Agalma: an automated *de novo* transcriptome assembly pipeline

Mark Howison

CCV/EPSCoR Bioinformatics Workshop
October 19, 2012

Motivation

- “I have 30 million paired-end RNA-Seq reads and no reference genome. What next?”
- Automating *de novo* transcriptome assembly:
  - Filtering and assembling paired-end Illumina data
  - Collecting diagnostics and generating reports
  - Providing fault-tolerance
  - Profiling CPU and memory usage

Overview of pipeline

- ‘Sanitize’ reads according to quality criteria
- Estimate the insert size of the read pairs
- Remove ribosomal RNA
- Assemble

Common paradigm:

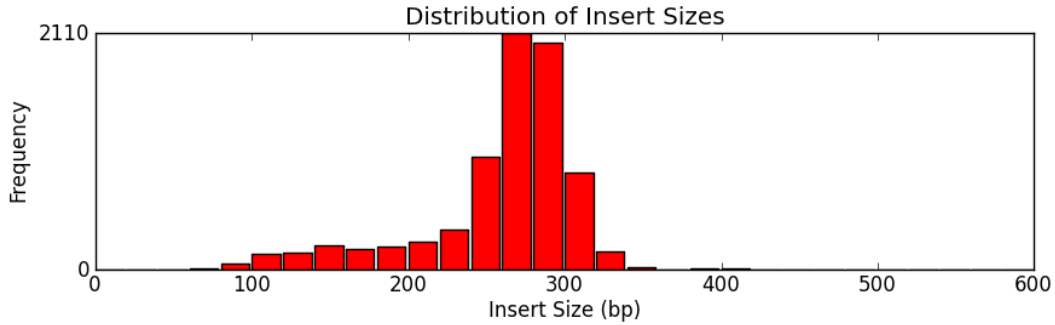
- Assemble a random subset of reads (Oases)
- Use the *subassembly* in a *mapping* (BLAST, Bowtie2)

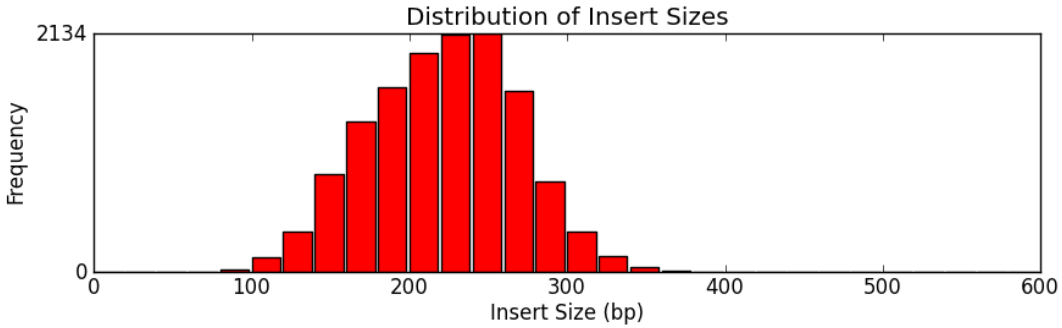
Sanitize

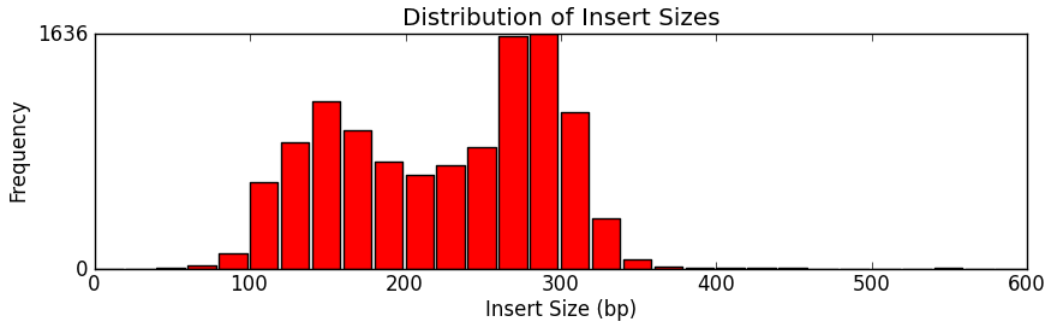
- Randomizes the order of read pairs
- Runs FastQC
- Discards reads that:
  - fall below a mean quality threshold
  - contain Illumina adapter sequences
  - have skewed base composition
    (any base is < 5% or > 60% of the sequence)

Estimate insert size

- *Subassembly*:
  100K high-quality reads
- *Mapping* (Bowtie2):
  10K read pairs against subassembly
- Extract estimated gap between pairs from SAM output
- Mean and stdev are used by downstream tools





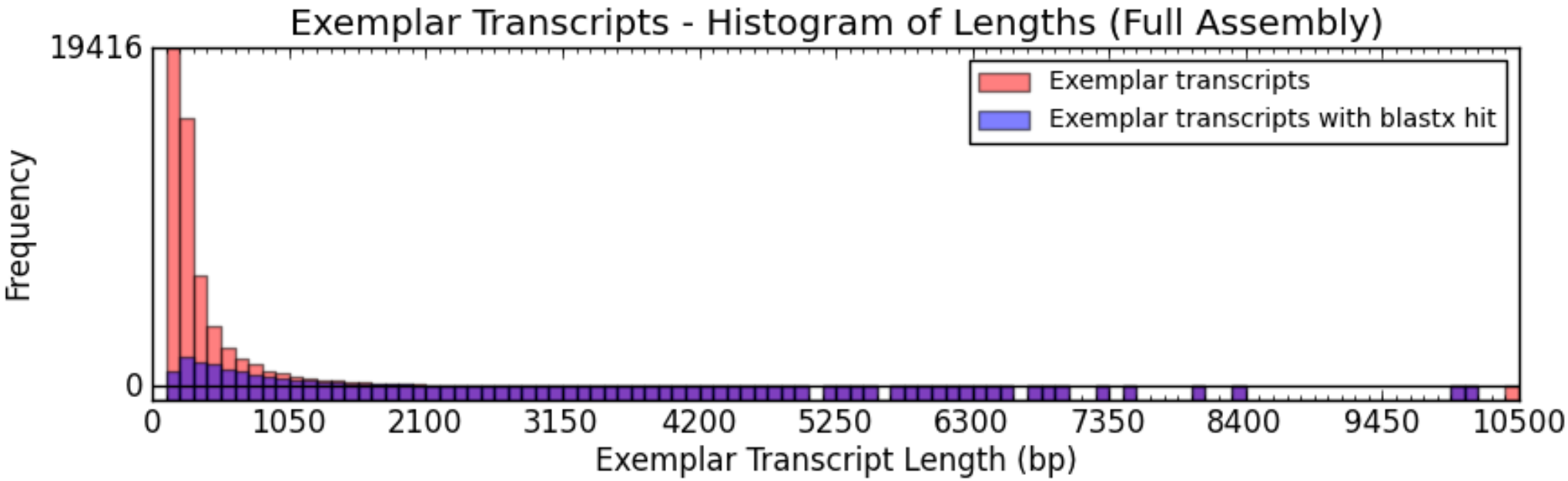


Remove ribosomal RNA

- *Subassemblies*:
  500 to 1M reads (log scaled)
- *Mapping* (BLASTN):
  subassemblies against curated rRNA sequences from a related species
  - Identify an “exemplar” sequence for each rRNA gene
- *Mapping* (Bowtie2):
  rRNA subassemblies against all reads
  - Excludes all reads with an rRNA hit

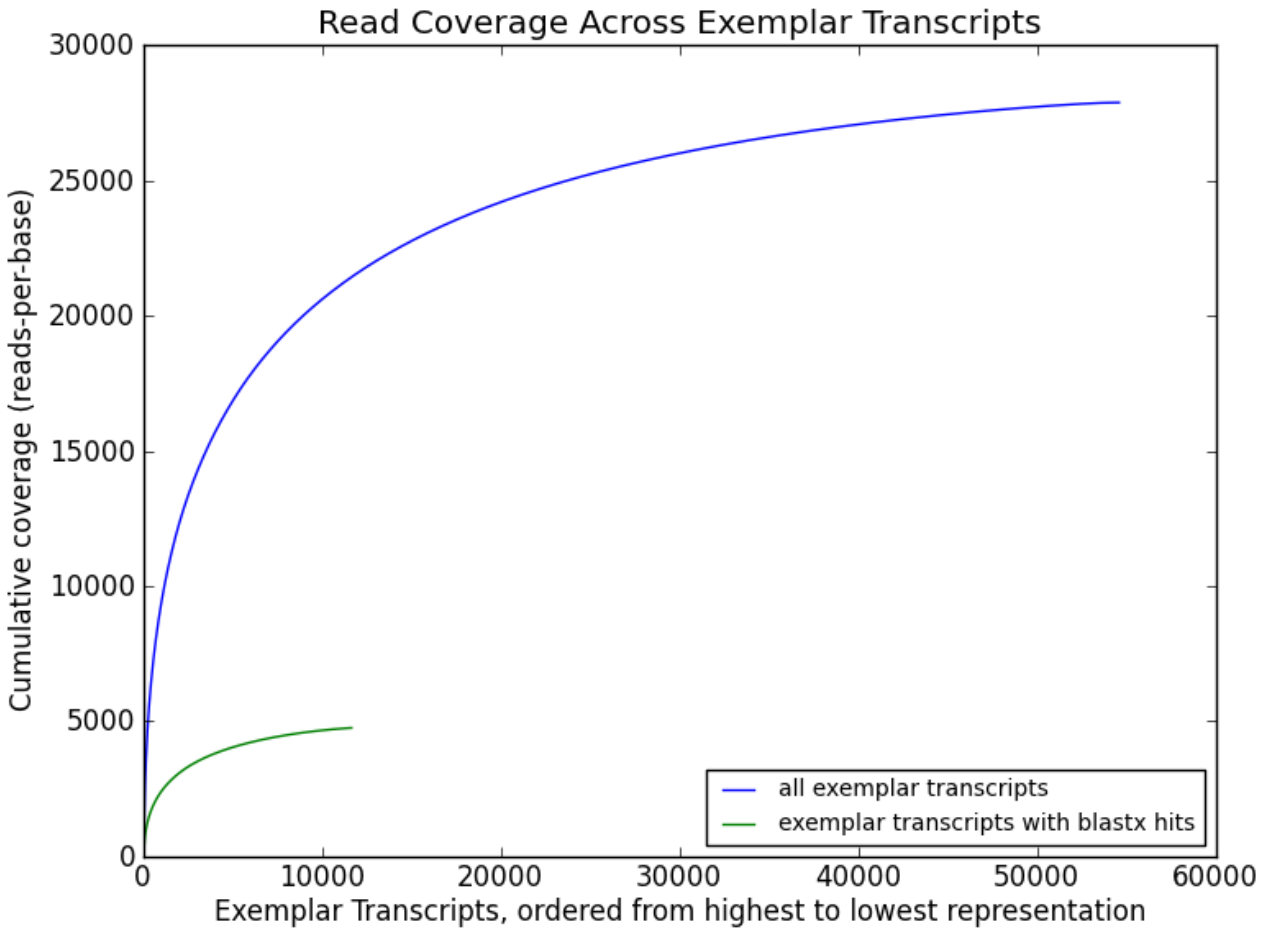
Assemble transcripts

- Filters reads again at a higher mean quality threshold
- Supports different assembly “protocols” (Oases, Trinity)
- Screens out rRNA and vector contaminates
- BLASTX against SwissProt: how much is “real”?



Assemble transcripts

- Map reads onto assembly to generate coverage map



Future directions

- More assembly protocols (e.g. Trinity)
- Assembly comparison
  - Within or between assemblers
- Normalizing reads prior to assembly
  - Reduces variation in read coverage
- Translation, annotation, phylogenomics…

Availability

- Source code is available on Github under a GPL license:

  https://github.com/caseywdunn/agalma

- On Oscar:

```
$ module load agalma
```

- Please contact me if you would like help using Agalma!

Acknowledgement

- Agalma developers:
  - Casey Dunn
  - Nick Sinnott-Armstrong
  - Felipe Zapata
- Thanks to Stefan Siebert, Steve Haddock, Warren Francis, and Lingsheng Dong for their feedback
- Funding from NSF [0844596, 0844596, 1004057]