
Welcome to ~~~~~Shorty version: 2.0~~~~~
---

1. Directory Structure

Directory Name	Contents
bak	Saves backup
bin*	Executables are stored here
conf*	Configuration files (details to follow)
dat*	Data (input) files (details to follow)
doc	Documentation
obj*	Object files stored here
out*	Output files, e.g. contigs, logs, etc.
scripts	Shell scripts to automate many tasks
src*	All source code and Makefile
tmp	Temporary files/source code
tools	Third party applications, e.g. MUMmer
utils	Some standalone utility programs

*Required for the core assembler
*Make sure that these directories are created.

2. Input Format
2.1 SOLiD

SOLiD data should be in "color space" format rather than the traditional "letter space" format (http://marketing.appliedbiosystems.com/images/Product_Microsites/Solid_Knowledge_MS/pdf/SOLiD_Dibase_Sequencing_and_Color_Space_Analysis.pdf, http://marketing.appliedbiosystems.com/images/Product_Microsites/Solid_Knowledge/Download/SOLiD_Data_Format_and_File_Definitions_Guide.pdf). Here is a snapshot of an input file:

```
....
....
>a450249-
GTTAGGAGAACGCGAATACTCGCGTC
>b448559-
CCCGTCCAACCGCGTGCTCCCTCCCT
>a1446565-
CTCAGACAATGTTGAGTATCTTGATG
>b1444916-
```

```
GACTATGTCGGTATCCGTCTACAGGT
>a1631829+
GAGATGTAATAAAATCATGCTCTACT
>b1633866+
TTTTTCACCTGACTTCAAGGTCAGAT
>a4429468-
CCTGTCTCACGATTGATATCACCGTA
>b4427384-
CTACAAATTGGAAGTCTATACACATC
>a424775+
TTCAGTCCAGCAACACTGTCCCTGCC
>b426531+
AAAGCAGTGAAGGCTGACAACAGTGC
....
....
```

Data should be sampled as mate paired data. For example, in the above sample input, consecutive reads starting with 'a' and 'b' are paired reads where 'a...' is the read from the 3' end and 'b...' is the read from the 5' end of the sequence.

2.2 Illumina/Solexa

Illumina data is available in fastq format as below.

```
@SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
GGGTGATGGCCGCTGCCGATGGCGTCAAATCCCACC
+SRR001666.1 071112_SLXA-EAS1_s_7:5:1:817:345 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIII9IG9IC
@SRR001666.2 071112_SLXA-EAS1_s_7:5:1:801:338 length=36
GTTCAGGGATACGACGTTTGTATTTTAAGAATCTGA
+SRR001666.2 071112_SLXA-EAS1_s_7:5:1:801:338 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIIII6IBI
@SRR001666.3 071112_SLXA-EAS1_s_7:5:1:839:309 length=36
GAATTTCAATACGGGTGACTTTAATCCCCCACGGGT
+SRR001666.3 071112_SLXA-EAS1_s_7:5:1:839:309 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII/
@SRR001666.4 071112_SLXA-EAS1_s_7:5:1:792:346 length=36
GATACCCAGAATACCAAACACCGTCTGCTGCATATC
+SRR001666.4 071112_SLXA-EAS1_s_7:5:1:792:346 length=36
IIIIIIIIIIIIIIIIIIIIIIIIIIIIII9I6II
@SRR001666.5 071112_SLXA-EAS1_s_7:5:1:765:373 length=36
GGTGTGTTGAGCTTGTCCAGCCTGGTGGATTCGATG
+SRR001666.5 071112_SLXA-EAS1_s_7:5:1:765:373 length=36
IIIIIIIIIIIIIIIIIIIIIIIIII;IIII8II
```

Before running through shorty assembler, the data should be converted to fasta format. Scripts are available to convert fastq format to fasta format. See the Additional Resources section below.

2.3 Helicos

Helicos data is automatically handled by shorty. A configuration file is created which automatically takes care of processing paired end reads

generated from Helicos. Please refer the Running section for specifying the configuration files.

2.4 Reference sequence

Reference Sequence input format:

Reference sequences are normally available in below format. The first line and all new line characters at the end of the string should be removed.

```
>chromosome Mycoplasma genitalium G-37 {Mycoplasma genitalium G-37}
TAAGTTATTATTTAGTTAATACTTTTAACAATATTATTAAGGTATTTAAAAAATACTATT
ATAGTATTTAACATAGTTAAATACCTTCCTTAATACTGTTAAATTATATTCAATCAATAC
ATATATAATATTATTAAAATACTTGATAAGTATTATTTAGATATTAGACAAATACTAATT
TTATATTGCTTTAATACTTAATAAATACTACTTATGTATTAAGTAAATATTACTGTAATA
CTAATAACAATATTATTACAATATGCTAGAATAATATTGCTAGTATCAATAATTACTAAT
ATAGTATTAGGAAAATACCATAATAATATTTCTACATAATACTAAGTTAATACTATGTGT
AGAATAATAAATAATCAGATTAAAAAAATTTTATTTATCTGAAACATATTTAATCAATTG
AACTGATTATTTTCAGCAGTAATAATTACATATGTACATAGTACATATGTAAAATATCAT
TAATTTCTGTTATATATAATAGTATCTATTTTAGAGAGTATTAATTATTACTATAATTAA
```

3. Running the Assembler

3.1 Downloading
Download shorty available at this website.
Unzip the shorty into a directory
Now download Mummer 3.20 from http://mummer.sourceforge.net/
Install Mummer by following the steps in the INSTALL file of Mummer directory. Download any required pre-requisities required for Mummer.

3.2 Building
The first thing should be to build the assembler with the following command from the root directory:
./build conf/conf-file bin/shorty-assembler
'conf-file' is a configuration file used to set various parameters. File 'src/global.*' also specifies some of the parameters. They will be combined soon in a future release.
'ab-strict' : Used for ABI Solid Data
'helicos-strict' : Used for Helicos data
'solexa-strict' : Used for Illumina/Solexa data

3.3 Running

A sample run can be like this:
bin/shorty-assembler -o12 -r5 -s3 -l100 -L580077 -t5 shorty-dat/ab/m_genitalium/seed.0 shorty-dat/ab/m_genitalium/pair100_2270_350.fasta .mg.0.100

A brief description of the parameters:
-o the amount of minimum overlap between reads to be considered as significant
-r number of time a read can be re-used
-s number of errors allowed while mapping a read onto a seed

-l minimum length of a contig produced in the output
-L length of the reference sequence (as estimation is OK)
-t output thickness desired

The last three arguments are: seed file (letter space), input read file, extension used for all output files.

Seed file : Seed file can be created from the reference sequence. It is used for finding mapping reads during the initial stages of assembly.
Input read file : Input read file should correspond to one of the input format mentioned above.
Extension : Can be any string useful for users to distinguish various runs.Download shorty available at this website.

3.4 Output
After the assembler is finished, the output files will be stored in out/ directory. The most notable output files are:

- contigs.ext : Output all the contigs in FASTA format (only those that pass the threshold length mentioned in the previous step). A sample may look like this:

....
....
>contig_asm_0
CAAGGACTGCTGTCGAAGACAAAAGTTTACTGGCCGAAGCGATGCTAGGCTACTCACCCTTGATCGCTGAGAT
GGAAAATGCATTCAAGTAGGAAACCAATAAGAGGACTGTGGGCGCTGACTACTACGTAGGTAATGTGTTCTAT
TGAATGACCAGTTGCTGCGCGTTGCAGGCCGCG
>contig_asm_1
ATATGTCTACGGGATCGATTTTTCTCGTGTAGAGGGGAGCGAGCAGATACGCCGGTTGATGCCCGCATAGAGA
TGGGTCTCGCCGGTATCAGCCAAAGCGCTGGACAAAGCCATTG
....
....

Note that "ext" is the extension you provided in the "Running Assembler" step.

- geom.ext : Detailed information about each contig. For example, a contig is described as follows:

>contig_asm_id
contig_id contig parent_seed_id generation_id left_or_right
[# of As # of Ts # of Gs # of Cs] .. repeat it for every base ..
An example:
>contig_asm_5861
5861
GGGAAACATGTCATCACACCTTTGATGACTGTGAGGGGACAACCCAAATCATACAATTATGAACATCTCTTAT
TCT
GAAATGTCCTAAAGATCACACTTTTTTCGTTCGATTTCAGTTAGTAGGTCGAACTCGCAAAGGTGAAGTCTTT
GTTGATCC
GAGATTCACAACGGCTCACTAA 5820 47 1
0 0 1 0 0 0 1 0 0 0 2 0 3 0 0 0 6 0 0 0 6 0 0 0 0 0 0 6 6 0 0 0 0 6 0 0 0
0 7 0 0
 7 0 0 0 0 0 7 7 0 0 0 0 7 0 0 0 0 0 7 8 0 0 0 0 0 0 8 8 0 0 0 0 0 0 9 0
0 0 9 0
9 0 0 0 9 0 0 0 9 0 0 0 0 9 0 9 0 0 0 0 10 0 0 0 0 9 0 9 0 0 0 0 0 0 8 0
7 0 0 0

0 6 0 0 6 0 0 0 0 6 0 8 0 0 0 0 0 8 0 0 0 8 0 0 0 8 0 0 0 8 0 8 0 0 0 0 0
0 8 8 0
 0 0 7 0 0 0 0 0 0 8 0 0 0 8 0 0 0 7 8 0 0 0 8 0 0 0 8 0 0 0 0 9 0 0 0 0
0 9 11 0
 0 0 0 10 0 0 12 0 0 0 0 0 0 12 12 0 0 0 12 0 0 0 0 10 0 0 0 11 0 0 11 0
0 0 0 9
0 0 0 0 9 0 8 0 0 0 8 0 0 0 0 0 0 8 9 0 0 0 0 9 0 0 0 0 0 11 0 11 0 0 0 0
0 10 0
10 0 0 0 10 0 0 9 0 0 0 0 11 0 0 0 11 0 0 0 0 0 11 0 11 0 0 0 0 9 0 9 0 0
0 7 0 0
 0 8 0 0 0 0 8 0 0 0 0 8 0 0 8 0 0 0 0 0 8 0 0 0 9 0 9 0 0 10 0 0 0 10 0
0 0 10 0
 0 0 0 0 10 0 9 0 0 0 0 11 0 0 0 0 0 10 11 0 0 0 0 0 0 11 11 0 0 0 0 0 0
12 0 13
0 0 0 13 0 0 0 13 0 0 0 12 0 0 0 13 0 0 0 13 0 0 0 0 0 13 0 0 13 0 0 12 0
0 0 12
0 0 0 0 0 12 0 0 12 0 13 0 0 0 0 12 0 0 0 12 0 0 0 11 0 0 0 0 0 11 12 0 0
0 0 0 1
3 0 0 13 0 0 0 13 0 0 13 0 0 0 0 0 13 0 0 13 0 0 13 0 0 0 0 0 13 0 0 0 12
0 0 10
0 0 0 0 0 10 0 0 11 0 11 0 0 0 11 0 0 0 0 0 0 11 0 12 0 0 0 0 0 12 0 0 12
0 0 0 0
 12 13 0 0 0 12 0 0 0 12 0 0 0 0 0 12 0 0 0 12 0 0 12 0 0 0 0 11 0 10 0 0
0 11 0
0 0 0 0 9 0 0 8 0 0 0 0 0 8 0 9 0 0 0 10 0 0 0 9 0 0 0 0 11 0 0 11 0 0 0
11 0 0 0
 0 10 0 10 0 0 0 0 10 0 0 0 0 0 10 0 0 0 9 0 0 9 0 9 0 0 0 0 0 9 0 8 0 0
0 0 7 0
0 0 7 0 0 0 0 0 7 7 0 0 0 0 0 0 7 7 0 0 0 7 0 0 0 0 0 0 6 0 0 6 0 0 0 6 0
0 0 0 5
 0 4 0 0 0 0 0 3 3 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 0

- geo.ext: Another file describing which reads actually made each contig along with their ids, offsets and orientation that were used to form the contig.

4 Running the Geographer

This is the step where you will enrich your contigs you got in the previous step and they will become significantly longer. Note that you can combine output from multiple runs of the assembler in this step.

4.1 Building the geographer

The first thing should be to build the geographer with the following command from the root directory:

```
./build conf/conf-file bin/shorty-geography
```

'conf-file' is a configuration file used to set various parameters. File 'src/global.*' also specifies some of the parameters. They will be combined soon in a future release.

4.2 Running the geographer

A sample run can be like this:

```
bin/shorty-geography -o12 -l100 dat/e-coli-clean.dat .tmp 3 out/geom.big2
out/geo.big2 out/geom.big3 out/geo.big3 out/geom.big4 out/geo.big4
```

A brief description of the parameters:
-o the amount of minimum overlap between reads to be considered as significant
-l minimum length of a contig produced in the output
The third parameter is the input read file (color space), fourth one is the extension used for all the output files. The rest of the parameters describe the input geo.* and geom.* files which are produced from the assembler. For example, in this sample run, we used output from three runs of the assembler.

4.3 Output
A similar set of output files (contigs*, geo* geom*) like the assembler will be produced in the out/ directory.

5. Running the Assmebler end to end

run-all script in scripts directory allows one to run shorty assembler, geography and collect contig stats by running through Mummer.

The arguments to run-all are explained below.
scripts/run-all 12 5 3 100 shorty-dat/ab/m_genitalium/org.seq 5 shorty-dat/ab/m_genitalium/seed.0 shorty-dat/ab/m_genitalium/pair100_2270_350.fasta .mg.0.100
1 overlap of reads
2 max read reuse
3 substition error allowed in seed mapping
4 min output contig length
5 reference sequence file
6 desired thickness of output
7 seed file
8 reads file
9 output extension
(Each one of the above corresponds to the argument specified after run-all)

Output:
contigs[ext] : Contigs generated by shorty-assembler
contigs-enriched[ext] : Contigs enriched using shorty-geography
contig-map[ext].eps : Contig length mapping to the reference coverage
contigs-len[ext].eps : Contig length mapping to the accuracy.
contig-map-enriched[ext].eps : Enriched Contig length mapping to the reference coverage(Produced by shorty-geography)
contigs-len-enriched[ext].eps : Contig length mapping to the accuracy.(Produced by shorty-geography)

Additional resources:

1. Conversion between one format to another format

   Command: all2y.pl informat outfile outfileformat < infile

2. Converting FASTQ(Solexa Illumina) Data to FASTA:

Command: fastq2fasta.pl sequence_file > outfile

3. Converting CSFASTA(ABI Solid) Data to FASTA

   Command: program csfasta shift
            (if shift is 1, the first base is omitted in the output)

4. Converting Fasta(Solexa Illumina) Data to FASTA

   Command: script.pl <infile> <outfile>