26 March 2012 – BIT815 exercises

I completed a Velvet assembly of simulated paired-end 50-nt Illumina reads (produced using dwgsim, as described in the 19 March Outline and Exercise notes available from the web page at http://www4.ncsu.edu/~rosswhet/BIT815/Spring2012/list.html) together with filtered, clipped reads from the Sequence Read Archive file (SRR077393.fastq.gz), available at the course S3 bucket (<http://ncsubit815.s3-website-us-east-1.amazonaws.com>). I put the contigs.fa file produced by Velvet on the S3 bucket site, along with SAM format alignment files of each of the read types (simulated Illumina short paired-end reads, and real 454 reads after removal of adapter sequences and trimming to the highest-quality subset of the sequences).

Here are some notes I took in the process of doing the assembly and mapping steps:

#SRR077393 reads are mate pair reads of 454 data, with an adapter sequence, TCGTATAACTTCGTATAATGTATGCTATACGAAGTTATTACG

#between the two ends of the mate-pair library.

#Download SRR077393.fastq.gz and DPC4571.fasta from s3-website...

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> SRR077393.fastq.gz

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> DPC4571.fasta

# Install a text editor and fastx-toolkit

# Uncompress the SRR077393.fastq.gz file and pipe the output to fastx\_clipper to remove adapter sequence:

sudo apt-get install fastx-toolkit gedit

gunzip -c SRR077393.fastq.gz | fastx\_clipper -a TCGTATAACTTCGTATAATGTATGCTATACGAAGTTATTACG -l 50 -v -Q33 -o clipped.fq

# Returns comment about blank line as last line of file – OK

# Run fastqc to check quality scores and k-mer distribution:

fastqc clipped.fq

# Shows aberrant k-mer distribution to nt 20, poor quality after ~320

# Use fastx\_trimmer to trim off first 20 nt and any sequence after 320

fastx\_trimmer -i clipped.fq -Q33 -f 21 -l 320 -o cliptrim.fq

# Download fastaNamesSizes.pl and plotLengthDistribution.R from

# ncsubit815.s3-website..., and make executable

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> fastaNamesSizes.pl

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> plotLengthDistribution.R

chmod u+x fastaNamesSizes.pl

chmod u+x plotLengthDistribution.R

# Convert the clipped, trimmed reads to fasta and find read lengths

fastq\_to\_fasta -i cliptrim.fq -Q33 -o cliptrim.fa

./fastaNamesSizes.pl cliptrim.fa > readstats.out

# 227157 sequences, total length 53233187.

# Minimum len: 30. Max: 300. Average: 234

# Plot read length distribution with R script:

Rscript plotLengthDistribution.R readstats.out

# Produces a histogram of read lengths readstats.out.png

#Download velvet from EBI, extract with tar -xzf velvet\_1.2.03.tgz

wget <http://www.ebi.ac.uk/~zerbino/velvet/velvet_1.2.03.tgz>

tar –xzf velvet\_1.2.03.tgz

#Change to velvet\_1.2.03 directory and compile with make command.

cd velvet\_1.2.03/

make

# Change back to ubuntu directory; create simulated paired-end reads of 50 bp length using:

cd ~

dwgsim -e 0.001-0.01 -E 0.001-0.01 -d 180 -s 10 -N 1000000 -1 50 -2 50 -r 0.00001 -R 0.00001 -X 0.01 -y 0.01 -n 0 -S 2 DPC4571.fasta PEreads

# Create an output directory; hash the 454 and short reads together with k=31:

mkdir output

velvet\_1.2.03/velveth output 31 -fasta -long cliptrim.fa -fastq -shortPaired PEreads.bwa.read1.fastq PEreads.bwa.read2.fastq

# Run velvetg to assemble into first-draft assembly, using automatic detection of expected coverage and a minimum contig length of 300 bp:

velvet\_1.2.03/velvetg output -ins\_length 280 -exp\_cov auto -min\_contig\_lgth 300

# Assembly ran for about 8.5 hours, and used a maximum of about 4 Gb RAM, and reported:

Final graph has 458548 nodes and n50 of 22, max 18421, total 6374057,  
using 210400/2227157 reads

# Mapping simulated reads to the reference: first index the reference genome using the command

bwa index -a is –p DPC DPC4571.fasta

# Now align each of the files of simulated reads to the genome separately:

bwa aln DPC PEreads.bwa.read1.fastq > PE1.sai

bwa aln DPC PEreads.bwa.read2.fastq > PE2.sai

# Then merge the two reads from the paired-end library into a single SAM output file:

bwa sampe DPC PE1.sai PE2.sai PEreads.bwa.read1.fastq PEreads.bwa.read2.fastq > PE.sam

# Mapping real 454 reads to the reference, using the ‘aln’ function of BWA:

bwa aln DPC cliptrim.fa > 454.sai

bwa samse DPC 454.sai cliptrim.fa > 454aln.sam

# Mapping real 454 reads to the reference, using the ‘bwasw’ function of BWA:

bwa bwasw DPC cliptrim.fa > 454sw.sam

All the output files (contigs.fa, the statistics file from the Velvet assembly, and the SAM format alignment files can be downloaded from the course S3 bucket using:

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> contigs.fa

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> stats.txt

wget <http://ncsubit815.s3-website-us-east-1.amazonaws.com> \*.sam

To examine these files, open the Firefox browser on your instance (in the NX Client window, look in the Applications menu at the upper left, then under sub-menu Internet) and download the Tablet genome browser from <http://bioinf.scri.ac.uk/tablet/>. Click on the link that reads Linux (64-bit) to download a shell script that will install Tablet on your local instance. This will create a folder called Tablet, with a red icon for the Tablet program. Double-click on this icon (in the graphical interface) to start the program. Go to the Open Assembly menu, browse to the PE.sam file for the Primary Assembly File, and the DPC4571.fasta sequence for the reference file. Compare the mapping of the 454 reads in the 454aln.sam and 454sw.sam files – is one clearly better than the other? What does the BWA page on Sourceforge.net say about the ‘aln’ and ‘bwasw’ commands?

Exercise

Following the examples given in the notes above, download the DPC4571.fasta reference sequence from the course S3 bucket, create an index using the BWA ‘index’ command, and then try mapping the contigs.fa file to the reference genome sequence using the BWA ‘bwasw’ command, which is better suited to long sequences than the BWA. If that produces a SAM output file, view it with Tablet to see how the alignments of contigs to the reference genome look. The summary statistics provided at the end of the Velvet run indicate that the assembly is about 3x larger than the reference genome – do you see evidence of this using Tablet?

Another program for aligning assembled sequences to a reference genome is Mummer. You can download the source code to your instance using the Firefox browser on the instance – point the browser to <http://mummer.sourceforge.net> and click the Distribution link, then (on the next page) click the Download MUMmer3.23.tar.gz link. This will take you to a page that has a blank form;

**YOU DO NOT HAVE TO FILL OUT THE FORM TO DOWNLOAD THE SOFTWARE.**

A timer at the top of that page will count down 5 seconds until your download starts – just wait, and save the tar.gz archive to the ubuntu folder on your instance when the dialog window opens and prompts for the location to save the file to.

Unpack the archive by executing

tar –xzf MUMmer3.23.tar.gz

in a terminal window, while you are in the ubuntu directory. Change to the MUMmer3.23 directory, view the INSTALL file, and follow the directions to compile and install the program.

The manual is available in html format online at <http://mummer.sourceforge.net/manual>. We will use the NUCMER program to align the DPC4571.fasta reference sequence to the contigs.fa file produced by Velvet. Execute the commands

nucmer –o DPC4571.fasta contigs.fa

mapview –f pdf out.coords

This should produce a PDF file containing a plot showing the alignments of contigs produced by Velvet to the reference sequence. How does this view of the alignment compare with that shown by Tablet?