SOMA V2.0

1. Installation
2. Input Format
3. Running SOMA
4. Output Format
5. Using Multple Maps
6. Special Cases (Scaffolds, Multiple Chromosomes)
7. Combining Optical Maps and Assembly Graph
8. On-SOMA (Online version of SOMA)
9. Intermediate Files and Debugging
10. Choosing Enzyme for Optical Map

1. Installation

After downloading the tarball, run the following commands:

```
gunzip soma-v2.tar.gz
tar -xvf soma-v2.tar
cd soma-v2; make install
```

2. Input Format

In the **bin** directory, running soma.pl without any parameters gives the following output:

```
./soma.pl

USAGE: soma.pl <fasta-file> <map-summary> <output-dir>

fasta-file      = Contig/Scaffold sequences in multi-fasta format
map-summary     = Summary of genome-wide optical map
output-dir      = Directory where the input, log and output information will be
written
```

The sequences are assumed to be in FASTA format, while the map can be in the older **summary.txt** format or the newer XML format (**summary.xml**, see the **test** directory for examples). The restriction enzyme information is obtained directly from the map file. Note that the output directory is created if it does not exist.

3. Running SOMA

Running soma.pl automatically parses the input files, runs the sequence matching program match, places the uniquely matched sequences (place_rest.pl), finds a globally optimal placement for the rest of the sequences (schedule) and combines the results to produce an output scaffold and summary of results (collect_result.pl). The pipeline can be tested with the following command:

```
./soma.pl ../test/sequences.fa ../test/AflII-summary.txt ../test/AflII

Step 1: Creating input files
Step 2: Matching sequences to the map
Step 3: Resolving unique placements
Step 4: Finding optimal placement for the rest
Step 5: Collecting results
```

4. Output Format

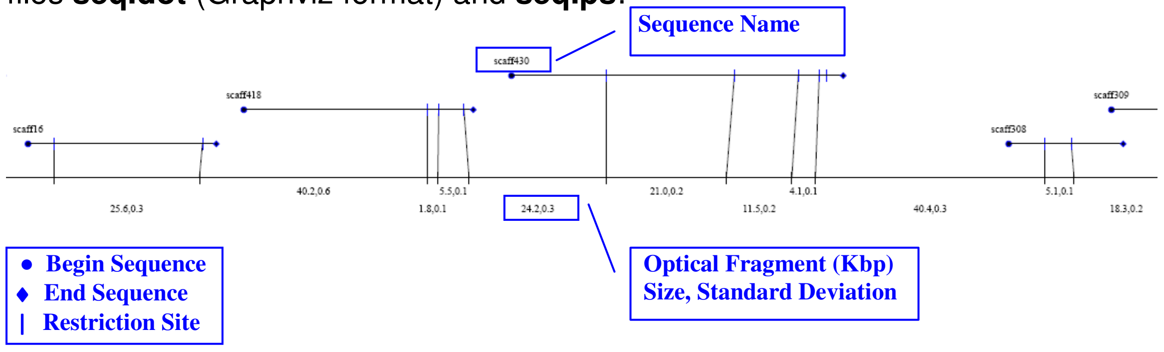
All the output from SOMA including log files and intermediate files are written to the output directory. The main output is the scaffold information in the file **seq.scaff**:

>Scaffold-name Number-of-contigs Ungapped-size Gapped-size

```
>all_seq 48 3907682 4404883
contig00049 EB 531074 1 337 MATCH
contig00026 BE 59798 808 207 SCHEDULE
contig00039 BE 78310 1969 732 MATCH
contig00015 BE 86740 3457 570 MATCH
contig00206 BE 87122 2785 341 FILTER
```

Contig-name Orientation Contig-size Gap-to-next-contig Standard-deviation-of-gap Placement-algorithm

The codes for the orientation field are **BE** and **EB** indicating that the corresponding sequence is placed begin-to-end or end-to-begin. A self-explanatory summary of results can be found in the file **seq.summary** and a log of the execution (with any error reports) can be found in the file **log**. A graphical view of the alignment of the sequences to the optical map can be found in the files **seq.dot** (Graphviz format) and **seq.ps**:



5. Using Multiple Maps

To obtain a global placement of sequences using multiple optical maps, first run *soma.pl* with each map separately. The script *get_joint_scaff.pl* (in the directory **multiple_maps**) can then be used to merge the resulting scaffolds into a consistent global scaffold:

```
./get_joint_scaff.pl
```

```
USAGE: get_joint_scaff.pl <dir1> <dir2> <output> <anchor>

dir1    = Directory containing SOMA output for first enzyme
dir2    = Directory containing SOMA output for second enzyme
output  = Directory to which the files "joint.scaff" and "log" will be written
anchor  = Name of contig to use as "anchor" for merging placements. Set this to
1 to try all anchors.
```

The current version of the script can only handle two maps but the algorithm extends to any number of maps and the next release of the script will handle these cases. The script *get_joint_scaff.pl* relies on an "anchor" sequence to merge two scaffolds and should therefore typically be first run in a "discovery mode" (anchor is set to 1):

```
./get_joint_scaff.pl ../test/AflII ../test/NheI ../test 1

Step 1: Reading in reliable contig placements
Step 2: Using an anchor to find a common orientation & order for placements
Placement using anchor: contig00049 (size=531074)
Contigs=45, Bases=3971207, Direction-Mismatch=0, Location-Mismatch=0,Overlaps=0
Placement using anchor: contig00073 (size=240016)
Contigs=44, Bases=3953069, Direction-Mismatch=0, Location-Mismatch=0,Overlaps=1
Placement using anchor: contig00261 (size=193158)
Contigs=45, Bases=3971207, Direction-Mismatch=0, Location-Mismatch=0,Overlaps=0
Placement using anchor: contig00248 (size=177954)
Contigs=45, Bases=3971207, Direction-Mismatch=0, Location-Mismatch=0,Overlaps=0
Placement using anchor: contig00116 (size=163449)
```

The output in the discovery mode can then be used to select the anchor sequence that provides the best scaffold in terms of <u>contigs</u> placed, <u>bases</u> in scaffold, minimizing <u>direction</u> and <u>location</u> mismatches and <u>overlap</u>ping sequence placements:

```
./get_joint_scaff.pl ../test/AflII ../test/NheI ../test contig00049

Step 1: Reading in reliable contig placements
Step 2: Using an anchor to find a common orientation & order for placements
Step 3: Reading in possibly ambiguous placements
Step 4: Resolving ambiguous placements by merging placement information from
the maps
```

The merged scaffold is in the file **joint.scaff** and more detailed information about the merge process can be found in the **log** file.

6. Special Cases (Scaffolds, Multiple Chromosomes)

The current version of SOMA can automatically parse an XML map file detailing maps for multiple chromosomes and simultaneously aligns sequences to these maps. The final output of SOMA is also split into multiple scaffolds accordingly. The script to merge results from multiple maps is however not adapted to these scaffolds. SOMA can be used to align scaffolds to optical maps with great success due to its robust matching procedure. The next version will have more support for this task as well.

7. Combining Optical Maps and Assembly Graph

The directory **assembly_graph** contains scripts that can be used to combine the scaffold obtained using SOMA with information from the graph structure implicit in the output of most *de novo* assembly algorithms. The script get_graph.pl can be used to first extract this information from the **acefile** output of a Newbler assembly:

```
./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 (Note: this doesn't have to be a scaffold obtained using SOMA):

```
./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/joint.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/joint.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 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.

8. On-SOMA (Online version of SOMA)

An online version of SOMA can be accessed at http://www.cbcb.umd.edu/soma. The website is based on SOMA v1.0 and will soon be updated to accommodate other tools available as part of SOMA v2.0.

9. Choosing Enzymes for Optical Map

A script to do this using simulations and the mapping procedure in SOMA will be available soon.

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.