How we use deepTools

The majority of samples that we handle within our facility come from ChIP-seq experiments, therefore you will find many examples from ChIP-seq analyses. This does not mean that deepTools is restricted to ChIP-seq data analysis, but some tools, such as *bamFingerprint* specifically address ChIP-seq-issues. (That being said, we do process quite a bit of RNA-seq, other -seq and genomic sequencing data using deepTools, too.)

As depicted in the figure down below, our work usually begins with one or more FASTQ file(s) of deeply-sequenced samples. After a first quality control using FASTQC, we align the reads to the reference genome, e.g. using bowtie2. We then use deepTools to assess the quality of the aligned reads:

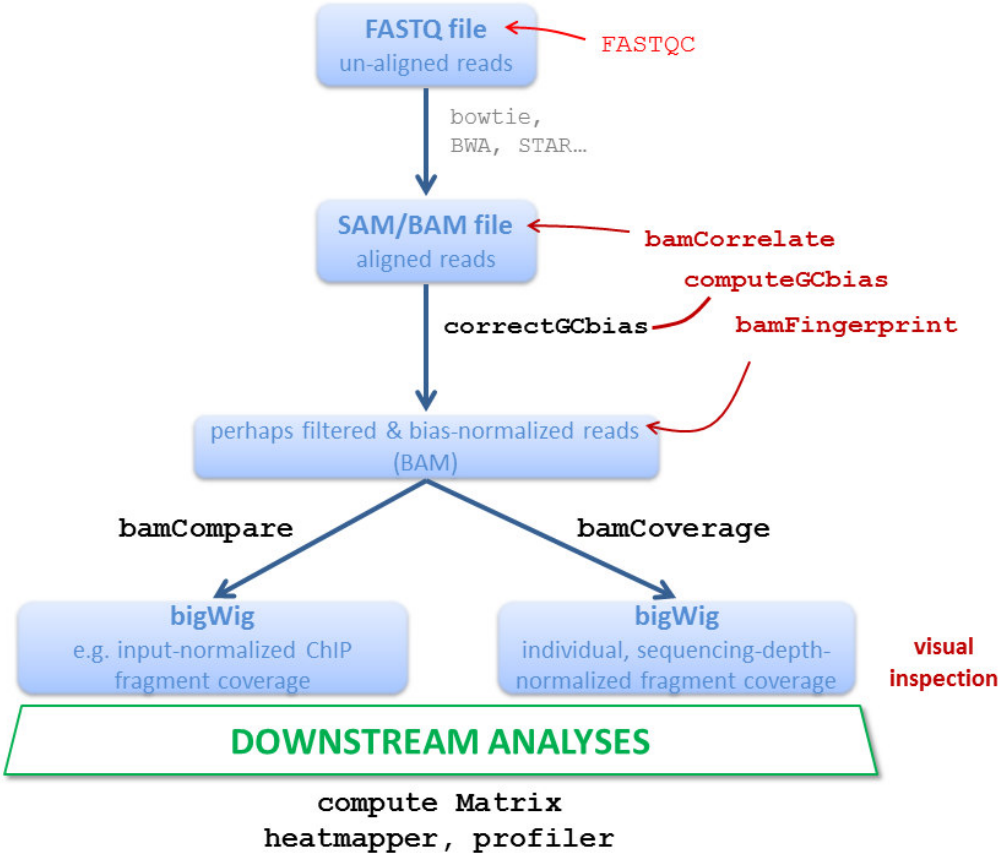
1. **Correlation between BAM files** (*bamCorrelate*). This is a very basic test to see whether the sequenced and aligned reads meet your expectations. We use this check to assess the reproducibility - either between replicates and/or between different experiments that might have used the same antibody/the same cell type etc. For instance, replicates should correlate better than differently treated samples.
2. **GC bias check** (*computeGCbias*). Many sequencing protocols require several rounds of PCR-based amplification of the DNA to be sequenced. Unfortunately, most DNA polymerases used for PCR introduce significant GC biases as they prefer to amplify GC-rich templates. Depending on the sample (preparation), the GC bias can vary significantly and we routinely check its extent. In case we need to compare files with different GC biases, we use the *correctGCbias* module to match the GC bias. See the paper by Benjamini and Speed for many insights into this problem.
3. **Assessing the ChIP strength**. This is a QC we do to get a feeling for the signal-to-noise ratio in samples from ChIP-seq experiments. It is based on the insights published by Diaz et al..

Once we're satisfied by the basic quality checks, we normally **convert the large BAM files into a leaner data format, typically bigWig**. bigWig files have several advantages over BAM files that mainly stem from their significantly decreased size: - useful for data sharing & storage - intuitive visualization in Genome Browsers (e.g. UCSC Genome Browser, IGV) - more efficient downstream analyses are possible

The deepTools modules *bamCompare* and *bamCoverage* do not only allow the simple conversion from BAM to bigWig (or bedGraph for that matter), **the main reason why we developed those tools was that we wanted to be able to *normalize* the read coverages** so that we could compare different samples despite differences in sequencing depth, GC biases and so on.

Finally, once all the files have passed our visual inspections, the fun of downstream analyses with *heatmapper* and *profiler* can begin!

Here's a visual summary of our average workflow - deepTools modules are indicated in bold letters, alternative software such as FASTQC and bowtie are noted in regular font. Everything written in red is related to quality control (QC) of the samples.



General options and parameters of deepTools

Once you select a tool, you will see that almost every option has a brief description of its purpose. There are some options that you will encounter over and over again, so it's important that you understand their implications. Most of these options are related to the **computation of read coverages**.

- **Length of average fragment size**: For high-throughput sequencing of short reads, the cells' DNA is typically sheared and fragments of a certain size are selected to be sequenced. Very often, this will be between 200 to 300 bp. From each of these fragments, the reads one obtains will only represent the first (and/or) last 30-50 bp (depending on the chosen read length). When calculating coverages, we therefore extend the reads to match the original fragment size. In the case of paired-end sequencing, the exact fragment size will be known, for single-end sequencing every read will be extended to the same length.
- **bin size**: In order to create a continuous profile of read coverages along the genome, we need to divide the genome into regions of equal length for which the number of overlapping reads is counted. The size of this window could be 1 bp, however, for practical reasons we usually chose a bin size between 10 to 50 bp (depending on the depth of sequencing, the size of the genome and the desired resolution).
- **Minimum mapping quality**: This is an optional parameter ("Advanced options"). If you set the Minimum mapping quality to 10, all reads with a mapping quality below 10 will not be taken into consideration for the read coverage computation.
- **ignore duplicates**: This an optional parameter, too, that will filter out reads that have the exact same start and end point (thought to be PCR artefacts). You should absolutely **not** select this option if you have a BAM file that you corrected for GC bias.
- **missing Data as zero**: If this option is selected, regions where no overlapping reads are found will be included as regions with coverage = 0. Note that this is different from including those regions with "no

coverage". Imagine 4 genome bins with read coverages of 0,1,2,3 - the average of these four bins will be 6/4 = 1.5. If the first bin would have been marked with "no coverage", it would not have been included in the calculation for the average read coverage, thus the result would have been 6/3 = 2.

References

This tool suite is developed by the Bioinformatics Facility at the Max Planck Institute for Immunobiology and Epigenetics, Freiburg.