QC of aligned reads

Table of Contents

bamCorrelate

This tool is useful to assess the overall similarity of different BAM files. A typical application is to check the correlation between replicates or published data sets.

What it does

The tool splits the genomes into bins of a given length. For each bin, the number of reads found in each BAM file is counted and a correlation of the read coverages is computed for all pairs of BAM files.

Important parameters

bamCorrelate can be run in 2 modes: *bins* and *bed*.

In the bins mode, the correlation is computed based on **randomly sampled bins of equal length**. The user has to specify the *number* of bins. This is useful to assess the overall similarity of BAM files, but outliers, such as heavily biased regions have the potential to skew the correlation values.

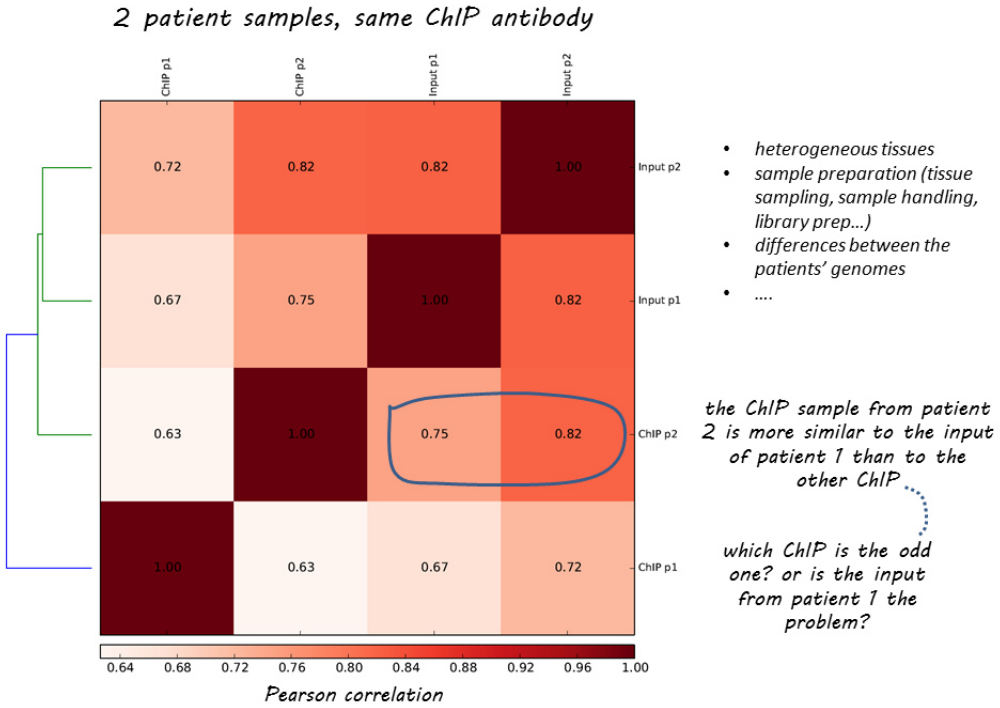
In the BED-file options, the user supplies a list of genomic regions in BED format in addition to (a) BAM file(s). bamCorrelate subsequently uses this list to compare the read coverages for these regions only. This can be used, for example, to compare the ChIP-seq coverages of two different samples for a set of peak regions.

Output files:

- **diagnostic plot** the plot produced by bamCorrelate is a clustered heatmap displaying the values for each pair-wise correlation, see below for an example
- **data matrix** (optional) in case you want to plot the correlation values using a different program, e.g. R, this matrix can be used

Example Figures

Here is the result of running bamCorrelate. We supplied four BAM files that were generated from 2 patients - for each patient, there is an input and a ChIP-seq sample (from GSE32222).

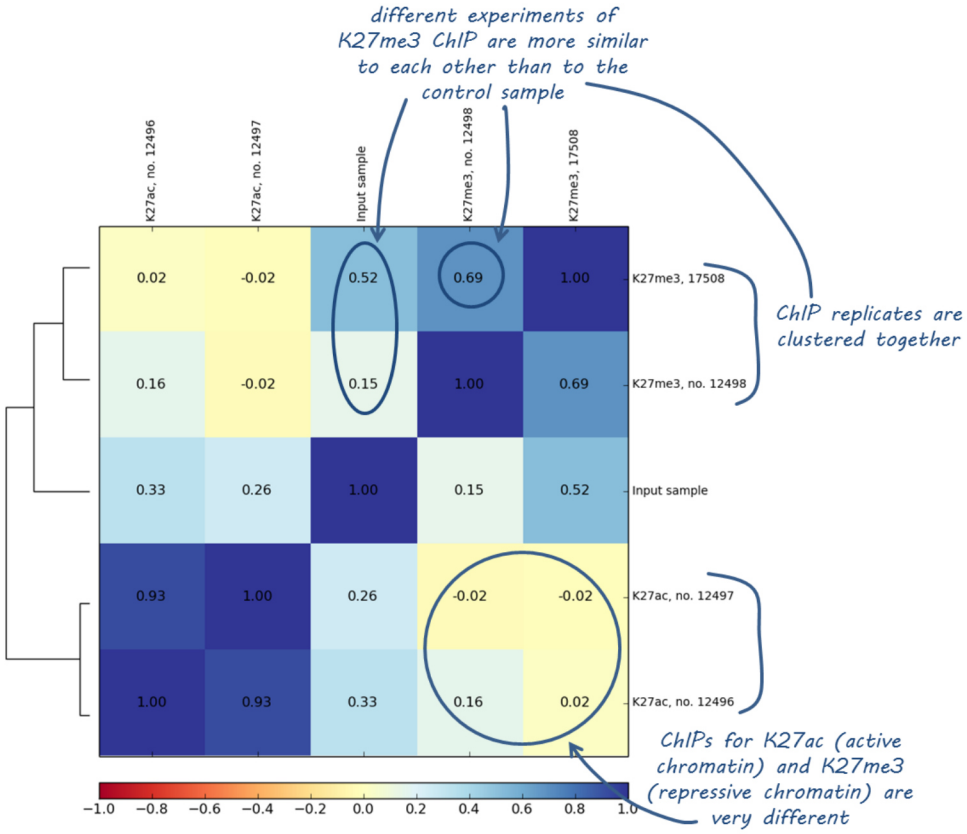


You can supply any number of BAM files that you would like to compare. In Galaxy, you simply have to click "Add BAM file", in the command line you simply list all files one after the other (you can give meaningful name via the --label option).

Here's the command that was used with the command line version:

$ deepTools-1.5.2/bin/bamCorrelate bins --fragmentLength 200 --bamfiles GSM798383_SLX-1201.250.s_4.bwa.homo_sapiens_f.bam GSM798384_SLX-1881.334.s_1.bwa.homo_sapiens_f.bam GSM798406_SLX-1202.250.s_1.bwa.homo_sapiens_f.bam GSM798407_SLX-1880.337.s_8.bwa.homo_sapiens_f.bam --labels "ChIP p1" "ChIP p2" "Input p1" "Input p2" --plotFile /eva_data /deeptools_manual/bamCorrelate_bad2.pdf --corMethod pearson

Here is another example of ChIP samples where H3K27ac was ChIPed by the same experimentator for different cell populations while H3K27me was performed with the same antibody, but at different times. You can see that the correlation between the K27ac replicates is much higher than for the H3K27me3 samples, however, for both histone marks, the ChIP-seq experiments are more similar to each other than to the other ChIP or to the input. In fact, the signals of K27ac and K27me3 are almost not correlated at all which supports the notion that their biological function is also quite opposing.



computeGCbias

This tool computes the GC bias using the method proposed by Benjamini and Speed (see below for more explanations).

What it does

The basic assumption of the GC bias diagnosis is that an ideal sample should show a uniform distribution of sequenced reads across the genome, i.e. all regions of the genome should be similarly well sequenced.

computeGCbias estimates how many reads with what kind of GC content one should have sequenced given an organism's genome GC content. The calculations are based on the methods published by Benjamini and Speed. The tool first determines how many regions the specific reference genome contains for each amount of GC content, i.e. how many regions in the genome have 50% GC (or 10% GC or 90% GC or...). For this, it samples a large number of equally sized genome bins and counts how many times we see a bin with 50% GC (or 10% GC or 90% or...). These **expected values are independent of any sequencing, but they do depend on the respective reference genome**. This means, that the expected values will, of course, differ between mouse and fruit fly due to their genome's different GC contents, but it also means that strong biases in the reference genome assembly might lead to a false positive diagnosis of GC bias.

After the expected values, the tool samples the BAM file of sequenced reads. Instead of noting how many

genomic regions there are per GC content, we now count the **reads per GC content**.

Output files

- **Diagnostic plot**
  - box plot of absolute read numbers per genomic GC bin
  - x-y plot of observed/expected read ratios per genomic GC content bin
- **Data matrix**
  - to be used for GC correction with *correctGCbias*

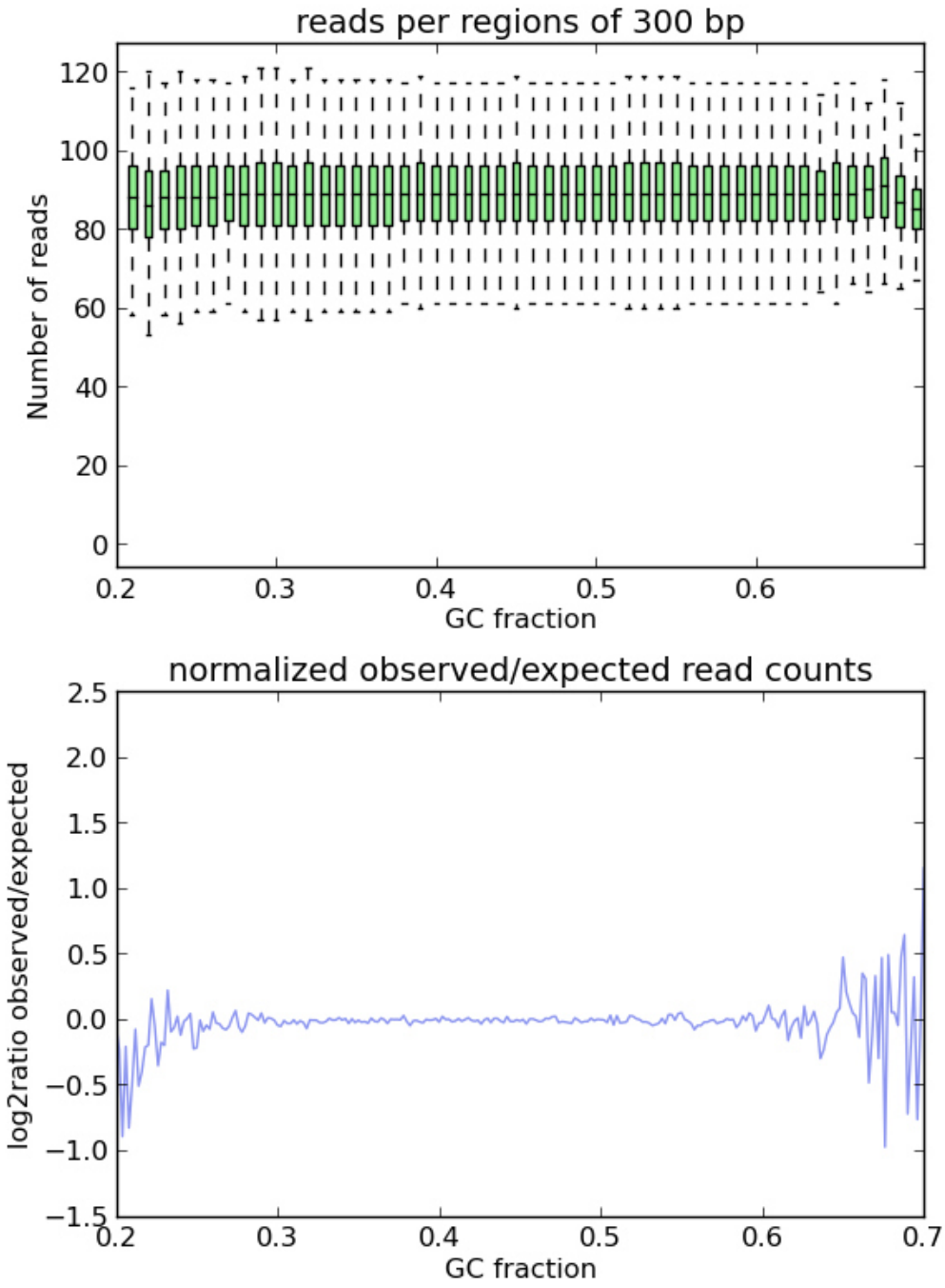
What the plots tell you

In an ideal sample without GC bias, the ratio of observed/expected values should be close to 1 for all GC content bins.

However, due to PCR (over)amplifications, the majority of ChIP samples usually shows a significant bias towards reads with high GC content (>50%) and a depletion of reads from GC-poor regions.

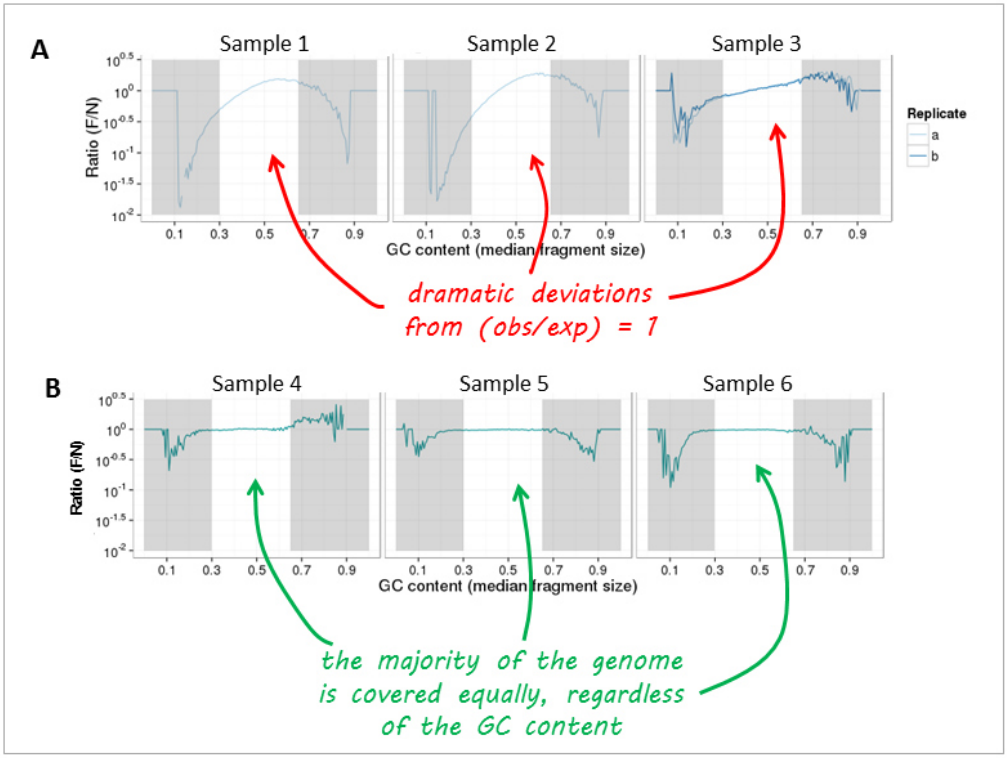
Example figures

Let's start with an ideal case. The following plots were generate with computeGCbias using simulated reads from the Drosophila genome.



As you can see, both plots do not show enrichments or depletions for specific GC content bins.

Now, let's have a look at real-life data from genomic DNA sequencing. Panels A and B can be clearly distinguished and the major change that took place between the experiments underlying the plots was that the samples in panel A were prepared with too many PCR cycles and a standard polymerase whereas the samples of panel B were subjected to very few rounds of amplification using a high fidelity DNA polymerase.



bamFingerprint

What it does

This tool is based on a method developed by Diaz et al.. For factors that will enrich well-defined, rather narrow regions (e.g. transcription factors such as p300), the resulting plot can be used to assess the strength of a ChIP, i.e. whether the signal of the enrichment can be clearly distinguished from the background.

The tool first samples indexed BAM files and counts all reads overlapping a window (bin) of specified length. These counts are then sorted according to their rank and the cumulative sum of read counts is plotted.

Output files:

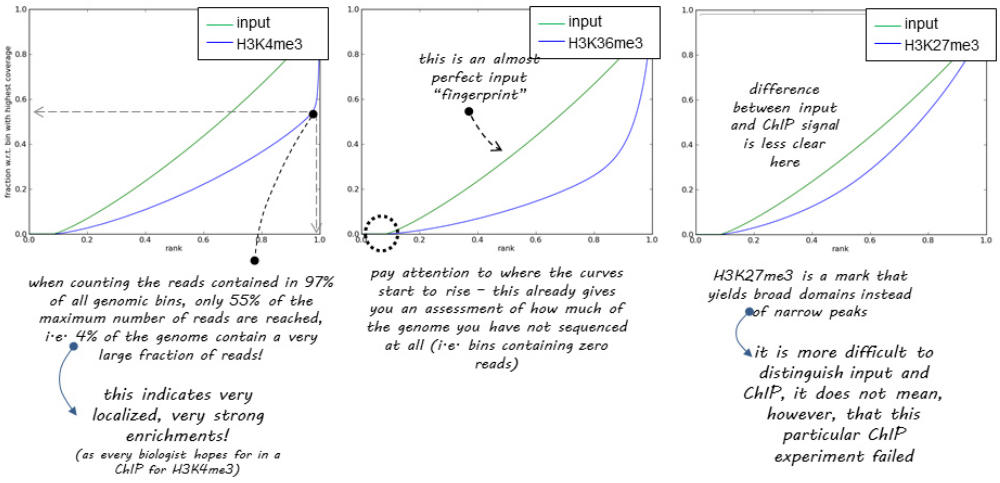
- **Diagnostic plot**
- **Data matrix** of raw counts (optional)

What the plots tell you

An ideal input with perfect uniform distribution of reads along the genome (i.e. without enrichments in open chromatin etc.) should generate a straight diagonal line. A very specific and strong ChIP enrichment will be indicated by a prominent and steep rise of the cumulative sum towards the highest rank. This means that a big chunk of reads from the ChIP sample is located in few bins which corresponds to high, narrow enrichments seen for transcription factors.

Example figures

Here you see three different fingerprint plots that we routinely generate to check the outcome of ChIP-seq experiments. We chose these examples to show you how the nature of the ChIP signal (narrow and high vs. wide and not extremely high) is reflected in the "fingerprint" plots. Please note that for reasons we ourselves cannot recall anymore, these plots go by the name of "fingerprints" in our facility, but the idea underlying these plots came from Diaz et al.



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