Finishing Scripts V1.0

1. *In-silico* Finishing Using Contig Graph
2. Filling Gaps with Finishing Reads
3. Suggesting Contigs for Large Gaps
4. Picking Primers
5. Scaffold to Sequence

1. *In-silico* Finishing Using Contig Graph

A scaffold obtained from *de novo* assembly or through other methods (for e.g. optical map based scaffolding using SOMA, or using mate-pairs and AMOS-Hybrid) can be further refined using information from the graph structure implicit in the output of most *de novo* assembly algorithms. In our experience we found this to be quite useful for closing gaps in the scaffold *in silico*. The first step, of course is to extract this information from the assembly - the script get_graph.pl can be used to extract a contig graph from the **acefile** output of a Newbler assembly (newer versions of Newbler automatically output this as 454ContigGraph.txt):

```
./get_graph.pl ../test/454Contigs.ace

Step 1: Parsing the Acefile.
Step 2: Collecting the link information.
Step 3: Writing out the graph (Average Coverage = 22.48)
```

The contig graph that is extracted can be viewed using the corresponding **.dot** file or **.ps** file (**454Contigs.dot** and **454Contigs.ps** in this case). Additional information is also written to the **.graph** file which mimics the Newbler Contig Graph file (and log to the **.log** file). The script parse_links.pl can then be used to combine this information with a scaffold file:

```
./parse_links.pl

USAGE: parse_links.pl <scaff> <graph> <outdir>

scaff   = File containing scaffold information in .scaff format
graph   = File containing contig graph information
outdir  = Directory to which the files "result.scaff" and "log" will be written
```

The resulting scaffold is written to the file **result.scaff** with an accompanying **log** file. Note that the script can be run in two modes, a default interactive mode that provides evidence from the contig graph for manual verification:

```
./parse_links.pl ../test/seq.scaff ../test/454Contigs.graph ../test
Step 1: Reading the graph in.
Step 2: Comparing the graph to the scaffold.
Gap between contig00108 & contig00062 can be closed (Gap Size = 765).
Contig Graph Path: <30 links> contig00117,Size=768 <26 links>
Close Gap? (Y/N):
```

as well as a fast mode that assumes that all paths found are good:

```
./parse_links.pl ../test/seq.scaff ../test/454Contigs.graph ../test 1
Step 1: Reading the graph in.
Step 2: Comparing the graph to the scaffold.
Step 3: Identifying contigs placed in the scaffold that are likely to be
repeats.
Successfully closed 37 gaps!
```

The length of paths to look for is currently upper-bounded at 15.

2. Filling Gaps with Finishing Reads

The script fill_gaps.pl can take a scaffold and finishing sequences and contigs and integrate them with the existing scaffold:

```
./fill_gaps.pl

USAGE: fill_gaps.pl <scaff> <seqfile> <outdir>

scaff   = File containing scaffold information in .scaff format
seqfile = Multi-Fasta file containing contig sequences
outdir  = Directory to which the files "result.scaff" and "log" will be written
```

The script relies on the program nucmer to reliably match sequence ends and then confirms that they match the expected orientations in the scaffold. It can also test if adjacent contigs in the scaffold actually overlap. The script is typically run in an interactive mode:

```
./fill_gaps.pl ../test/result.scaff ../test/finishing.fa ../test

Step 1: Matching sequences to each other using Nucmer.
Step 2: Testing to see if these matches fill any gaps.

Looking at Gap: contig00263 EB contig00033 BE
Found sequence NS2457_33B_263B_R that seems to span gap! Nucmer evidence:
1       153     449     601     153     153     100.00  68580   601     0.22
25.46   contig00263     NS2457_33B_263B_R
1       141     141     1       141     141     100.00  601     11966   23.46
1.18    NS2457_33B_263B_R       contig00033
Good Enough? (Y/N): Y
```

Start1 End1 Start2 End2 Match-Length1 Match-Length2 Percent-Identity Length1 Length2 Fraction1 Fraction2 Sequence1 Sequence2

The length of paths to look for is currently upper-bounded at 15 and sequences that are already in the scaffold are excluded from the paths. The script currently looks only for the shortest path in terms of sequences used.

3. Suggesting Contigs for Large Gaps

The script *suggest_contigs.pl* can suggest sequences that could possibly fill large gaps in a scaffold based on the base composition of the ends of contigs (currently GC content of 1Kbp ends):

```
./suggest_contigs.pl

USAGE: suggest_contigs.pl <scaff> <seqfile>

scaff   = File containing scaffold information in .scaff format
seqfile = Multi-Fasta file containing contigs and finishing sequences

The program will suggest contigs that are likely to be in large gaps in the
scaffold based on the agreement of GC content of the ends.
```

Here is some sample output from the script:

```
./suggest_contigs.pl ../test/seq.scaff ../test/sequences.fa

Step 1: Computing GC content of 1Kbp ends.
Step 2: Testing to see if any of these fill the gaps.
Candidates: contig00270 contig00245 contig00198 contig00271 contig00043
contig00277 contig00254 contig00242 contig00258 contig00031 contig00025
contig00235 contig00243 contig00038 contig00233 contig00259 contig00029
contig00239

Considering gap: contig00241 BE contig00212 EB 6008
Fits: contig00277 EB 6897 (GC info: 0.529 0.518 0.475 0.47)
--------------------------------------------------------------------------------
Considering gap: contig00195 EB contig00263 EB 27406
Fits: contig00245 BE 23842 (GC info: 0.485 0.479 0.482 0.486)
Fits: contig00245 EB 23842 (GC info: 0.485 0.482 0.479 0.486)
Fits: contig00038 BE 16972 (GC info: 0.485 0.491 0.484 0.486)
```

Fits: Sequence-Name Orientation Size (GC info: Gap-Start-GC Sequence-Start-GC Sequence-End-GC Gap-End-GC

The information provided by the script can then be used to design primers and experiments to test these candidate sequences.

4. Picking Primers

A set of scripts to pick good primers will be added soon to this package.

5. Scaffold to Sequence

The script *scaff2fasta.pl* can be used to convert a **.scaff** file into a representation of the genomic sequences that they encode:

```
./scaff2fasta.pl

USAGE: scaff2fasta.pl <scaff> <seqfile> <outfile>

scaff   = File containing scaffold information in .scaff format
```

```
seqfile = Multi-Fasta file containing contig sequences
outfile = File to which sequences corresponding to the scaffolds will be output
```

The default is to create a sequence with the orf-terminator sequence "NNNNNTTAATTAATTAANNNNN" between contigs (when gap-size is > 0). An additional argument of 1 will create a sequence with as many "N"'s as the estimated gap-size.

Acknowledgments

We are very grateful to Thomas Jarvie at Roche for providing us with a test set of contigs from a Newbler assembly of 454 Sequences for E. coli K12.