Bin-by-sam Summary

Overview

- This script outputs a table of read coverage by bin across a reference sequence, using a directory sam files created using samtools as input.
- It can also output a measure of relative coverage compared to a control dataset. There can be two types of control data: either a control file is indicated or the mean of all files in the directory is calculated and used as the control values.
- A more detailed description of the relative coverage calculation can be found in the Output area below.
- This script also outputs a second small file containing the number of successfully mapped reads, indicated by a flag of 0 or 16 in the sam file.
- IMPORTANT: This script aims at identifying RELATIVE difference in coverage between samples. It is not adequate to determine ploidy levels. Samples of different ploidies, if they don't contain any dosage variation (large insertions, deletions or chromosomal aneuploidies) will look identical.

Initial Step: Collecting the .sam files

This is a collection of samtools alignment files ending with .sam that contain a sequence header and mapped reads in text format. The sam files ***must*** end with the exact string "_aln.sam".

The sequence header is a list of the reference sequences (chromosomes, scaffolds, contigs, amplicons etc) and their lengths. The form appears as the following.

```
@SQ    SN:Chr01       LN:50495391
@SQ    SN:Chr02       LN:25263035
@SQ    SN:Chr03       LN:21816808
@SQ    SN:Chr04       LN:24267051
@SQ    SN:Chr05       LN:25890704
```

There is one of these lines for every reference, including scaffolds and other reference types. Seeing a header containing both types of reference molecules is common:

```
@SQ    SN:Chr16       LN:14494361
@SQ    SN:Chr17       LN:16080358
@SQ    SN:Chr18       LN:16958300
@SQ    SN:Chr19       LN:15942145
@SQ    SN:scaffold_20  LN:948134
@SQ    SN:scaffold_21  LN:931759
@SQ    SN:scaffold_22  LN:1028913
```

Following the header, is the mapping information for all of the sequencing reads. This appears in the standard samtools format. As an example, this is a 50 bp sequencing read mapped to Chromosome 17 in the forward direction:

DJB77P1:564:C3016ACXX:8:1101:8769:3590 0 Chr17 3396702 37 50M * 0 0 CTGTCTTGTGTAAATTCGCAGAAGTTATGATTTATCATGTTATGTCATGA @C@FFFFFCFCDDBGEEHIIIBHGFHHIIGFGGGICHIIBFGIIIEHGCG XT:A:U NM:i:0 X0:i:1 X1:i:0 XM:i:0 XO:i:0 XG:i:0 MD:Z:50

There is a line for every read, a bitwise flag of 0 or 16 in the second column is a valid mapped read, all other flags indicate that the read was not mapped correctly.

Organizing the sam files.
All .sam files to be included in the bin-by-sam must be placed in the same directory. You must run the program in the same directory as the .sam files. The program itself can be in a different directory, but the command must be run while in the directory with the sam files. It is recommended the directory be empty otherwise. Please note all files must be aligned to the exact same reference file, using slightly different versions of the same reference is not acceptable. Also, all files MUST end in "_aln.sam".

Preparing to Run the Program: Selecting parameters
The program bin-by-sam.py has several required and a few optional parameters that must be selected at runtime.

Required Parameters

1. -o, output file name. A result file with this name will be generated with the collected bins, and another file with the name readcounts-<selected-name> will be generated with read counts per sample. It is good to include all relevant information into the output file name so that one can later remember which parameters were used. For example: Experiment3_100kbbins_Uniques_3mm.txt.

2. -s, bin size (an integer for length of bin in bps). Each reference sequence will be divided into consecutive, non-overlapping bins of a certain size and all reads will be assigned to a single bin. The -s parameter allows you to set the size of the bin. This can be set to be very small, which allows narrow precision but slows the runtime and increases noise, or to be very large. If a size larger than the largest chromosome / refseq is selected, the reads will be binned on a per chromosome/reference basis.

Optional Parameters

- -c, "control_aln.sam" where "control_aln.sam" is the filename corresponding to the sample to use as a control, if a control is to be used. This is an option to use a control sam file for relative percent coverage calculations. In that case, relative coverage will be computed against the coverage of the sample specified instead of the global average. The control sample will be automatically set to the basal ploidy value (-p option below) and relative coverage values for all other samples will be calculated relative to the control sample. Default is no control file.

- -u. This option limits the analysis to reads flagged by samtools as “unique”, i.e. unambiguously mapped. Such reads bear samtools unique flag (XT:A:U).

- -m “integer”. This option limits the analysis to reads exhibiting a maximum of “integer” mismatches to the reference genome. The number of mismatches is indicated in field field 15 of the .sam file XM:i:(Number of mismatches). The typical and default value used for this is 5.

- -b, this option inserts empty lines between reference sequences in the result table. This is primarily for downstream analyses allowing the reference sequences to be more easily visualized in graphs. It is not recommended if the reference sequence contain a large number of small contigs as opposed to a few assembled chromosomes.

- -r “remove_file.txt”. This points to a file in sam header format of reference sequences to ignore, there is an included example file Remove-Sample.txt in the archive. This can be useful for excluding short contigs with high repeat content of organellar chromosomes for example.

- -p “integer” to indicate the background ploidy level. The default is 2 (diploid), this is used as the multiplier in the relative coverage calculation.

- -C. This option should be used if relative coverage information is not required. In this case, only raw read counts are output, not relative coverage. This option cannot be used when a control library is specified.

Using Parameters

Parameters are specified by adding them after the program name.
Output file name and bin size are required parameters, which can be specified like this:
./bin-by-sam.py -o outputfilename.txt -s bin-size-number

For example, for an output file named out.txt and a 1MB bin size, the command would be:
./bin-by-sam.py -o out.txt -s 1000000

To view the possible parameters at any point, simply use the -h or --help parameter, like this:
./bin-by-sam.py -h **or** ./bin-by-sam.py --help

Optional parameters are specified in the same way as the required ones.
For example, to add breaks between chromosomes and specify a maximum of 6 snps, (while still using out.txt and a 1MB bin size) the command would be:
./bin-by-sam.py -o out.txt -s 1000000 -b -m 6

Running and Output
When running, the screen will cycle through progress as it scans the .sam files in the current directory, this is normal and means the program is running correctly.

When finished, two new files should have appeared in the current working directory:

1. An out.txt file, with the binning information for the sample.sam files present in the directory.
2. A readcounts-out.txt file, with the read counts from each .sam file for quick referencing. This is the number of reads and reads per MB used for the analysis.

The output text file has a specific format.

1. The first column indicates the Chromosome/Reference name
2. Columns 2 and 3 indicate the start and end positions of the bin.
3. Next come the number of reads per bin for each sample file (as many columns as there are samples).
4. Unless the "-C" option was activated, another set of columns will be present, indicating relative coverage calculated in one of two ways;
    a. If no control file is used, read coverage values will be normalized to the mean coverage of all samples.
    b. In a control library is used, read coverage values will be normalized to the read coverage values of the control sample.

Note that by default 4a and 4b are both calculated for a diploid individual, to correct for a different ploidy use the ploidy parameter -p to specify a background ploidy other than 2.

How Relative Mean Coverage values are calculated.

1) Percentage calculations.

For each sample and each bin, the percentage of reads mapping to that particular bin is recorded.

For example, for sample A and bin X:
%Reads = % reads in bin X = # reads mapped to bin X / total # reads for sample A.

2) Normalization.

If there is no control sample:

<u>For each bin, the mean %Reads is calculated across all samples.</u>

<u>For each bin, the normalized coverage for each sample is calculated as follows:</u>

Norm. Cov. = %Reads / Mean %Reads

If there is a control sample:

For each bin, the control %Reads is calculated for the control sample (%Reads(control).

For each bin, the normalized coverage for each sample is calculated as follows:
Norm. Cov. = %Reads / %Reads(control)

3) Ploidy adjustment

For visualization purposes, it is easier to adjust values to background ploidy of the samples. The final normalized coverage values are thus adjusted as follows;
Adj. Cov = Norm. Cov x ploidy modifier.

As a sample output, for two samples names S1 and S2, the header and an output row for no control would look like:

Chrom	Strt	End	S1	S2	S1/NA	S2/NA
Chr01	1	1000000	1486	1022	2.133	1.867

And an example with S2 as the control would look like:

Chrom	Strt	End	S1	S2	S1/S2	S2/S2
Chr01	1	1000000	1486	1022	2.286	2.0