Transcript quantification

We will be using RSEM to quantify the expression levels of the transcripts that have been assembled by Trinity. But because we do not have proper biological replicates for the Red Sisken RNAseq data, we will be using *Candida glabrata* transcriptome data. The paper from which these data are derived examined *C. glabrata* in two conditions, nutrient rich (wt) and under nitrosative stress (GNSO).

Make sure that you are in your `/pool/genomics/<username>/RNAseq_SMSC/` directory.

To use RSEM, we will use the wrapper script included in the Trinity package called `align_and_estimate_abundance.pl`.

This script makes it very easy to take Trinity output and run RSEM.

Create another job file with QSubGen. This will be a serial job and you should reserve 4GB of memory. It will also run fine in the short queue.

Load the trinity module and enter the following program command:

```
align_and_estimate_abundance.pl --seqType fq \
         --left data/diff_ex/GSNO_SRR1582648_1.fastq \
         --right data/diff_ex/GSNO_SRR1582648_2.fastq  \
         --transcripts data/diff_ex/trinity_out_dir.Trinity.fasta  \
         --est_method RSEM  --aln_method bowtie \
         --trinity_mode --prep_reference --coordsort_bam \
         --output_dir GSNO_SRR1582648.RSEM
```

Save the script into a file called `trinity_rsem_GNSO_SRR1582648.job`. Since there are six biological replicates, we'll need to make six of these files -- one for each replicate.

Take some time to create the five other files for the other treatments. Be sure to change both the names of the reads, and the `--output_dir` according to the sample name. Double check that you are calling both the correct reads and using the correct names. It is always important to check twice, run once.

Now you can submit each of these jobs using `qsub job_file_name`. This is the magic of parallel computing clusters--you can run many jobs at once.

These jobs should run pretty fast. Once they are finished, you can check the output with, e.g.:

`$ head GSNO_SRR1582648.RSEM/RSEM.genes.results | column -t`

Your output should look something like:

```
gene_id             transcript_id(s)        length  effective_length    expected_coun
t  TPM     FPKM
TRINITY_DN100_c0_g1 TRINITY_DN100_c0_g1_i1  253.00  123.83              2.00
   1014.01 4866.18
TRINITY_DN103_c0_g1 TRINITY_DN103_c0_g1_i1  524.00  394.48              57.00
   9071.81 43534.98
TRINITY_DN103_c1_g1 TRINITY_DN103_c1_g1_i1  152.00  27.95               3.00
   6739.18 32340.83
TRINITY_DN104_c0_g1 TRINITY_DN104_c0_g1_i1  174.00  47.05               0.00
   0.00    0.00
TRINITY_DN105_c0_g1 TRINITY_DN105_c0_g1_i1  221.00  92.16               0.00
   0.00    0.00
TRINITY_DN105_c1_g1 TRINITY_DN105_c1_g1_i1  238.00  108.94              1.00
   576.30  2765.60
TRINITY_DN107_c0_g1 TRINITY_DN107_c0_g1_i1  161.00  35.48               1.00
   1769.44 8491.42
TRINITY_DN108_c0_g1 TRINITY_DN108_c0_g1_i1  190.00  62.01               0.00
   0.00    0.00
TRINITY_DN108_c1_g1 TRINITY_DN108_c1_g1_i1  195.00  66.79               1.00
   940.01  4511.05
```

Generate a transcript counts matrix and perform cross-sample normalization

Each rsem file holds the expression estimates for each of the samples. Now we will use these samples to create a counts matrix and to perform cross-sample normalization. The script included in the Trinity packages does this for you according to the TMM method. If you want to read more about this normalization method, you can do so in this paper, "A scaling normalization method for differential expression analysis of RNA-seq data."

Go ahead and generate a job file using the QSubGen. Assuming that you set up the remaining 5 jobs the same way that you did the first job, your job command would be:

```
abundance_estimates_to_matrix.pl --est_method RSEM \
      --gene_trans_map data/diff_ex/trinity_out_dir.Trinity.fasta.gene_trans_map \
      --out_prefix Trinity_trans --name_sample_by_basedir \
      GSNO_SRR1582648.RSEM/RSEM.isoforms.results \
      GSNO_SRR1582646.RSEM/RSEM.isoforms.results \
      GSNO_SRR1582647.RSEM/RSEM.isoforms.results \
      wt_SRR1582649.RSEM/RSEM.isoforms.results \
      wt_SRR1582651.RSEM/RSEM.isoforms.results \
      wt_SRR1582650.RSEM/RSEM.isoforms.results
```

Choose a serial key and 2GB of memory. Either save the job file and transfer it to Hydra, or copy and paste the text into your terminal window with `nano`. Subit the job.

This command will run estimates for both genes and isoforms. The output of this command will produce:

- RSEM.isoform.counts.matrix : the estimated RNA-Seq fragment counts (raw counts)
- RSEM.isoform.TPM.not_cross_norm : a matrix of TPM expression values (not cross-sample normalized)
- RSEM.isoform.TMM.EXPR.matrix : a matrix of TMM-normalized expression values

First lets look at the at the first 20 lines of the isoform counts.

```
$ head -20 Trinity_trans.isoform.counts.matrix | column -t
```

The resulting output will look someting like this:

```
GSNO_SRR1582648         GSNO_SRR1582646  GSNO_SRR1582647  wt_SRR1582649  wt_SRR158265
1  wt_SRR1582650
TRINITY_DN270_c0_g1_i1  0.00             0.00             0.00           1.00
   0.00           1.00
TRINITY_DN286_c0_g1_i1  38.00            36.00            39.00          5.00
   7.00           8.00
TRINITY_DN472_c0_g1_i1  2.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN269_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN378_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN258_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN392_c0_g1_i1  0.00             0.00             2.00           0.00
   0.00           0.00
TRINITY_DN407_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN328_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN61_c0_g1_i1   0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN221_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN260_c1_g1_i1  3.00             4.00             7.00           8.00
   8.00           5.00
TRINITY_DN357_c0_g1_i1  0.00             0.00             0.00           0.00
   0.00           0.00
TRINITY_DN63_c1_g1_i1   19.00            26.00            33.00          3.00
   4.00           3.00
TRINITY_DN401_c0_g1_i1  2.00             3.00             4.00           3.00
   6.00           2.00
TRINITY_DN637_c0_g1_i1  0.00             0.00             0.00           1.00
   2.00           1.00
TRINITY_DN288_c0_g1_i1  0.00             0.00             1.00           0.00
   0.00           0.00
TRINITY_DN625_c0_g1_i1  0.00             0.00             1.00           2.00
   3.00           1.00
TRINITY_DN97_c0_g1_i1   0.00             0.00             0.00           0.00
   0.00           0.00
```

Now look at the output generated for the isoforms from the TMM normalized counts:

```
$ head -20 Trinity_trans.isoform.TMM.EXPR.matrix | column -t
```

The output from this file will look a bit different. As described in the paper linked to above, this normalization method assumes that most transcripts are not differentially expressed and linearly scales the values with that assumption in mind.

```
GSNO_SRR1582648         GSNO_SRR1582646  GSNO_SRR1582647  wt_SRR1582649  wt_SRR158265
1  wt_SRR1582650
TRINITY_DN270_c0_g1_i1  0.000            0.000            0.000          1382.140
   0.000          1599.862
TRINITY_DN286_c0_g1_i1  19831.723        15748.460        16547.800      2036.568
   3442.755       3763.621
TRINITY_DN472_c0_g1_i1  3139.756         0.000            0.000          0.000
   0.000          0.000
TRINITY_DN269_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN378_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN258_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN392_c0_g1_i1  0.000            0.000            2511.305       0.000
   0.000          0.000
TRINITY_DN407_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN328_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN61_c0_g1_i1   0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN221_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN260_c1_g1_i1  1701.427         1902.282         3228.057       3540.731
   4288.504       2556.256
TRINITY_DN357_c0_g1_i1  0.000            0.000            0.000          0.000
   0.000          0.000
TRINITY_DN63_c1_g1_i1   11708.546        13440.787        16537.510      1442.567
   2337.418       1666.528
TRINITY_DN401_c0_g1_i1  1288.233         1621.389         2095.379       1507.745
   3671.449       1161.284
TRINITY_DN637_c0_g1_i1  0.000            0.000            0.000          1155.640
   2970.275       1337.249
TRINITY_DN288_c0_g1_i1  0.000            0.000            243.398        0.000
   0.000          0.000
TRINITY_DN625_c0_g1_i1  0.000            0.000            1048.765       2006.101
   3821.984       1160.229
TRINITY_DN97_c0_g1_i1   0.000            0.000            0.000          0.000
   0.000          0.000
```

We can also examine the generated matrices for genes.

```
$ head -20 Trinity_trans.gene.counts.matrix | column -tt
```

```
$ head -20 Trinity_trans.gene.TMM.EXPR.matrix | column -t
```

Now that we have expression values we can estimate some new statistics. We will use the expression quantification values to calculate the ExN50, which is restricted to only the most highly expressed transcripts. This is more informative of the quality of the assembly since it only uses the transcripts that have suitable coverage.

The file that you generate from QSubGen should choose the short queue and the default RAM. Load the `bioinformatics/trinity/2.6.6` module. The command that you should use is:

```
contig_ExN50_statistic.pl data/diff_ex/Trinity_trans.TMM.EXPR.matrix \
data/diff_ex/trinity_out_dir.Trinity.fasta > ExN50.stats
```

When your job is completed, you will see a new output file called, `ExN50.stats`. Go ahead and take a look at your new file with:

```
$ cat ExN50.stats | column -t
```

Your output should look like this:

```
#E    min_expr   E-N50  num_transcripts
E2    48510.841  470    1
E4    35522.525  470    2
E6    35522.525  329    3
E7    35522.525  470    4
E9    33698.327  453    5
E11   28346.056  453    6
E13   22770.403  453    7
E14   22770.403  416    8
E16   22770.403  417    9
E17   22770.403  417    10
E18   22770.403  417    11
E20   22770.403  416    12
E21   22770.403  416    13
E23   22770.403  417    14
E24   16940.365  416    15
E25   16940.365  417    16
E27   16940.365  453    17
```

```
E28   16940.365  459    18
E29   16940.365  470    19
E30   16940.365  459    20
E31   16940.365  459    21
E32   16940.365  453    22
E33   16940.365  459    23
E35   16940.365  459    24
E36   13602.022  470    25
E37   13602.022  470    26
E38   13602.022  470    27
E39   12842.900  470    29
E40   12842.900  470    30
E41   12842.900  470    31
E42   12842.900  470    32
E43   12842.900  470    33
E44   12842.900  470    34
E45   12842.900  470    36
E46   12842.900  470    37
E47   12842.900  470    38
E48   12099.821  473    40
E49   12099.821  476    41
E50   12099.821  476    43
E51   11837.930  485    44
E52   9272.015   485    46
E53   9272.015   489    48
E54   9272.015   489    49
E55   9272.015   489    51
E56   9272.015   490    53
E57   9272.015   490    55
E58   9272.015   489    57
E59   7363.329   485    59
E60   7363.329   485    61
E61   7363.329   489    64
E62   7363.329   489    66
E63   7363.329   489    68
E64   5522.466   485    71
E65   5522.466   476    74
E66   4922.224   473    76
E67   4623.903   473    79
E68   4623.903   473    82
E69   4623.903   470    86
E70   4623.903   470    89
E71   4623.903   470    92
E72   4225.842   468    96
E73   4225.842   459    99
```

```
E74   4225.842   456    103
E75   3584.598   453    107
E76   3584.598   453    111
E77   3584.598   443    115
E78   3584.598   440    119
E79   3584.598   434    124
E80   3584.598   422    128
E81   3584.598   422    133
E82   3584.598   420    138
E83   3024.328   417    144
E84   3024.328   416    150
E85   3024.328   416    156
E86   1835.197   415    162
E87   1835.197   404    168
E88   1835.197   388    175
E89   1835.197   380    183
E90   1835.197   376    190
E91   1398.490   373    198
E92   1398.490   362    207
E93   1398.490   355    215
E94   1398.490   341    224
E95   1398.490   337    234
E96   1398.490   329    245
E97   1398.490   328    256
E98   1211.604   320    269
E99   932.114    308    285
E100  0.265      320    324
E100  0          285    690
```

Now you can look at the N50 stats scaled by the genes with the highest expression. If, for example, you only wanted to see the N50 for the 60% highest expressed genes, it would be ~485.