0005 - Sablefish Genome Project - Lab Notebook

Goal: Build a sablefish (*Anoplopoma fimbria*) genome
PIs:
Krista Nichols

2016/04/20

Koop Masurca Genome Run. Using Koop's original raw reads with the MaSuRCA Genome assembler (v3.1.3)

Ran on node18

/data/ggoetz/sablefish/koop_masurca

Config File:

```
DATA
PE= l1 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L1_SZAXPI013927-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L1_SZAXPI013927-169_2.fq
PE= l2 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L2_SZAXPI013928-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L2_SZAXPI013928-169_2.fq
PE= l3 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L3_SZAXPI013926-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L3_SZAXPI013926-169_2.fq
PE= l4 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L4_SZAXPI013929-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L4_SZAXPI013929-169_2.fq
#JUMP= sh 3600 200  /FULL_PATH/short_1.fastq  /FULL_PATH/short_2.fastq
#OTHER=/FULL_PATH/file.frg
END

PARAMETERS
#this is k-mer size for deBruijn graph values between 25 and 101 are supported,
auto will compute the optimal size based on the read data and GC content
GRAPH_KMER_SIZE = auto
#set this to 1 for Illumina-only assemblies and to 0 if you have 1x or more
long (Sanger, 454) reads, you can also set this to 0 for large data sets with
high jumping clone coverage, e.g. >50x
USE_LINKING_MATES = 0
```

```
#this parameter is useful if you have too many jumping library mates. Typically
set it to 60 for bacteria and 300 for the other organisms
LIMIT_JUMP_COVERAGE = 300
#these are the additional parameters to Celera Assembler.  do not worry about
performance, number or processors or batch sizes -- these are computed
automatically.
#set cgwErrorRate=0.25 for bacteria and 0.1<=cgwErrorRate<=0.15 for other
organisms.
CA_PARAMETERS = cgwErrorRate=0.15 ovlMemory=4GB
#minimum count k-mers used in error correction 1 means all k-mers are used.
one can increase to 2 if coverage >100
KMER_COUNT_THRESHOLD = 1
#auto-detected number of cpus to use
NUM_THREADS = 16
#this is mandatory jellyfish hash size -- a safe value is
estimated_genome_size*estimated_coverage
JF_SIZE = 200000000
#this specifies if we do (1) or do not (0) want to trim long runs of
homopolymers (e.g. GGGGGGGG) from 3' read ends, use it for high GC genomes
DO_HOMOPOLYMER_TRIM = 0
END
```

2016/05/09

Running Reapr (v1.0.18) on original Koop assembly. Made adjustments (-n 8) so that it would run faster.
Running on node18.

/data/ggoetz/sablefish/reapr/koop

Run Script:

```
#!/bin/bash

source /share/bioinformatics/biotools_setup.sh

reapr smaltmap -n 8 new_assembly.fa koop_R1.fq koop_R2.fq koop.bam \
    > reapr.step1.log 2>&1
```

2016/05/11

Running second step of Reapr (v1.0.18) on original Koop assembly

Run Script:

```
#!/bin/bash

source /share/bioinformatics/biotools_setup.sh

reapr pipeline \
    new_assembly.fa \
    koop.bam \
    pipeline_results \
    > reapr.step2.log 2>&1
```

2016/05/12

Reapr completed on original Koop assembly

Summary Report:

```
Stats for original assembly '00.assembly.fa':
Total length: 699326415
Number of sequences: 208506
Mean sequence length: 3353.99
Length of longest sequence: 66922
N50 = 5156, n = 39740
N60 = 4117, n = 54919
N70 = 3216, n = 74109
N80 = 2362, n = 99446
N90 = 1519, n = 136020
N100 = 500, n = 208506
Number of gaps: 12537
Total gap length: 43195
Error free bases: 71.74% (501706689 of 699326415 bases)

12106 errors:
FCD errors within a contig: 10455
FCD errors over a gap: 1550
Low fragment coverage within a contig: 101
Low fragment coverage over a gap: 0

231752 warnings:
Low score regions: 0
Links: 207433
Soft clip: 10511
```

```
Collapsed repeats: 1185
Low read coverage: 18
Low perfect coverage: 0
Wrong read orientation: 12605

Stats for broken assembly '04.break.broken_assembly.fa':
Total length: 699374210
Number of sequences: 210092
Mean sequence length: 3328.90
Length of longest sequence: 66922
N50 = 5144, n = 39852
N60 = 4107, n = 55072
N70 = 3208, n = 74313
N80 = 2355, n = 99714
N90 = 1515, n = 136397
N100 = 107, n = 210092
Number of gaps: 21465
Total gap length: 6428721
```

2016/05/16

Starting Reapr (v1.0.18) on the Koop Masurca Genome
Running on node18

/data/ggoetz/sablefish/reapr/koop_masurca

Run Script:

```
#!/bin/bash

source /share/bioinformatics/biotools_setup.sh

reapr smaltmap -n 8 genome.scf.fasta koop_R1.fq koop_R2.fq koop.bam \
    > reapr.step1.log 2>&1
```

2016/05/18

Running step 2 of Reapr (v1.0.18) on Koop Masurca Genome

Run Script:

```
#!/bin/bash
```

```
source /share/bioinformatics/biotools_setup.sh

reapr pipeline \
    genome.scf.fasta \
    koop.bam \
    pipeline_results \
    > reapr.step2.log 2>&1
```

Job finished before end of the day.

Summary Report:

```
Stats for original assembly '00.assembly.fa':
Total length: 660788920
Number of sequences: 130134
Mean sequence length: 5077.76
Length of longest sequence: 215234
N50 = 17426, n = 10084
N60 = 12935, n = 14485
N70 = 9038, n = 20579
N80 = 5430, n = 29934
N90 = 2027, n = 49294
N100 = 91, n = 130134
Number of gaps: 19376
Total gap length: 1243536
Error free bases: 70.87% (468316741 of 660788920 bases)

56659 errors:
FCD errors within a contig: 41778
FCD errors over a gap: 6279
Low fragment coverage within a contig: 2674
Low fragment coverage over a gap: 5928

221874 warnings:
Low score regions: 2
Links: 136046
Soft clip: 9595
Collapsed repeats: 1694
Low read coverage: 858
Low perfect coverage: 0
Wrong read orientation: 73679

Stats for broken assembly '04.break.broken_assembly.fa':
Total length: 661703338
Number of sequences: 144097
Mean sequence length: 4592.07
Length of longest sequence: 168709
```

```
N50 = 14166, n = 12658
N60 = 10603, n = 18053
N70 = 7520, n = 25448
N80 = 4561, n = 36660
N90 = 1780, n = 59433
N100 = 100, n = 144097
Number of gaps: 49950
Total gap length: 29094171
```

2016/05/26

Re-running the Koop + MaSuRCA assembly using slightly different configuration file (changed USE_LINKING_MATES to 1 and increased JF_SIZE). Using MaSuRCA (v 3.1.3).

config.txt:

```
# example configuration file

# DATA is specified as type {PE,JUMP,OTHER} and 5 fields:
# 1)two_letter_prefix 2)mean 3)stdev 4)fastq(.gz)_fwd_reads
# 5)fastq(.gz)_rev_reads. The PE reads are always assumed to be
# innies, i.e. --->.<---, and JUMP are assumed to be outties
# <---.--->. If there are any jump libraries that are innies, such as
# longjump, specify them as JUMP and specify NEGATIVE mean. Reverse reads
# are optional for PE libraries and mandatory for JUMP libraries. Any
# OTHER sequence data (454, Sanger, Ion torrent, etc) must be first
# converted into Celera Assembler compatible .frg files (see
# http://wgs-assembler.sourceforge.com)
DATA
PE= l1 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L1_SZAXPI013927-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L1_SZAXPI013927-169_2.fq
PE= l2 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L2_SZAXPI013928-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L2_SZAXPI013928-169_2.fq
PE= l3 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L3_SZAXPI013926-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L3_SZAXPI013926-169_2.fq
PE= l4 445 82
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L4_SZAXPI013929-169_1.fq
/data/ggoetz/sablefish/120903_I312_FCD1CNUACXX_L4_SZAXPI013929-169_2.fq
#JUMP= sh 3600 200  /FULL_PATH/short_1.fastq  /FULL_PATH/short_2.fastq
#OTHER=/FULL_PATH/file.frg
END

PARAMETERS
```

```
#this is k-mer size for deBruijn graph values between 25 and 101 are supported,
auto will compute the optimal size based on the read data and GC content
GRAPH_KMER_SIZE = auto
#set this to 1 for Illumina-only assemblies and to 0 if you have 1x or more
long (Sanger, 454) reads, you can also set this to 0 for large data sets with
high jumping clone coverage, e.g. >50x
USE_LINKING_MATES = 1
#this parameter is useful if you have too many jumping library mates. Typically
set it to 60 for bacteria and 300 for the other organisms
LIMIT_JUMP_COVERAGE = 300
#these are the additional parameters to Celera Assembler.  do not worry about
performance, number or processors or batch sizes -- these are computed
automatically.
#set cgwErrorRate=0.25 for bacteria and 0.1<=cgwErrorRate<=0.15 for other
organisms.
CA_PARAMETERS = cgwErrorRate=0.15 ovlMemory=4GB
#minimum count k-mers used in error correction 1 means all k-mers are used.
one can increase to 2 if coverage >100
KMER_COUNT_THRESHOLD = 1
#auto-detected number of cpus to use
NUM_THREADS = 16
#this is mandatory jellyfish hash size -- a safe value is
estimated_genome_size*estimated_coverage
#JF_SIZE = 200000000
JF_SIZE = 6600000000
#this specifies if we do (1) or do not (0) want to trim long runs of
homopolymers (e.g. GGGGGGGG) from 3' read ends, use it for high GC genomes
DO_HOMOPOLYMER_TRIM = 0
END
```

Also running FastQC on the various reads.

Koop L1 R1
Koop L1 R2
Koop L2 R1
Koop L2 R2
Koop L3 R1
Koop L3 R2
Koop L4 R1
Koop L4 R2

Jump 3KB R1
Jump 3KB R2

2016/06/02

Second attempt at Koop + MaSuRCA genome assembly failed. Not sure exactly why.

assemble.log:

```
[Wed Jun  1 15:45:59 PDT 2016] CA failed, check output under CA/ and runCA3.out
```

runCA3.out:

```
Analyzing edge eid:44136256 A:42140907 B:42167131 wgt:1       &R  ori:O qua:  1
trstd:0 con:0 dst:150 std:13.3227  (209161187,209161186)
Analyzing edge eid:46988765 A:42140907 B:42167131 wgt:6       &E  ori:O qua:  1
trstd:0 con:0 dst:231 std:5.18522
* No overlap found between 981822 and 42140907.  Fail.
ContigContainment failed.
cgw: LeastSquaresGaps_CGW.C:1419: RecomputeOffsetsStatus
RecomputeOffsetsInScaffold(ScaffoldGraphT*, CIScaffoldT*, int, int, int):
Assertion `0' failed.

----------------------------------------
Failure message:

scaffolder failed
```

According to CA Wiki, we could try restarting this step either via runCA or the cgw command.

Used the MaSuRCA method for restarting assembly, deleted the 7-0-CGW folder then rebuilt the assemble.sh command (masurca config.txt). Restarted the assembly using the new assemble.sh command.

2016/06/03

Assembly failed again at the same step. Attempting to restart this time using the cgw command directly.

cgw command, pulled from runlog command file

```
/share/bioinformatics/MaSuRCA-3.1.3-CentOS6/CA/Linux-amd64/bin/cgw \
  -j 1 \
  -k 5 \
  -r 5 \
  -s 2 \
  -z \
  -P 2 \
  -B 2078489 \
  -m 100 \
  -g /data/ggoetz/sablefish/koop_masurca_run2/CA/genome.gkpStore \
  -t /data/ggoetz/sablefish/koop_masurca_run2/CA/genome.tigStore \
  -o /data/ggoetz/sablefish/koop_masurca_run2/CA/7-0-CGW/genome \
  > cgw.out 2>&1
```

2016/06/06

The cgw appears to have finished. Trying to restart the assembler using the MaSuRCA method (rebuilding assemble.sh then starting it).

2016/06/07

MaSuRCA finished, created 161751 contigs, 137149 scaffolds, and 18026 superreads.

Quick summary calculated from lengths of sequences for both scaffolds and contigs.

	Min	1stQt	Median	Mean	3rdQt	Max
Contigs	64	587	1215	4117	4268	146500
Scaffolds	94	598	1185	4865	4390	218000

If we remove any sequence with length less then 200bp we get the following numbers. Total counts, 159682 contigs and 136036 scaffolds.

	Min	1stQt	Median	Mean	3rdQt	Max
Contigs	200	602	1240	4169	4351	146500
Scaffolds	200	607	1200	4903	4450	218000

Started reapr on filtered set of contigs, only on contigs with lengths greater than or equal to 200bp.

reapr, step1 script:

```
#!/bin/bash

source /share/bioinformatics/biotools_setup.sh

reapr \
    smaltmap \
    -n 8 \
    genome.scf.fasta.filtered.fa \
    koop_R1.fq \
    koop_R2.fq \
    koop.bam \
    > reapr.step1.log 2>&1
```

2016/06/09

reapr step1 finished this morning at about 6am after running for almost 48 hours. Starting step2.

reapr step2 script:

```
#!/bin/bash

source /share/bioinformatics/biotools_setup.sh

reapr pipeline \
    genome.scf.fasta.filtered.fa \
    koop.bam \
    pipeline_results \
    > reapr.step2.log 2>&1
```

2016/06/10

reapr step2 finished yesterday after running about 11 hours. Here is a quick results summary.

```
Stats for original assembly '00.assembly.fa':
Total length: 667040889
Number of sequences: 136036
Mean sequence length: 4903.41
Length of longest sequence: 217961
N50 = 16926, n = 10331
N60 = 12460, n = 14933
N70 = 8656, n = 21336
N80 = 5163, n = 31211
N90 = 1890, n = 52110
N100 = 200, n = 136036
Number of gaps: 24602
Total gap length: 1238909
```

```
Error free bases: 71.40% (476258831 of 667040889 bases)

54968 errors:
FCD errors within a contig: 37182
FCD errors over a gap: 6213
Low fragment coverage within a contig: 2832
Low fragment coverage over a gap: 8741

233860 warnings:
Low score regions: 2
Links: 151588
Soft clip: 7957
Collapsed repeats: 1803
Low read coverage: 1001
Low perfect coverage: 0
Wrong read orientation: 71509

Stats for broken assembly '04.break.broken_assembly.fa':
Total length: 667361443
Number of sequences: 152324
Mean sequence length: 4381.20
Length of longest sequence: 168705
N50 = 13171, n = 13723
N60 = 9880, n = 19573
N70 = 6956, n = 27598
N80 = 4209, n = 39827
N90 = 1671, n = 64642
N100 = 100, n = 152324
Number of gaps: 48128
Total gap length: 25145085
```