



4^{ABiMS}

04/10/2020

Assemblage de-novo de transcriptome Trinity

Ecole EBAII 2020

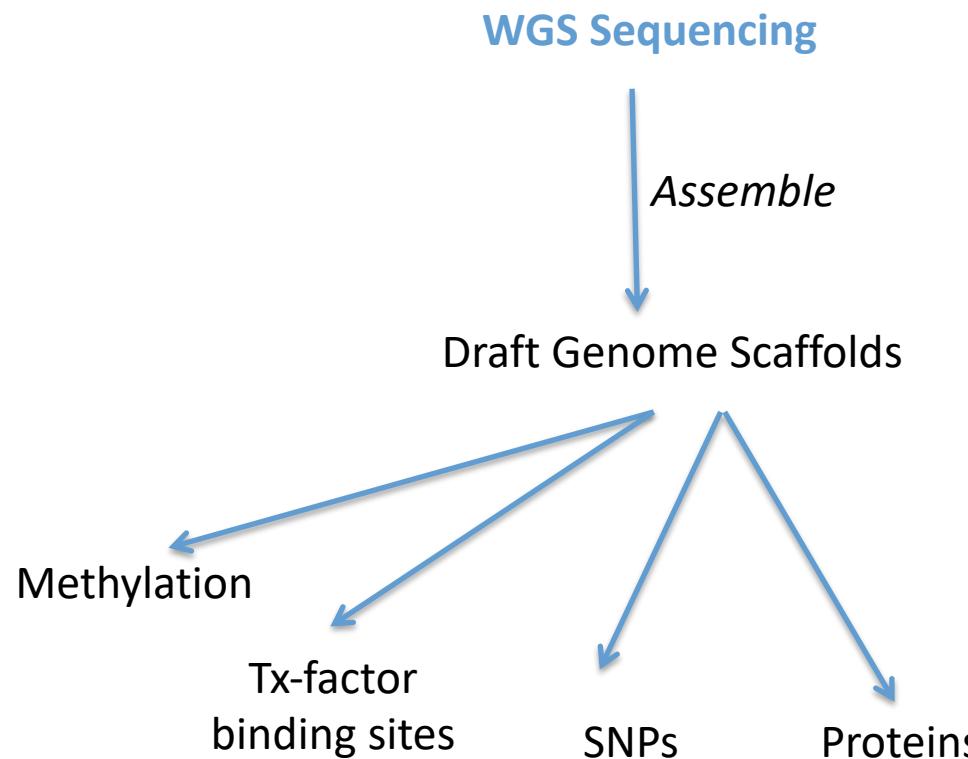
ABiMS – Station Biologique Roscoff



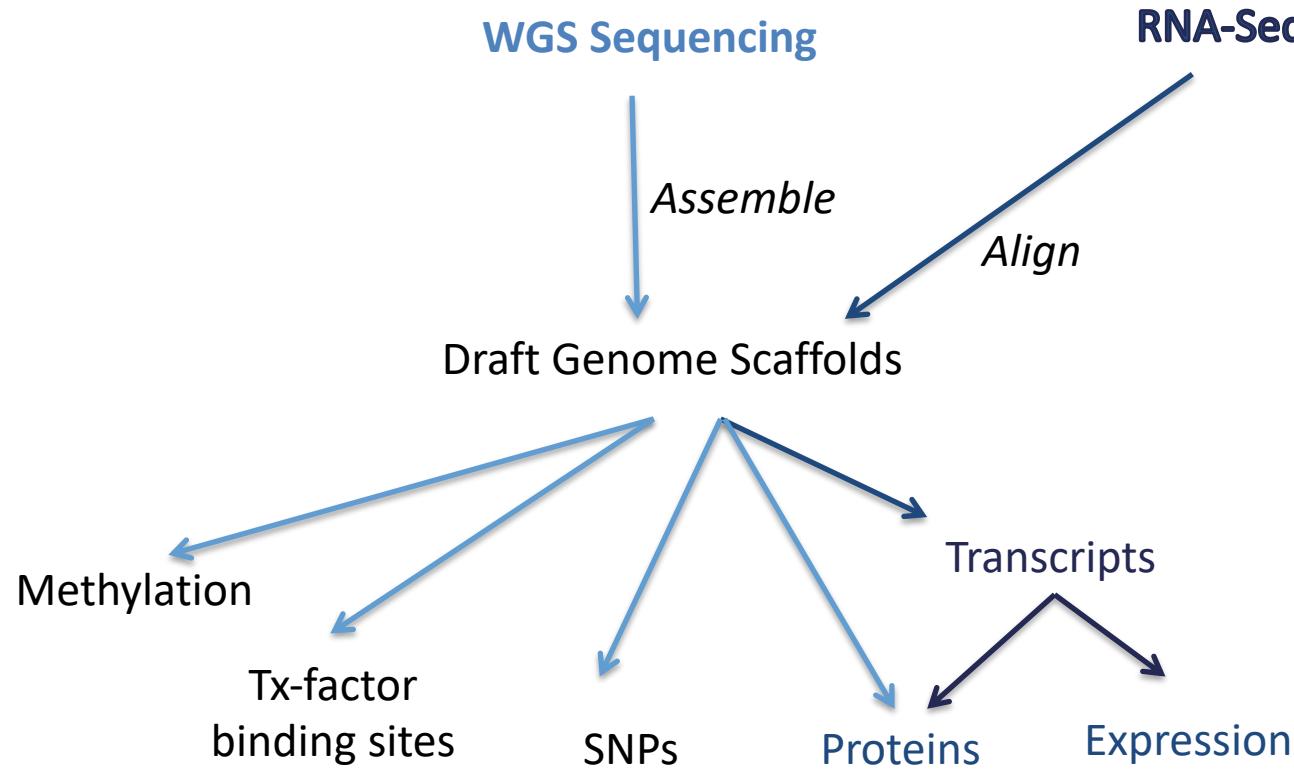
This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. [\[link\]](#)



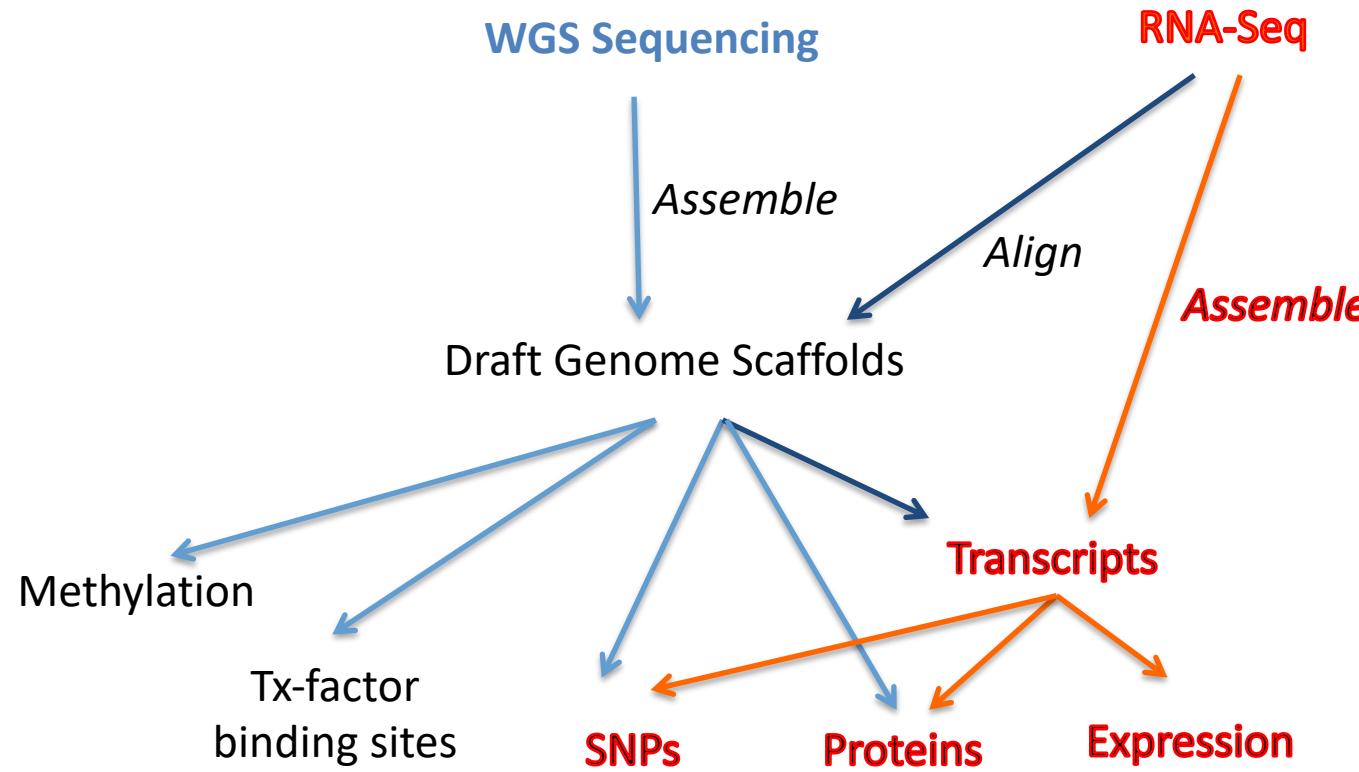
A Paradigm for Genomic Research



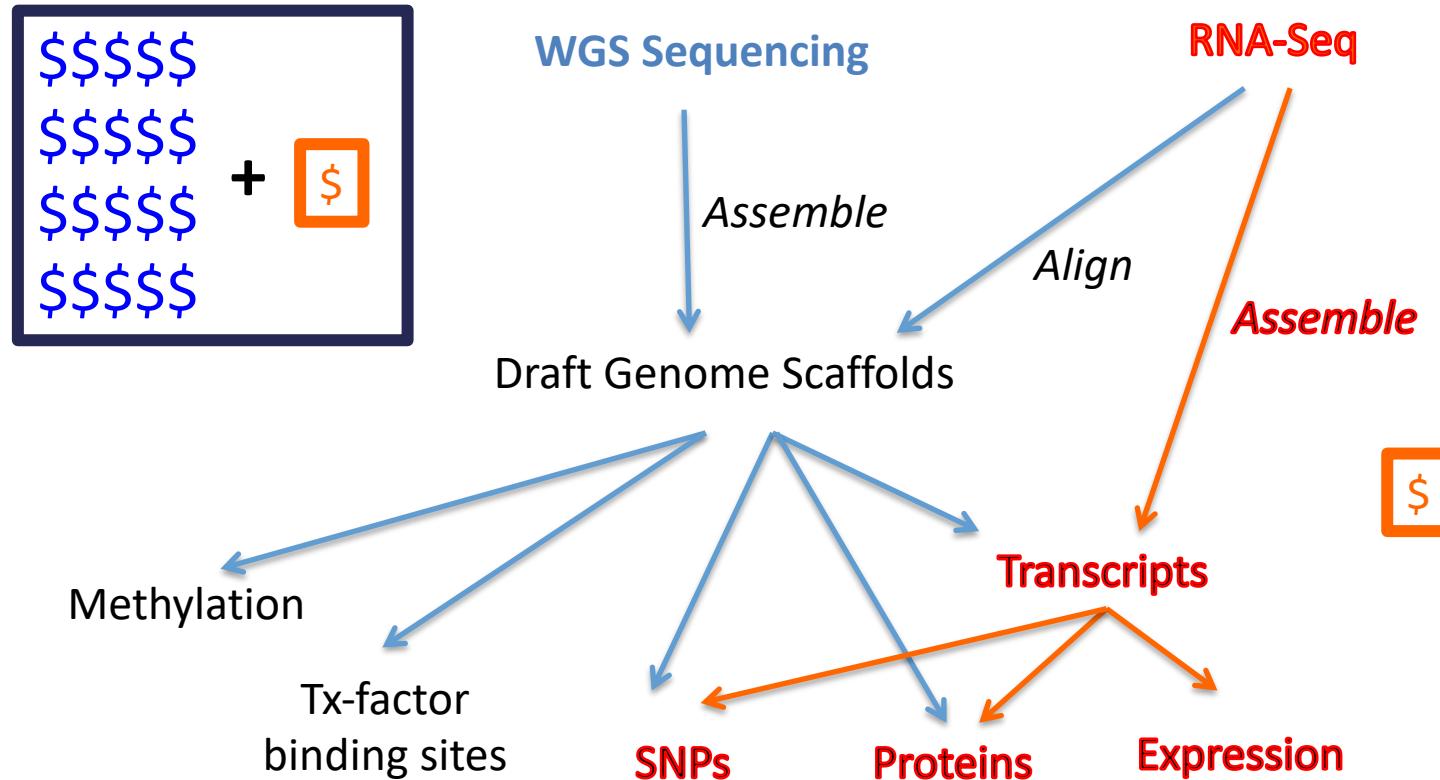
A Paradigm for Genomic Research



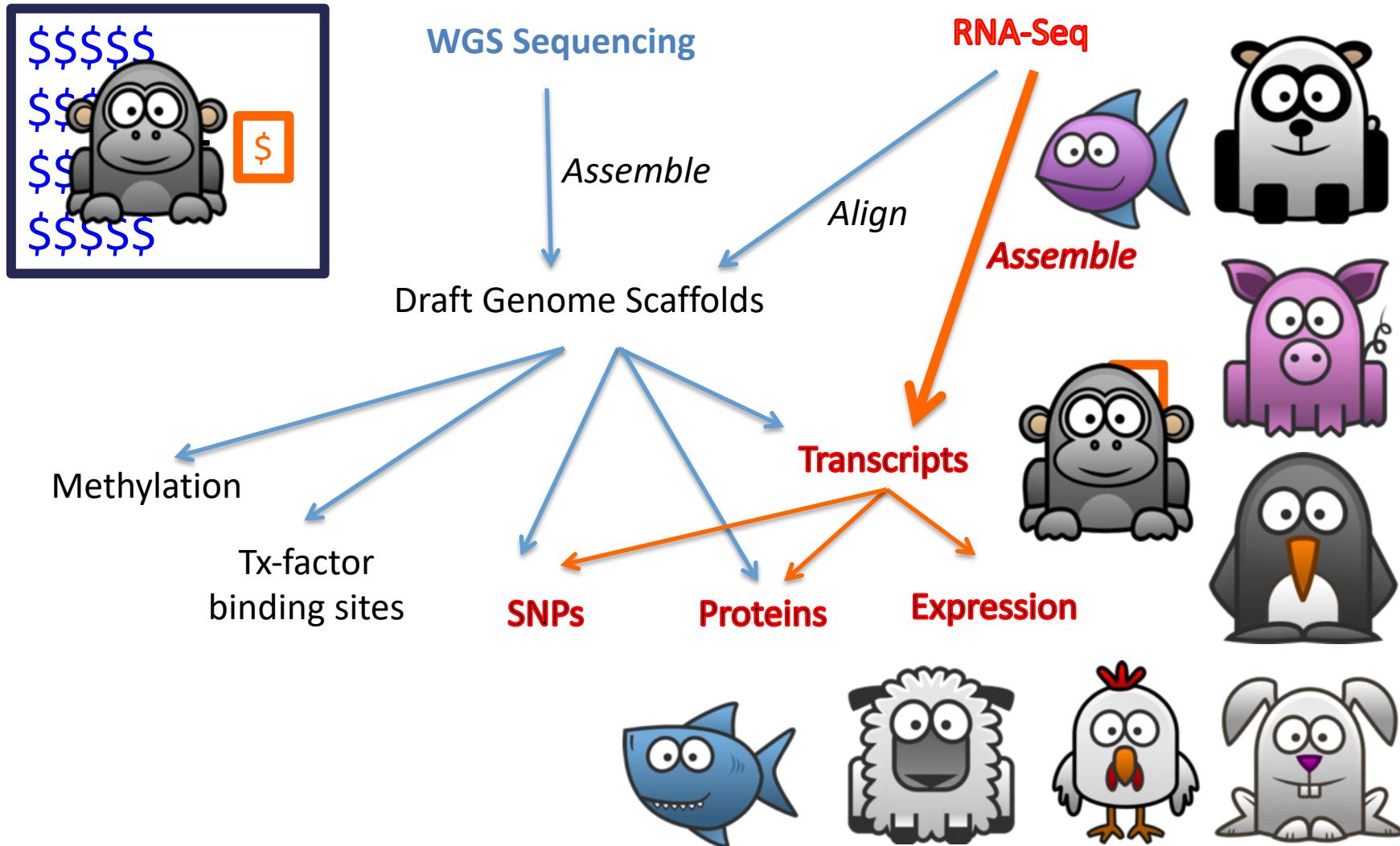
A Maturing Paradigm for Transcriptome Research



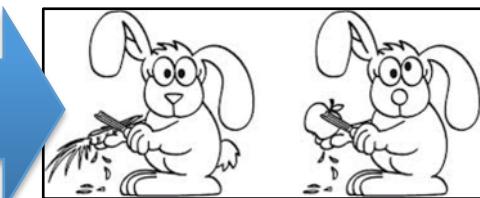
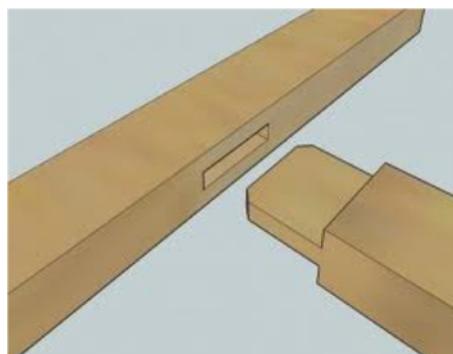
A Maturing Paradigm for Transcriptome Research



A Maturing Paradigm for Transcriptome Research



RNA Seq de novo analysis workflow



Data Cleaning



- Unknown nucleotides
- Bad quality nucleotides
- Adaptors and primers sub-sequences
- Poly A/T tails
- Low complexity sequences
- rRNA sequences
- Contaminant sequences
- Short length sequences

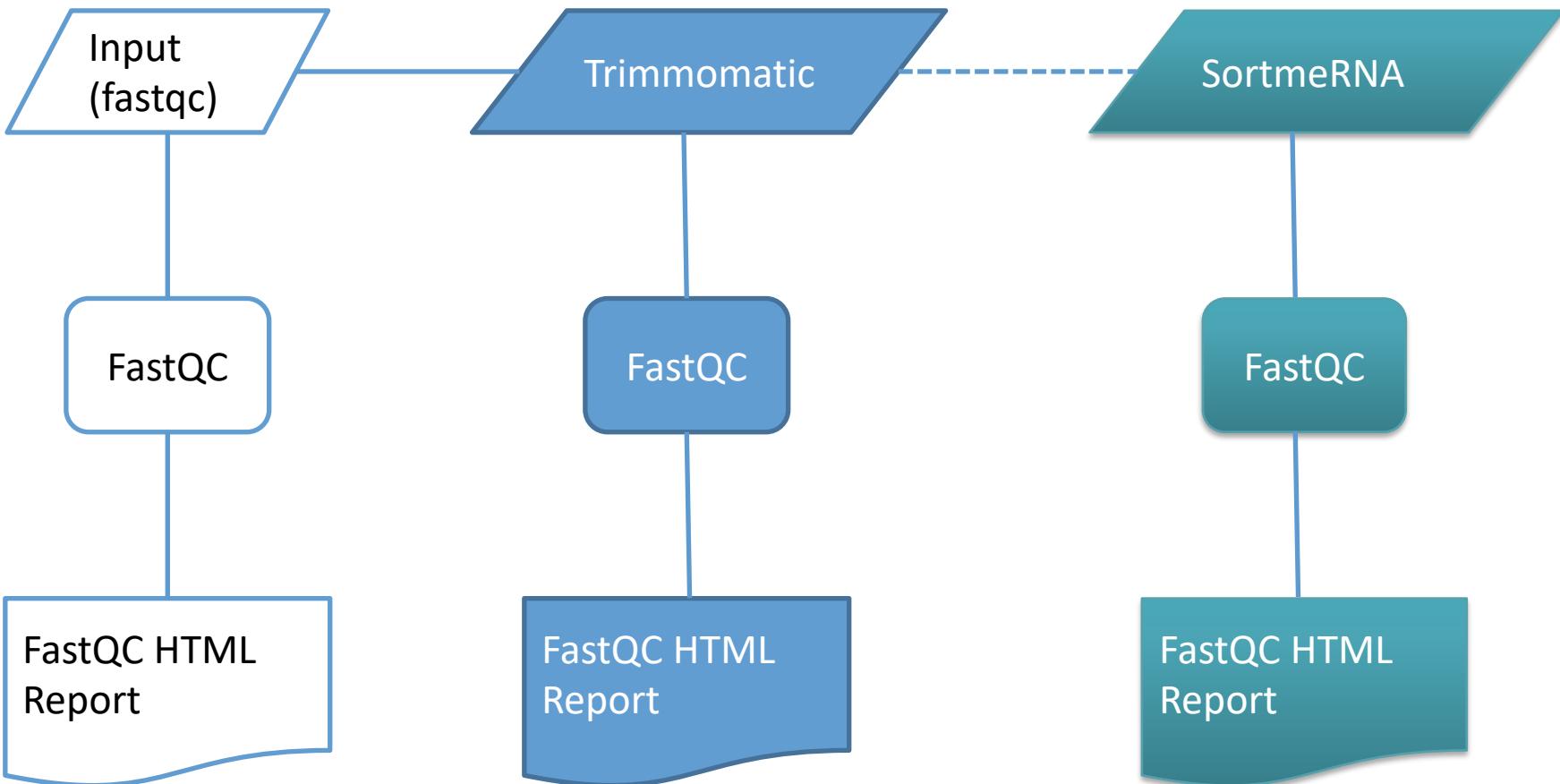
But also:

- Removing singletons
- In-silico normalization
- Sequencing errors correction
- ...

Bias should be corrected in reverse order of their generation

1. Sequencing biases (bad quality, unknowns)
2. Library preparation
 - Adaptors and primers sequences
 - Poly A/T tails
3. Biological sample (low complexity, rRNA, contaminants)

Data cleaning



Trimmomatic command

```
java -jar trimmomatic.jar PE -phred33
\ lib1_1.fastq lib1_2.fastq           Raw reads
\ lib1_1.P.qtrim lib1_1.U.qtrim       Paired and unpaired reads1
\ lib1_2.P.qtrim lib1_2.U.qtrim       Paired and unpaired reads2
\ ILLUMINACLIP:illumina.fa:2:30:10  Adapters
\ SLIDINGWINDOW:4:15 LEADING:5 TRAILING:5 MINLEN:25
```

Input Read Pairs: 2 000 000
Both Surviving: 1 879 345 (93.97%)
Forward Only Surviving: 94 153 (4.71%)
Reverse Only Surviving: 18 098 (0.90%)
Dropped: 8 404 (0.42%)

TrimmomaticPE: Completed successfully

rRNA contamination

Prior to sequencing :

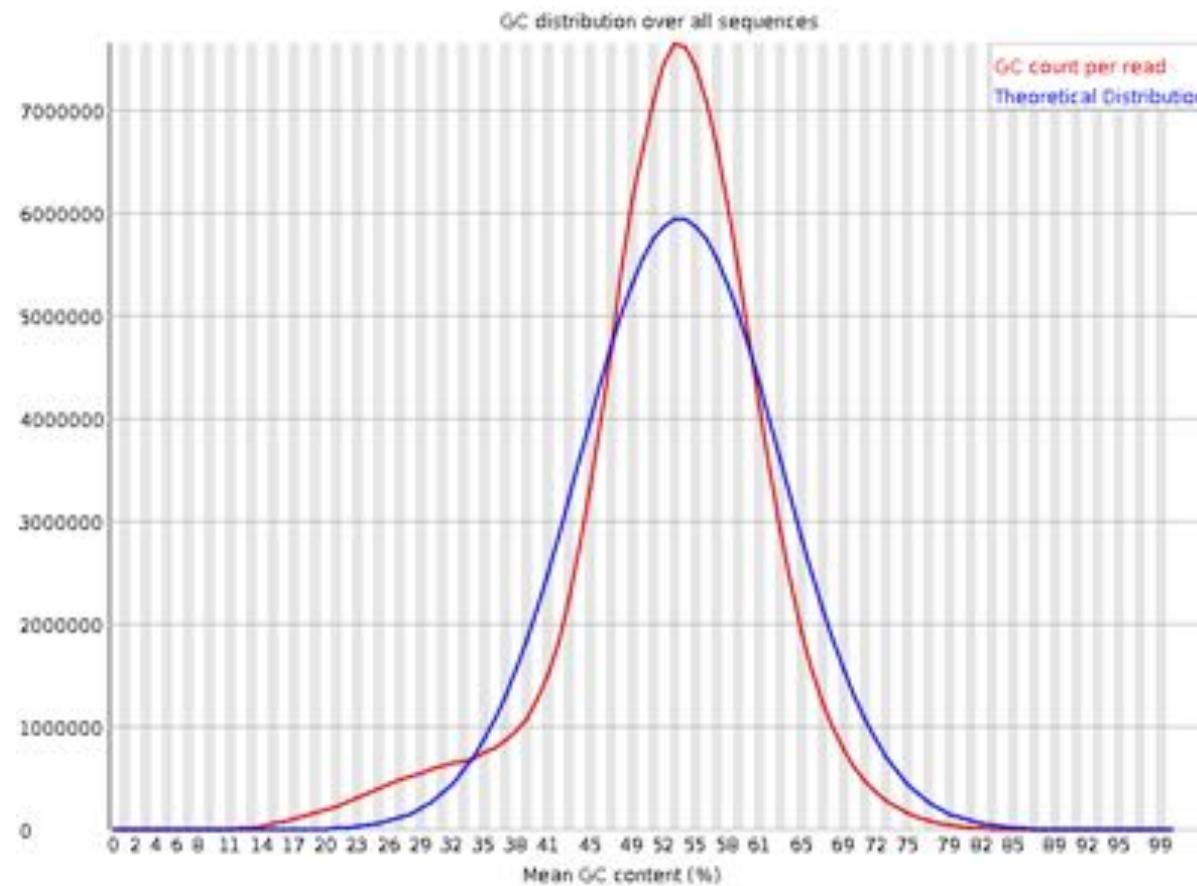
- Ribodepletion kits
- Selection polyA

After sequencing :

- Remove rRNA reads from raw reads
- Detect rRNA transcripts

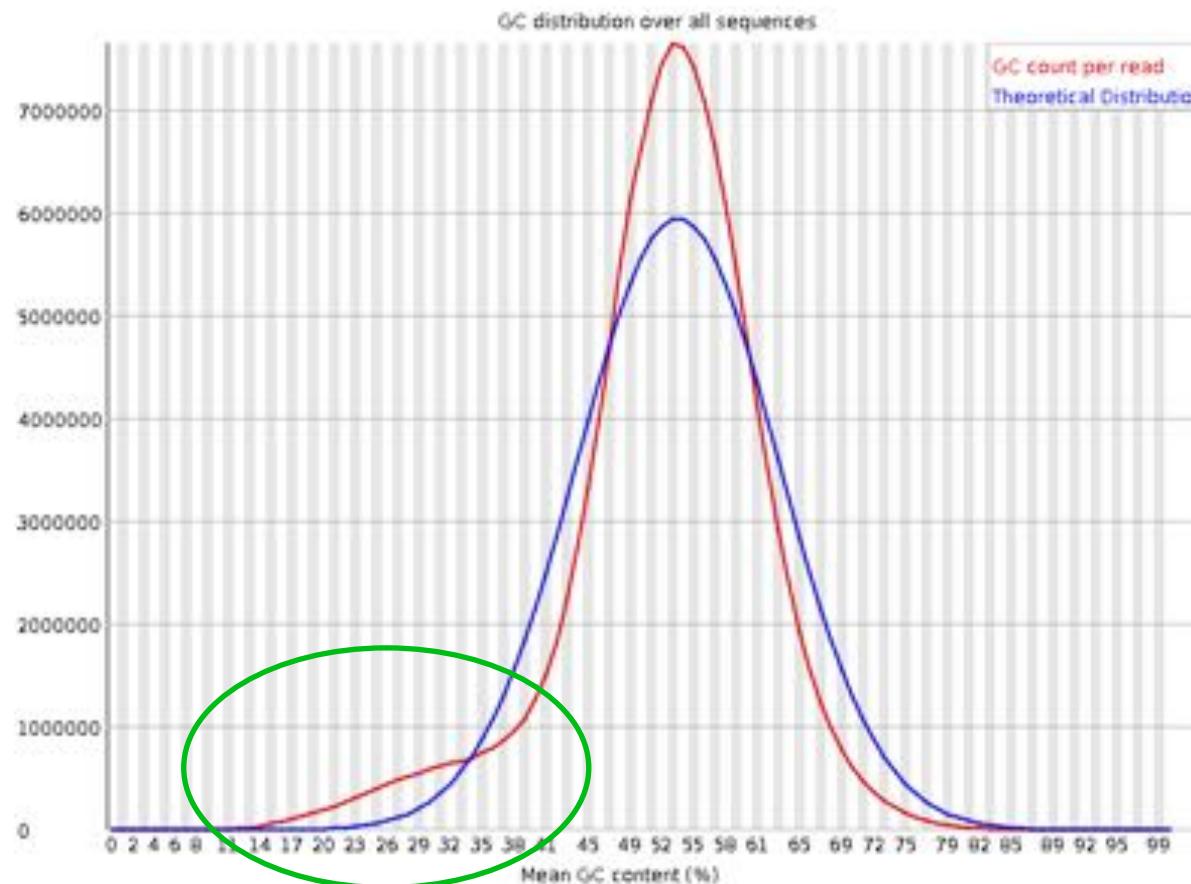
FastQC: Per sequence GC content

- A contamination ?



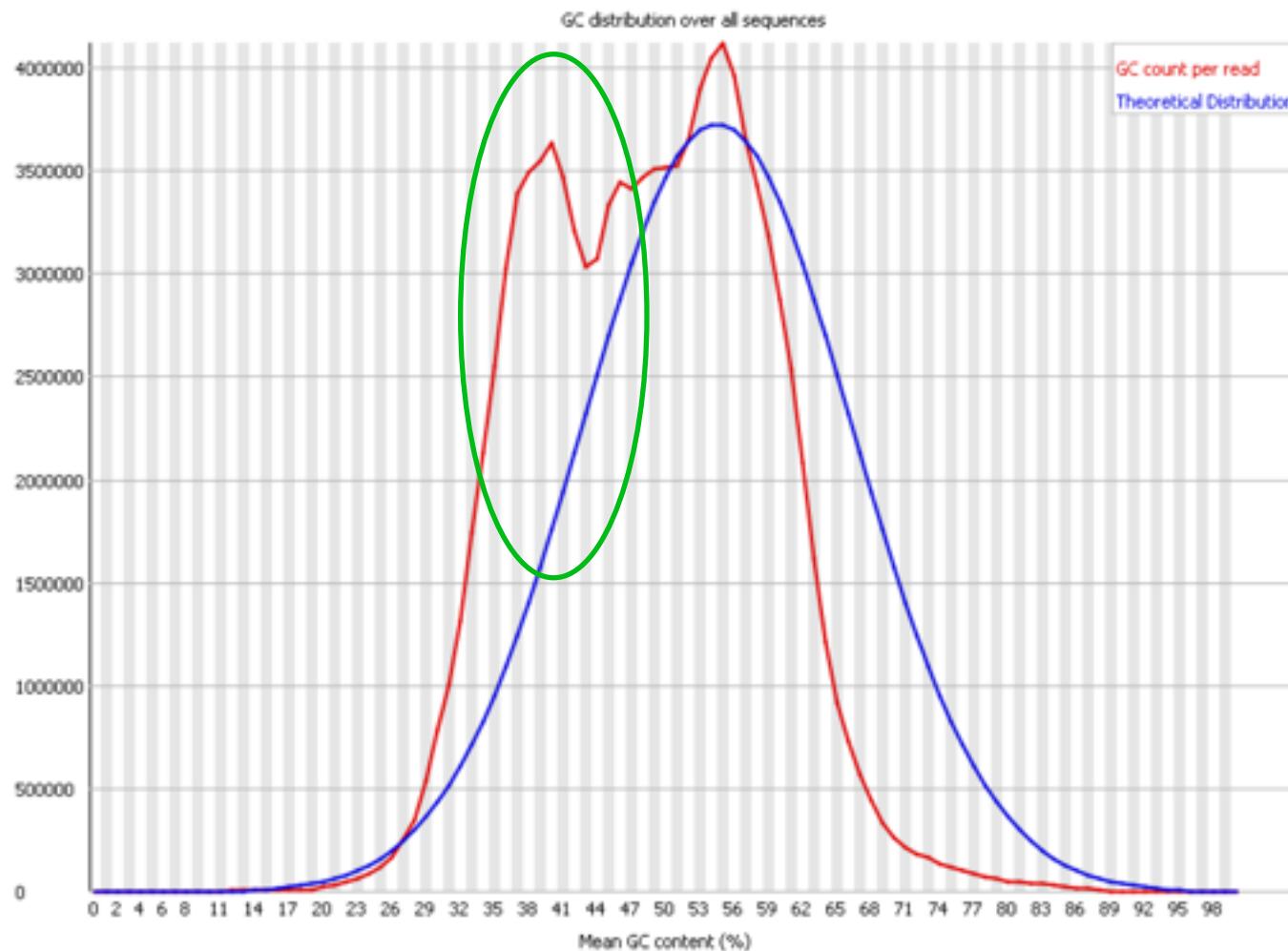
FastQC: Per sequence GC content

- A contamination ?

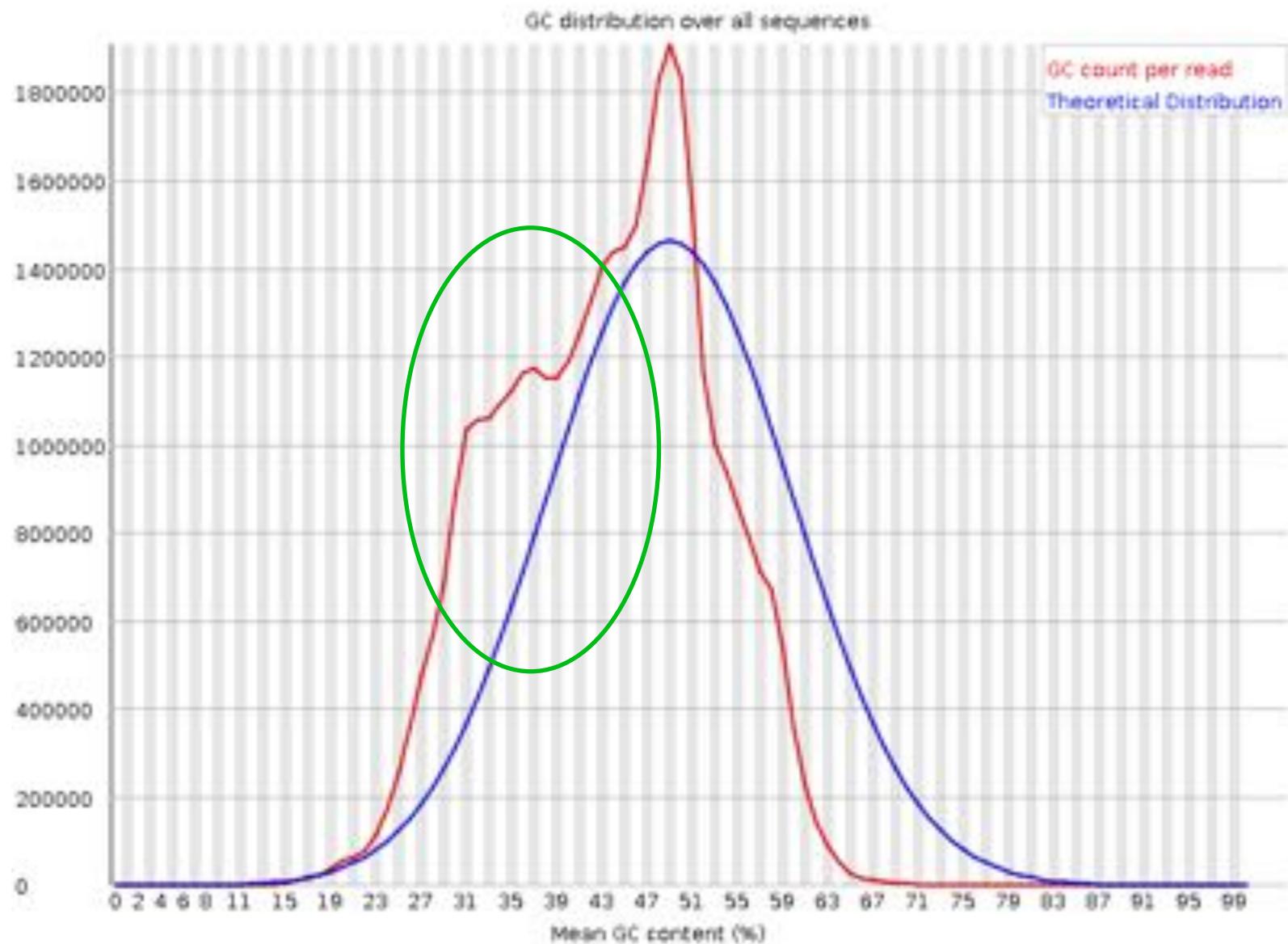


Can this be fixed ? Maybe...

FastQC: Per sequence GC content

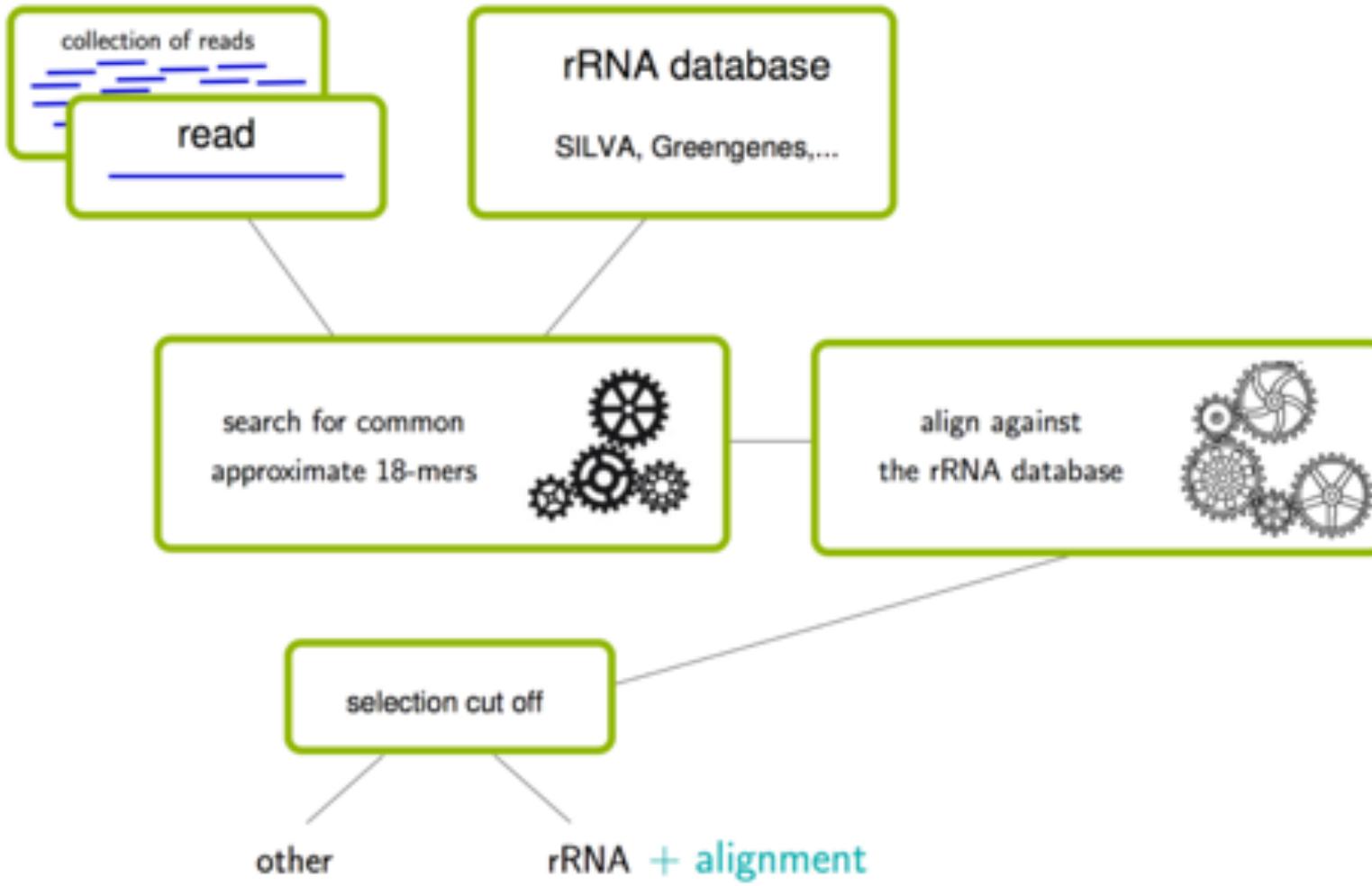


Third-party contamination : detection



Sabellaria alveolata : mantle transcriptome

SortMeRNA



SortMeRNA commands

```
> merge-paired-reads.sh read_1.fq read_2.fq read-interleaved.fq
```

Reference DB

```
>sortmerna --fastx -a 4 --log --paired_out -e 0.1 --id 0.97 --coverage 0.97  
  \--ref silva-bac-16s-id90.fasta,silva-bac-16s-id90:  
  \silva-bac-23s-id98.fasta,silva-bac-23s-id98:  
  \silva-euk-18s-id95.fasta,silva-euk-18s-id95:  
  \silva-euk-28s-id98.fasta,silva-euk-28s-id98:  
  \rfam-5s-database-id98.fasta,rfam-5s-database-id98:  
  \rfam-5.8s-database-id98.fasta,rfam-5.8s-database-id98  
  --reads read-interleaved.fq --other output_mRNA.fastq  
  fastq --aligned output_aligned.fastq
```

```
>unmerge-paired-reads.sh output_mRNA.fastq read-sortmerna_1.fq read-sortmerna_2.fq
```

SortMeRNA results

Results:

Total reads = 34 196 864

Total reads for de novo clustering = 4 084 914

Total reads passing E-value threshold = 30 122 173 (88.08%)

Total reads failing E-value threshold = 4 074 691 (11.92%)

Minimum read length = 150

Maximum read length = 150

Mean read length = 150

By database:

silva-bac-16s-id90.fasta 6.95%

silva-bac-23s-id98.fasta 18.75%

silva-euk-18s-id95.fasta 9.97%

silva-euk-28s-id98.fasta 52.42%

rfam-5s-database-id98.fasta 0.00%

rfam-5.8s-database-id98.fasta 0.00%

Total reads passing %id and %coverage thresholds = 26 037 259

Detect rRNA transcripts : RNAMMER



The program uses hidden Markov models trained on data from the 5S ribosomal RNA database and the European ribosomal RNA database project

#	seqname	source	feature	start	end	score	+/-	frame	attribute	
# -----										
##gff-version2##source-version RNAmmer-1.2##date 2009-11-16										
#	AE000511	RNAmmer-1.2	rRNA	448462		448577		49.2		+
#		.	5s_rRNA							
AE000511	RNAmmer-1.2	rRNA		1473564	1473679	49.2		-		5s_rRNA
AE000511	RNAmmer-1.2	rRNA		1045067	1045183	40.3		+		5s_rRNA
AE000511	RNAmmer-1.2	rRNA		445339		448223		3056.5		+
	.	23s_rRNA								
AE000511	RNAmmer-1.2	rRNA		1473918	1476803	3032.8		-		23s_rRNA
AE000511	RNAmmer-1.2	rRNA		1207586	1209074	1801.4		-		16s_rRNA
AE000511	RNAmmer-1.2	rRNA		1511140	1512627	1803.6		-		16s_rRNA

Lagesen K, Hallin PF, Rødland E, Stærfeldt HH, Rognes T Ussery DW [RNAmmer: consistent annotation of rRNA genes in genomic sequences](#)

Nucleic Acids Res. 2007 Apr 22.

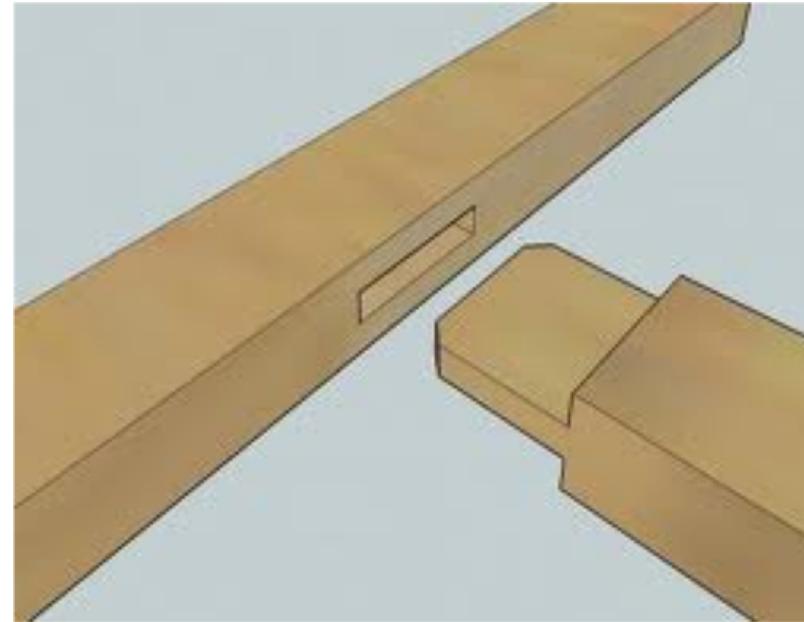
Alternative Barrnap :
<https://github.com/tseemann/barrnap>

RNAmmer - Barnap

```
> Trinotate-3.0.1/util/rnammer_support/RnammerTranscriptome.pl  
--transcriptome Assembly.fasta --org_type (arc|bac|euk) --  
path_to_rnammer /usr/local/genome2/rnammer/rnammer
```

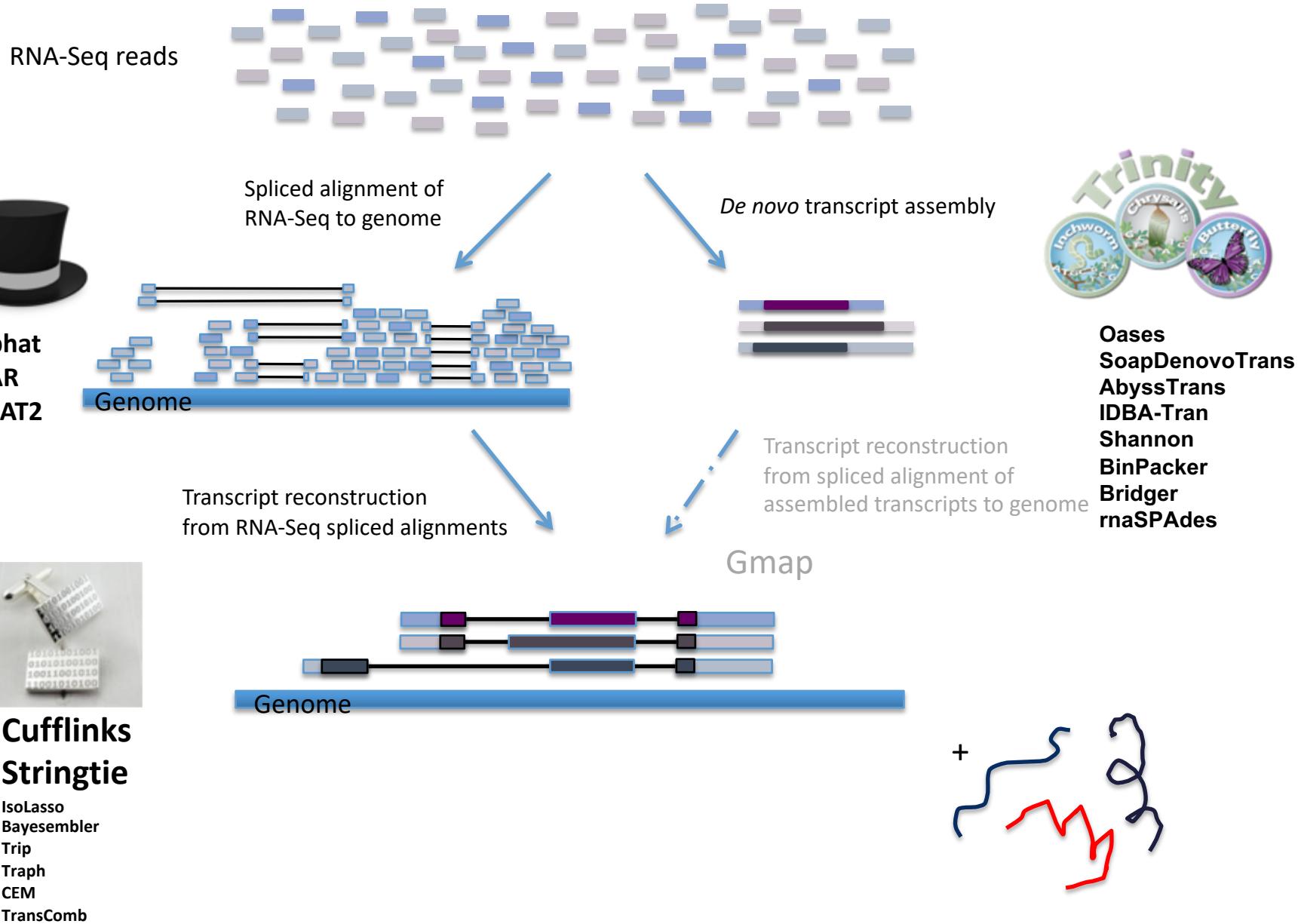
```
>bedtools getfasta -fi Assembly.fasta -bed  
rnammer_predictions.gff > transcripts_rrna.fasta
```

```
> /usr/local/genome2/barnap-master2/bin/barnap --kingdom bac  
--threads 10 --outfasta rrna_bact.fasta Assembly.fasta
```

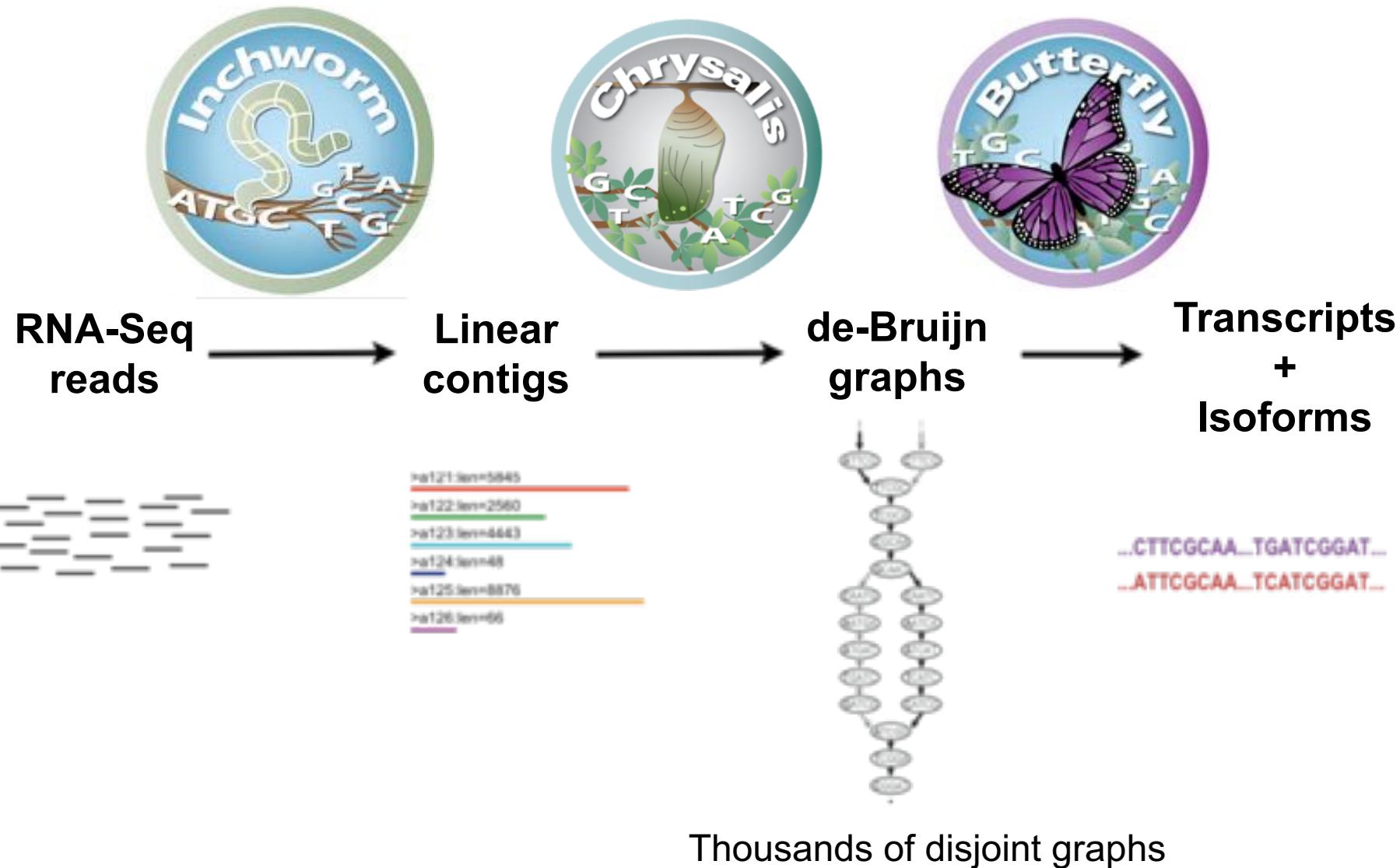


TRANSCRIPTOME ASSEMBLY STRATEGIES

Contemporary strategies for transcript reconstruction from RNA-Seq



Trinity – How it works:

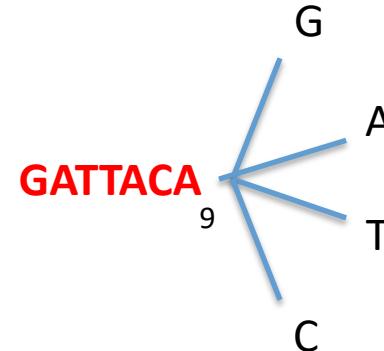


Inchworm Algorithm

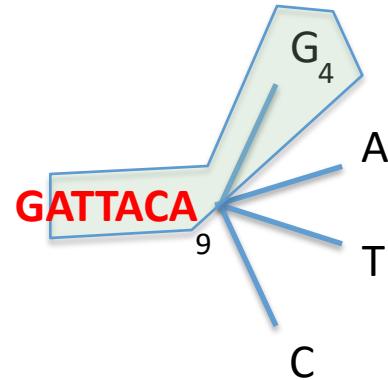
Decompose all reads into overlapping Kmers (25-mers) and count them : Jellyfish

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

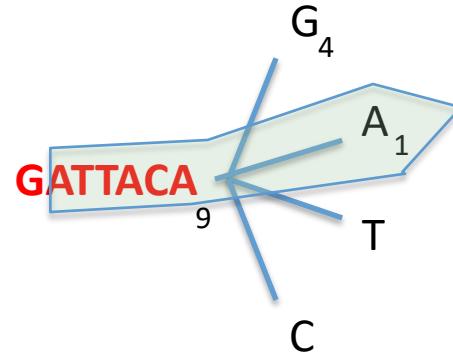
Extend kmer at 3' end, guided by coverage.



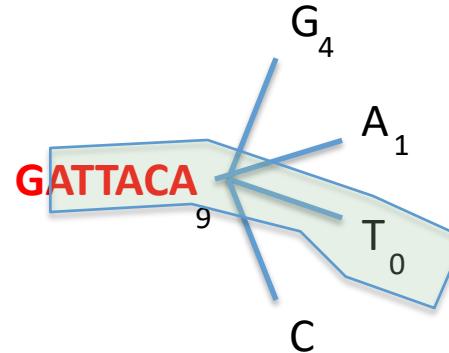
Inchworm Algorithm

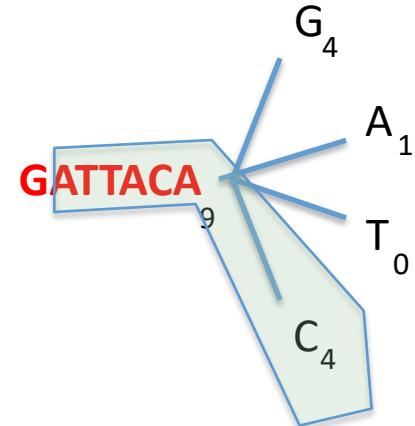


Inchworm Algorithm

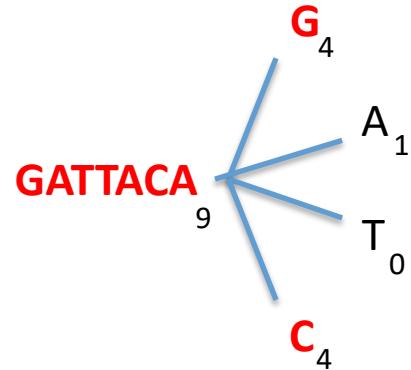


Inchworm Algorithm

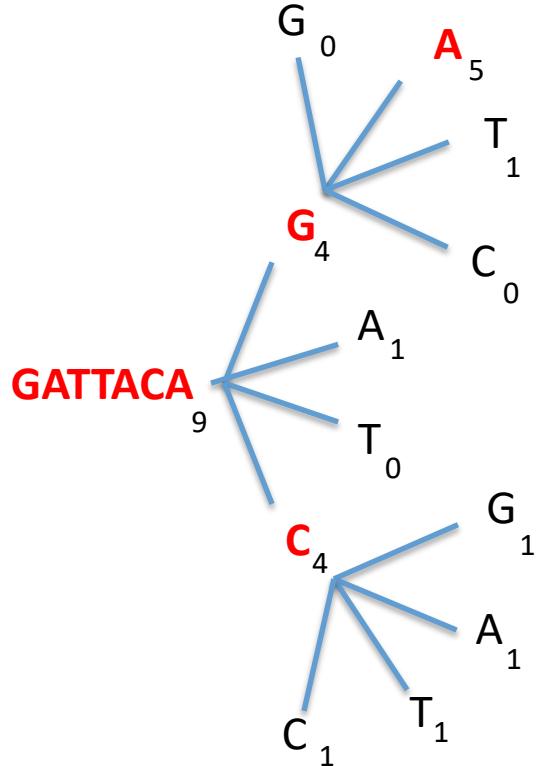




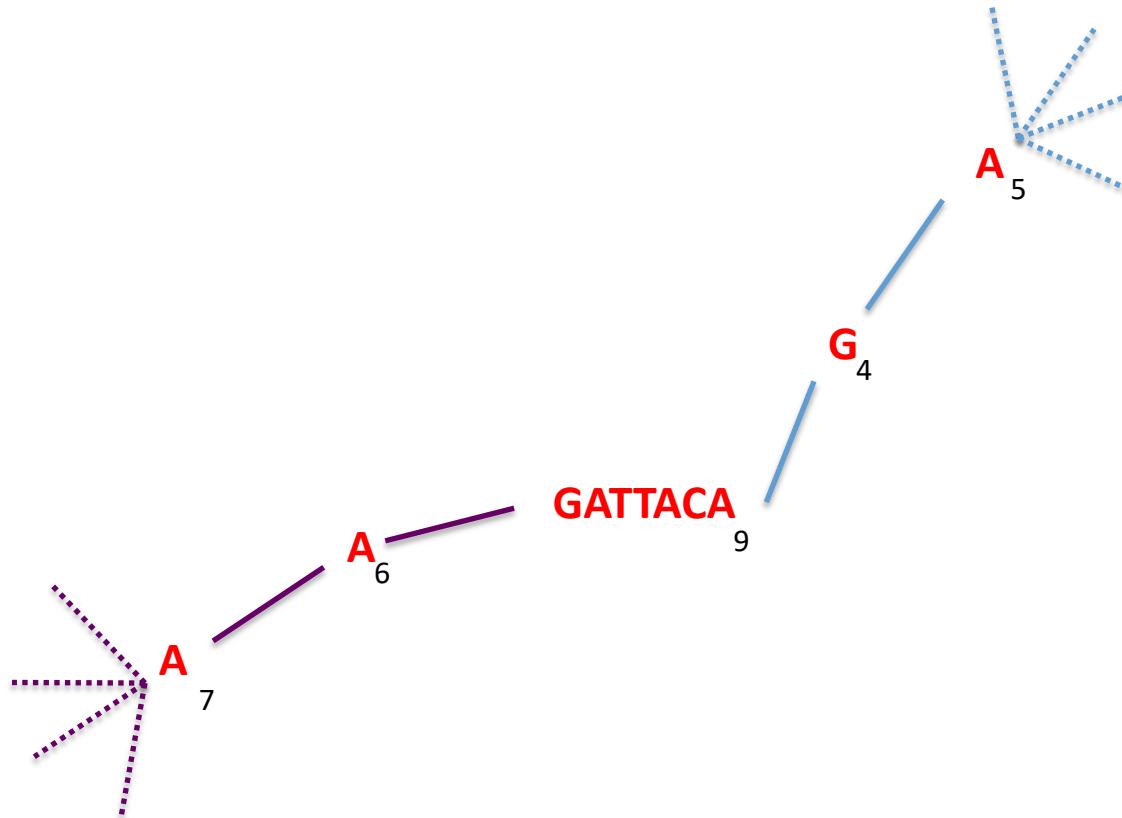
Inchworm Algorithm



Inchworm Algorithm



Inchworm Algorithm



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

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

Inchworm Contigs from Alt-Spliced Transcripts

Expressed isoforms



Expression

(low)
(high)



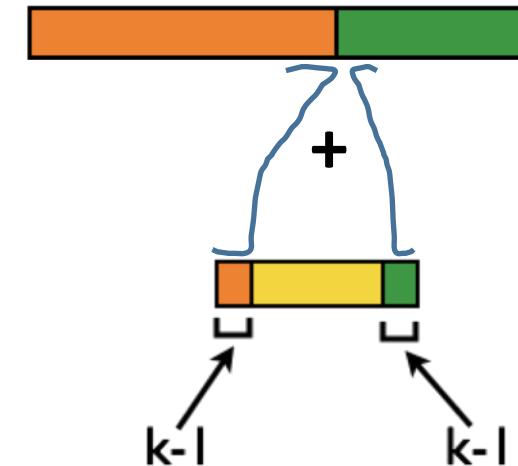
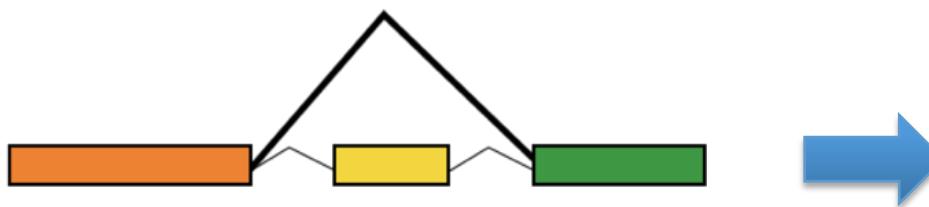
Inchworm Contigs from Alt-Spliced Transcripts



Expressed isoforms



Expression

(low)
(high)

Inchworm can only report contigs derived from unique kmers.

Alternatively spliced transcripts :

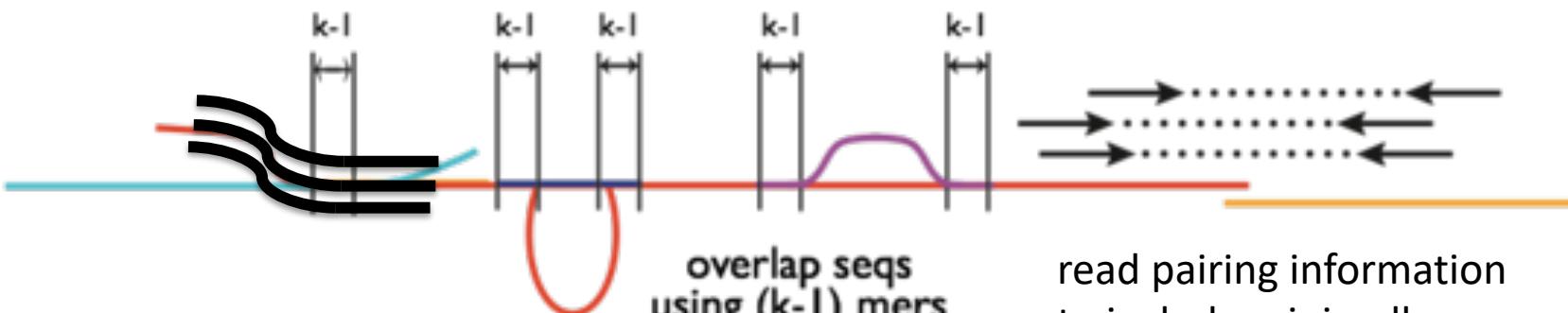
- the more highly expressed transcript may be reported as a single contig,
- the parts that are different in the alternative isoform are reported separately.

Chrysalis



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

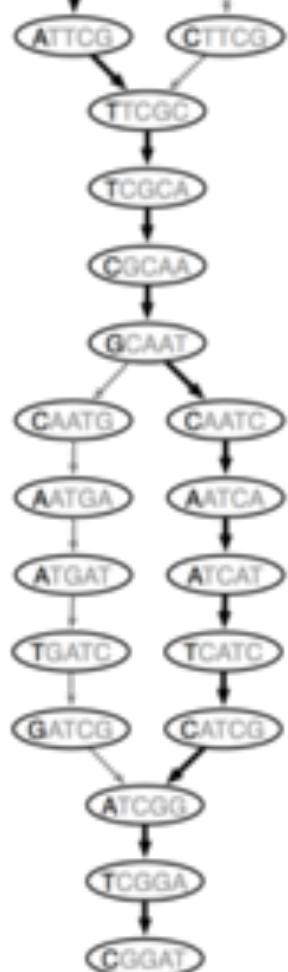
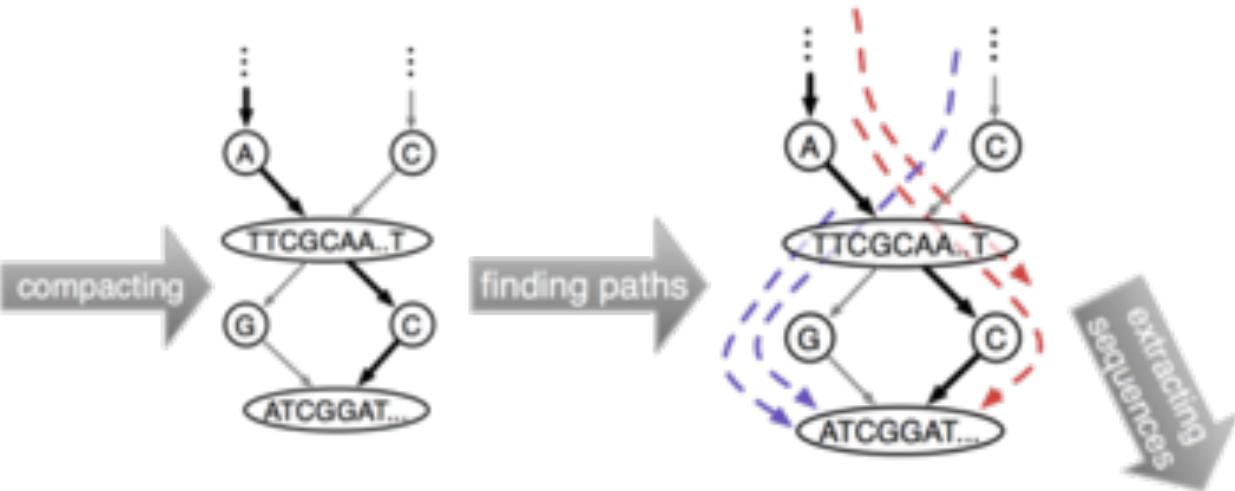
Integrate (clustering)
Isoforms via k-1 overlaps
Verify via “welds”



Build de Bruijn Graphs
(ideally, one per gene)

read pairing information
to include minimally
overlapping contigs

Butterfly

de Bruijn
graphcompact
graphcompact
graph with
reads

sequences

..CTTCGCAA..TGATCGGAT...
..ATTCGCAA..TCATCGGAT...

Trinity usage and options

Typical Trinity command

```
Trinity --seqType fq --max_memory 50G  
\--left A_rep1_left.fq --right A_rep1_right.fq --CPU 4
```

```
Trinity --seqType fq --max_memory 50G --single single.fq --  
CPU 4
```

Running a typical Trinity job requires ~1 hour and ~1G RAM per ~1 million PE reads.

The assembled transcripts will be found at 'trinity_out_dir/Trinity.fasta'.

Results

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

```
>TRINITY_DN889_c0_g1_i1 len=259 path=[ 473:0-258 ] [-1, 473, -2]
GAACAATGTCTACACTGTCTCAACTTGGATGACAAGGAACCTTCATTGGCTCAAGCTAA
CTACAATTCATCTCTGAAACCAGATATTGAAGAAATCAAGGATACTGTCCCTAGCGCTGT
GCTGGCTCCACAATACTACAAACACATTCTCAGCTGACCCAAGTGCAGTCAGTCAGTC
TAACATCTTGCACCAGAGGCCACTATGTCCATGGCTGCTCCAGCTAATGCTTAGAAA
CTCTTCATTAAACTCTCCT

>TRINITY_DN810_c0_g1_i2 len=226 path=[ 407:0-225 ] [-1, 407, -2]
GATGATATCAACAATGAGACTTGTGAACCAGGTGAAGAAAATCTTCTTGATGCGAC
CTAGGTGAAATTGAAAGATTGTACGCTAACTGGTGGAAAGAACTACCAAGAGTTCAGCCA
TTTACGCTGTCAAGTGTAAACCCAGATTGAAGATAATAAGAAAATTGGCTGACCTCGGA
```

TRINITY_DNW|cX_gY_iZ (until release 2.0 **cX_gY_iZ** previously **compX_cY_seqZ**)

TRINITY_DNW|cX 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 (**gY**).: trinity genes

Each subgraph then gives rise to path sequences (**iZ**). : trinity isoforms

(**path**) list of vertices in the compacted graph that represent the final transcript sequence and the range within the given assembled sequence that those nodes correspond to.

Trinity statistics

```
TRINITY_HOME/util/TrinityStats.pl Trinity.fasta
```

```
#####
## Counts of transcripts, etc.
#####
Total trinity 'genes': 7648
Total trinity transcripts: 7719
Percent GC: 38.88
#####
Stats based on ALL transcript contigs:
#####
Contig N10: 4318
Contig N20: 3395
Contig N30: 2863
Contig N40: 2466
Contig N50: 2065
Median contig length: 1038
Average contig: 1354.26
Total assembled bases: 10453524
#####
## Stats based on ONLY LONGEST ISOFORM per 'GENE':
#####
Contig N10: 4317
Contig N20: 3375
Contig N30: 2850
Contig N40: 2458
Contig N50: 2060
Median contig length: 1044
Average contig: 1354.49
Total assembled bases: 10359175
```

Trinity usage and options

Typical Trinity command with multiple samples

```
Trinity --seqType fq --max_memory 50G --CPU 4
\--left A_rep1_left.fq,A_rep2_left.fq
\--right A_rep1_right.fq,A_rep2_right.fq
```

sample.txt

cond_A	cond_A_rep1	A_rep1_left.fq	A_rep1_right.fq
cond_A	cond_A_rep2	A_rep2_left.fq	A_rep2_right.fq
cond_A	cond_A_rep3	A_rep3_left.fq	A_rep3_right.fq
cond_B	cond_B_rep1	B_rep1_left.fq	B_rep1_right.fq
cond_B	cond_B_rep2	B_rep2_left.fq	B_rep2_right.fq
cond_B	cond_B_rep3	B_rep3_left.fq	B_rep3_right.fq

```
Trinity --seqType fq --max_memory 50G --CPU 4
\--samples_file sample.txt
```

Trinity « genome guided »

If your RNA-Seq **sample differs sufficiently** from your reference genome and you'd like **to capture variations** within your assembled transcripts

De novo assembly is restricted to only those reads that map to the genome.

The advantage is that **reads that share sequence in common but map to distinct parts of the genome** will be targeted separately for assembly.

The disadvantage is that reads that do not map to the genome will not be incorporated into the assembly.

-> Unmapped reads can, however, be targeted for a separate genome-free de novo assembly.

Genome guided Trinity command

```
Trinity --genome_guided_bam rnaseq_alignments.csorted.bam --  
max_memory 50G --genome_guided_max_intron 10000 --CPU 6
```

The assembled transcripts will be found at 'trinity_out_dir/Trinity-GG.fasta'.

Trinity « longreads »

```
Trinity --seqType fq --max_memory 50G --CPU 4  
\--samples_file sample.txt --long_reads contigs.fasta
```

contigs.fasta:

fasta file containing error-corrected or circular consensus (CCS) PacBio reads

In short, the Trinity v2.4.0 version uses the pacbio reads mostly for path tracing in a graph that's built based on the illumina reads (not build using illumina AND pacbio) .

Under development

Trinity including trimming and normalisation

- Trimming

```
Trinity --seqType fq --max_memory 50G --CPU 4
--samples_file sample.txt --trimmomatic
--quality_trimming_params "ILLUMINACLIP:illumina.fa:2:30:10
SLIDINGWINDOW:4:15 LEADING:5 TRAILING:5 MINLEN:25"
```

Trinity including trimming and normalisation

- Trimming

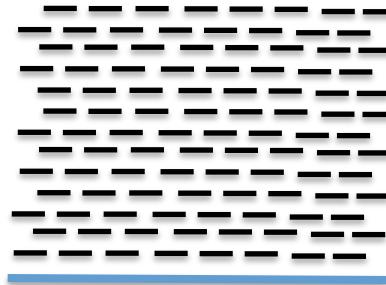
```
Trinity --seqType fq --max_memory 50G --CPU 4
--samples_file sample.txt --trimmomatic
--quality_trimming_params "ILLUMINACLIP:illumina.fa:2:30:10
SLIDINGWINDOW:4:15 LEADING:5 TRAILING:5 MINLEN:25"
```

- Normalisation:

- By definition RNAseq display a wide range of expressions
Very low expressed → Very highly expressed transcripts
 - The information given by reads from high expression transcripts is redundant, and very high coverage also brings more sequencing errors
 - De-novo assemblers do not benefit from coverage increase beyond a certain point (> 200 millions reads) , and fewer data means quicker assemblies
- How to decrease coverage of highly expressed transcripts without decreasing that of low expressed transcripts ?

In silico normalization of reads

High



Moderate



Low



NGS reads normalization (by Trinity)

1. Count kmers in all the data (Jellyfish):
 - with $k = 25$
2. For each read, compute the median, average and stdev kmers coverage
3. Accept a read with a probability of:
$$\max \text{ coverage}/\text{median}$$

NGS reads normalization (by Trinity)

3. Accept a read with a probability of:

e.g. with *max coverage* = 30

Read_A: *median coverage* = 60 → $\frac{\text{max_coverage}}{\text{median}} = 0.5$

→ Read_A has a 50% chance of being kept

Read_B: *median coverage* = 10 → $\frac{\text{max_coverage}}{\text{median}} = 3$

→ Read_B has a 300% chance of being kept ;-)
→ Read_B will be kept

NGS reads normalization (by Trinity)

3. Accept a read with a probability of:

Reads coming from a highly expressed transcript and are several times more covered than the threshold.

- ➔ Its information is also contained by other reads.
- ➔ So it has less chance to be kept.

Reads coming from a low expressed transcript, way below the threshold.

- ➔ Its information is not very redundant, need it for the assembly.
- ➔ So it will absolutely be kept

NGS reads normalization (by Trinity)

1. Count kmers in all the data (Jellyfish):
 - with $k = 25$
2. For each read, compute the median, average and std dev kmers coverage
3. Accept a read with a probability of: $\text{maxcov}/\text{median}$
4. Remove a read if: $\text{standartdev}/\text{average} \text{ (CV)} > 1$ (100%)

A high variability in a read kmer coverage means there is probably a lot of sequencing errors in this read

Stand alone normalisation

```
$TRINITY_HOME/util/insilico_read_normalization.pl  
\ --seqType fq --JM 1G --max_cov 50  
\ --left lib1_1.P.qtrim --right lib2_2.P.qtrim  
\ --pairs_together --output insil_norm_ex
```

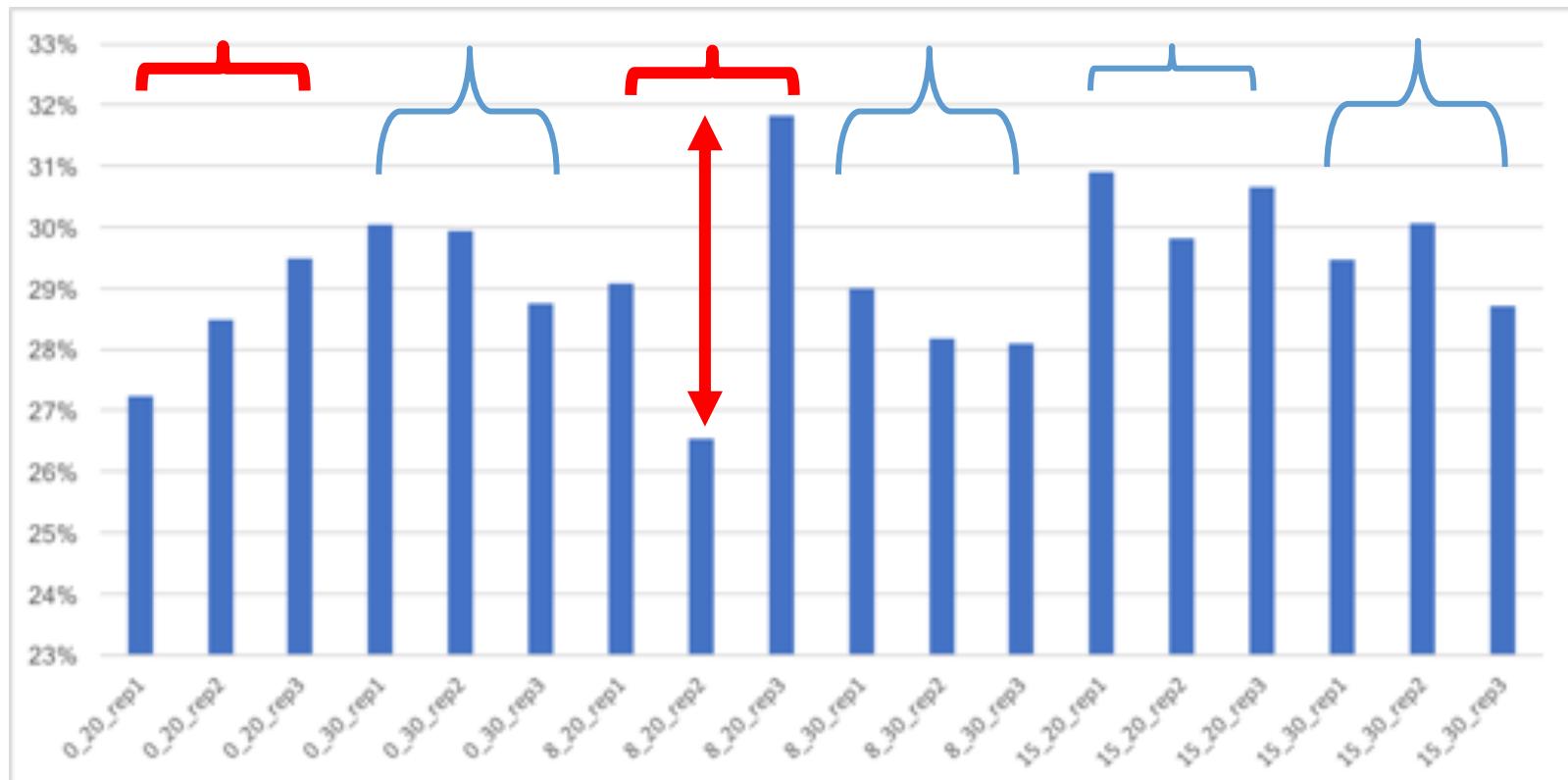
1189570 / 1879312 = 63.30% reads selected during normalization.
1094 / 1879312 = 0.06% reads discarded as likely aberrant based on coverage profiles.

Normalization complete. See outputs:

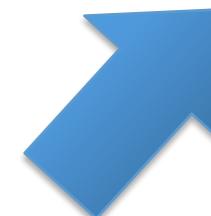
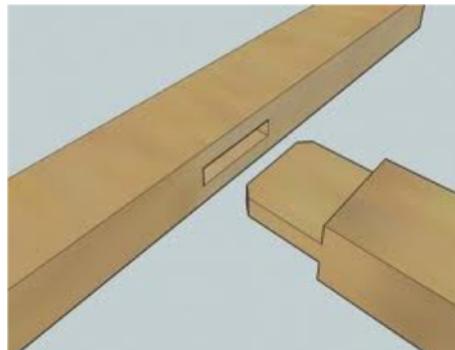
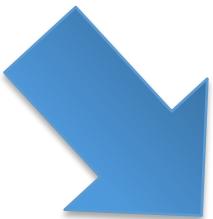
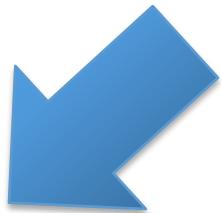
insil_norm_ex/lib1_1.P.qtrim.normalized_K25_C50_pctSD200.fq
insil_norm_ex/lib1_2.P.qtrim.normalized_K25_C50_pctSD200.fq

Trinity normalisation

```
Trinity --seqType fq --max_memory 50G --CPU 4  
--samples_file sample.txt --trimmomatic  
--quality_trimming_params "ILLUMINACLIP:illumina.fa:2:30:10  
SLIDINGWINDOW:4:5 LEADING:5 TRAILING:5 MINLEN:25  
--normalize_by_read_set
```



RNA Seq analysis





Transcriptome assembly

ASSEMBLY QUALITY ASSESSMENT AND CLEANING

Assembly quality assessment

- Generating Assembly metrics
- Comparing the assembled sequences to the reads used to generate them (reference-free)
- Aligning the sequences of conserved gene domains found in mRNA transcripts to transcriptomes or genomes of closely related species (reference-based).

Metrics

- The number of contigs in the assembly
- The size of the smallest contig
- The size of the largest contig
- The number of bases included in the assembly
- The mean length of the contigs
- The number of contigs <200 bases
- The number of contigs >1,000 bases
- The number of contigs >10,000 bases
- The number of contigs that had an open reading frame
- The mean % of the contig covered by the ORF
- NX (e.g. N50): the largest contig size at which at least X% of bases are contained in contigs at least this length
- % Of bases that are G or C
- GC skew
- AT skew
- The number of bases that are N
- The proportion of bases that are N
- The total linguistic complexity of the assembly

De novo Transcriptome Assembly is Prone to Certain Types of Errors

Error type	Transcripts	Assembly	Read evidence
Family collapse	geneAA geneAB geneAC n=3	n=1	
Chimerism	geneC geneB n=2	n=1	
Unsupported insertion	n=1	n=1	no reads align to insertion
Incompleteness	n=1	n=1	read pairs align off end of contig
Fragmentation	n=1	n=4	bridging read pairs
Local misassembly	n=1	n=1	read pairs in wrong orientation
Redundancy	n=1	n=3	all reads assign to best contig

Realignment metrics

The Assembly is a sum-up.

The realignment rate gives how much of the initial information is inside the contigs.

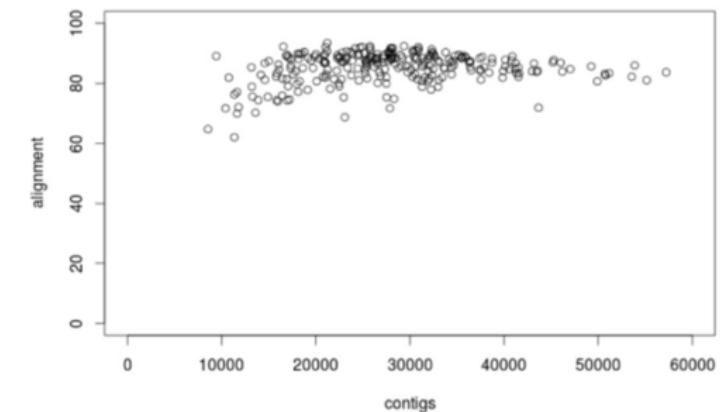
-> compute percentage of reads mapped

Factors affecting realignment rate:

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

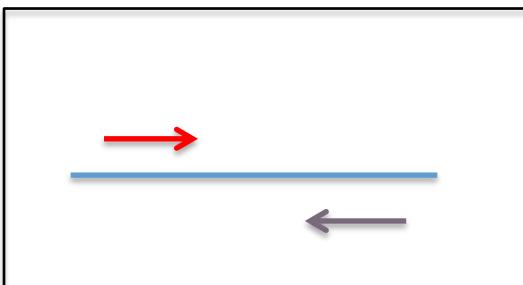
Realignment metrics

A typical ‘good’ assembly has ~80 % reads mapping to the assembly and ~80% are properly paired.

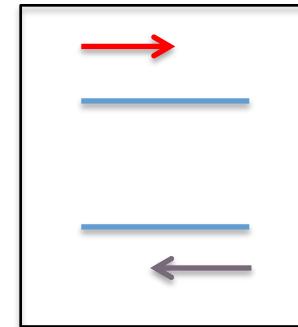


Given read pair:   Possible mapping contexts in the Trinity assembly are reported:

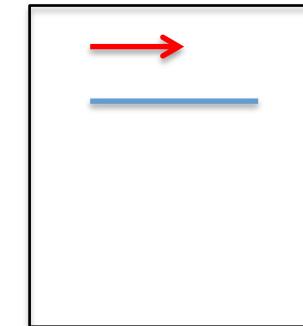
Proper pairs



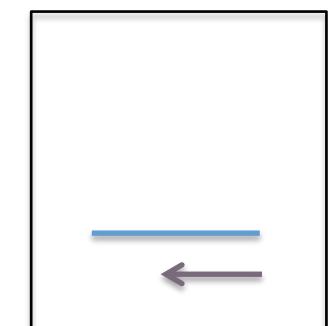
Improper pairs



Left only



Right only



Assembly evaluation : read remapping

Alignment methods : bowtie2 -RSEM

```
$TRINITY_HOME/util/align_and_estimate_abundance.pl --seqType fq  
--transcripts Trinity.fasta --est_method RSEM --aln_method bowtie2  
--prep_reference --trinity_mode --samples_file samples.txt --  
seqType fq
```

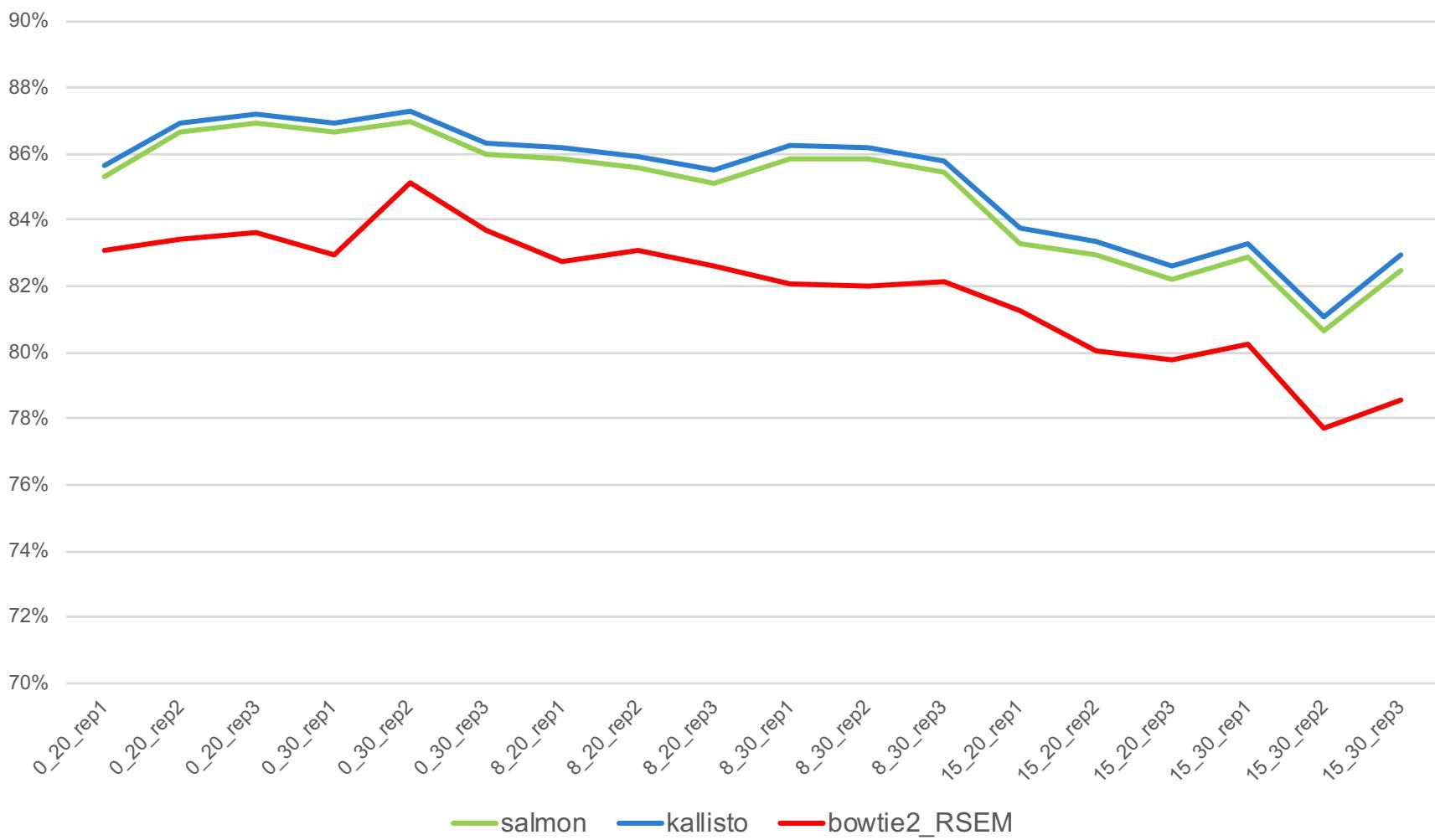
Pseudo-Alignment methods : kallisto

```
$TRINITY_HOME/util/align_and_estimate_abundance.pl --seqType fq  
--transcripts Trinity.fasta --est_method kallisto --prep_reference  
--trinity_mode --samples_file samples.txt --seqType fq
```

Pseudo-Alignment methods : salmon

```
$TRINITY_HOME/util/align_and_estimate_abundance.pl --seqType fq  
--transcripts Trinity.fasta --est_method salmon --prep_reference --  
trinity_mode --samples_file samples.txt --seqType fq
```

Realignment metrics



Assembly evaluation : read remapping

Pseudo-Alignment methods : kallisto (salmon : quant.sf ; quant.sf.genes)

```
head cond_A_rep1/abundance.tsv | column -t
```

Or

```
head cond_A_rep1/abundance.tsv.genes | column -t
```

target_id	length	eff_length	est_counts	tpm
TRINITY_DN144_c0_g1_i1	4833	4703.42	138	16.266
TRINITY_DN144_c0_g2_i1	2228	2098.42	0.000103136	2.72479e-05
TRINITY_DN179_c0_g1_i1	1524	1394.42	227	90.2502
TRINITY_DN159_c0_g1_i1	659	529.534	7.75713	8.12123
TRINITY_DN159_c0_g2_i1	247	119.949	0.24287	1.12251
TRINITY_DN153_c0_g1_i1	2378	2248.42	16	3.9451
TRINITY_DN130_c0_g1_i1	215	89.2898	776	4818.09
TRINITY_DN130_c1_g1_i1	295	166.986	216	717.115
TRINITY_DN106_c0_g1_i1	4442	4312.42	390	50.137

target_id	length	eff_length	est_counts	tpm
TRINITY_DN2774_c0_g1	2926.00	2796.42	31.00	6.15
TRINITY_DN5482_c0_g1	3064.00	2934.42	344.00	64.99
TRINITY_DN6803_c0_g1	1439.00	1309.42	1379.00	583.85
TRINITY_DN386_c0_g2	4279.00	4149.42	3.23	0.43
TRINITY_DN23_c0_g2	632.00	502.53	9.99	11.02
TRINITY_DN5348_c0_g1	2091.00	1961.42	264.00	74.62
TRINITY_DN5222_c0_g1	2416.00	2286.42	148.00	35.89
TRINITY_DN4680_c0_g1	1420.00	1290.42	167.00	71.75
TRINITY_DN2900_c0_g1	283.00	155.12	1.00	3.57

Expression matrix construction

```
$TRINITY_HOME/util/abundance_estimates_to_matrix.pl
\ --est_method kallisto --out_prefix Trinity_trans
\ --name_sample_by_basedir
\ cond_A_rep1/abundance.tsv
\ cond_A_rep2/abundance.tsv
\ cond_B_rep1/abundance.tsv
\ cond_B_rep2/abundance.tsv
```

Two matrices,

- one containing the estimated counts,
- one containing the TPM expression values that are cross-sample normalized using the TMM method.

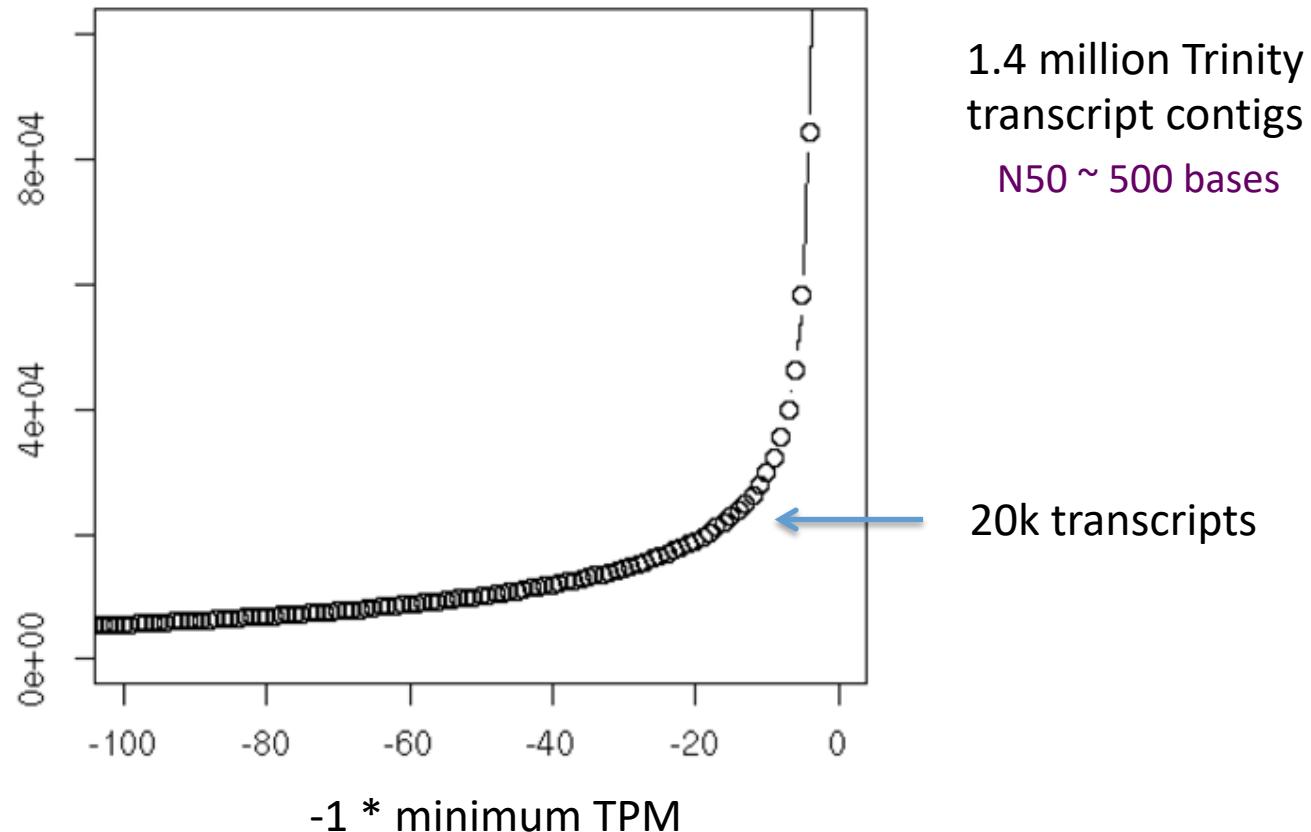
TMM normalization assumes that most transcripts are not differentially expressed, and linearly scales the expression values of samples to better enforce this property.

[A scaling normalization method for differential expression analysis of RNA-Seq data, Robinson and Oshlack, Genome Biology 2010.](#)

Alternative to N50 ?

Often, most assembled transcripts are *very* lowly expressed
(How many ‘transcripts & genes’ are there really?)

Cumulative
of
Transcripts

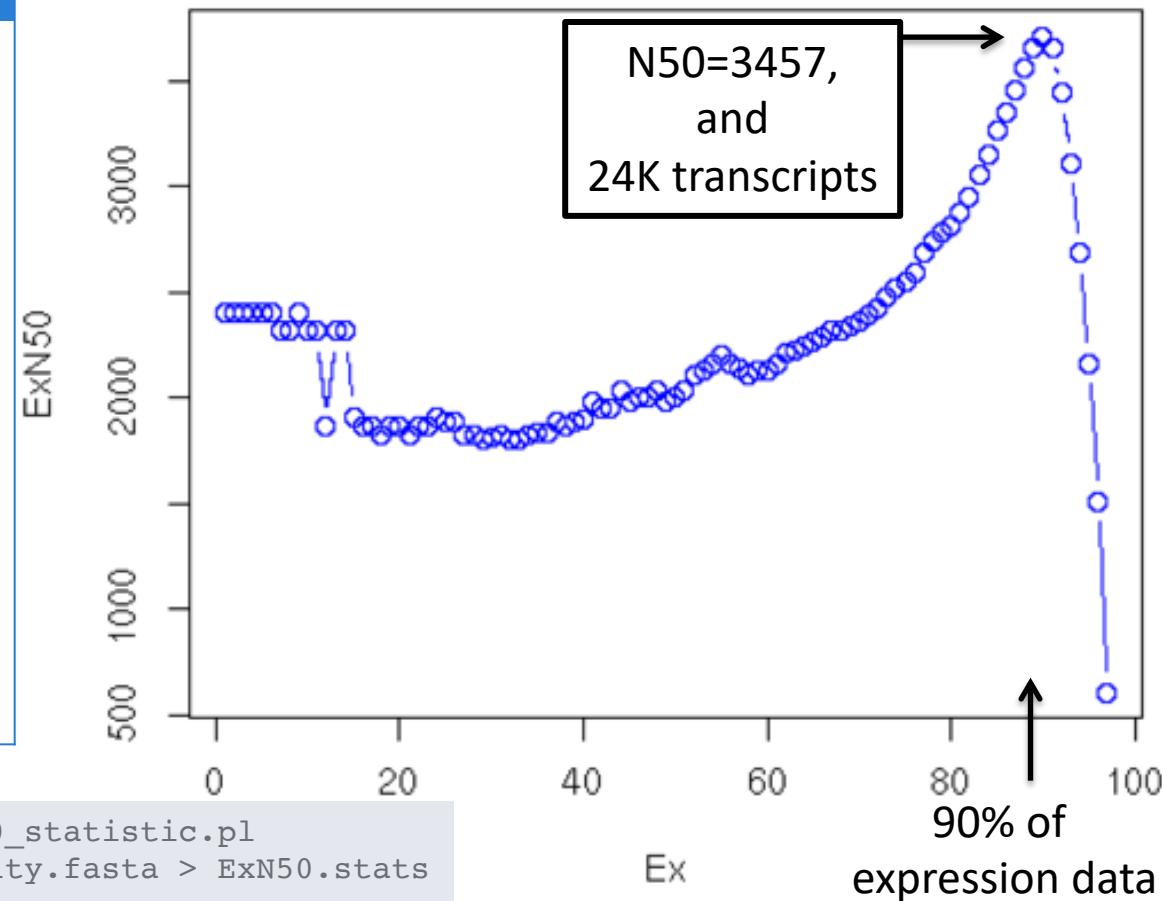


Alternative to N50 : ExN50 – E90N50

Compute N50 Based on the Top-most Highly Expressed Transcripts (ExN50)

- Sort contigs by expression value, descendingly.
- Compute N50 given minimum % total expression data thresholds => ExN50

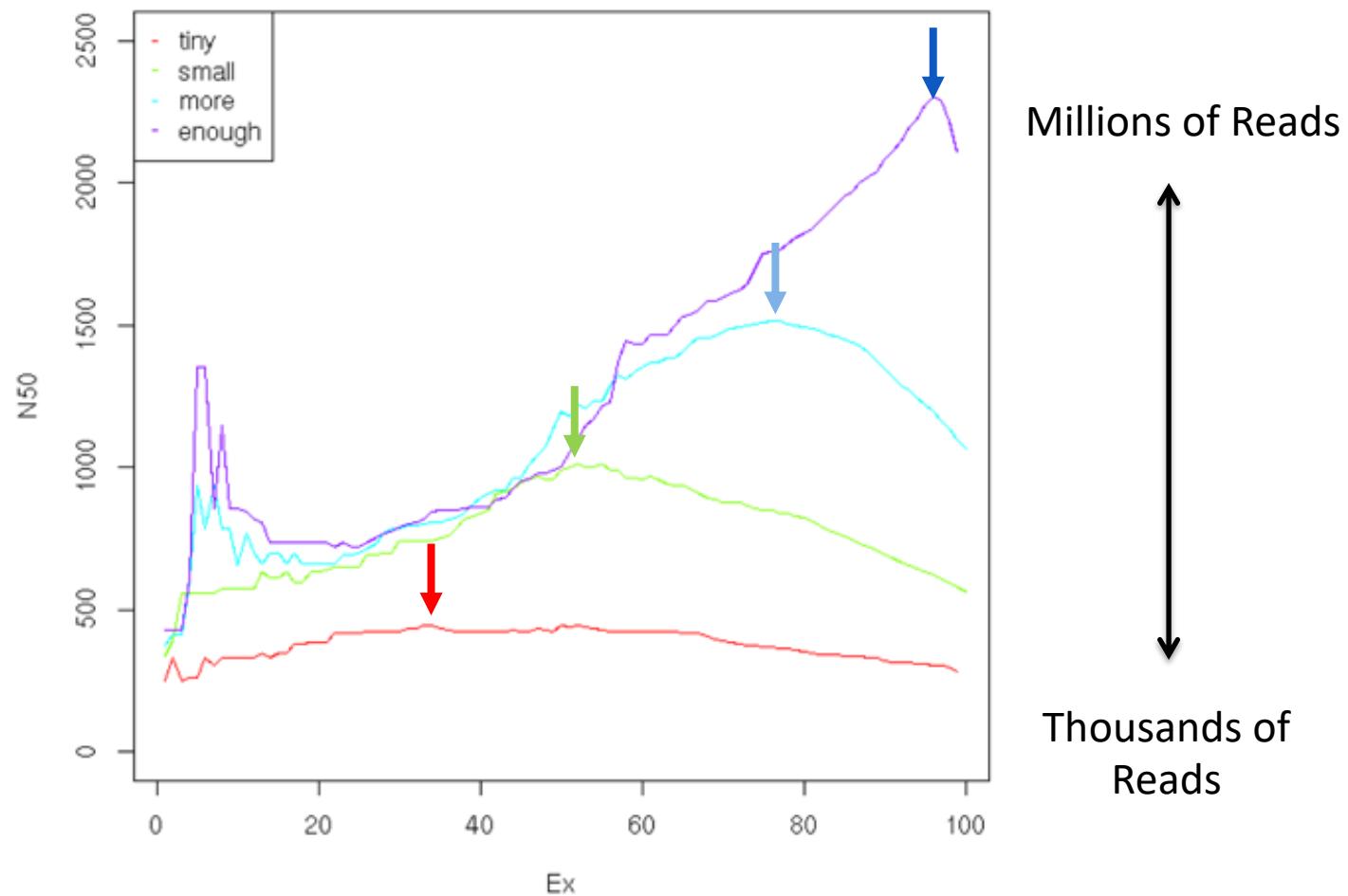
#E	min_expr	E-N50	num_transcripts
E2	89129.251	2397	1
E3	89129.251	2397	2
E5	66030.692	2397	3
E6	66030.692	2397	4
E8	66030.692	2397	5
...
E86	9.187	3056	12309
E87	7.044	3149	14261
E88	6.136	3261	16646
E89	4.538	3351	19635
E90	3.939	3457	23471
E91	3.077	3560	28583
E92	2.208	3655	35832
E93	1.287	3706	47061
...
E97	0.235	2683	275376
E98	0.164	2163	428285
E99	0.128	1512	668589
E100	0	606	1554055



```
$TRINITY_HOME/util/misc/contig_ExN50_statistic.pl
\Trinity_trans.TMM.EXPR.matrix Trinity.fasta > ExN50.stats
```

Ex

ExN50 Profiles for Different Trinity Assemblies Using Different Read Depths



Note shift in ExN50 profiles as you assemble more and more reads.

Tools to evaluate transcriptomes

Transrate: understand your transcriptome assembly. <http://hibberdlab.com/transrate>

Transrate analyses a transcriptome assembly in three key ways:

- by inspecting the contig sequences
- by mapping reads to the contigs and inspecting the alignments
- by aligning the contigs against proteins or transcripts from a related species and inspecting the alignments
 - Assemblies score
 - Contigs score
 - Optimised assemblies score (filter out bad contigs from an assembly, leaving you with only the well-assembled ones)



BUSCO analysis

CEGMA (<http://korflab.ucdavis.edu/datasets/cegma/>)

HMM:s for 248 core eukaryotic genes aligned to your assembly to assess completeness of gene space

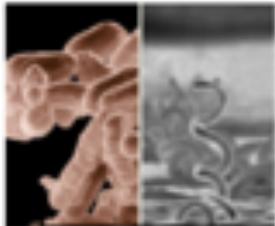
“complete”: 70% aligned

“partial”: 30% aligned

BUSCO(<http://busco.ezlab.org/>)

Assessing genome assembly and annotation completeness with Benchmarking Universal Single-Copy Orthologs

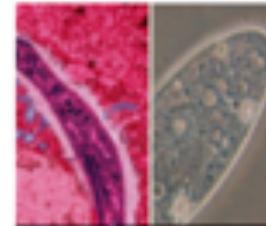
Datasets (Beta versions, updated sets and additional lineages coming soon)



Bacteria sets



Eukarya sets



Protists sets



Metazoa sets



Fungi sets



Plants set

Arthropods:



Vertebrates:



Fungi:



Bacteria:



Metazoans:



&



&



Eukaryotes:



&



&



&



Plants:



Early access available upon [request](#).

BUSCO Results

```
# BUSCO was run in mode: transcriptome EUKARYOTES
```

```
C:86.5%[S:48.2%,D:38.3%],F:7.6%,M:5.9%,n:303
```

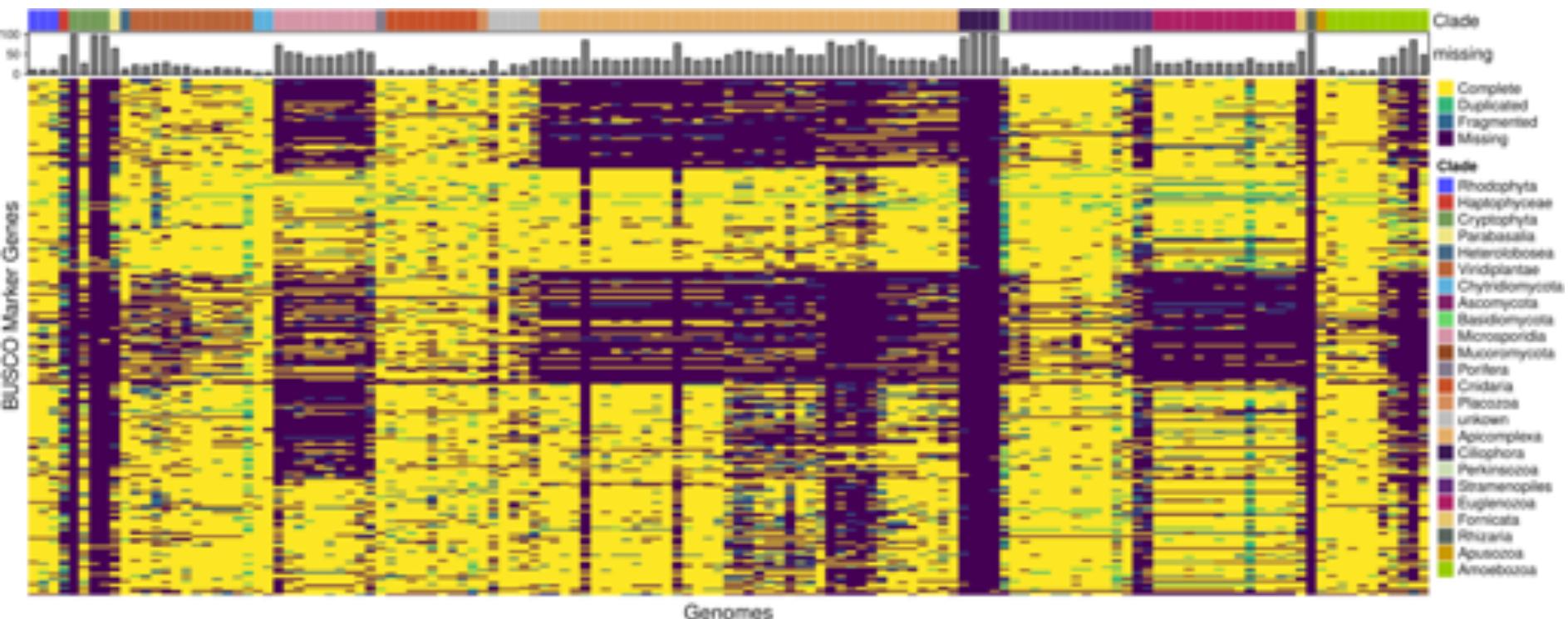
```
262 Complete BUSCOs (C)
146 Complete and single-copy BUSCOs (S)
116 Complete and duplicated BUSCOs (D)
23 Fragmented BUSCOs (F)
18 Missing BUSCOs (M)
303 Total BUSCO groups searched
```

```
# BUSCO was run in mode: transcriptome PLANT
```

```
C:13.9%[S:8.1%,D:5.8%],F:2.0%,M:84.1%,n:1440
```

```
200 Complete BUSCOs (C)
117 Complete and single-copy BUSCOs (S)
83 Complete and duplicated BUSCOs (D)
29 Fragmented BUSCOs (F)
1211 Missing BUSCOs (M)
1440 Total BUSCO groups searched
```

BUSCO limitation



Rob Finn - Paul Saari EBI : Personal comm.

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**
- **SOAPdenovo-trans**,
- **rnaSPADES**

New de novo transcriptome assemblers

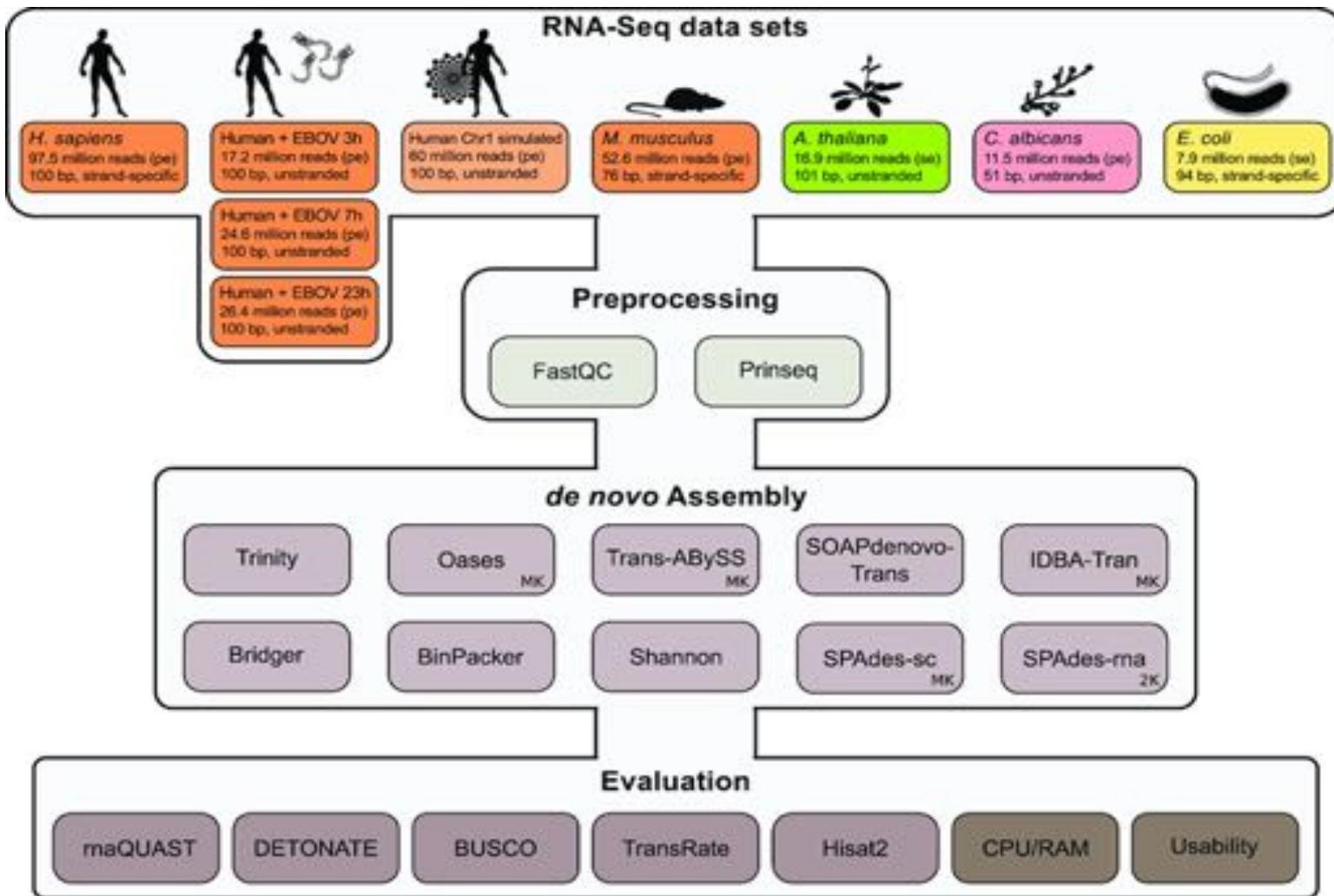
- IDBA-Tran (Peng et al., Bioinf., 2014)
- IDBA-MTP (Peng et al., RECOMB 2014)
- SOAPdenovo-Trans (Xie et al., Bioinf., 2014)
- Fu et al., ICCABS, 2014
- StringTie (Pertea et al., Nat. Biotech., 2015)
- Bermuda (Tang et al., ACM, 2015)
- Bridger (Chang et al., Gen. Biol. 2015)
- BinPacker (Liu et al. PLOS Comp Biol, 2016)
- Frama (Bens M et al., BMC Genomics 2016)
- rnaSPAdes (Bushanova et al., *GigaScience* 2019)



Assemblers comparison

- 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
- (Vijay et al., 2013) Challenges and strategies in transcriptome assembly and differential gene expression quantification. A comprehensive in silico assessment of RNA-seq experiments. *Molecular ecology*. PMID: 22998089
- (Haas et al., 2013) De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. *Nature protocols*. PMID: 23845962
- (Lu et al., 2013) Comparative study of de novo assembly and genome-guided assembly strategies for transcriptome reconstruction based on RNA-Seq. *Sci China Life Sci.*
- 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
- (He et al., 2015) Optimal assembly strategies of transcriptome related to ploidies of eukaryotic organisms. *BMC genomics*. DOI: 10.1186/s12864-014-1192-7
- S. B. Rana, F. J. Zadlock IV, Z. Zhang, W. R. Murphy, and C. S. Bentivegna, “Comparison of De Novo Transcriptome Assemblers and k-mer Strategies Using the Killifish, *Fundulus heteroclitus*,” *PLoS ONE*, vol. 11, no. 4, p. e0153104, Apr. 2016.
- (Wang and Gribkov, 2016) Comprehensive evaluation of de novo transcriptome assembly programs and their effects on differential gene expression analysis. *Bioinformatics*. PMID: 27694201
- **M. Hölzer and M. Marz, “De novo transcriptome assembly: A comprehensive cross-species comparison of short-read RNA-Seq assemblers,” *Gigascience*, vol. 8, no. 5, pp. 57–16, May 2019.**

Assemblers comparison



Assemblers comparison

