Samtools & Gbrowse2 for NGS

10/19/10
What is Samtools?

• A supported format for alignments
  • SAM - human-readable, scriptable
  • BAM - compact, indexed alignments
• A software toolset
• An API
Samtools and Friends

- Aligners
- Raw reads
- Viewers/Browsers
- Language bindings
- SNP/CNV/SV detection
- Transformers
- Peak finders

SAM/BAM Format

http://samtools.sourceforge.net/swlist.shtml
Samtools and Friends

Aligners

- BFAST
- Bowtie & TopHat
- BWA
- Mosaik
- Novoalign
- SSAHA2

Raw reads

SAM/BAM Format

Language bindings

Viewers/Browsers

Peak finders

SNP/CNV/SV detection

Transformers

http://samtools.sourceforge.net/swlist.shtml
Samtools and Friends

FindPeaks (Vancouver)
MACS
QuEST (Stanford)

Peak finders

http://samtools.sourceforge.net/swlist.shtml
Samtools and Friends

Aligners

Language bindings

Bio-SamTools (Perl)
Picard (Java)
BamTools (C++)
Pysam (Python)
Samtools-Ruby (Ruby)
cl-sam (Common Lisp)
R-Samtools

SNP/CNV/SV detection

Transformers

SAM/BAM Format

Viewers/Browsers

Raw reads
Samtools and Friends

**SV Detection**
Breakdancer (WashU)

**SNP pipelines:**
Genome Analysis Toolkit (Broad)

**CNV Detection**
CNVer (Toronto)

Aligners

SNP/CNV/SV detection

SAM/BAM Format

Transformers

Peak finders

Viewers/Browsers

Raw reads

http://samtools.sourceforge.net/swlist.shtml
Simple Pileup

```bash
samtools pileup -cf ex1.fa ex1.bam
```
Verbose Pileup

samtools-0.1.8/samtools pileup -v -c -f yeasties.fa SRR003681.sorted.bam > SRR003681.pileup

I do not understand the columns in the pileup output.

This is explained in the manual page. Or briefly (when you invoke pileup with the −c option):

1. reference sequence name
2. reference coordinate
3. reference base, or ‘*’ for an indel line
4. genotype where heterozygotes are encoded in the IUB code: M=A/C, R=A/G, W=A/T, S=C/G, Y=C/T and K=G/T; indels are indicated by, for example, */+A, −A/* or +CC−C. There is no difference between */+A or +A/*.
5. Phred-scaled likelihood that the genotype is wrong, which is also called ‘consensus quality’.
6. Phred-scaled likelihood that the genotype is identical to the reference, which is also called ‘SNP quality’. Suppose the reference base is A and in alignment we see 17 G and 3 A. We will get a low consensus quality because it is difficult to distinguish an A/G heterozygote from a G/G homozygote. We will get a high SNP quality, though, because the evidence of a SNP is very strong.
7. root mean square (RMS) mapping quality
8. # reads covering the position
9. read bases at a SNP line (check the manual page for more information); the 1st indel allele otherwise
10. base quality at a SNP line; the 2nd indel allele otherwise
11. indel line only: # reads directly supporting the 1st indel allele
12. indel line only: # reads directly supporting the 2nd indel allele
13. indel line only: # reads supporting a third indel allele

If pileup is invoked without ‘−c’, indel lines and columns between 3 and 7 inclusive will not be outputted.
Samtools and Friends

- Aligners

- Viewers/Browsers
  - gbrowse2
  - Tablet

- Raw reads
- Peak finders
- SAM/BAM Format
- Transformers
Sam Output

QNAME = name of read
FLAG  = Bitwise FLAG ($2^{18}-1$)
RNAME = Reference sequence name
POS   = Position (1-based)
MAPQ  = Mapping Quality (Phred-based)
CIGAR = CIGAR STRING
MRNM  = Mate Reference Sequence
MPOS  = 1-based Mate Position of the other seq
ISIZE = Inferred Insert Size
SEQ   = Sequence reported on the + strand
QUAL  = Quality scores (ASCII-33 = Phred)
TAG   = TAG

<table>
<thead>
<tr>
<th>Tag</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM</td>
<td>Edit distance</td>
</tr>
<tr>
<td>MD</td>
<td>Mismatching positions/bases</td>
</tr>
<tr>
<td>AS</td>
<td>Alignment score</td>
</tr>
<tr>
<td>XB</td>
<td>Number of best hits</td>
</tr>
<tr>
<td>X1</td>
<td>Number of suboptimal hits found by BWA</td>
</tr>
<tr>
<td>XN</td>
<td>Number of ambiguous bases found by BWA</td>
</tr>
<tr>
<td>XM</td>
<td>Number of mismatches in the reference</td>
</tr>
<tr>
<td>XO</td>
<td>Number of gap opens</td>
</tr>
<tr>
<td>XG</td>
<td>Number of gap extensions</td>
</tr>
<tr>
<td>XT</td>
<td>Type: Unique/Repeat/N/Mate-swap</td>
</tr>
<tr>
<td>XS</td>
<td>Suboptimal alignment score</td>
</tr>
<tr>
<td>XF</td>
<td>Support from forward/reverse alignment</td>
</tr>
<tr>
<td>XF</td>
<td>Number of supporting seeds</td>
</tr>
</tbody>
</table>
High-level API

```perl
use Bio::DB::Sam;

# high level API
my $sam = Bio::DB::Sam->new(-bam => "data/ex1.bam",
                           -fasta => "data/ex1.fa",
                          );

my @targets = $sam->seq_ids;
my @alignments = $sam->get_features_by_location(-seq_id => 'seq2',
                                                 -start => 500,
                                                 -end => 800);

for my $a (@alignments) {
    # where does the alignment start in the reference sequence
    my $seqid = $a->seq_id;
    my $start = $a->start;
    my $end = $a->end;
    my $strand = $a->strand;
    my $cigar = $a->cigar_str;
    my $paired = $a->get_tag_values('PAIRED');

    # where does the alignment start in the query sequence
    my $query_start = $a->query->start;
    my $query_end = $a->query->end;

    my $ref_dna = $a->dna;
    # reference sequence bases
    my $query_dna = $a->query->dna;
    # query sequence bases

    my @scores = $a->qscore;
    # per-base quality scores
    my @match_qual = $a->qual;
    # quality of the match
}

my @pairs = $sam->get_features_by_location(-type => 'read_pair',
                                           -seq_id => 'seq2',
                                           -start => 500,
                                           -end => 800);

for my $pair (@pairs) {
    my $length = $pair->length;
    # insert length
    my ($first_mate,$second_mate) = $pair->get_SeqFeatures;
    my $first_start = $first_mate->start;
    my $second_start = $second_mate->start;
}
```

Low-level API

```perl
# low level API
my $bam = Bio::DB::Bam->open('/path/to/bamfile');
my $header = $bam->header;
my $target_count = $header->n_target;
my $target_names = $header->target_name;
while (my $align = $bam->read1) {
    my $seqid = $target_names->[align->tid];
    my $start = $align->pos+1;
    my $end = $align->calend;
    my $cigar = $align->cigar_str;
}

my $index = Bio::DB::Bam->index_open('/path/to/bamfile');
my $index = Bio::DB::Bam->index_open_in_scfewd('/path/to/bamfile');

my $callback = sub {
    my $alignment = shift;
    my $start = $alignment->start;
    my $end = $alignment->end;
    my $seqid = $target_names->[alignment->tid];
    print $alignment->qname," aligns to $seqid:$start..$end\n";
}

my $header = $index->header;
$index->fetch($bam, $header->parse_region('seq2'), $callback);
```

http://search.cpan.org/~lds/Bio-SamTools/lib/Bio/DB/Sam.pm
Problem Set

Get Bowtie for Mac (unzip, make). You may use another SAM-compliant aligner if you are adventurous.

http://bowtie-bio.sourceforge.net/

Download the reference sequences that came with gbrowse2

http://localhost/gbrowse2/databases/yeast_scaffolds/

Concatenate them:
cat chr1.fa chr2.fa > yeasties.fa

Build the indices for those two reference sequences:
bowtie-build yeasties.fa yeasties

Download this short read archive:
http://infoserver.cshl.edu/LectureNotes/ngs2/SRR003681.subset.fq

Study the section on FASTQ encoding: http://en.wikipedia.org/wiki/FASTQ_format#Encoding

Type 'tail SRR003681.subset.fq' to see some less than perfect sequences at the end of the file. Can you tell what encoding was used for these sequences?

bowtie-0.12.7/bowtie --sam yeasties SRR003681.subset.fq > SRR003681.sam

Can you explain to my grandmother what Bowtie's default alignment settings are?
http://bowtie-bio.sourceforge.net/manual.shtml#the--n-alignment-mode
Problem Set

Copy Samtools to your current directory
```bash
cp /Network/Servers/infoserver.cshl.edu/Users/Shared/samtools-0.1.8 .
```

These three incantations are always necessary to get a sorted, index BAM from a SAM file:

```bash
samtools-0.1.8/samtools view -b -S -T yeasties.fa SRR003681.sam > SRR003681.bam
samtools-0.1.8/samtools sort SRR003681.bam SRR003681.sorted
samtools-0.1.8/samtools index SRR003681.sorted.bam
```

The script calculates coverage depth at each position of chrI.
Use your perl skills to calculate the maximum, minimum, and average depth.
The script is located at: http://infoserver.cshl.edu/LectureNotes/ngs2/getCoverage.pl

```perl
#!/usr/bin/perl
use strict;
use warnings;
use Bio::DB::Sam;

# high level API
my $sam = Bio::DB::Sam->new(-bam => "SRR003681.sorted.bam",
                           -fasta => "yeasties.fa",
);;

my $segment = $sam->segment(-seq_id => 'chrI');
my ($coverage) = $segment->features('coverage');
my @data_points = $coverage->coverage;
foreach my $depth (@data_points){
    print $depth."
";
}```
Copy the following four files into a new directory called bamfiles:
- SRR003681.sorted.bam
- SRR003681.sorted.bam.bai
- yeasties.fa
- yeasties.fa.fai

Move this directory: (do you have admin privileges on your computer yet?)
sudo mv bamstuff /Library/WebServer/Documents/gbrowse2/databases/

Add this to the database definitions section of /etc/gbrowse2/yeast_chr1+2.conf
[SRR003681:database]
db_adaptor = Bio::DB::Sam
db_args = -fasta /Library/Webserver/Documents/gbrowse2/databases/bamfiles/yeast_chromosomes.fa
-bam /Library/Webserver/Documents/gbrowse2/databases/bamfiles/SRR003681.sorted.bam
search options = default

Add these to the tracks section of /etc/gbrowse2/yeast_chr1+2.conf

[Cov eryXyplot]
feature = coverage
glyph = wiggle_xyplot
database = SRR003681
height = 50
fgcolor = black
biclo r_pivot = 20
pos_color = blue
neg_color = red
key = Coverage (xyplot)
category = NGS
label = 0

[Reads]
feature = match
glyph = segments
draw_target = 1
show_mismatch = 1
mismatch_color = red
database = SRR003681
bcolor_pivot = 20
bgcolor = blue
fgcolor = white
height = 5
label density = 50
bump = fast
key = Reads
category = NGS

Go to http://localhost/cgi-bin/gb2/gbrowse/yeast_advanced/
Turn on the NGS tracks.
NOTE this is the advanced yeast gbrowse instance not the basic one

Having problems?
tail /var/log/apache2/error_log