Assemblage de-novo de transcriptome
Trinity
Ecole ITMO 2017

ABiMS – Station Biologique Roscoff
INTRODUCTION
A Paradigm for Genomic Research

WGS Sequencing

Assemble

Draft Genome Scaffolds

Methylation

Tx-factor binding sites

SNPs

Proteins
A Paradigm for Genomic Research

WGS Sequencing → Assemble → Draft Genome Scaffolds

- Methylation
- Tx-factor binding sites
- SNPs

→ Transcripts
→ Proteins
→ Expression

RNA-Seq → Align
A *Maturing* Paradigm for Transcriptome Research

WGS Sequencing → Assemble

Assemble → Draft Genome Scaffolds

Draft Genome Scaffolds → Align

Align → RNA-Seq

RNA-Seq → Assemble

Assemble → Transcripts

Transcripts → Expression

Expression → SNPs

SNPs → Proteins

Proteins → Methylation

Methylation → Tx-factor binding sites

Tx-factor binding sites → Draft Genome Scaffolds
A *Maturing* Paradigm for Transcriptome Research

WGS Sequencing

Assemble

Align

Draft Genome Scaffolds

Methylation

Tx-factor binding sites

SNPs

Proteins

Expression

Transcripts

RNA-Seq

+
A Maturing Paradigm for Transcriptome Research

WGS Sequencing → Assemble → Draft Genome Scaffolds

RNA-Seq → Align → Assemble

Methylation

Tx-factor binding sites

SNPs → Proteins → Expression
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

Input (fastqc)

FastQC

FastQC HTML Report

Trimmomatic

FastQC

FastQC HTML Report

SortmeRNA

FastQC

FastQC HTML Report
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:4:5 LEADING:5 TRAILING:5 MINLEN:25
```
FastQC: Per sequence GC content

- A contamination?
FastQC: Per sequence GC content

- A contamination?

Can this be fixed? Maybe…
FastQC: Per sequence GC content

GC distribution over all sequences

- GC count per read
- Theoretical Distribution

Mean GC concen (%)
Prior to sequencing:
- Ribodepletion kits

After sequencing:
- Remove rRNA reads from raw reads
- Detect rRNA transcripts
SortMeRNA

collection of reads

read

rRNA database
SILVA, Greengenes,...

search for common approximate 18-mers

align against the rRNA database

selection cut off

other

rRNA + alignment
SortMeRNA commands

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

>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
TRANSCRIPTOME ASSEMBLY STRATEGIES
Contemporary strategies for transcript reconstruction from RNA-Seq

RNA-Seq reads

Spliced alignment of RNA-Seq to genome

De novo transcript assembly

Transcript reconstruction from spliced alignment of assembled transcripts to genome

Gmap

Tophat
STAR
HISAT2

Genome

Transcript reconstruction from RNA-Seq spliced alignments

Cufflinks
Stringtie
IsoLasso
Bayesembler
Trip
Traph
CEM
TransComb

Oases
SoapDenovoTrans
AbyssTrans
IDBA-Trans
Shannon
BinPacker
Bridger
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!!)
Trinity – How it works:

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

Thousands of disjoint graphs
Inchworm Contigs from Alt-Spliced Transcripts

Expressed isoforms

Isoform A

Isoform B

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.
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
Trinity usage and options

**Typical Trinity command**

Trinity --seqType fq --max_memory 50G --left A_repl_left.fq  --right A_repl_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

```plaintext
>TRINITY_DN889_c0_g1_i1  len=259  path=[473:0–258] [-1, 473, -2]
GAACAATGTCTACACTGCTTTCAACTTGGGATGACAAGGAACTTTTCATTGGCTCAAGCTAA
CTACAATTCCATCTCTGAAACAGATATTGAAGAAATCAAGGATACTGTGCCTCGCTGTT
GCTGGCCTCCACAATACTACAACACATTTCTCAGCTGACCCAAACTGCCACTGCACTGG
TAACATCTTTGCAACAGAGGCCACTATGTCCATGGCCTGCTCCAGCTAATGCTTTCTAGAA
CTCTTCATTAAACTCTCCT

>TRINITY_DN810_c0_g1_i2  len=226  path=[407:0–225] [-1, 407, -2]
GATGATATCAACAATGAGACTTTGTGAAACCAGTGGAGAAATCAAGGATACTGTCCCTAG
CTAGGTGAATTGAAGATTGTACGCTAATGTGTAACGCTGCGTACTTCCAGCCA
TTTACGCTGTCAAGTGTAACCGATATTGAACCGAGATAATAAGAGAAATTGGCTGACCTCGGA
```

**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**

```bash
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

<table>
<thead>
<tr>
<th>cond</th>
<th>cond_A_rep</th>
<th>A_left</th>
<th>A_right</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A_rep1</td>
<td>A_rep1_left.fq</td>
<td>A_rep1_right.fq</td>
</tr>
<tr>
<td>B</td>
<td>B_rep1</td>
<td>B_rep1_left.fq</td>
<td>B_rep1_right.fq</td>
</tr>
<tr>
<td>B</td>
<td>B_rep2</td>
<td>B_rep2_left.fq</td>
<td>B_rep2_right.fq</td>
</tr>
<tr>
<td>B</td>
<td>B_rep3</td>
<td>B_rep3_left.fq</td>
<td>B_rep3_right.fq</td>
</tr>
</tbody>
</table>

Trinity --seqType fq --max_memory 50G --CPU 4
  --samples_file sample.txt
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).
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:4:5 LEADING:5 TRAILING:5 MINLEN:25"
```

- **Normalisation:**
  - By definition RNAseq display a wide range of expressions
    Very low expressed \(\rightarrow\) 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

\(\rightarrow\) How to decrease coverage of highly expressed transcripts without decreasing that of low expressed transcripts?
In silico normalization of reads

High

Moderate

Low
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:
   \[
   \text{max coverage/median}
   \]
3. Accept a read with a probability of:

e.g. with $max\ coverage = 30$

$\text{Read}_A$: $median\ coverage = 60 \rightarrow \frac{max\ coverage}{median} = 0.5$

$\Rightarrow$ Read_A has a 50% chance of being kept

$\text{Read}_B$: $median\ coverage = 10 \rightarrow \frac{max\ coverage}{median} = 3$

$\Rightarrow$ Read_B has a 300% chance of being kept ;-

$\Rightarrow$ Read_B will be kept
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.
1. Count kmers in all the data (Jellyfish):
   - with $k = 25$

2. For each read, compute the median, average and standard deviation kmers coverage

3. Accept a read with a probability of: $\frac{\text{maxcov}}{\text{median}}$

4. Remove a read if: $\frac{\text{standarddev}}{\text{average}} \ (\text{CV}) > 1$ (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

- Assembly metrics
- Reads mapping back rate
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

<table>
<thead>
<tr>
<th>Error type</th>
<th>Transcripts</th>
<th>Assembly</th>
<th>Read evidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Family collapse</td>
<td>geneAA, geneAB,</td>
<td></td>
<td>[Graph showing bases in reads and coverage]</td>
</tr>
<tr>
<td></td>
<td>geneAC, n=3</td>
<td>n=1</td>
<td></td>
</tr>
<tr>
<td>Chimerism</td>
<td>geneC, n=2</td>
<td>n=1</td>
<td>[Graph showing no reads align to insertion]</td>
</tr>
<tr>
<td>Unsupported insertion</td>
<td>n=1</td>
<td>n=1</td>
<td>[Graph showing read pairs align off end of contig]</td>
</tr>
<tr>
<td>Incompleteness</td>
<td></td>
<td>n=1</td>
<td>[Graph showing bridging read pairs]</td>
</tr>
<tr>
<td>Fragmentation</td>
<td></td>
<td>n=1</td>
<td>[Graph showing read pairs in wrong orientation]</td>
</tr>
<tr>
<td>Local misassembly</td>
<td></td>
<td>n=1</td>
<td>[Graph showing all reads assign to best contig]</td>
</tr>
<tr>
<td>Redundancy</td>
<td></td>
<td>n=1</td>
<td></td>
</tr>
</tbody>
</table>

Smith-Unna et al. Genome Research, 2016
The assembly is a sum-up. The realignment rate gives how much of the initial information is inside the contigs.

- align reads against assembly generated transcripts
- compute percentage of reads mapped
Factors affecting realignment rate:

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

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:
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

- salmon
- kallisto
- bowtie2_RSEM
Assembly evaluation: read remapping

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

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

<table>
<thead>
<tr>
<th>target_id</th>
<th>length</th>
<th>eff_length</th>
<th>est_counts</th>
<th>tpm</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRINITY_DN2774_c0_g1</td>
<td>2926.00</td>
<td>2796.42</td>
<td>31.00</td>
<td>6.15</td>
</tr>
<tr>
<td>TRINITY_DN5482_c0_g1</td>
<td>3064.00</td>
<td>2934.42</td>
<td>344.00</td>
<td>64.99</td>
</tr>
<tr>
<td>TRINITY_DN6803_c0_g1</td>
<td>1439.00</td>
<td>1309.42</td>
<td>1379.00</td>
<td>583.85</td>
</tr>
<tr>
<td>TRINITY_DN386_c0_g2</td>
<td>4279.00</td>
<td>4149.42</td>
<td>3.23</td>
<td>0.43</td>
</tr>
<tr>
<td>TRINITY_DN23_c0_g2</td>
<td>632.00</td>
<td>502.53</td>
<td>9.99</td>
<td>11.02</td>
</tr>
<tr>
<td>TRINITY_DN5348_c0_g1</td>
<td>2091.00</td>
<td>1961.42</td>
<td>264.00</td>
<td>74.62</td>
</tr>
<tr>
<td>TRINITY_DN5222_c0_g1</td>
<td>2416.00</td>
<td>2286.42</td>
<td>148.00</td>
<td>35.89</td>
</tr>
<tr>
<td>TRINITY_DN4680_c0_g1</td>
<td>1420.00</td>
<td>1290.42</td>
<td>167.00</td>
<td>71.75</td>
</tr>
<tr>
<td>TRINITY_DN2900_c0_g1</td>
<td>283.00</td>
<td>155.12</td>
<td>1.00</td>
<td>3.57</td>
</tr>
</tbody>
</table>
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.
Often, most assembled transcripts are *very* lowly expressed (How many ‘transcripts & genes’ are there really?)

* Salamander transcriptome

Cumulative # of Transcripts

1.4 million Trinity transcript contigs
N50 ~ 500 bases

20k transcripts

Expression
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

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

$TRINITY_HOME/util/misc/contig_ExN50_statistic.pl
\Trinity_trans.TMM.EXPR.matrix Trinity.fasta > ExN50.stats
Note shift in ExN50 profiles as you assemble more and more reads.

* Candida transcriptome
Since a reference genome is not available, the quality of computer-assembled contigs may be verified:

- by comparing the assembled sequences to the reads used to generate them (reference-free)

- by aligning the sequences of conserved gene domains found in mRNA transcripts to transcriptomes or genomes of closely related species (reference-based).
Tools to evaluate transcriptomes

**Transrate**: understand your transcriptome assembly. [http://hibberdlab.com/transrate](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)

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

“complete”: 70% aligned
“partial”: 30% aligned


Assessing genome assembly and annotation completeness with Benchmarking Universal Single-Copy Orthologs
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
• **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**
Assemblers comparison


This study compared four transcriptome assembly methods,
- a de novo assembler (Trinity)
- two transcriptome re-assembly strategies utilizing proteomic and genomic resources from closely related species (reference-based re-assembly and TransPS)
- a genome-guided assembler (Cufflinks)

« However, our results emphasize the efficacy of de novo assembly, which can be as effective as genome-guided assembly when the reference genome assembly is fragmented.

If a genome assembly and sufficient computational resources are available, it can be beneficial to combine de novo and genome-guided assemblies »
Assemblers comparison


• (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.


• (He et al., 2015) Optimal assembly strategies of transcriptome related to ploidies of eukaryotic organisms. BMC genomics. DOI: 10.1186/s12864-014-1192-7


• (Wang and Gribskov, 2016) Comprehensive evaluation of de novo transcriptome assembly programs and their effects on differential gene expression analysis. Bioinformatics. PMID: 27694201
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)