Week 12, Lecture 24

István Albert
Bioinformatics Consulting Center
Penn State
Midterm project report: ReadSeq data format conversion

**Readseq: Read & reformat biosequences .. web service**

Readseq was written originally around 1989 a component of a sequence analysis program, but when I added a small, simple command-line interface, it took on a life of its own as a conversion program for bioinformatics. It’s main contribution to bioinformatics is it takes on the job of guessing what your input biosequence data format is, and converting it to what your software knows how to handle. See here readseq-help.html and Readseq-help.html for extended help.

What you need to use it is

1. readseq.jar the java archive of this program. *(don’t unpack this jar on your computer)*.
2. a Java runtime system on your computer, versions 1.1, 1.2, 1.3, 1.4, 1.5.
3. for OLD MacOS 9-, take also ReadseqApp, a small Mac application to run readseq without a command line (B> do un-binhex and un-stuffit this one).

This version includes a Graphic User Interface (GUI) for those who prefer not to learn the many command line options, or who’s workstation lacks a command-line interface (which probably includes most but the hardy bioinformaticians :). Double-click the readseq.jar file to launch or Run as

```
java -jar readseq.jar
```

If you do like command line interfaces, see the docs, the brief doc is

```
java -jar readseq.jar help
java -cp readseq.jar run --for command-line
```

**Note: not all conversions are valid!**

Usually you can only go from a complex format to a simpler one.
BWA vs Bowtie 1 vs Bowtie 2

- Testing on simulated reads BWA and realistic ones bowtie
- For both sensitivity (mapping rate) and specificity (correct alignments)
- We’ll keep a score along the way
- We all need to remember: both tools are triumphs of human ingenuity!
Short Read Archive

Command line toolkit to convert from the sra format to fastq, ssf, csfasta formats
About me

My name is Heng Li, a research scientist at the Broad Institute, working with David Reich and David Altshuler. My current interests include the analysis of new sequencing data, population genetics and phylogenetics. I am the principal developer of several projects including SAMtools, BWA, MAQ, TreeSoft and TreeFam with most of them started when I was a postdoctoral fellow of Richard Durbin at the Wellcome Trust Sanger Institute. I was also involved in several projects when I was in BGI (Beijing Genomics Institute, Chinese Academy of Science) between 2002 and 2006. These projects include rice finishing, silkworm sequencing, chicken variation study, heterozygote detection for capillary reads, gene finding and CAT alignment software.

At the time I am writing this page my wife, my little daughter and I are living in Boston, United States. Our motherland is China, forever.

Selected Publications
Clone the wgsim repository on github

get wgsim from GitHub

```
$ cd src
$ git clone git://github.com/lh3/wgsm.git
```

Compile wgsm

```
$ cd wgsm/
$ gcc -g -O2 -Wall -o wgsm wgsm.c -lz -lm
```
Install the bowties

Cannot install on cygwin! Mac Linux only
Building indices for each tool

```bash
# create a separate index for each tool
mkdir bow1
mkdir bow2
cp ~/refs/yeast.fasta bow1/
cp ~/refs/yeast.fasta bow2/
```

```bash
pwd
/home/ialbert/work/24/bow1

~/.src/bowtie-0.12.7/bowtie-build yeast.fasta yeast1

```

```bash
pwd
/home/ialbert/work/24/bow2
~/.src/bowtie2-2.0.0-beta3/bowtie2-build yeast.fasta yeast2
```
Running the aligners

```
1 echo 'BWA'
2 time bwa aln ~/refs/yeast.fasta 1.fq > 1.aln
3 time bwa samse ~/refs/yeast.fasta 1.aln 1.fq > 1.bwa.sam
4
5 echo 'Bowtie 1'
6 time bowtie bow1/yeast1 -S 1.fq > 1.bowtie1.sam
7
8 echo 'Bowtie 2'
9 time bowtie2 -x bow2/yeast2 1.fq > 1.bowtie2.sam
```
$ wgsim_eval.pl

Usage: wgsim_eval.pl <command> <arguments>

Command: alneval               evaluate alignment in the SAM format
         vareval               evaluate variant calls in the pileup format
         unique                keep the top scoring hit in SAM
         uniqcmp               compare two alignments without multiple hits

$ wgsim_eval.pl alneval

Usage: wgsim_eval.pl alneval [options] <in.sam>

Options: -p  print wrong alignments
         -g  INT   correct if within INT of the true coordinate

$
Quality reports

get alignment reports

```
$ wgsim_eval.pl alneval 1.bwa.sam
03x 1 / 87557 87557 1.142e-05
02x 0 / 3554 91111 1.098e-05
01x 5 / 97 91208 6.578e-05
00x 3854 / 5223 96431 4.003e-02
$
```

```
$ # in bowtie1 all alignments have the map
$ # quality of 255
$
$ wgsim_eval.pl alneval 1.bowtie1.sam | head -1
25x 3927 / 93045 93045 4.221e-02
$
```

```
$ wgsim_eval.pl alneval 1.bowtie2.sam
04x 2 / 88819 88819 2.252e-05
03x 0 / 0 88819 2.252e-05
02x 0 / 1032 89851 2.226e-05
01x 3 / 345 90196 5.543e-05
00x 4002 / 6753 96949 4.133e-02
$
```
Redirect standard error to output

```
# two streams output/error
# to redirect the error to output
wgsim_eval.pl alneval -p 1.bwa.sam 2>&1 | wc -l
```

get alignment reports

```
wgsim_eval.pl alneval 1.bwa.sam
03x 1 / 87557 87557 1.142e-05
02x 0 / 3554 91111 1.098e-05
01x 5 / 97 91208 6.578e-05
00x 3854 / 5223 96431 4.003e-02
```
An R program to plot it

```r
# Read tables
bwa = read.table('bwa.txt', header=F)
bow1 = read.table('bow1.txt', header=F)
bow2 = read.table('bow2.txt', header=F)

# Plotting
plot(bow2$V6, bow2$V5/100000, 'b', col="blue", lwd=5, ann=F)
lines(bwa$V6, bwa$V5/100000, 'b', col="red", pch=22, lwd=5)
lines(bow1$V6, bow1$V5/100000, 'b', col="green", pch=22, lwd=5)
legend(0.0, 0.96, c('Bowtie2', 'BWA', 'Bowtie1'), cex=0.8, col=c("blue", "red", "green"), lty=1:1:1)
title(xlab='False positive rate')
title(ylab='Percent of reads mapped')
```
Homework 24

• Install **R** (optionally **Rstudio**) (will be using them during the next two lectures)

• Install and then run **wgsim_eval.pl** script on one of your SAM files that came from a **wgsim** simulator

• Inspect the CIGAR strings of reads that were classified incorrectly and see if there is a pattern to them