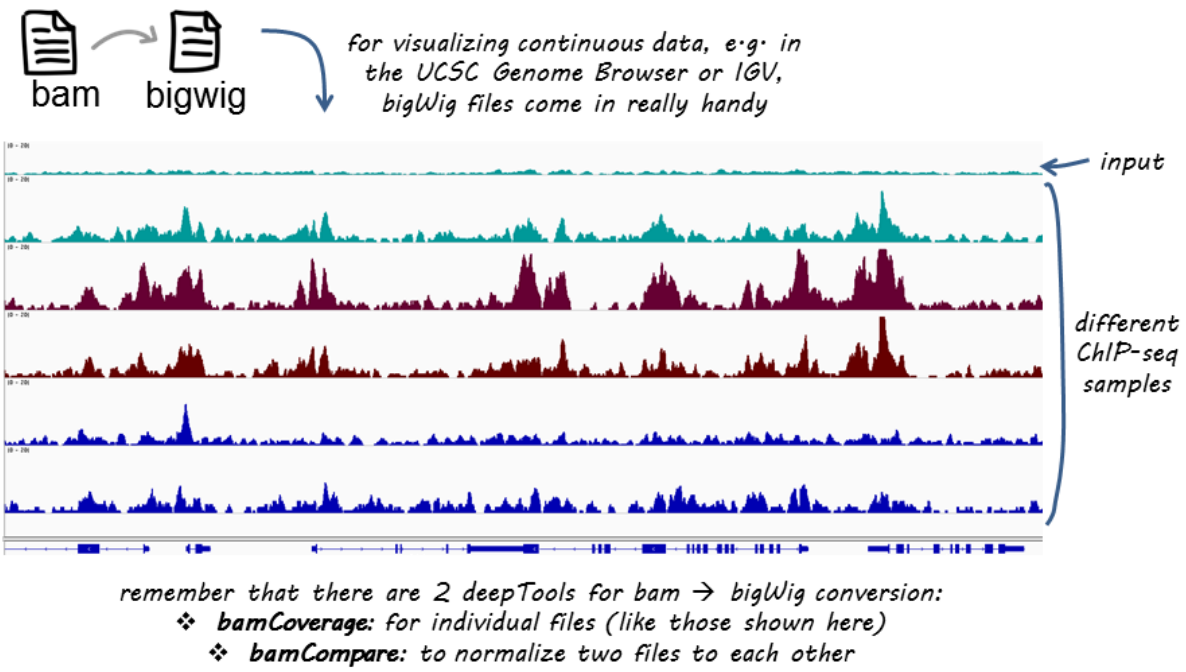
Normalization of BAM files

deepTools contains 3 tools for the normalization of BAM files:

1. **correctGCbias**: if you would like to normalize your read distributions to fit the expected GC values, you can use the output from computeGCbias and produce a GC-corrected BAM-file.
2. **bamCoverage**: this tool converts a *single* BAM file into a bigWig file, enabling you to normalize for sequencing depth.
3. **bamCompare**: like bamCoverage, this tool produces a normalized bigWig file, but it takes 2 BAM files, normalizes them for sequencing depth and subsequently performs a mathematical operation of your choice, i.e. it can output the ratio of the read coverages in both files or the like.



correctGCbias

What it does

This tool requires the **output from computeGCBias** to correct the given BAM files according to the method proposed by Benjamini and Speed.

correctGCbias will remove reads from regions with too high coverage compared to the expected values (typically GC-rich regions) and will add reads to regions where too few reads are seen (typically AT-rich regions).

The resulting BAM files can be used in any downstream analyses, but **be aware that you should not filter out duplicates from here on** (duplicate removal would eliminate those reads that were added to reach the expected number of reads for GC-depleted regions).

output

- **GC-normalized BAM file**

bamCoverage

What it does

Given a BAM file, this tool generates a bigWig or bedGraph file of fragment or read coverages. The way the method works is by first calculating all the number of reads (either extended to match the fragment length or not) that overlap each bin in the genome. Bins with zero counts are skipped, i.e. not added to the output file. The resulting read counts can be normalized using either a given scaling factor, the RPKM formula or to get a 1x depth of coverage (RPGC).

- RPKM:
  - reads per kilobase per million reads
  - The formula is: RPKM (per bin) = number of reads per bin / (number of mapped reads (in millions) * bin length (kp))
- RPGC:
  - reads per genomic content
  - used to normalize reads to 1x depth of coverage
  - sequencing depth is defined as: (total number of mapped reads * fragment length) / effective genome size

output

- **coverage file** either in bigWig or bedGraph format

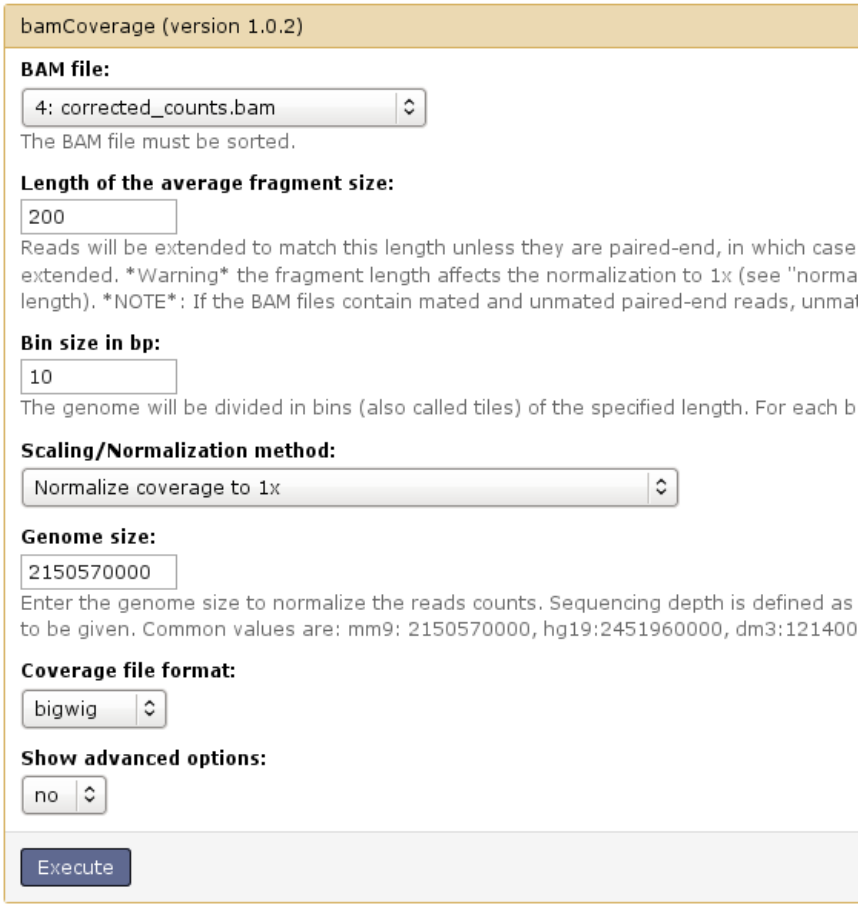
Usage

Here's an exemplary command to generate a single bigWig file out of a single BAM file via the command line:

```
$/deepTools-1.5/bin/bamCoverage --bam corrected_counts.bam \
--binSize 10 --normalizeTo1x 2150570000 --fragmentLength 200 \
-o Coverage.GCcorrected.SeqDepthNorm.bw --ignoreForNormalization chrX
```

- The bin size **(-bs)** can be chosen completely to your liking. The smaller it is, the bigger your file will be.
- This was a mouse sample, therefore the effective genome size for mouse had to be indicated once it was decided that the file should be normalize to 1x coverage.
- Chromosome X was excluded from sampling the regions for normalization as the sample was from a male mouse that therefore contained pairs of autosome, but only a single X chromosome.
- The fragment length of 200 bp is only the fall-back option of bamCoverage as the sample provided here was done with paired-end sequencing. Only in case of singletons will bamCoverage resort to the user-specified fragment length.
- --ignoreDuplicates - important! in case where you normalized for GC bias using correctGCbias, you should absolutely **NOT** set this parameter

Using deepTools Galaxy, this is what you would have done (pay attention to the hints on the command line as well!):



bamCompare

What it does

This tool compares **two** BAM files based on the number of mapped reads. To compare the BAM files, the genome is partitioned into bins of equal size, the reads are counted for each bin and each BAM file and finally, a summarizing value is reported.This value can be the ratio of the number of reads per bin, the log2 of the ratio or the difference. This tool can normalize the number of reads on each BAM file using the SES method proposed by Diaz et al. Normalization based on read counts is also available. If paired-end reads are present, the fragment length reported in the BAM file is used by default.

output file

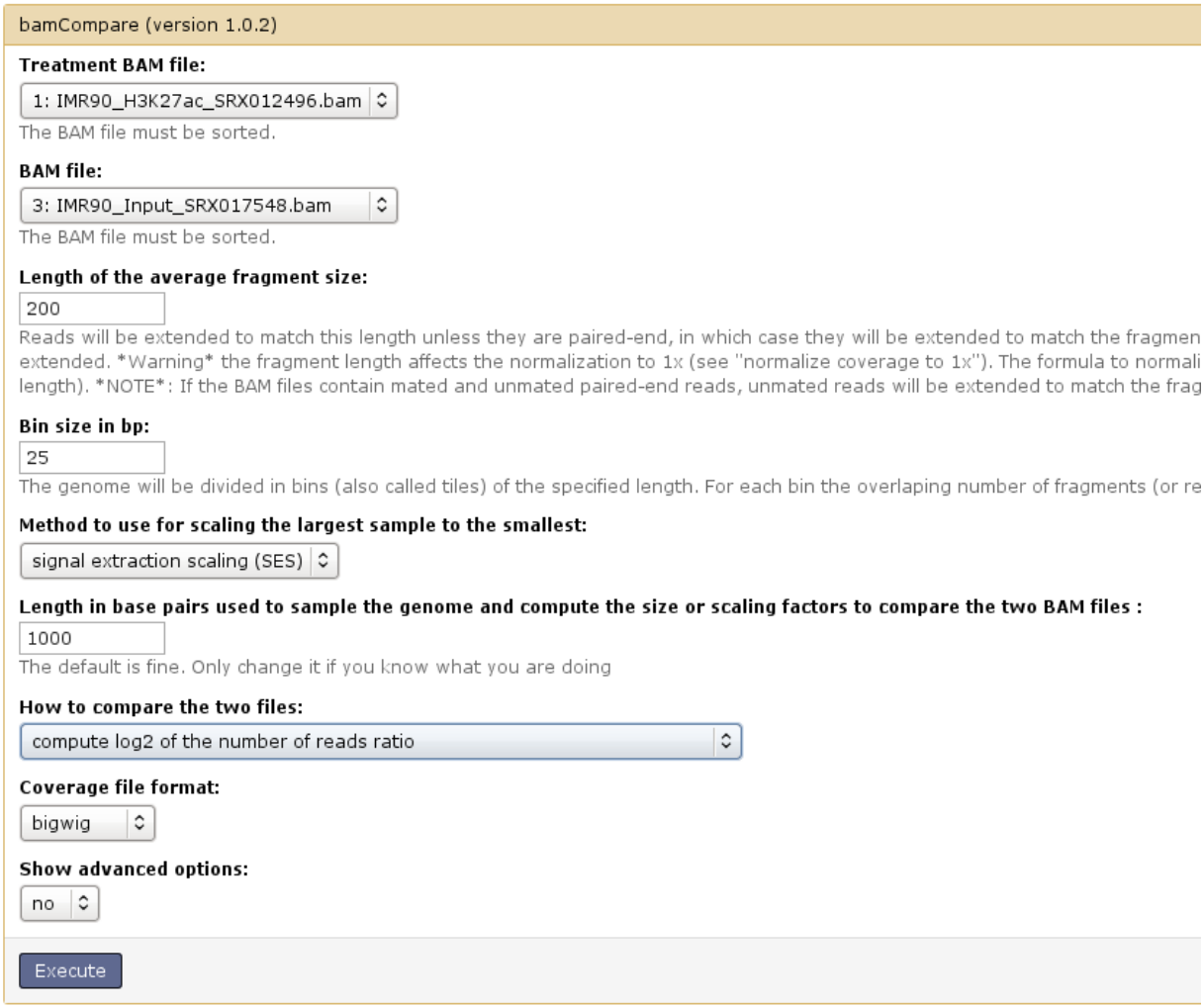
- same as for bamCoverage, except that you now obtain **1** coverage file that is based on **2** BAM files.

Usage

Here's an example command that generated the log2(ChIP/Input) values via the command line.

```
$ /deepTools-1.5/bin/bamCompare --bamfile1 ChIP.bam -bamfile2 Input.bam \
--binSize 25 --fragmentLength 200 --missingDataAsZero no \
--ratio log2 --scaleFactorsMethod SES -o log2ratio_ChIP_vs_Input.bw
```

The Galaxy equivalent:



Note that the option "missing Data As Zero" can be found within the "advanced options" (default: no).

- like for bamCoverage, the bin size is completely up to the user
- the fragment size (-f) will only be taken into consideration for reads without mates
- the SES method (see below) was used for normalization as the ChIP sample was done for a histone mark with highly localized enrichments (similar to the left-most plot of the bamFingerprint-examples)

Some (more) parameters to pay special attention to

--scaleFactorsMethod (in Galaxy: "Method to use for scaling the largest sample to the smallest")

Here, you can choose how you would like to normalize to account for variation in sequencing depths. We provide:

- the simple normalization **total read count**
- the more sophisticated signal extraction (SES) method proposed by Diaz et al. for the normalization of ChIP-seq samples. **We recommend to use SES only for those cases where the distinction between input and ChIP is very clear in the bamFingerprint plots**. This is usually the case for transcription factors and sharply defined histone marks such as H3K4me3.

--ratio (in Galaxy: "How to compare the two files")

Here, you get to choose how you want the two input files to be compared, e.g. by taking the ratio or by subtracting the second BAM file from the first BAM file etc. In case you do want to subtract one sample from the other, you will have to choose whether you want to normalize to 1x coverage (--normalizeTo1x) or to **r**eads **p**er **k**ilobase (--normalizeUsingRPKM; similar to RNA-seq normalization schemes).