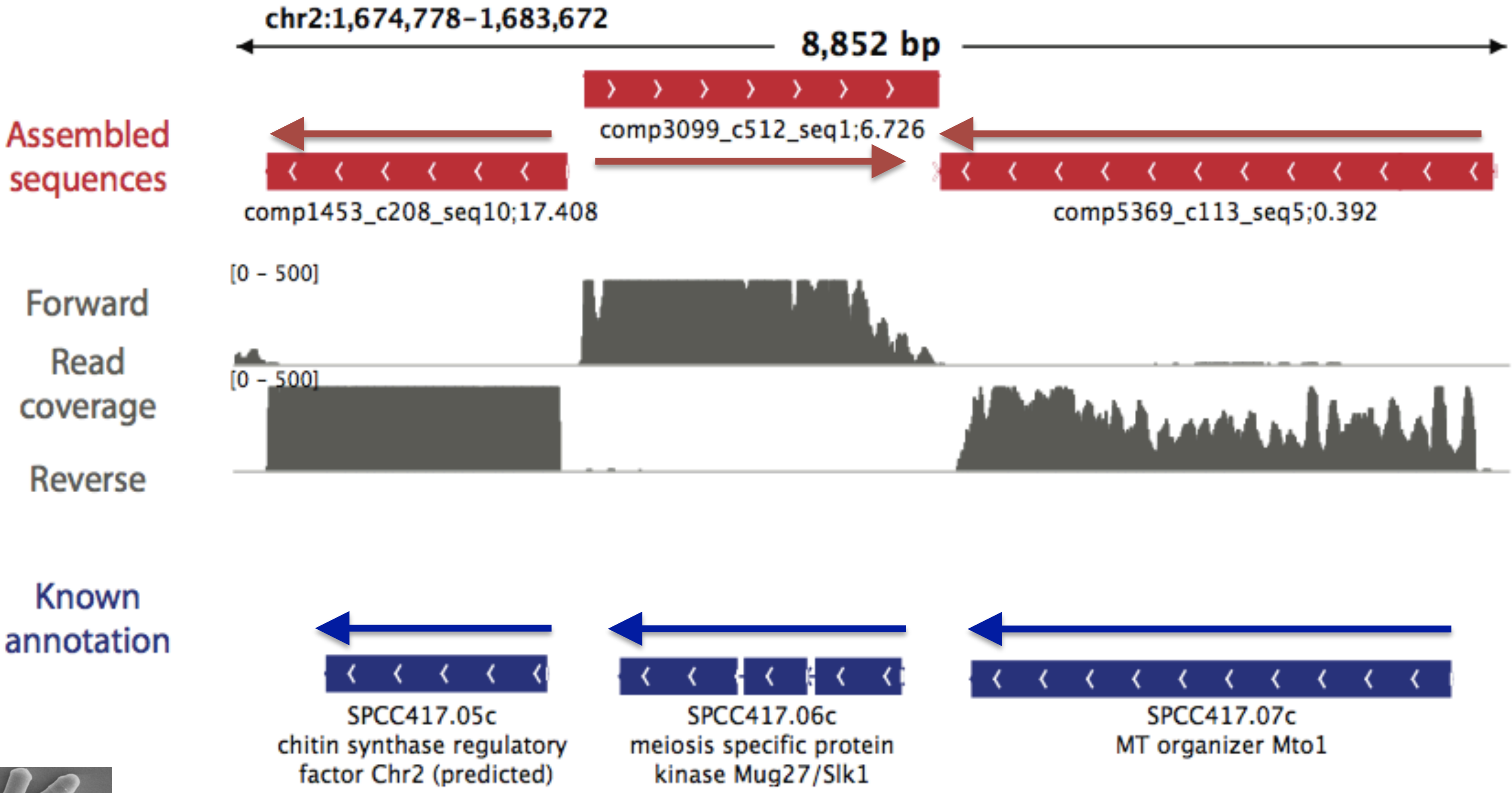
Overlapping UTRs from Opposite Strands

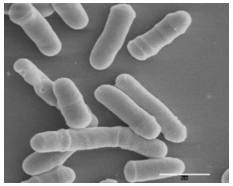


Schizosacharomyces pombe
(fission yeast)

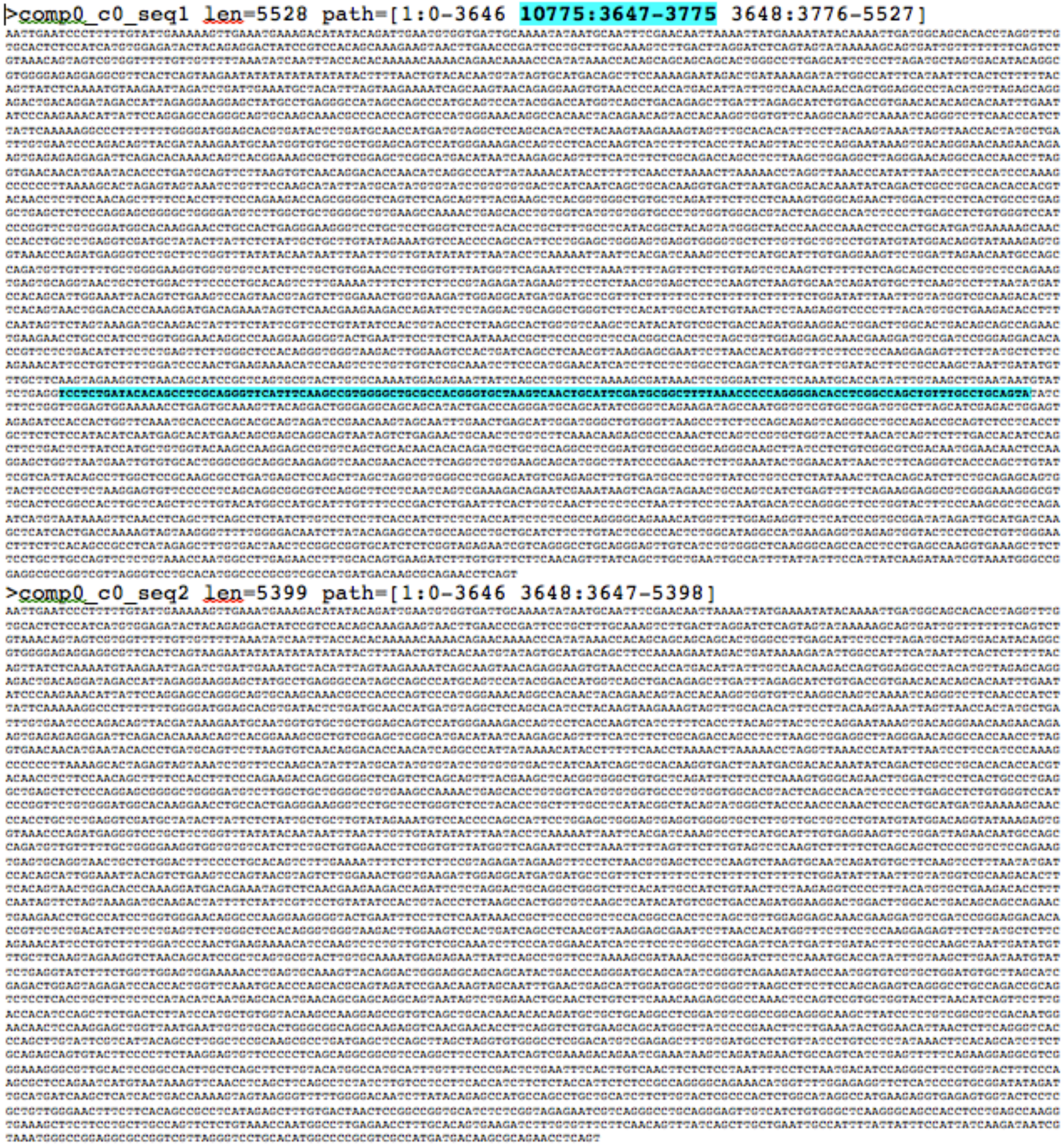


Antisense-dominated Transcription





Trinity output: a multi-fasta file



```
>comp0_c0_seq1 len=5528 path=[1:0-3646 10775:3647-3775 3648:3776-5527]
```

```
>comp0_c0_seq2 len=5399 path=[1:0-3646 3648:3647-5398]
```

We are on a Coffee Break & Networking Session