LSU Bioinformatics Workshop Day 2

the chaos is real, I appreciate your patience!

loving the questions and comments!

ppt slides from Day 1 in the Google Drive (lsu_bioinformatics_day1_lecture.pdf)

Tuesday morning: shell scripting commands, navigating on the server

Tuesday afternoon: LSU Herbarium tour, transcriptome assembly and QC

shell script and job submission exercises

RNA-Seq workflow

mRNA

cDNA

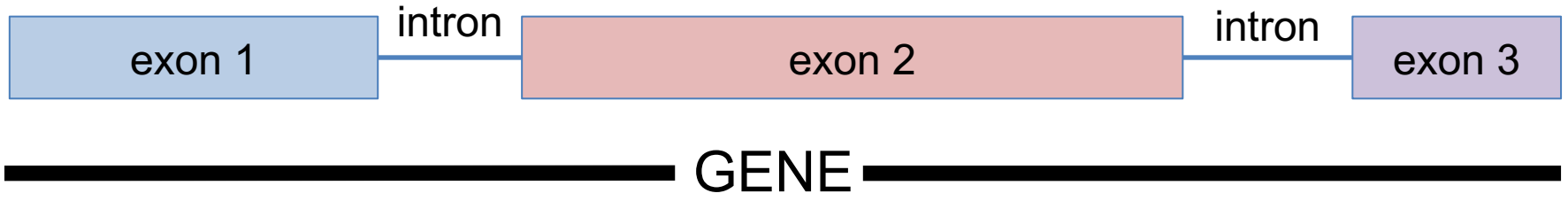
sequence the copies

assemble transcriptome

transcriptome assembly in Trinity

step 1: inchworm

- assembles the RNA-seq data into unique sequences of transcripts
- generates full-length transcripts for a dominant isoform, also reports unique isoforms

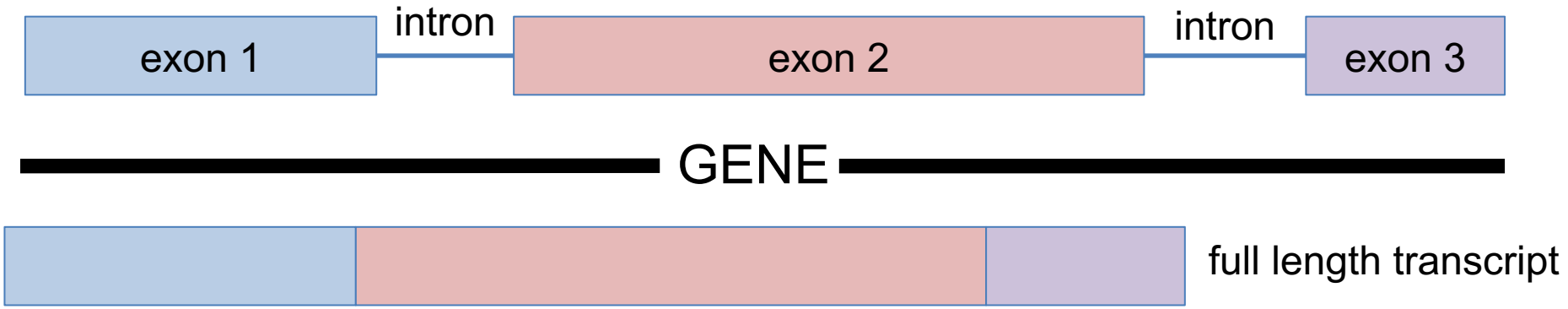


https://www.nature.com/articles/nbt.1883
https://github.com/trinityrnaseq/trinityrnaseq/wiki

transcriptome assembly in Trinity

step 1: inchworm

- assembles the RNA-seq data into unique sequences of transcripts
- generates full-length transcripts for a dominant isoform, also reports unique isoforms

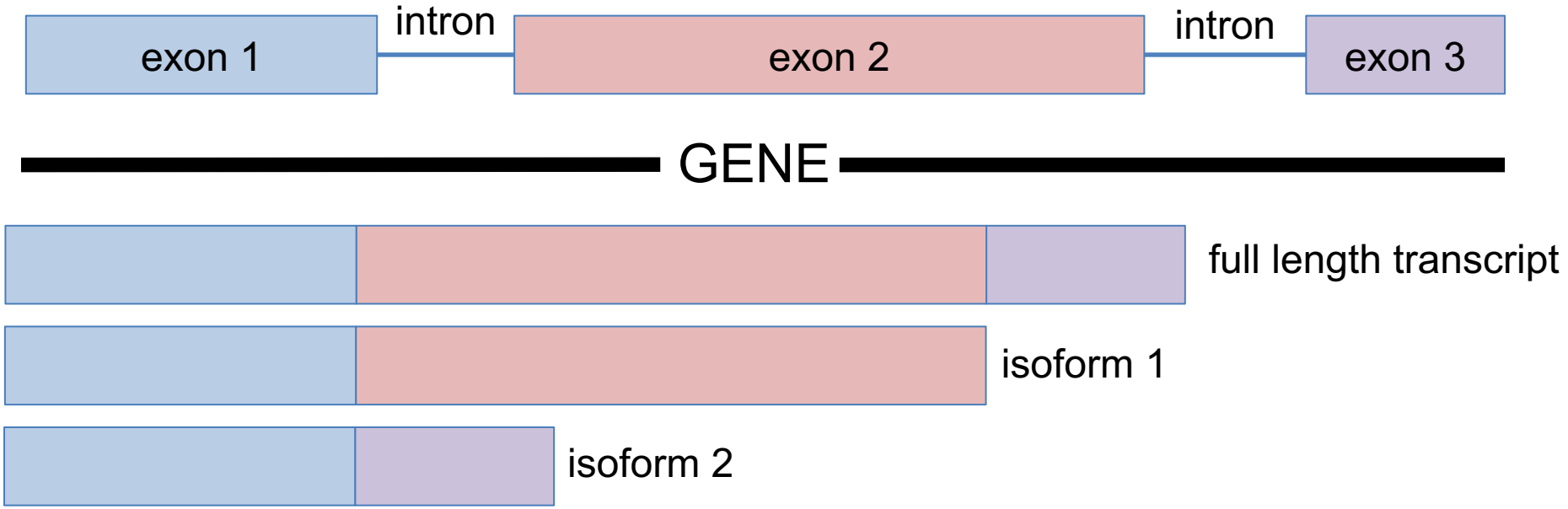


https://www.nature.com/articles/nbt.1883
https://github.com/trinityrnaseq/trinityrnaseq/wiki

transcriptome assembly in Trinity

step 1: inchworm

- assembles the RNA-seq data into unique sequences of transcripts
- generates full-length transcripts for a dominant isoform, also reports unique isoforms



https://www.nature.com/articles/nbt.1883
https://github.com/trinityrnaseq/trinityrnaseq/wiki

transcriptome assembly in Trinity

step 1: inchworm

- assembles the RNA-seq data into unique sequences of transcripts
- generates full-length transcripts for a dominant isoform, also reports unique isoforms

step 2: chrysalis

- clusters inchworm output and constructs de Bruijn graphs
- each cluster/graph represents the full transcriptional complexity for a given gene

step 3: butterfly

- processes de Bruijn graphs in parallel, tracing the paths that reads take within the graph, reporting full-length transcripts for alternatively spliced isoforms, and teasing apart transcripts that corresponds to paralogous genes.

https://www.nature.com/articles/nbt.1883
https://github.com/trinityrnaseq/trinityrnaseq/wiki

transcriptome assembly in Trinity

transcriptome assembly is computationally intensive, with large memory requirements

we will run Trinity with truncated input fastq files so jobs need less memory and do not take days to complete

https://www.nature.com/articles/nbt.1883
https://github.com/trinityrnaseq/trinityrnaseq/wiki

transcriptome quality control - how good is our assembly?

Benchmarking Universal Single-Copy Orthologs - BUSCO:

- compares a transcriptome or genome assembly to a set of single-copy core genes expected for a specified group (e.g., vertebrates, metazoans, eukaryotes)
- returns summary statistics that indicate how complete the assembly is (e.g., number of missing or duplicated genes)
- also reports general stats like assembly length, number of sequences, N50 value
  (N50 = 25Kb means that 50% of the sequences in the assembly are at least 25Kb long)

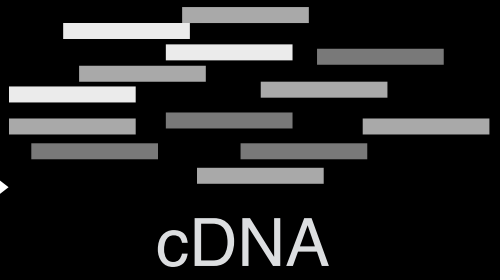
gVolante:

a free webserver that runs the BUSCO analysis quickly and easily

https://academic.oup.com/bioinformatics/article/31/19/3210/211866
https://gvolante.riken.jp/

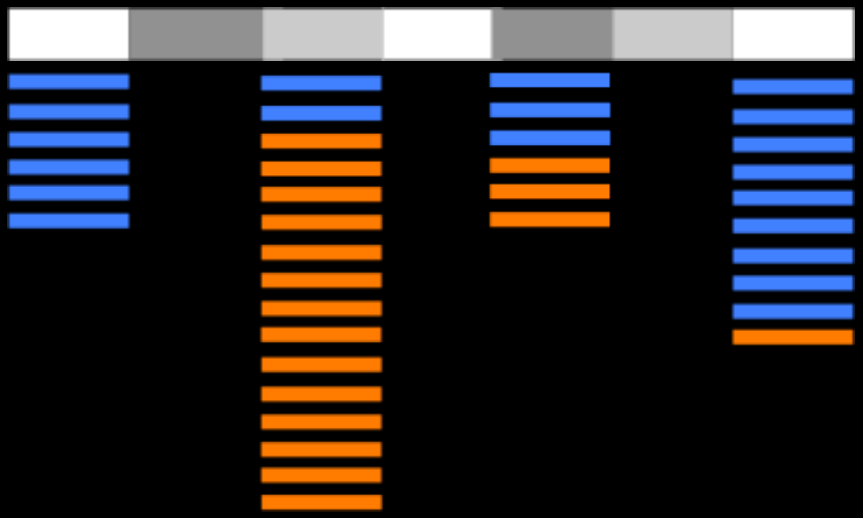
RNA-Seq workflow

mRNA



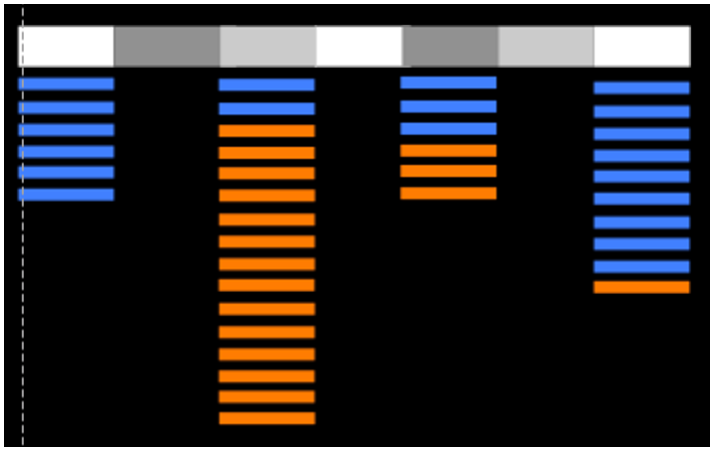
cDNA

count expressed transcripts



north population

south population



RSEM: mapping reads to the transcriptome

- user-friendly software package for quantifying gene and isoform abundances from single-end or paired-end RNA-Seq data
- program runs in multiple steps
- the output is a matrix of read counts for each transcript/gene or isoform

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-12-323
https://deweylab.github.io/RSEM/README.html

trinity, busco, rsem exercises

00-README

- text file created in each directory where you are running an analysis
- contains detailed information about what happened during the run
- should also contain, or point to, the exact command lines you used to run the job

```
These files were generated using the files truncated to 5 million reads: cvar.trinity
Also ran analyses on the full (_full) infiles and on files truncated to 1 million reads
(_1mill)

First run terminated in an error about bowtie, which is odd bc the slurm.out file says
bowtie got loaded:
/work/mdebia1/00-SUB/trin_cvar.err.bowtie_error
"Loading mvapich2/2.3.3/intel-19.0.5
  Loading requirement: intel/19.0.5
Loading trinity/2.12.0/intel-19.0.5
  Loading requirement: gcc/9.3.0 bowtie2/2.3.5.1/gcc-9.3.0
    samtools/1.10/intel-19.0.5"

ERROR in trin.out: Use of uninitialized value $bowtie2_build_path in concatenation (.) or
string at /usr/local/packages/trinity/2.12.0/intel-19.0.5/Trinity line 2052.

Found this post on the Trinity github page that suggests using the --no_bowtie flag, which
worked in the next run
https://github.com/trinityrnaseq/trinityrnaseq/issues/1020

full command lines are here:
/work/mdebia1/00-SUB/trinity.pbs
```