Trinity

Trinity is a de Bruijn graph based *de novo* assembler for RNA seq data. It has been published in 2011 by Grabherr et al. in Nature Biotechnology. As with all our software tools, Trinity is not necessarily the all time best solution for assembling RNA seq data, however, it is certainly a good start. You can find a presentation of how to run Trinity assemblies HERE.

Haas et al. 2013 published a protocol that describes the use of Trinity in the reconstruction of transcripts using a de novo assembly strategey. Follow this LINK to access the publication.

Running Trinity

To run a de novo transcriptome assembly Trinity requires the reads, file type (Fastq) and the maximum amount of memory it may use.

Example code:

```
Trinity --seqType fq --max_memory 64G --left <Path/to/forward/reads.fq> \
--right <Path/to/reversed/reads.fq> --output
<path/to/output/location/>Trinity/ --verbose --CPU 12 --no_normalize_reads
```

Think about which of the parameters are important and why - maybe you can leave some of them out. Also check your version of Trinity - if options are not available that indicates that you have not installed the latest version (ask your tutor for help). For more options visit the Trinity site or open the --help in the terminal. It also should be mentioned that when running the script over the slurm controller, the memory and CPU count should be equal to the amount stated in the script.

Output

The final transcriptome assembly is documented in the "**Trinity.fasta**" file in the output directory that was given.

An example Fasta entry for one of the transcripts is formatted like so:

```
>TRINITY_DN1000|c115_g5_i1 len=247 path=[31015:0-148 23018:149-246]
 AATCTTTTTTGGTATTGGCAGTACTGTGCTCTGGGTAGTGATTAGGGCAAAAGAAGACAC
 ACAATAAAGAACCAGGTGTTAGACGTCAGCAAGTCAAGGCCTTGGTTCTCAGCAGACAGA
 AGACAGCCCTTCTCAATCCTCATCCCTTCCCTGAACAGACATGTCTTCTGCAAGCTTCTC
 CAAGTCAGTTGTTCACAGGAACATCATCAGAATAAATTTGAAATTATGATTAGTATCTGA
 TAAAGCA
```

The header encodes the Trinity 'gene' and 'isoform' information. In the example above, the header 'TRINITY_DN1000|c115_g5_i1' indicates Trinity read cluster 'TRINITY_DN1000|c115', gene 'g5', and isoform 'i1'. Because a given run of trinity involves many many clusters of reads, each of which are assembled separately, and because the 'gene' numberings are unique within a given processed read cluster, the 'gene' identifier should be considered an aggregate of the read cluster and corresponding

gene identifier, which in this case would be 'TRINITY_DN1000|c115_g5'.
The Path information stored in the header (“path=[31015:0-148 23018:149-246]”) indicates the path traversed in the Trinity compacted de Bruijn graph to construct that transcript.

Visit the following LINK to access further information about the Trinity output.

Additional Information

Trinity run-time depends on a number of factors including the number of reads to be assembled and the complexity of the transcript graphs. The assembly from start to finish can take anywhere from ~1/2 hour to 1 hour per million reads.

For a quick Nx statistic run “TrinintyStats.pl”

```
/path/to/TrinityStats/TrinityStats.pl  Trinity.fasta
```

This will print out the conventional N50 statistics for your Transcripts and also the stats based on single isoforms (“genes”).
More info can be found here.