Assembly using newbler

Learning points:

- How to start a newbler assembly
- Understanding newbler output
- Basic understanding of parameters

Where is what

All 454 files are located in the folder `/data/assembly/454`.

A short description of the files we will use today:

`HUFU8GI02_30X.sff`	GS FLX+ shotgun reads	172 000 reads, 139 Mbp
`HUFU8GI04.sff`	GS FLX+ shotgun reads	244 000 reads, 195 Mbp
`FJ4Q8OU01_18X.sff`	GS FLX Titanium 8kb MP reads	518 000 reads, 163 Mbp

`HUFU8GI02_30X` contains GS FLX+ reads (peak at 800 bases). `FJ4Q8OU01_18X` contains GS FLX Titanium reads (peak length around 500 bases) from a 8 kb mate pair library.

Quality Control of 454 reads using PRINSEQ

The analysis was done before the course started. If you want to have a look at the demo data, go to the prinseq website:

http://edwards.sdsu.edu/cgi-bin/prinseq/prinseq.cgi?access=1

Choose 'Access Data' and enter the following data ID: **31333439343234323539**

Shotgun reads only assembly

We will perform an assembly using newbler and 30x coverage of the the 454 GS FLX+ shotgun reads, first *without* the 8kb Mate Pair reads. This file will be used for this first assembly:

```
/data/assembly/454/HUFU8GI02_30X.sff
```

To run an assembly, use the `runAssembly` command described below. Please note:

- `runAssembly` is a program that sets up a newbler assembly and starts newbler
- the command should be written on one line
- replace **`projectname`** (see below) with something of your own choosing; this will be the name of the folder the assembly will appear in ('`-o`' stands for output folder)
- the rest of the command is the path to one or more files to be used as input; in this case, we use one file
- although one can use multiple cpus (through the '`-cpu`' flag) to speed up the assembly, do not do this today (you only have two available).
- *do not close the terminal window* until the assembly is done, as you will cancel it
- the whole process will, for this dataset, take around 15 minutes

First, create a new folder

```
/home/yourusername/assembly/newbler
```

Use this command

```
runAssembly -o projectname
    /data/assembly/454/HUFU8GI02_30X.sff
```

The output that newbler sends to the screen during assembly is explained in my blog at http://contig.wordpress.com/2010/02/09/how-newbler-works/
It can be also found in the file called 454NewblerProgress.txt after the assembly is done.

While the assembly is running, follow the output to the screen and answer the following questions:

Question	Your answer
How many bases were there in total available for the assembly?	
How much coverage is that (taking a total genome size of 4.6 Mbp)?	

Assembly metrics

After the assembly is done, have a look at the 454NewblerMetrics.txt file. For more details of what the different parts of this file mean, check http://contig.wordpress.com/2010/03/11/newbler-output-i-the-454newblermetrics-txt-file.

When newbler reports the number of reads (or bases) for a file, it writes this as:

numberOfReads = xxx[raw], yyy[after filtering];

Question	Your answer
How many bases did newbler remove during filtration for the input file?	
What was the N50 length of the (large) contigs? The length of the longest contig?	
What did newbler determine to be the coverage (peak depth)? And the genome size? Look in the section labeled 'Alignment depths' further down in the file.	
What do you think 'Q40PlusBases" means?	

To have a look at the sequences newbler produced. ‘LargeContigs’ are those that are 500 bp and more, ‘AllContigs’ those from 100 bp (including all the ‘Large’ ones).

To have a quick look at the lengths of the different contigs, write:

```
grep 'contig' 454LargeContigs.fna |less
```

`grep` is a really useful command. It will show the lines from a file that match a certain pattern. The pattern can, but does not have to be, in between ‘quotation marks’ (single or double). CAREFUL, if you want to select lines beginning with the ‘>’ symbol, you HAVE to use quotation marks:

```
grep '>' 454LargeContigs.fna
```

You will see that the sequences are no longer shown, just the fasta headers (sequence descriptors). Note that the largest contigs appear first.

Adding mate pairs

The term ‘mate pair’ is what we refer to when we mean long-insert libraries, as opposed to paired end reads, which are from short-insert libraries (and are genedated with a more simple protocol). Unfortunately, 454 choose the term paired end for their mate pair reads. Here we will stick with mate pair for consistency.

We will perform an assembly using the same 454 GS FLX+ shotgun reads, supplemented with 18x coverage of the 8kb mate pair reads. This will generate scaffolds in addition to the contigs. Newbler doesn’t need to be told the that a file contains mate pair reads, as it determines this from the linker sequence. If at least a certain fraction of the reads in a file contain the mate pair linker, the file is deemed coming from a mate pair library. There will always be a fraction of reads without linker, due to the library preparation process.

Perform the following:

```
runAssembly -o projectname \
     /data/assembly/454/HUFU8GI02_30X.sff \
     /data/assembly/454/FJ4Q8OU01_18X.sff
```

How many reads did newbler find in the mate pair file? Try to explain the number *after trimming*	
What was the percentage of reads in the mate pair file that contained the linker (was a mate pair read)?	
What was the N50 length of the (large) contigs for this assembly? The length of the longest contig?	
What was the N50 length of the scaffold(s)? The length of the longest scaffold?	
How many gaps are there in the longest scaffold?	
What did newbler determine to be the coverage (‘peak depth)? And the genome size?	
What did newbler determine to be the insert size for the mate pair library?	

Use the assemblathon stats script on the scaffolds (and contigs) of this assembly

```
assemblathon_stats.pl -s 4.6 454Scaffolds.fna >
metrics.txt
```

Assembly viewing

The 454Contigs.ace file represents the assembly with contigs, and all read alignments (but not the scaffolding information), which is why it usually is quite large. We can view the alignments using the program Tablet. More information on the ace file can be found at http://bozeman.mbt.washington.edu/consed/distributions/README.19.0.txt.
For this second assembly, open the 454Contigs.ace file in tablet.

- can you spot a homopolymer difference between one or more reads and the consensus? Look for both undercalls (one base too few) and overcalls (one too many)
- play around with the different elements of this program (slider bars etc), also have a look at the largest contig

Mauve assembly metrics

Run mauve assembly metrics on the scaffolds. **Remember to copy sequence files to this folder:**

```
~/assembly/mauve
```

Take a look at the final alignment. How well do the contigs (or scaffolds) align to the reference genome?