labibi Documentation

Release 0.1

C. Titus Brown

December 27, 2013

Contents

Notes and materials from Davis Masterclass on assembly and the associated PacBio sequence assembly workshop.

Authors: Holly Bik, C. Titus Brown, Nick Loman, Lex Nederbragt, and Jared Simpson.

Please see *#UCD – Assemble! Outputs* for our summary outputs!

CHAPTER 1

Reference list

General references

NGS wikibook: http://en.wikibooks.org/wiki/Next_Generation_Sequencing_(NGS) NGS field guide http://www.molecularecologist.com/next-gen-fieldguide-2013

Looking at 16S and whole transcriptome:

Radax R, Rattei T, Lanzen A, Bayer C, Rapp HT, Urich T, Schleper C (2012) Metatranscriptomics of the marine sponge Geodia barretti: tackling phylogeny and function of its microbial community. Environmental Microbiology 14(5): 1308-1324. doi: 10.1111/j.1462-2920.2012.02714.x.

Assembling symbiont communities with digital normalization:

K Goffredi S, Yi H, Zhang Q, Klann JE, Struve IA, Vrijenhoek RC, Brown CT. Genomic versatility and functional variation between two dominant heterotrophic symbionts of deep-sea Osedax worms. ISME J. 2013 Nov 14. doi: 10.1038/ismej.2013.201.

Assembling messy eukaryotic genomes with digital normalization:

Schwarz EM, Korhonen PK, Campbell BE, Young ND, Jex AR, Jabbar A, Hall RS, Mondal A, Howe AC, Pell J, Hofmann A, Boag PR, Zhu XQ, Gregory TR, Loukas A, Williams BA, Antoshechkin I, Brown CT, Sternberg PW, Gasser RB. The genome and developmental transcriptome of the strongylid nematode Haemonchus contortus. Genome Biol. 2013 Aug 28;14(8):R89.

Assembling soil with partitioning and digital normalization:

Assembling large, complex environmental genomes. Howe et al., arXiv (http://arxiv.org/abs/1212.2832).

Variant calling (including structural variants):

Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G. De novo assembly and genotyping of variants using colored de Bruijn graphs. Nat Genet. 2012 Jan 8;44(2):226-32. doi: 10.1038/ng.1028. PubMed PMID: 22231483; PubMed Central PMCID: PMC3272472.

Iqbal Z, Turner I, McVean G. High-throughput microbial population genomics using the Cortex variation assembler. Bioinformatics. 2013 Jan 15;29(2):275-6. doi: 10.1093/bioinformatics/bts673. Epub 2012 Nov 19. PubMed PMID: 23172865; PubMed Central PMCID: PMC3546798.

Leggett RM, Ramirez-Gonzalez RH, Verweij W, Kawashima CG, Iqbal Z, Jones JD, Caccamo M, Maclean D. Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de bruijn graphs. PLoS One. 2013;8(3):e60058. doi: 10.1371/journal.pone.0060058. Epub 2013 Mar 25. PubMed PMID: 23536903; PubMed Central PMCID: PMC3607606.

Using chromatin interactions data (Hi-C) for scaffolding assemblies:

> Chromosome-scale scaffolding of de novo genome assemblies based on chromatin interactions http://www.nature.com/nbt/journal/vaop/ncurrent/full/nbt.2727.html
>
> High-throughput genome scaffolding from in vivo DNA interaction frequency http://www.nature.com/nbt/journal/vaop/ncurrent/full/nbt.2768.html

Trimming/filtering references:

> On the optimal trimming of high-throughput mRNA sequence data. MacManes. http://biorxiv.org/content/early/2013/11/14/000422

Q/A Resources

> http://www.biostars.org/
>
> http://seqanswers.com/

Binning: Genomes from Metangenomes

> CONCOCT: Clustering cONtigs on COverage and ComposiTion http://arxiv.org/abs/1312.4038
>
> Multi-metagenome scripts http://madsalbertsen.github.io/multi-metagenome/
>
> Genome sequences of rare, uncultured bacteria obtained by differential coverage binning of multiple metagenomes http://www.nature.com/nbt/journal/v31/n6/full/nbt.2579.html Also see presentation slides here: http://www.slideshare.net/MadsAlbertsen/20131202-mads-albertsen-extracting-genomes-from-metagenomes
>
> The human gut and groundwater harbor non-photosynthetic bacteria belonging to a new candidate phylum sibling to Cyanobacteria http://elife.elifesciences.org/content/2/e01102
>
> Time series community genomics analysis reveals rapid shifts in bacterial species, strains, and phage during infant gut colonization http://genome.cshlp.org/content/23/1/111.full

CHAPTER 2

Tools

Coral, error correction that uses Illumina to correct 454:

http://www.cs.helsinki.fi/u/lmsalmel/coral/

Kraken, de novo adapter discovery:

http://biorxiv.org/content/early/2013/11/14/000422

SSPACE for scaffolding:

http://www.baseclear.com/landingpages/basetools-a-wide-range-of-bioinformatics-solutions/sspacev12/

BESST for scaffolding:

https://github.com/ksahlin/BESST

CHAPTER 3

Other links

How to pronounce “De Bruijn”:

http://thegenomefactory.blogspot.com.au/2013/08/how-to-pronounce-de-bruijn.html

http://www.biostars.org/p/7186/

Adaptor trim or die: Nextera libraries & trimming (by Nick Loman)

http://pathogenomics.bham.ac.uk/blog/2013/04/adaptor-trim-or-die-experiences-with-nextera-libraries/

The khmer protocols: assembling & annotating mRNAseq and metagenomes in the cloud

http://khmer-protocols.readthedocs.org

(Blog post)

How much sequencing is needed for ...? (Titus Brown)

http://ivory.idyll.org/blog/how-much-sequencing-is-needed.html

CHAPTER 4

Notes on using khmer to do stuff

4.1 Software

You'll need khmer and its dependencies installed:

```
cd /root
pip install -e git+https://github.com/ged-lab/khmer.git@kmer_error_profile#egg=khmer
```

The khmer src (including sandbox scripts and data) will then be placed in */root/src/khmer/*.

If installing elsewhere you may need to modify *install [...]* to *install –user [...]*

We'll be using the khmer command line scripts.

4.2 Data

Download two data sets:

```
cd /mnt
curl -O https://s3.amazonaws.com/public.ged.msu.edu/ecoli_ref-5m.fastq.gz
curl -O https://s3.amazonaws.com/public.ged.msu.edu/ecoliMG1655.fa.gz
```

(The sequencing data set is the first 5m reads from Chitsaz et al., 2011.)

4.3 Graphs

There are a bunch of graphs in an IPython Notebook; you can view the rendered notebook in your Web viewer *or* you can download it to work with on your Amazon machine (https:// + machine name, password 'beacon', and on the command line:

```
cd /usr/local/notebooks
curl -O https://raw.github.com/ngs-docs/2013-davis-assembly/master/2013-davis-assembly-khmer.ipynb
```

4.4 0. Assembling mRNAseq and metagenome data in the cloud

See Eel Pond mRNAseq assembly protocol and Kalamazoo metagenome assembly protocols.

Also see NGS course materials and other workshop materials.

Note 'Using screen' and 'Installing Dropbox'.

4.5 1. Estimating genomic richness/non-repetitive genome content

If you want to estimate the non-repetitive sequence content of your sample, you can use the three-pass assembly protocol (see Kalamazoo)

```
normalize-by-median.py -k 20 -C 20 -x 2e9 -N 4 --savehash C20.kh <FASTA/FASTQ/GZ/BZ2 files>
filter-abund.py C20.kh *.keep
normalize-by-median.py -k 20 -C 5 -x 2e9 -N 4 *.keep.abundfilt
```

Finally, get the complete stats on the remaining reads:

```
python /path/to/khmer/sandbox/readstats.py *.keep.abundfilt.keep
```

Divide the total number of bases remaining by 5 and you'll have an estimate of your total genome size. (Technically, the total number of nodes in your De Bruijn graph. :)

Note that the '-x' and '-N' parameters, above, should multiple to be under the total amount of memory on your computer. Scripts will complain if this is set too low.

4.5.1 Example

Run:

```
cd /mnt
mkdir richness
cd richness
normalize-by-median.py -k 20 -C 20 -x 5e8 -N 4 --savehash C20.kh ../ecoli_ref-5m.fastq.gz
filter-abund.py C20.kh *.keep
normalize-by-median.py -k 20 -C 5 -x 5e8 -N 4 *.keep.abundfilt

python /root/src/khmer/sandbox/readstats.py *.keep.abundfilt.keep
```

For this data, you should get:

```
33151414 bp / 385233 seqs; 86.1 average length -- total
```

which works out to:

```
385233 * 86.1 / 5 = 6633712.26
```

so the predicted genome size of E. coli is 6.6 Mbp. Not too far off... the true answer is 4.5 Mbp.

4.5.2 Looking at diginorm coverage

First, we need to install the BWA aligner:

```
cd /root
wget -O bwa-0.7.5.tar.bz2 http://sourceforge.net/projects/bio-bwa/files/bwa-0.7.5a.tar.bz2/download

tar xvfj bwa-0.7.5.tar.bz2
cd bwa-0.7.5a
make

cp bwa /usr/local/bin
```

We also need a new version of samtools:

```
cd /root
curl -O -L http://sourceforge.net/projects/samtools/files/samtools/0.1.19/samtools-0.1.19.tar.bz2
tar xvfj samtools-0.1.19.tar.bz2
cd samtools-0.1.19
make
cp samtools /usr/local/bin
cp bcftools/bcftools /usr/local/bin
cd misc/
cp *.pl maq2sam-long maq2sam-short md5fa md5sum-lite wgsim /usr/local/bin/
```

Next, map the raw reads:

```
cd /mnt/richness
gunzip ../ecoliMG1655.fa.gz
bwa index ../ecoliMG1655.fa

bwa aln ../ecoliMG1655.fa ../ecoli_ref-5m.fastq.gz > raw.sai
bwa samse ../ecoliMG1655.fa raw.sai ../ecoli_ref-5m.fastq.gz > raw.sam

samtools faidx ../ecoliMG1655.fa
samtools import ../ecoliMG1655.fa.fai raw.sam raw.bam
samtools sort raw.bam raw.sorted
samtools index raw.sorted.bam
```

and map the diginormed reads:

```
bwa aln ../ecoliMG1655.fa ecoli_ref-5m.fastq.gz.keep.abundfilt.keep > dn.sai
bwa samse ../ecoliMG1655.fa dn.sai ecoli_ref-5m.fastq.gz.keep.abundfilt.keep > dn.sam

samtools import ../ecoliMG1655.fa.fai dn.sam dn.bam
samtools sort dn.bam dn.sorted
samtools index dn.sorted.bam
```

Now, view with 'samtools tview':

```
samtools tview raw.sorted.bam ../ecoliMG1655.fa
```

or:

```
samtools tview dn.sorted.bam ../ecoliMG1655.fa
```

4.6 2. Generating a saturation curve for shotgun data

You can use normalize-by-median.py to assess information saturation in your shotgun data:

```
normalize-by-median.py -k 20 -C 5 -x 2e9 -N 4 -R report.txt <FASTA/FASTQ/GZ/BZ2 files>
```

'report.txt' column 1 will then contain the number of reads examined, and column 2 will contain the number of reads kept. When this levels off, you've saturated to a coverage of 5.

4.6.1 Example

Run:

```
cd /mnt
mkdir saturate
cd saturate
normalize-by-median.py -k 20 -C 5 -x 5e8 -N 4 -R ecoli_5m-report.txt ../ecoli_ref-5m.fastq.gz
```

The results are in '/mnt/saturate/ecoli_5m-report.txt'.

4.7 3. Generating a coverage plot/coverage spectrum without a reference

First, load all the k-mers in the data set into a counting file:

```
load-into-counting.py -k 20 -x 2e9 -N 4 counts.kh reads.fq
```

Then calculate the spectrum of read coverages:

```
python /path/to/khmer/sandbox/calc-median-distribution.py counts.kh reads.fq reads.hist
```

4.7.1 Example 1

First, let's do it for a small data set with interesting coverage:

```
cd /mnt
mkdir cover1
cd cover1
load-into-counting.py -k 20 -x 1e8 -N 4 counts.kh /root/src/khmer/data/stamps-reads.fa.gz
python /root/src/khmer/sandbox/calc-median-distribution.py counts.kh /root/src/khmer/data/stamps-reac
```

The results are in '/mnt/cover1/reads.hist'.

4.7.2 Example 2

Now try E. coli:

```
cd /mnt
mkdir cover2
cd cover2
load-into-counting.py -k 20 -x 1e9 -N 4 counts.kh ../ecoli_ref-5m.fastq.gz
python /root/src/khmer/sandbox/calc-median-distribution.py counts.kh ../ecoli_ref-5m.fastq.gz reads.h
```

The results are in '/mnt/cover2/reads.hist'.

4.8 4. Generating an error profile for individual shotgun reads

calc-error-profile.py reads.fq

For example,

```
cd /mnt
mkdir error
cd error

calc-error-profile.py ../ecoli_ref-5m.fastq.gz
```

The results will be in '/mnt/error/ecoli_ref-5m.fastq.gz.errhist'.

4.9 5. Assembly-filtered k-mer spectrum

Do:

```
cd /mnt
mkdir kmercov
cd kmercov

load-into-counting.py -k 20 -x 1e7 -N 4 counts.kh /root/src/khmer/data/stamps-reads.fa.gz
abundance-dist.py counts.kh /root/src/khmer/data/stamps-genomes.fa counts.out
```

The results will be in '/mnt/kmercov/counts.out'.

4.10 6. Partitioning

Do:

```
cd /mnt
mkdir part
cd part

do-partition.py -k 32 -x 1e7 -N 4 stamps /root/src/khmer/data/stamps-reads.fa.gz

extract-partitions.py -X 1000 stamps *.part
```

You will now have two files, stamps.group0000.fa and stamps.group0001.fa, that represent the two "species" in the data set. To examine them, let's graph their abundances:

```
abundance-dist-single.py -k 20 -x 1e7 -N 4 stamps.group0000.fa group0.hist
abundance-dist-single.py -k 20 -x 1e7 -N 4 stamps.group0001.fa group1.hist
```

The hist files will be in '/mnt/part/group0.hist' and '/mnt/part/group1.hist'.

4.11 7. A small demo set for khmer protocols/mRNAseq

To get started on the mRNAseq protocol with a small data set, do:

```
mkdir /data
cd /data
curl -O http://athyra.idyll.org/~t/mrnaseq-subset.tar
tar xvf mrnaseq-subset.tar
```

and then start at "Link your data into a working directory". (You'll need to install the software at the top, too.)

4.12 8. A small demo set for khmer protocols/metagenomics

To get started on the metagenomics protocol with a small data set, do:

```
mkdir /data
cd /data
curl -O http://athyra.idyll.org/~t/metag-subset.tar
tar xvf metag-subset.tar
```

and then start at "Link your data into a working directory". (You'll need to install the software at the top, too.)

CHAPTER 5

Metagenomics Practical

5.1 Metagenomics assemblies with Ray

Ray is a particularly interesting genome assembler due to several unusual features:

- It can scale to arbitrary numbers of processors and machines by distributing its assembly graph
- It has several functions specific to metagenome assembly 'Ray Meta'
- Ray's author, `@sebhtml` is incredibly responsive on Twitter :)
- Ray will happily mix input from several different sequencing techniques, e.g. Illumina and 454
- If run with the `write-kmers` option enabled, the resulting assembly graph may be viewed using the separate Ray Cloud Browser software

5.2 Installing Ray

5.2.1 Dependencies

```
sudo apt-get install build-essential
sudo apt-get install git
sudo apt-get install openmpi1.6-bin openmpi1.6-common libopenmpi1.6-dev
```

5.2.2 Installing Ray from source code

```
git clone https://github.com/sebhtml/ray
git clone https://github.com/sebhtml/RayPlatform

cd ray
HAVE_LIBZ=y MAXKMERLENGTH=64 make
```

You can add this to your PATH:

```
export PATH=$PATH:`pwd`
```

A simple command-line for multi-processor execution:

For paired-end reads:

```
mpirun -np 8 Ray -k 31 -p pair1.fastq.gz pair2.fastq.gz -o output_directory
```

For interleaved paired-end reads:

```
mpirun -np 8 Ray -k 31 -i pairs.fastq.gz -o output_directory
```

For single-end reads:

```
mpirun -np 8 Ray -k 31 -s reads.fastq.gz -o output_directory
```

If you want to run Ray Cloud Browser, you will want the `-write-kmers` option:

```
mpirun -np 8 Ray -write-kmers -k 31 -p pair1.fastq.gz pair2.fastq.gz -o output directory
```

If you run via a cluster, i.e. StarCluster, mpirun can be set to execute on multiple machines, e.g.:

```
mpirun -np 8 -H host1,host2,host3,host4 -k 31 -p pair1.fastq.gz pair2.fastq.gz -o x
```

For more command-line options, see:

https://github.com/sebhtml/ray/blob/master/MANUAL_PAGE.txt

5.2.3 Ray Cloud Browser

Here is a useful script to set up Ray Cloud Browser from a kmers.txt and Contigs.fasta file:

```
#!/bin/bash
tag=$1
kmerfile=$2
contigfile=$3
mapid=$4
sectionid=$5

RayCloudBrowser-client create-map $kmerfile $tag.dat
RayCloudBrowser-client add-map config.json "$tag" $tag.dat
RayCloudBrowser-client create-section $contigfile $tag-contigs.dat
RayCloudBrowser-client create-map-annotations-with-section $tag.dat $tag-contigs.dat $sectionid
RayCloudBrowser-client add-section config.json $mapid "$tag Contigs" $tag-contigs.dat
```

CHAPTER 6

Assembly QC

Enable HTTPS in the security group.

6.1 Installing software

BWA aligner

Do:

```
cd /root
wget -O bwa-0.7.5.tar.bz2 http://sourceforge.net/projects/bio-bwa/files/bwa-0.7.5a.tar.bz2/download

tar xvfj bwa-0.7.5.tar.bz2
cd bwa-0.7.5a
make

cp bwa /usr/local/bin
```

samtools

Do:

```
cd /root
curl -L http://sourceforge.net/projects/samtools/files/latest/download?source=files >samtools.tar.bz2
tar xjf samtools.tar.bz2
mv samtools-* samtools-latest
cd samtools-latest/
make
cp samtools bcftools/bcftools misc/* /usr/local/bin
```

FRCAlign

First:

```
apt-get -y install libbz2-dev libboost-dev libboost-iostreams-dev libboost-program-options-dev libboo
```

Do:

```
 cd /root
 git clone https://github.com/vezzi/FRC_align
```

```
 cd FRC_align
 cd src/samtools;
 make
 cd ..
 cd ..
./configure
 make install
```

REAPR

Do:

```
cd /root
curl -O ftp://ftp.sanger.ac.uk/pub4/resources/software/reapr/Reapr_1.0.16.tar.gz
tar xzf Reapr_1.0.16.tar.gz
cd Reapr_1.0.16

export PERL_MM_USE_DEFAULT=1
export PERL_EXTUTILS_AUTOINSTALL=--defaultdeps
export MAKEFLAGS='-j4'
perl -MCPAN -e 'install File::Spec::Link'

./install.sh
ln -s /root/Reapr_1.0.16/reapr /usr/local/bin/reapr
```

Python modules

Do:

```
pip install biopython
pip install pysam
```

Install R

Do:

```
apt-get install r-base
```

scaffoldgap2bed

do:

```
cd /usr/local/bin
curl -O https://raw.github.com/lexnederbragt/sequencetools/master/scaffoldgap2bed.py
chmod 770 scaffoldgap2bed.py
```

An IPython notebook

do:

```
cd /usr/local/notebooks
curl -O https://raw.github.com/lexnederbragt/INF-BIO9120_fall2013_de_novo_assembly/master/practicals/
```

6.2 Data files

Create a new volume based on snapshot *snap-78cf1764* and attach it to your running instance via the Amazon EC2 management interface. When it is attached, remember the partition (e.g. sdf is *xvdf*) and mount like:

```
mkdir /data2
mount /dev/xvdf /data2
```

6.3 Practicals handouts

Use the following practicals from https://github.com/lexnederbragt/INF-BIO9120_fall2013_de_novo_assembly/tree/master/practicals

Mapping reads to an assembly

Evaluating assemblies with FRCbam

Assembly improvement using REAPR

Note that you'll have to adjust file paths and we're skipping a few things (e.g. SNP calling)

6.4 Downloading data from the VM

use scp:

do:

```
scp -i /path/to/keyfile.pem root@ec-xx-xx-xx-.compute-1.amazonaws.com:/path/to/data ./
```

For IGV, download:

- velvet fasta file
- bam and bai files
- bed and gff files once you have them

—

To connect to the IPython Notebook interface, connect to

    https://ec2-machine-name

(ignore/accept the security warning) and use the password 'beacon'.

CHAPTER 7

Blobology

The steps for generating taxon-annotated 'blob' plots are as follows:

- Subsample your contigs to a reasonable number (to speed the process for very big assemblies)
- Perform rough taxonomic assignment (e.g. using BLASTN against NCBI's `nt` database)
- Map reads back to contigs (e.g. using Bowtie2 or BWA)
- Generate a file with GC, taxon and coverage data for plotting
- Plot graph in R using ggplot2

Full instructions for running Blobology and more advanced usage can be found at:

https://github.com/sujaikumar/assemblage

Also see the paper:

> Sujai Kumar, Martin Jones, Georgios Koutsovoulos, Michael Clarke and Mark Blaxter Blobology: exploring raw genome data for contaminants, symbionts, and parasites using taxon-annotated GC-coverage plots doi: 10.3389/fgene.2013.00237 http://www.frontiersin.org/Journal/10.3389/fgene.2013.00237/abstract

Although this tutorial uses the assemblage github repository, the repository for this paper is at https://github.com/blaxterlab/blobology The scripts at blaxterlab/blobology can be easier to use if you already have bam or coverage files.

https://github.com/sujaikumar/assemblage also has other scripts/workflows relevant to genome assembly, annotation, finding conserved non-coding elements, etc.

Also please see Sujai Kumar's recent presentation at Beatles and Bioinformatics:

http://www.youtube.com/watch?v=tSul_qDwvN4&t=3h10m50s

7.1 Installing dependencies

BLAST+ for taxonomic assignment

```
sudo apt-get install ncbi-blast+
```

Install R and required packages (requires R >=2.14).

```
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-base-core-dbg_3.0.2-1_amd64.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-base-core_3.0.2-1_amd64.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-base-dev_3.0.2-1_all.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-base-html_3.0.2-1_all.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-base_3.0.2-1_all.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-doc-html_3.0.2-1_all.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-doc-info_3.0.2-1_all.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-doc-pdf_3.0.2-1_all.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-mathlib_3.0.2-1_amd64.deb
wget http://athyra.ged.msu.edu/~mcrusoe/debs/oneiric/r-recommended_3.0.2-1_all.deb
dpkg -i r-base* r-base-core_* r-recommended* r-mathlib*
R
install.packages(c("codetools", "MASS", "ggplot2"))
```

After ths point, the following dependencies have already been installed on AMI public snapshot (in us-east): `snap-78cf1764` so if you mount this snapshot as a volume you can skip the following steps, but you do need to add the programs to your $PATH:

```
cd /mynewmount
source env.sh
```

We will use seqtk for subsampling contigs (can also be used for reads)

```
git clone https://github.com/lh3/seqtk.git
cd seqtk; make; cd ..
```

BWA for read mapping

```
sudo apt-get install bwa
```

Sujai Kumar's Assemblage for the scripts we need:

```
git clone https://github.com/sujaikumar/assemblage
```

The NCBI nt files, plus taxonomy information

```
wget ftp://ftp.ncbi.nih.gov/blast/db/nt.??.tar.gz
wget ftp://ftp.ncbi.nih.gov/pub/taxonomy/gi_taxid_nucl.dmp.gz
wget ftp://ftp.ncbi.nih.gov/pub/taxonomy/taxdump.tar.gz
```

7.2 Running blobology

Assume you have a contig file from your assembly called 'contigs.fasta'

First, subsample 10,000 contigs from the file:

```
seqtk sample contigs.fasta 10000 > contigs_10000.fasta
```

BLAST the reads

```
blastn -task megablast -query contigs_10000.fasta -db blast/nt -max_target_seqs 1 -outfmt 6 > contigs
```

Produce a taxonomy file

```
blast_taxonomy_report.pl \
-b contigs_10000.megablast.nt \
-nodes tax/nodes.dmp \
-names tax/names.dmp \
```

```
-gi_taxid_file tax/gi_taxid_nucl.dmp.gz \
-t genus=1 -t order=1 -t family=1 -t superfamily=1 -t kingdom=1 \
>contigs_10000.megablast.nt.taxon
```

Create a database of your contigs for BWA

```
bwa index contigs.fasta
```

Align the reads you used to make the assembly to the contigs database (for paired-end reads)

```
bwa mem contigs.fasta pair1.fasta.gz pair2.fasta.gz > samfile
```

If you have interleaved reads, or single-end, just omit the second FASTQ file

```
bwa mem contigs.fasta reads.fasta.gz > samfile
```

Create a file with coverage and GC information, this will be named according to your samfile name with .lencovgc.txt as the suffix.

Ensure you change the name of the files below before running the command.

```
sam_len_cov_gc_insert.pl -s samfile -f contigs.fasta

cat contigs_10000.megablast.nt.taxon samfile.lencovgc.txt |
perl -anF"\t" -e '
    chomp;
    /^(\S+).*\torder\t([^\t]+)/ and $o{$1} = $2 and next;
    if ($F[2] =~ /^\d+$/ ) { print "$_\t" . (exists $o{$F[1]} ? $o{$F[1]} : "Not annotated") . "\n" }
' >> lencovgc.taxon
```

If you want a different taxonomic level, e.g. species, download this script:

```
wget http://static.xbase.ac.uk/files/results/nick/make_blobology_file.py
python make_blobology_file.py contigs_10000.megablast.nt.taxon samfile.lencovgc.txt order > lencovgc
```

Just replace 'order' with another taxonomic level when you run make_blobology_file.py, e.g.: species, genus, family, superfamily, kingdom

Create the blobology plot

```
Rscript --vanilla ../bin/assemblage/blobology.R lencovgc.taxon 0.005 1 2
```

The final file is called lencovgc.taxon.png - you will need to download this to view it (e.g. with *sftp*, *scp* run on your local machine).

```
scp -i identity.pem root@server:/path/to/lencovgc.taxon.png .
```

CHAPTER 8

PacBio tutorials

8.1 Smrtportal AMI

Smrtportal/smrtanalysis is the software developed by Pacific Biosciences

- start a new amazon instance:
- select m3.2xlarge to get 8 cpus (instead of 4 with m1.xlarge) but NOT with the beacon AMI; instead, follow the instructions for setting up the smrtportal AMI from this pdf
- **NOTE** skip the filezilla step

8.2 Downloading data

We'll use data from a single smrtcell based on a size selected 20kb *E. coli* library

Do:

```
cd /opt/smrtanalysis/common/inputs_dropbox
wget http://files.pacb.com/datasets/secondary-analysis/ecoli-k12-P4C2-20KSS/ecoliK12.tar.gz
tar -vxzf ecoliK12.tar.gz
```

To import into smrtportal:

- in smrtportal, go to Home –> Import and Manage –> Import SMRT Cells
- click on common/inputs_dropbox
- click 'Scan' and OK

8.3 HGAP for assembly

- in smrtportal, go to Home –> Create New
- add your imported smrtcell by selecting it and clicking the triangle pointing to the right
- give your job a name
- for protocol, select RS_HGAP_Assembly.2 (*not* '.1')

- click 'Save'
- click 'Start'

Monitor the run, it will probably go overnight

Where is the data

Check /opt/smrtanalysis/common/jobs/016/016XXX, where 016XXX is the job ID from smrtportal. Results appear in the *data* folder

8.4 Base modification detection

Use:

- the same smrtcell data
- the RS_Modification_and_Motif_Analysis.1 protocol
- the 'e coli K12 MG1655' reference from the dropdown menu

8.5 Running smrtanalysis from the commandline

See this part of the Pacific Biosciences wiki: https://github.com/PacificBiosciences/SMRT-Analysis/wiki/SMRT-Pipe-Reference-Guide-v2.0#-using-the-command-line

- You'll need a settings xml file, which can only be obtained by setting up a smrtportal job with the correct protocol, and grabbing the settings.xml from /opt/smrtanalysis/common/jobs/016/016XXX, where 016XXX is the job ID from smrtportal
- You'll also need an input.fofn file ('file-of-filenames', which contains the full path to the bax.h5 (or bas.h5) file(s)
- use screen!

Do:

```
. /opt/smrtanalysis/etc/setup.sh
fofnToSmrtpipeInput.py input.fofn > input.xml
smrtpipe.py --params=settings.xml xml:input.xml
```

Or customise smrtpipe to use more cpus (if available):

```
smrtpipe.py -D NPROC=24 --params=settings.xml xml:input.xml
```

Results appear in the *data* folder

8.6 Running blasr to map reads

- upload the reference (e.g., the velvet assembly we used for QC/Validation)
- we will run blasr on a subset of the reads (using only one of the three bax.h5 files)
- samtools is included in the smrtportal distribution
- use screen!

Do:

```
. /opt/smrtanalysis/etc/setup.sh

blasr /opt/smrtanalysis/common/inputs_dropbox/ecoliK12/Analysis_Results/m130404_014004_sidney_c100506
/path/to/velvet_pe+mp.fa \
-minSubreadLength 1000 -bestn 1 -nproc 8 -sam -out pacbio_2_velvet_pe+mp_71.sam

samtools view -buS pacbio_2_velvet_pe+mp_71.sam | samtools sort - pacbio_2_velvet_pe+mp_71.sorted
samtools index pacbio_2_velvet_pe+mp_71.sorted
```

The bam and bai files can be added to the IGV browser

8.7 Running bridgemapper

Check out https://github.com/PacificBiosciences/Bioinformatics-Training/wiki/Bridgemapper

First, fix a 'bug' which makes one of the steps single-cpu instead of parallel.

Edit /opt/smrtanalysis/analysis/lib/python2.7/pbbridgemapper/bridgemapper.py: change lines **426 + 427** from:

```
stdout, stderr = runBlasr(affixesFastq, args.referenceFasta,
                          affixBlasrOutputPath)
```

To:

```
stdout, stderr = runBlasr(affixesFastq, args.referenceFasta,
                          affixBlasrOutputPath, nproc=args.nproc)
```

You can now run Bridgemapper through the smrtportal, which gives the optimal output for viewing in the PacBio genome browser SMRTview. However, this will take a long time. To do that, add the velvet assembly as reference through 'Home –> Import and Manage –> Manage Reference Sequences –> New'

Alternatively, run bridgemapper on a subset of the reads through the command line. Use screen!

Do:

```
. /opt/smrtanalysis/etc/setup.sh

pbbridgemapper --debug --nproc 8 /opt/smrtanalysis/common/inputs_dropbox/ecoliK12/Analysis_Results/m1
/path/to/velvet_pe+mp.fa bridgemapper_out
```

- –debug allows for seeing what bridgemapper is doing

To use SMRTview with the results, you'll need to make a smrtportal reference repo:

```
referenceUploader -c -p smrtpipe_references -n velvet_pe+mp -f /path/to/velvet_pe+mp.fa --saw='sawrit
```

- -c: create
- -p: folder for references
- -n: name of the reference
- -f: fasta file

Viewing with SMRTView

- download the split_reads.bridgemapper.gz and complete smrtpipe_references folder to your harddrive
- install SMRTView from https://github.com/PacificBiosciences/DevNet/wiki/SMRT-View
- choose 'File –> Open data from server' and select 'Files of type –> Reference Metadata'

- find the relevant reference.info.xml in the smrtpipe_references folder
- choose 'File –> add Tracks from server" and select 'Files of type –> As above and also gzipped files'
- add the split_reads.bridgemapper.gz file

Start browsing!

Genome binning with CONCOCT

CONCOCT is automated genome binning software that can use coverage information across multiple samples, composition and paired-end linkage information.

The manuscript describing the software is available on arXiv:

> CONCOCT: Clustering cONtigs on COverage and ComposiTion http://arxiv.org/abs/1312.4038

The Github repository is here:

> https://github.com/BinPro/CONCOCT/

The basic steps are as follows:

- Generate a co-assembly from all the reads in your dataset (or a subset if too large)
- Map reads back on a per-sample basis to the contigs to generate coverage information
- Run CONCOCT to produce clusters
- (optionally) Link clusters with paired-end reads
- Evaluate results, e.g. using taxonomic assignments of contigs and presence of conserved genes in clusters

To install CONCOCT, follow the instructions here:

https://github.com/BinPro/CONCOCT

A complete walk-through is available here:

https://github.com/BinPro/CONCOCT/blob/master/doc/complete_example.md

A repository containing the input and output files for a simple, worked example is available here:

https://github.com/BinPro/CONCOCT-test-data

A recent presentation by Chris Quince on CONCOCT is available to "watch again" from Beatles and Bioinformatics at this link:

http://www.youtube.com/watch?v=tSul_qDwvN4&t=1h15m32s

CHAPTER 10

#UCD – Assemble! Outputs

See *2013 UC Davis Assembly Masterclass - homepage* for more information on this workshop.

10.1 The Opinionated Guide to Sequencing and Assembly

Authors: Holly Bik, C. Titus Brown, Nick Loman, Lex Nederbragt, and Jared Simpson

Expiration date: 6/1/2014.

(Meta)genome	Goal	Dataset	Assembly strategy
Bacteria	(Near) completion	PacBio 100x	HGAP/Celera
Bacteria	Draft few contigs	Ilmn Nextera/TruSeq PE 2x250 c50x + Nextera MP 5kbp c50x	SPADES/MIRA
Bacteria	Draft 10s - 100(s) of contigs	Ilmn Nextera/TruSeq PE 2x250 c50x	SPADES/A5/MIRA
Small eukaryote up to 100 Mbp	contigs	Ilmn Nextera/TruSeq PE 2x250 c50x	SOAPdenovo, MIRA, SGA
Small eukaryote up to 100 Mbp	scaffolds	Ilmn Nextera/TruSeq PE 2x250 c50x + Nextera MP 3-10kbp c50x (each) Optional: PacBio	SOAPdenovo, MIRA, SGA (ALLPATHS_LG with right libraries) PBJelly and/or AHA
Eukaryote 100-500 Mbp	contigs	Ilmn Nextera/TruSeq PE 2x250 c50x	SOAPdenovo, SGA
Eukaryote 100-500 Mbp	scaffolds	Ilmn Nextera/TruSeq PE 2x250 MiSeq OR 2x150 HiSeq c50x; optional: multiple fr. lengths; Nextera MP 3-10kbp c50x (each); Optional: PacBio	SOAPdenovo, SGA, MaSuRCA, CA, Abyss (ALLPATHS_LG with right libraries) PBJelly and/or AHA
Eukaryotes over 500	contigs / non-repetitive components	Ilmn Nextera/TruSeq PE 2x250 MiSeq OR 2x150 HiSeq c50x	SOAPdenovo, SGA, diginorm + velvet
Eukaryotes over 500	scaffolds	as for 100-500 Mpb add more library types	SOAPdenovo, SGA, MaSuRCA, CA, Abyss (ALLPATHS_LG with right libraries) PBJelly and/or AHA
Metagenome low diversity (2-50 “species”)	Diversity estimates, gene mining	Ilmn Nextera/TruSeq PE 2x150 HiSeq (tip: long insert)	IDBA-UD, SPADES, MIRA
Metagenome low diversity (2-50 “species”)	Complete genomes	PacBio or Moleculo	IDBA-UD, diginorm + velvet/SGA, Ray
Metagenome medium diversity (50-500 “species”)	Diversity estimates, gene mining	Ilmn Nextera/TruSeq PE 2x150 HiSeq (tip: long insert)	IDBA-UD, diginorm + velvet/SGA, Ray
Metagenome high-diversity (e.g. soil, sediment)	Diversity estimates, gene mining	Ilmn Nextera/TruSeq PE 2x150 HiSeq (tip: long insert)	diginorm + velvet/SGA
Metatranscriptome	Expression, gene mining	Ilmn Nextera/TruSeq PE 2x150 HiSeq	diginorm + velvet/SGA, Ray?
Single-cell genome bacterial	Partial genome	Ilmn Nextera/TruSeq PE 2x250 c50x	SPADES
Single-cell genome eukaryote (protist)	Partial genome	Ilmn Nextera/TruSeq PE 2x250 c50x	SPADES?, diginorm + velvet/SGA/
RNA-seq	De novo transcriptome	Ilmn TruSeq/Nextera PE 2x100 HiSeq. 50 - 100 million reads per tissue, 300-500 bp fragment	Trinity

10.2 The 10+ Commandments of Assembly

Nick Loman (corr), et al.

1. Do I really need to assemble?
2. Good data is more important than choice of assembler.
3. Have a specific goal.
4. An assembly is a hypothesis to be tested.
5. Assembly programs are not haplotype aware.
6. More data may help.
7. If you haven't found contamination in your data you haven't looked hard enough.
8. A different assembler may help.
9. Make sure the assembly agrees with the reads that were used to put it together.
10. N50 is not a measure of quality.
11. But we don't have a measure of quality.
12. Avoid: Wheat, Fish, and Soil.
13. Trust contigs more than scaffolds more than gap filling.
14. The answer to your question may not be in your data.
15. A bad assembly that completes, is better than a good assembly that doesn't.

10.3 Our list of good practices in (meta)genome assembly

Lex Nederbragt (corr) et al.

- talk to the bioinformatician(s) before doing anything
- QC your reads with fastqc, preqc (khmer?)
- try different programs for assembly (not too many, but more than one)
- map the reads back to the assembly and use QC/Validation programs
- Use orthogonal data for QC/Validation * known genes * CEGMA/Phylosift * RNA-seq data * linkage map data/optical mapping data/fosmids or BAC data
- do blobology to figure out what you actually assembled (for any genome/metagenome)
- make reads, mapped reads and validation results available upon release of the genome (or before)
- make the genome assembly work reproducible

10.4 Thoughts on sequencing strategy

Lex Nederbragt (corr) et al.

When setting up a strategy for sequencing/assembly these are valuable to know:

- the goal of the project, e.g.

- gene-mining
- presence/absence of genes
- long scaffolds for synteny
- reference genome-standard (whatever that is)
- strain variation/haplotying
- genome size
- genome complexity, i.e. repeat content and level of heterozygosity
- number of chromosomes, extrachromosomal elements
- can a pure sample be obtained or will there be contaminating DNA?
- how much DNA can be obtained from a single individual/clonal line or will it be necessary to use DNA from a mix of different individuals (read: different genomes)?
- is there sequencing data available from the same individual or species?
- how close is the most closely related reference genome?

10.5 Questions (and some answers) about sequencing and assembly

10.5.1 General/unclassified

Question: How do I identify the correct sequencing platform and approach?

Answer: it's complicated! See *The Opinionated Guide to Sequencing and Assembly*.

Question: How do I assemble, call SNPs, and compare genetic elements?

Answers: (1) See *The Opinionated Guide to Sequencing and Assembly*; (2) We didn't cover that, but look at mapping-based approaches vs graph-based approaches to variant calling (Cortex, SGA); (3) use Mummer and Mauve to visualize.

Question: Should you do de novo assembly, mapping only, or guided assembly?

Answer: Nobody likes guided assemblies, and mapping is often not an option (because the reference is incomplete).

Question: Can you salvage a run without enough data?

Answer: Not really. At best, the preliminary data can be used to generate or refine a strategy for generating more data (e.g. see @Pop paper).

Question: How can I compare M/F expression levels in my critter?

Answer: Not covered.

Question: Should assembly workflows for complex metagenomes with a large fraction of eukaryotes use metagenome-specific tools or eukaryotic tools? Or, perhaps which steps in a workflow for complex metagenomes with a large fraction of eukaryotes should use euk tools and which steps should use metagenomic tools?

Answer: While the assembler choice may be different (see *The Opinionated Guide to Sequencing and Assembly*) almost everything else (read cleanup, QC, etc) can be pretty similar.

Question: What should cores support?

Answer: Obviously it depends, but your core may (or may not) have expertise in more complex sample prep. For example, mate pair library generation needs experienced cores.

10.5.2 Technology choice

Question: At what read length should we be moving towards OLC assemblers instead of De Bruijn?

There's no obvious answer; further investigation is needed. But, probably, 500bp - 1kb.

Question: What are the best tools and workflows for eukaryote assemblies? Do they differ according to predicted genome size (e.g. 80Mb nematodes vs. 1GB euk.)?

Answer: See *The Opinionated Guide to Sequencing and Assembly*

Question: How different are the workflows aimed at metagenomes (where the desired output is gene contigs) vs whole-genome assembly?

Answer: different assemblers may be useful, and the scaling considerations may be different. See *The Opinionated Guide to Sequencing and Assembly*. Note that contamination is endemic so it's probably the case that you have a metagenome anyway :).

Question: I would like to develop/obtain a scalable, cloud-based data anaylsis and visualization pipeline.

Answer: khmer-protocols is one attempt at doing this. Note Galaxy, DNANexus, etc. as well. Visualization is still lacking, although note IGV, Ray Cloud Browser.

10.5.3 Metagenomes and other mixtures

Question: How do we deal with "mixed assemblies" (e.g. an endosymbiont)? Is it better to filter reads or contigs? Does the presence of other sequences compromise assembly?

Answer: Blobology@@ will show you the size of the problem, but is not a solution. *If* you get good contig assembly (e.g. mapping shows that you've got most of the reads), it's better to filter contaminants out from the contigs because you'll have more signal; but you may have to filter the reads if you didn't get a good assembly. Different coverage levels from contamination may mislead single-genome assemblers and screw up your recall. If you have lots of contamination, SPADES, digital normalization, and/or various binning approaches (see CONCOCT) may be useful

Question: How can we assemble genomes from "messy" samples (non-clean tissue, data more like a metagenome)? How can we separate out potential symbionts and/or secondarily abundant microbes?

Answer: See previous question.

Question: How does metagenomic assembly collapse (or not) information from homologs between strains/species?

Answer: This is very situation/data specific and cannot easily be answered generally. Note that in general highly similar *protein* sequences will not have highly similar DNA sequences, and so will not be assembled together; we're really talking about DNA similarity (16s, repeats).

Question: What can one expect to find based on coverage depth?

Answer: Well, you need coverage. preqc (and maybe khmer) may help diagnose low coverage, highe error rates, etc.

Question: What kinds of contamination might you expect to see, and how do you deal with it?

Answer: See above! More work is needed.

Question: Is there a simple way to remove "kitomics", it fingerprints?

Answer: no. And do your negative controls thoroughly, folks.

Question: How can I remove endosymbiont (i.e. Wolbachia) reads from fly and butterfly data?

Answer: see above. no standard protocol.

Question: How can we get alpha-diversity, beta-diversity, key taxa at multiple levels, and functional genes for a variety of pathways?

Answer: phylosift and other such tools can help with all of this. HUMANn has good ways to extract functional genes, if you have the references.

10.5.4 Genome assembly

Question: How closely related are my insects? How can I tell?

Answer: Probably very ;). This is really a genotyping question... you can calculate this with k-mer distributions (talk to Jared), too, but in general this is complicated and not necessarily easy or straightforward.

Question: will inbreeding be useful or necessary?

Answer: Useful, but not necessary (just more $$ sequencing).

Question: How can we sequence individuals to chromosomes or at least obtain meaningful synteny maps?

Answer: See *The Opinionated Guide to Sequencing and Assembly*; also you may want to use optical maps, linkage maps, and long-read technology.

Question: How do we assemble Eukaryotes? Do we need to use multiple sequencing technologies? Can we use whole-genome amplification? Should we try to make new strain/haplotype-specific references? How about iteratively improving the reference genome with new sequences? How can we make population-specific "type" references? And what exactly is finishing, anyway? Also, do we need to worry about mapping bias?

Answer: For the first two, see *The Opinionated Guide to Sequencing and Assembly*. Try to avoid whole-genome amplification. There are no good tools available for making either a new reference genome or a "reference graph" (which is really what you want) but Titus may be working on this in the future. You can use several different tools for merging assemblies, improving references, etc – PBJelly with PacBio, GAA and other assembly merging tools, etc. But they don't necessarily work that well. As for finishing, nobody really does it. Finally, yes, you do need to worry about allelic mapping bias – see @@Wittkopp paper on RNAseq, etc.

10.5.5 Assembly outcomes, metrics & evaluation

Question: do we care about the difference between a 150 contig and a 100 contig assembly? What about 10 or 1000 contigs?

Answer: Well, it depends on your biological question. Generally, outcomes will be limited by your data and require careful planning; see *The 10+ Commandments of Assembly*.

Question: can I tell how large my genome is from the data?

Answer: preqc can tell you this, and khmer can estimate this (albeit badly).

Question: What do I need to do to improve genome quality?

Answer: that's more or less the topic of this entire workshop ;)

Question: Can we still use bad-quality data?

Answer: All data is to some extent bad quality data, so yes :). But it will limit the strength of your conclusions (see: 'science'). More usefully, data is randomly bad, you can work past it; if it's unknown systematic error, then more data is worse, and you're in trouble.

Question: What are good quality/completion metrics?

Answer: See "finishing", above. Also: reads mapped back; marker genes; cegma; REAPR; FRCbam; visual inspection; bridgemapper.

Question: What quality genome assembly can I expect?

Answer: preqc can help tell you this. Don't work on wheat, soil, or fish, though.

Question: What's a good way to evaluate completeness of your assembly?

Answer: (1) Mapping of reads. A high percentage is good, and tells you if there is room for improvement. "Treat your assembly as a hypothesis and see how well your reads match." (2) Bring in orthogonal evidence (see *Our list of good practices in (meta)genome assembly*).

10.5.6 Sample prep

Question: How really bad is Nextera (insertion bias and dual mode insert size) compared to TruSeq?

Answer: It can be quite a bit worse, or not. You just need to be aware that Nextera is messy in different ways (we need to strong arm Nick into writing a blog post!) Nextera is cheaper and easier, but has insertion bias and requires PCR. High GC organisms are worse with Nextera, and Illumina overall...

Question: how much is it worth striving for PCR-free library preps to avoid amplification bias? and/or going for single-molecule sequencing?

Answer: Depends on details: GC, your question, amount of DNA, etc. Right now PacBio (and Moleculo?) require a lot of input DNA, for example.

10.6 Predictions

It is hard to predict things, especially the future.

- Niels Bohr

Predictions:

- @ryneches: by Dec 31, 2015, Illumina reads will be longer than Sanger reads ever were. (> 1kb)
- @arturgreensward: by Dec 31, 2015, PacBio will be the only thing used to sequence bacterial genomes
- @pathogenomenick: by Dec 31, 2014, there will be two Nanopore platforms usefully available
- OH: by Dec 31, 2013, Mick will ask for an embargo on new assemblers
- @lexnederbragt: 2014-2015, diploid aware assemblers will emerge
- @pathogenomenick: by end of 2015, graph based variant calling will be default
- @ryneches: by 2016, computation will be considerably more expensive than sequencing
- @ctitusbrown: by 2016, the field will have solved the expensive computation problem and > 50% of non-biomedical computational analysis will be a commodity service (although it may not be a particularly good commodity service)

CHAPTER 11

Other references:

Starting up an EC2 instance