6- First Pass Assembly & QC

Thursday morning

Bernardo J. Clavijo
Richard Smith-Unna
Gonzalo Garcia
The K tradeoff

- **Longer kmers** are more unique in the target, **disentangling** the graph.
- **Smaller kmers** will overlap more often, favouring **contiguity**.
- Every read produces $L-k+1$ kmers.
  - Higher $k$ -> less coverage.
- Every single error affects $k$ kmers.
  - Higher $k$ -> more errors.
- A typical choice for 100bp reads is $k=71$. 
Running abyss as a first pass assembler

- It runs easily and can use both single and multi-host multiprocessing.
- Creates a ton of useful output, and a nice log.

```bash
runabyss.sh  nyc3574_k61-5.fa
coverage.hist  nyc3574_k61-5.path
nyc3574_k61-1.fa  nyc3574_k61-5.adj
nyc3574_k61-bubbles.fa  nyc3574_k61-6.fa
nyc3574_k61-1.adj  nyc3574_k61-contigs.fa
nyc3574_k61-2.adj  nyc3574_k61-6.dot
nyc3574_k61-1.path  nyc3574_k61-contigs.dot
nyc3574_k61-3.adj  lmp1-6.hist
nyc3574_k61-2.path  lmp1-6.dist.dot
nyc3574_k61-3.fa  lmp2-6.hist
nyc3574_k61-indel.fa  lmp2-6.dist.dot
nyc3574_k61-unitigs.fa  nyc3574_k61-6.path1.dot
pe1-3.hist  nyc3574_k61-6.path1
pe1-3.dist  nyc3574_k61-6.path2
nyc3574_k61-3.dist  nyc3574_k61-6.fa
nyc3574_k61-4.fa  nyc3574_k61-7.fa
nyc3574_k61-4.adj  nyc3574_k61-7.dot
nyc3574_k61-4.path1  nyc3574_k61-scaffolds.fa
nyc3574_k61-4.path2  nyc3574_k61-scaffolds.dot
nyc3574_k61-4.path3  nyc3574_k61-stats
nyc3574_k61-4.log  Redirected Log
```
Beware of N50

• N50 is the most used metric in assembly world… and it should not be:
  • Using contiguity as primary goal reward “risky joining”.
  • N50 is affected by filtering, and not very sensitive!
Contiguity stats

<table>
<thead>
<tr>
<th>n</th>
<th>n:200</th>
<th>n:N50</th>
<th>min</th>
<th>N80</th>
<th>N50</th>
<th>N20</th>
<th>max</th>
<th>sum</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>10773</td>
<td>1353</td>
<td>144</td>
<td>200</td>
<td>11170</td>
<td>25592</td>
<td>44031</td>
<td>96106</td>
<td>11.6e6</td>
<td>nyc3574_k61-unitigs.fa</td>
</tr>
<tr>
<td>8880</td>
<td>497</td>
<td>53</td>
<td>200</td>
<td>32554</td>
<td>66307</td>
<td>139116</td>
<td>322315</td>
<td>11.79e6</td>
<td>nyc3574_k61-contigs.fa</td>
</tr>
<tr>
<td>8615</td>
<td>232</td>
<td>8</td>
<td>200</td>
<td>269923</td>
<td>551245</td>
<td>1029531</td>
<td>1372216</td>
<td>11.74e6</td>
<td>nyc3574_k61-scaffolds.fa</td>
</tr>
</tbody>
</table>

- Don’t forget to check your “Ns” !!!
Fragment Sizes

![Graphs showing fragment sizes and their corresponding counts.]

- The top graph displays fragment sizes ranging from 0 to 400 in bins of 10 bp.
- The bottom graph shows fragment sizes ranging from -6000 to 0 in bins of 10 bp.

**Fragment Size**

- X-axis: Fragment Size (bins of 10bp)
- Y-axis: Fragment Count

**Fragment Size**

- X-axis: Fragment Size (bins of 10bp)
- Y-axis: Fragment Count
Read mapping stats

```
abyss-map -j150 -l61 /scratch/clavijob/yeast_tests/diploid/s_3_1_sequence.txt /scratch/clavijob/yeast_tests/diploid/s_3_2_sequence.txt nyc3574_k61-3.fa |
  |abyss-fixmate -h pel-3.hist |
  |sort -snk3 -k4 |
  |DistanceEst -j150 -k61 -l61 -s200 -n10 -o pel-3.dist pel-3.hist
Building the suffix array...
Building the Burrows-Wheeler transform...
Building the character occurrence table...
Mateless 0
Unaligned 71619 1.14%
Singleton 516328 8.19%
FR 4018144 63.8%
RF 28 0.000444%
FF 8285 0.131%
Different 1686337 26.8%
Total 6300741
```
Read mapping stats

```
abyss-map -j150 -161 /scratch/clavijob/yeast_tests/diploid/LIB3796_clipped_A_R1.fastq /scratch/clavijob/yeast_tests/diploid/LIB3796_clipped_A_R2.fastq nyc3574_k61-6.fa \
    |abyss-fixmate -h lmp1-6.hist \ 
    |sort -snk3 -k4 \ 
    |DistanceEst --dot -j150 -k61 -161 -s200 -n10 -o lmp1-6.dist.dot lmp1-6.hist
Building the suffix array...
Building the Burrows-Wheeler transform...
Building the character occurrence table...
Mateless  0
Unaligned  127754  6.8%
Singleton  828893  44.1%
FR        3191  0.17%
RF        668696  35.6%
FF        20536  1.09%
Different  230815  12.3%
Total     1879885
```
Checking content inclusion using KAT

- Just compare the frequency of kmers in the assembly to the reads spectrum.
Different assemblies and pre-processing

**Figure 5** Length and coverage-based metrics for chromosome 3AS assembled using kmer sizes between 41 and 99 as indicated on the x axis: — untrimmed data, —— trimmed reads (to quality score of Q30)

<table>
<thead>
<tr>
<th>Assembly</th>
<th>Number Of Hits To ESTs</th>
</tr>
</thead>
<tbody>
<tr>
<td>3AS untrimmed</td>
<td>1539</td>
</tr>
<tr>
<td>3AS trimmed</td>
<td>1132</td>
</tr>
</tbody>
</table>
Look for expected content

• Just BLAST the output to check what it looks like.

• But you can also try finding genes/markers/ETS:

Figure 7: short arm EST recovery

Figure 8: long arm EST recovery
Look for contaminants

- Contaminants including symbionts, mitochondria, chloroplast.
Look for contaminants

• Contaminants including symbionts, mitochondria, chloroplast.
What is the output of your first-pass assembly?

• Knowledge about the used datasets:
  • Are they clean? Can they be better?
  • How each one performs.
  • How they all interact.

• Knowledge about the target:
  • How it’s “complications” affect assembly. (also the datasets?)
  • Repeat structure?

• Refined choice of K.

• Baseline metrics (to be improved).
Questions?