Frequently asked questions

How can I do...?

- I have downloaded/received a BAM file - how do I generate a file I can look at in a Genome Browser?
- How can I assess the reproducibility of my sequencing replicates?
- How do I know whether my sample is GC biased? And if yes, how do I correct for it?
- How do I get an input-normalized ChIP-seq coverage file?
- How can I compare the ChIP strength for different ChIP experiments?
- How do I get a (clustered) heatmap of sequencing-depth-normalized read coverages around the transcription start site of all genes?
- How can I compare the average signal for X- and autosomal genes for 2 or more different sequencing experiments

Galaxy-specific questions

- I've reached my quota - what can I do to save some space?
- How can I use a published workflow?
- What is the best way to integrate the deepTools results with other downstream analyses (outside of Galaxy)?
- How can I determine basic parameters of a BAM file?

General deepTools-related questions

- I just want to try out a tool, how can I optimize the computation time?
- When should I exclude regions from computeGCbias?
- Does it speed up the computation if I limit bamCorrelate to one chromosome, but keep the same numbers and sizes of sampling bins?
- Copying from one history to another doesn't work for me - the data set simply doesn't show up in the target history!

Heatmapper

- How can I increase the resolution of the heatmap?
- How can I change the automatic labels of the clusters in a kmeans clustered heatmap?

External data

- How do I calculate the effective genome size for an organism that's not in your list?

How can I do...?

This section is meant to give you quick guidance to specific tasks you may want to perform. We're using screenshots from Galaxy here, if you're using the command-line version, you can easily follow the given examples by typing the program name and the help option (e.g. /deepTools/bin/bamCoverage --help) which will show you all the parameters and options, most of them named very similarly to those in Galaxy.

For each "recipe" here, you will find the screenshot of the tool and the input parameters on the left hand side(circles mark non-default, *user-specified entries*) and screenshots of the output on the right hand side. Do let us know if you spot things that are missing, should be explained better or are plain confusing!

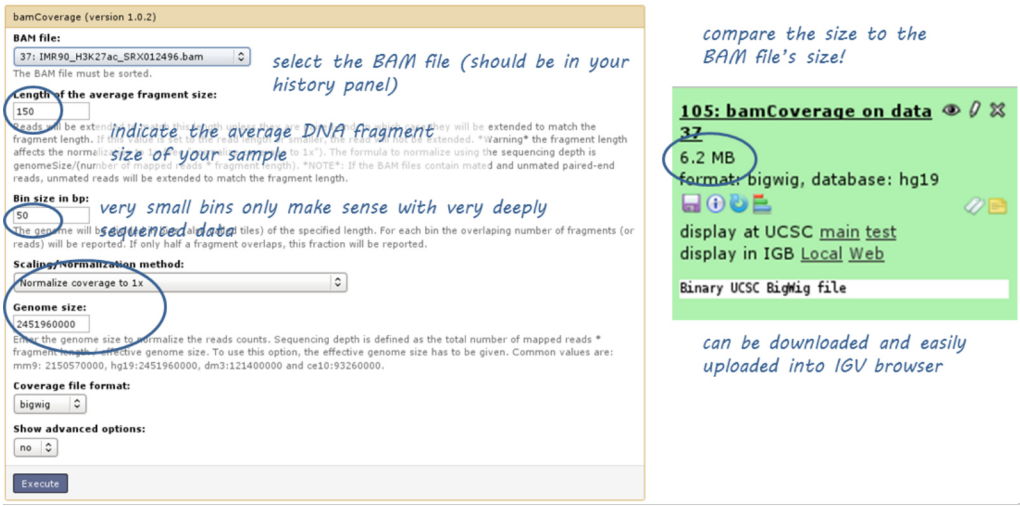
There are many more ways in which you can use deepTools Galaxy than those described here, so be creative once you're comfortable with using them. For detailed explanations of what the tools do, follow the links.

All recipes assume that you're working on the deepTools Galaxy and have uploaded your files.

I have downloaded/received a BAM file - how do I generate a file I can look at in a Genome Browser?

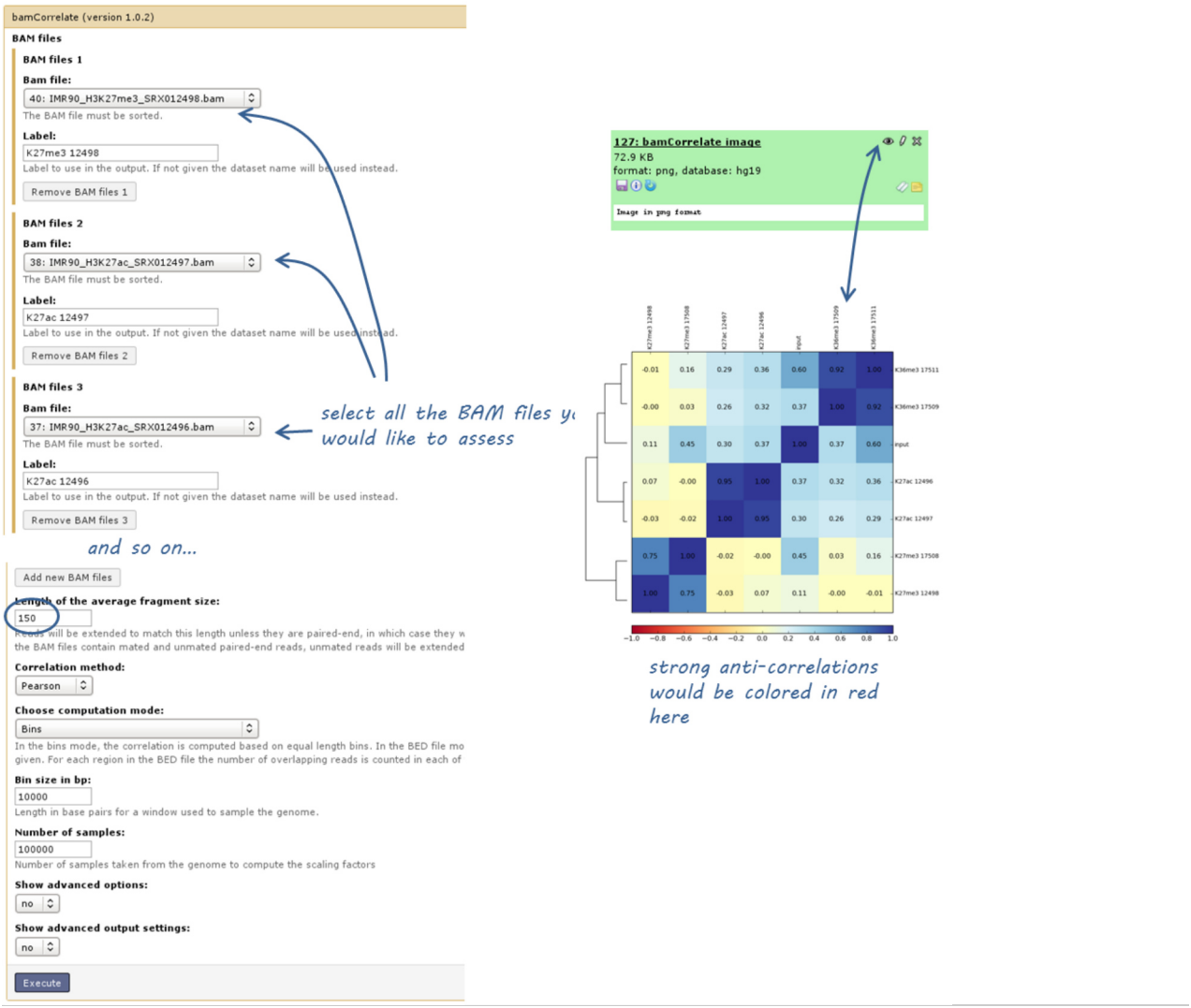
- tool: bamCoverage
- input: your BAM file

Note: BAM files can also be viewed in Genome Browsers, however, they're large and tend to freeze the applications. Generating bigWig files of read coverages will help you a lot in this regard. In addition, if you have more than one sample you'd like to look at, it is helpful to normalize all of them to 1x sequencing depth.



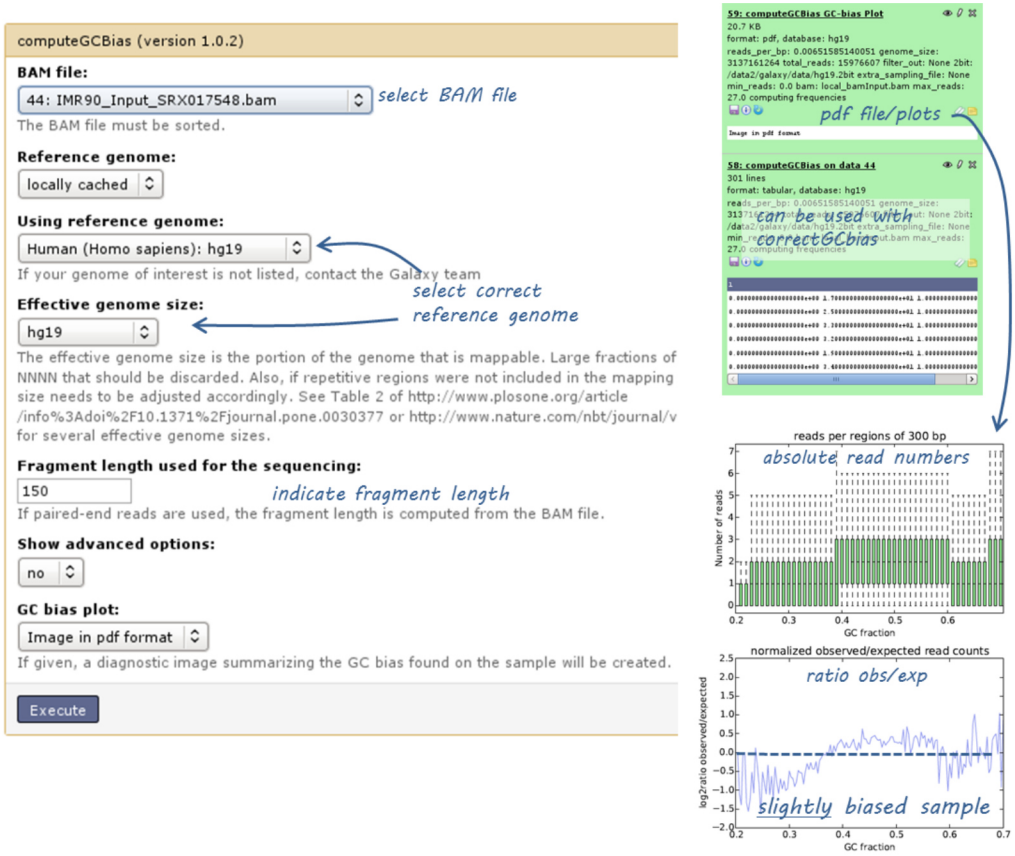
How can I assess the reproducibility of my sequencing replicates?

- tool: bamCorrelate
- input: BAM files
  - you can compare as many samples as you want - the more you put at the same time, the longer the computation takes
- output: heatmap of correlations - the closer two samples are to each other, the more similar their read coverages

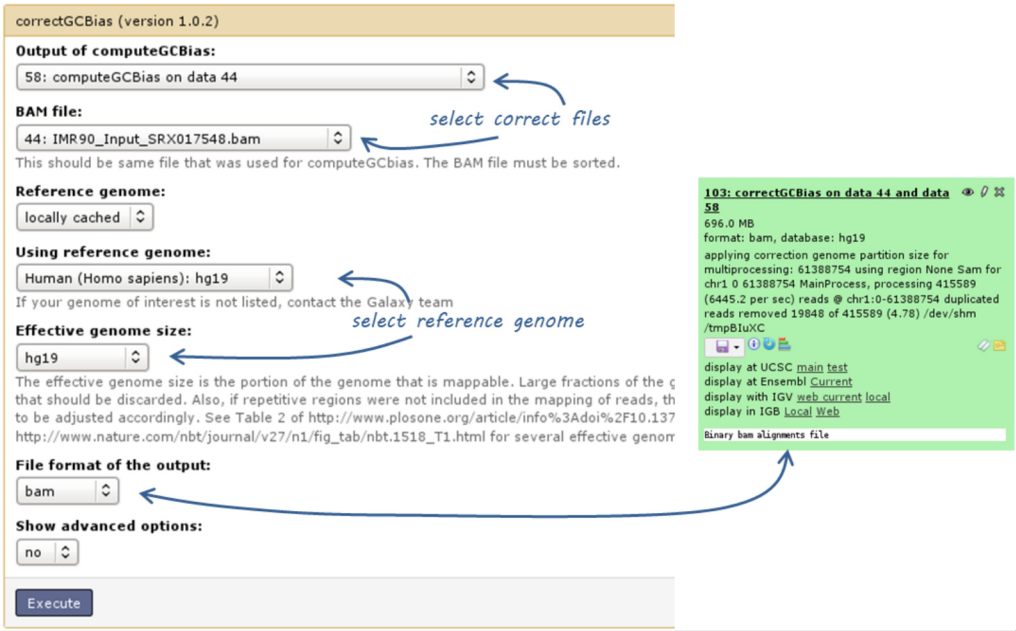


How do I know whether my sample is GC biased? And if yes, how do I correct for it?

- you need a BAM file of your sample in question
- use the tool computeGCbias on that BAM file (default settings, just make sure your reference genome and genome size are matching)

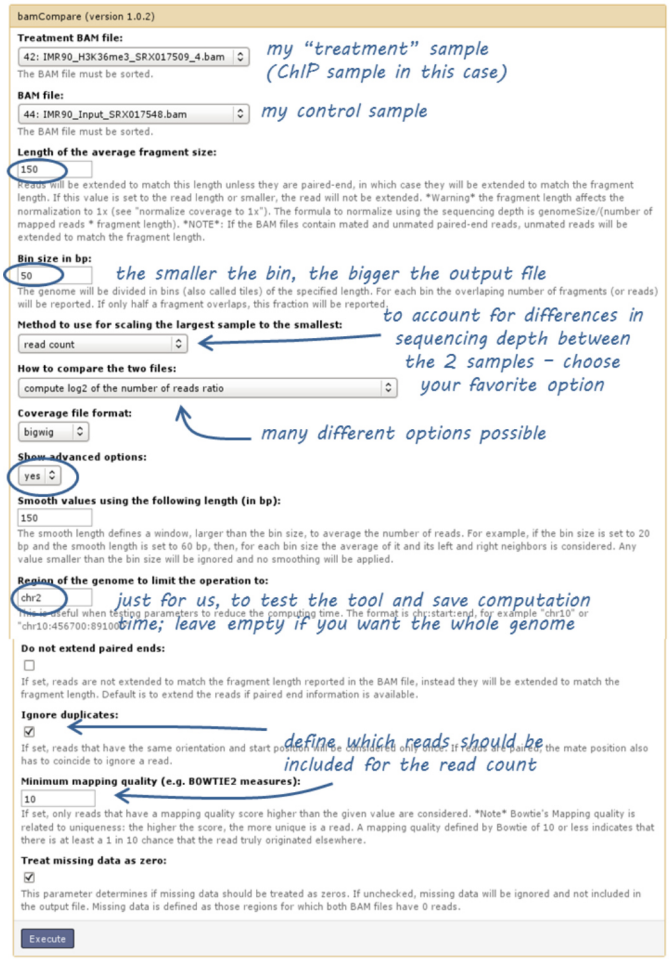


- have a look at the image that is produced and compare it to the examples here
- if your sample shows an almost linear increase in exp/obs coverage (on the log scale of the lower plot), then you should consider correcting the GC bias - *if* you think that the biological interpretation of this data would otherwise be compromised (e.g. by comparing it to another sample that does not have an inherent GC bias)
    - the GC bias can be corrected with the tool correctGCbias using the second output of the computeGCbias tool that you had to run anyway
    - CAUTION!! correctGCbias will add reads to otherwise depleted regions (typically GC-poor regions), that means that you should **not** remove duplicates in any downstream analyses based on the GC-corrected BAM file (we therefore recommend to remove duplicates before doing the correction so that only those duplicate reads are kept that were produced by the GC correction procedure)

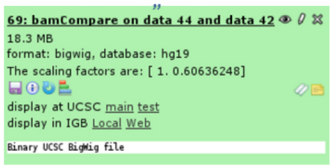


How do I get an input-normalized ChIP-seq coverage file?

1. you need two BAM files: one for the input, one for the ChIP-seq experiment
2. use the tool bamCompare with ChIP = treatment, input = control sample



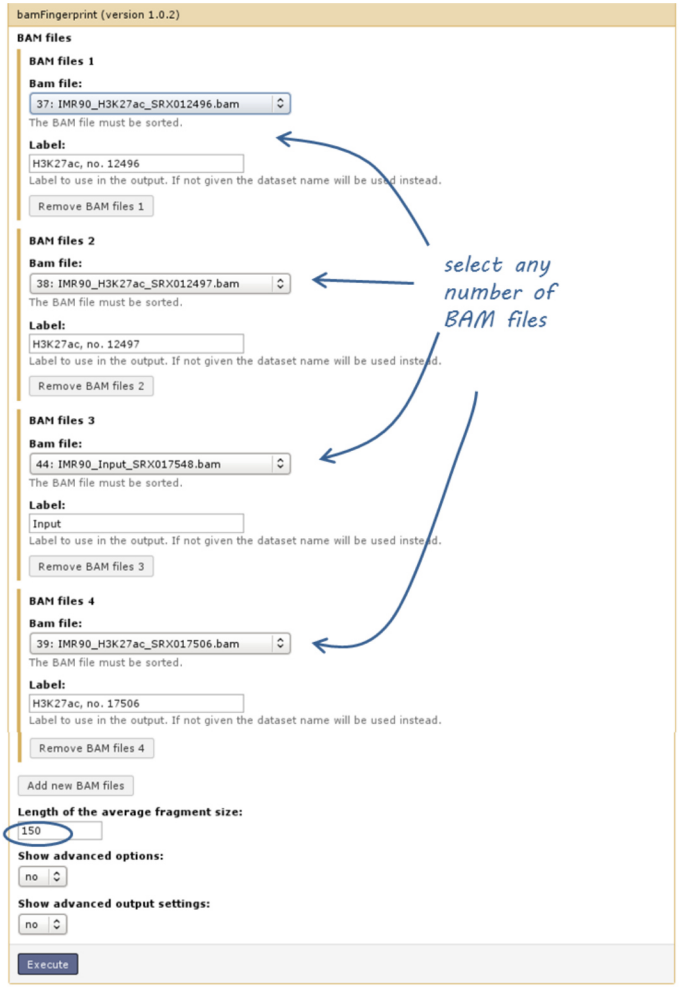
should give meaningful name, e.g. "log2ratio_ChIP_input_H3K27ac.bw"

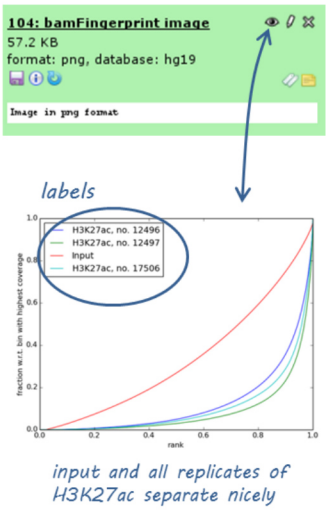


can be uploaded to UCSC or IGV browsers

How can I compare the ChIP strength for different ChIP experiments?

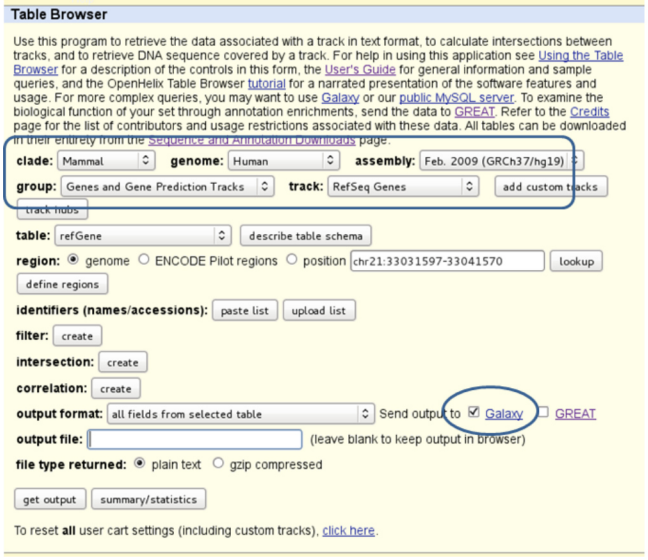
- tool: bamFingerprint
- input: as many BAM files as you'd like to compare. Make sure you get all the labels right!

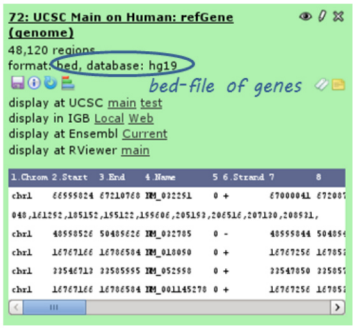




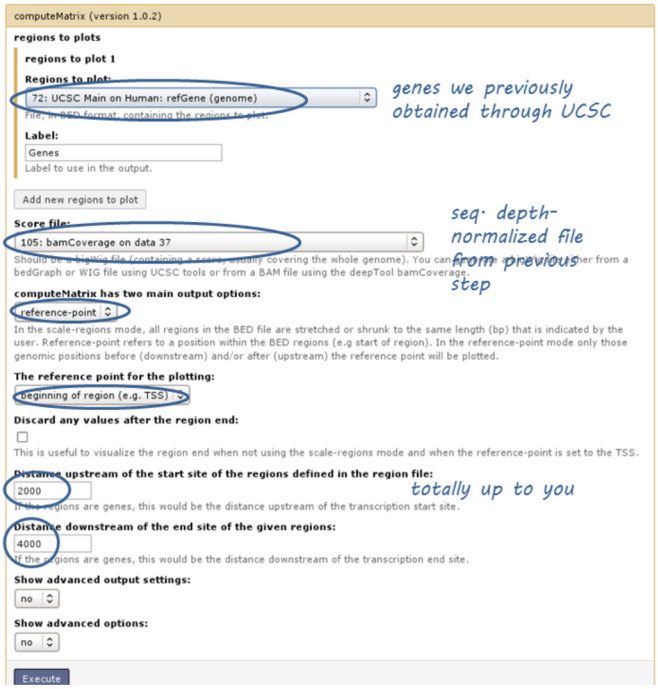
How do I get a (clustered) heatmap of sequencing-depth-normalized read coverages around the transcription start site of all genes?

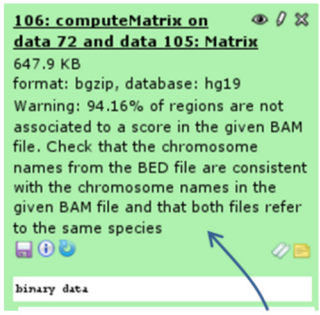
- if you want to start with a BAM file, begin by *generating the normalized read coverages* using the tool bamCoverage with the option "normalize to 1x sequencing depth" (make sure that you indicate the correct genome size) (1)
- you also need a BED or INTERVAL file of genes (you can obtain one via "Get Data" → "UCSC main table browser" → group: "Genes and Gene Predictions" → (e.g.) "RefSeqGenes" → send to Galaxy (2)





- use computeMatrix with the coverage file generated in (1) and the BED file from (2), indicate "reference-point" and whatever other option you would like to tune (3)





- use the output from (3) with heatmapper (if you would like to cluster the signals, choose "kmeans clustering" (last option of "advanced options") with a reasonable number of clusters)

heatmapper (version 1.0.2)

Matrix file from the computeMatrix tool:
106: computeMatrix on data 72 and data 105: Matrix

Show advanced output settings:
no

Show advanced options:
yes

Sort regions:
descending order
Whether the heatmap should present the regions sorted. The default is to sort in descending order based on the mean value per region.

Method used for sorting:
mean
For each row the method is computed.

Type of statistic that should be plotted in the summary image above the heatmap:
mean

Missing data color:
black
If 'Represent missing data as zero' is not set, such cases will be colored in black by default. By using this parameter a different color can be set. A value between 0 and 1 will be used for a gray scale (black is 0). Also color names can be used, see a list here: http://packages.python.org/ete2/reference/reference_svgcolors.html. Alternatively colors can be specified using the #rrggbb notation.

Color map to use for the heatmap:
winter reversed
Available color map names can be found here: http://www.astro.lsa.umich.edu/~msshin/science/code/matplotlib_cm/

Minimum value for the heatmap intensities. Leave empty for automatic values:

Maximum value for the heatmap intensities. Leave empty for automatic values:

Minimum value for the Y-axis of the summary plot. Leave empty for automatic values:

Maximum value for Y-axis of the summary plot. Leave empty for automatic values.

Description for the x-axis label:
distance from TSS (bp)

Description for the y-axis label for the top panel:
genes

Heatmap width in cm:
7.5
The minimum value is 1 and the maximum is 100.

Heatmap height in cm:
15.0
The minimum value is 3 and the maximum is 100.

What to show:
summary plot, heatmap and colorbar
The default is to include a summary or profile plot on top of the heatmap and a heatmap colorbar.

Label for the region start:
TSS
[only for scale-regions mode] Label shown in the plot for the start of the region. Default is TSS (transcription start site), but could be changed to anything, e.g. "peak start".

Label for the region end:
TES
[only for scale-regions mode] Label shown in the plot for the region end. Default is TES (transcription end site).

Reference point label:
TSS
[only for scale-regions mode] Label shown in the plot for the reference-point. Default is the same as the reference point selected (e.g. TSS), but could be anything, e.g. "peak start" etc.

Labels for the regions plotted in the heatmap:
genes
If more than one region is being plotted a list of labels separated by comma and limited by quotes, is required. For example, "label1, label2".

Title of the plot:
My clustered heatmap
Title of the plot, to be printed on top of the generated image. Leave blank for no title.

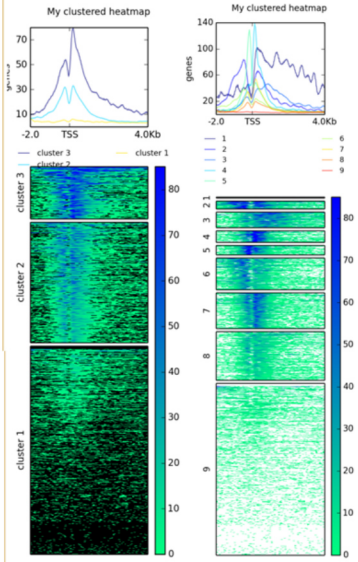
Do one plot per group:
When the region file contains groups separated by "#", the default is to plot the averages for the distinct plots in one plot. If this option is set, each group will get its own plot, stacked on top of each other.

Clustering algorithm:
Kmeans clustering

Number of clusters to compute:
3
When this option is set, then the matrix is split into clusters using the kmeans algorithm. Only works for data that is not grouped, otherwise only the first group will be clustered. If more specific clustering methods are required it is advisable to save the underlying matrix and run the clustering using other software. The plotting of the clustering may fail (Error: Segmentation fault) if a cluster has very few members compared to the total number or regions. (default: None).

Execute

117: heatmapper image
232.4 KB
format: png, database: hg19
Image in png format



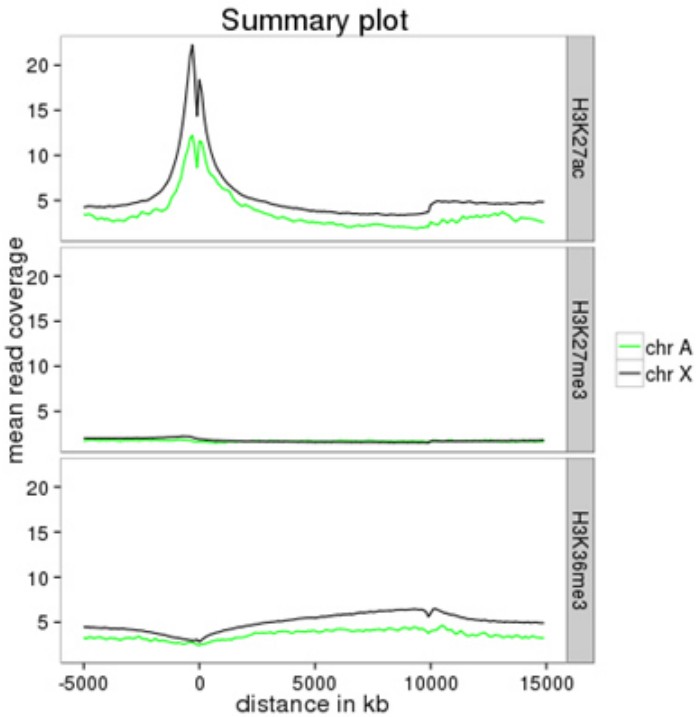
only two entries differed in heatmapper between these two plots:
- **color for missing data** (black vs. white)
- 3 vs. 9 **clusters**

note how the clustering can group genes with down- and upstream enrichments in addition to strong and weak signals

How can I compare the average signal for X- and autosomal genes for 2 or more different sequencing experiments?

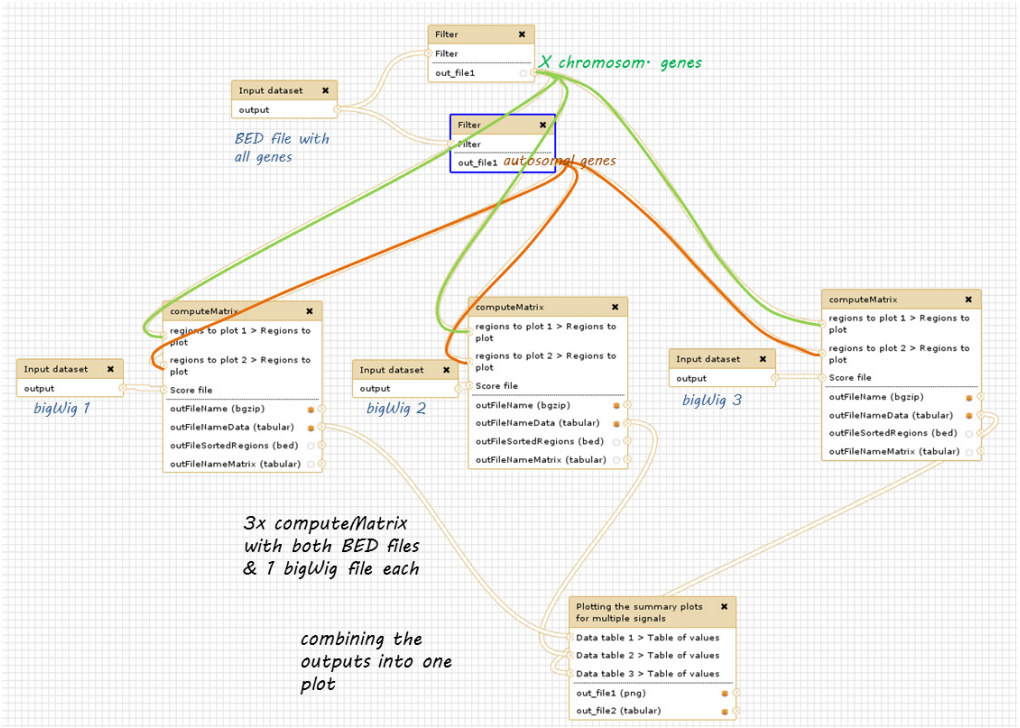
- you need two **BED files**: one with X-chromosomal and one with autosomal genes (1)
  - you can download a full list of genes via "Get Data" → "UCSC main table browser" → group:"Genes and Gene Predictions" → tracks: (e.g.) "RefSeqGenes" → send to Galaxy
  - then filter the full list twice using the tool "Filter data on any column using simple expressions"
    - first use the expression: c1=="chrX" to filter the list of all genes → this will generate a list of X-linked genes
    - then re-run the filtering, now with c1!="chrX" which will generate a list of genes that do not belong to chromosome X (!= indicates "not matching")
- you need **bigWig files** for each experiment (in case you only have BAM files, run bamCoverage on every BAM file first) (2)
- use computeMatrix for each signal file (bigWig) (you only need to specify all the parameters once, then use the re-run button underneath the first data set and just replace the signal file with the next one) (3)
  - supply both filtered BED files (click on "Add new regions to plot" once) and label them
  - indicate the corresponding signal file
  - make sure to **re-name** every data set in the history once computeMatrix is done so that you can easily keep track of which matrix was based on which bigWig file (you can always find these information by clicking on the i-button in the respective data set)
- now use profiler for every file you generated with computeMatrix (4)
  - important: display the "advanced output options" and select "save the data underlying the average profile" → this will generate a table in addition to the summary plot images
- now you have at least 2 separate images of profiles - one for each bigWig file - you can either leave it like this or use another script that will plot all the summary plots in one image at once (5)
  - this tool is called "Plotting the summary plots for multiple signals"
  - it uses the tables generated by profiler in (4)
  - for each group of genes (in this case, X and autosomal genes = 2 groups), you can assign a color

The result could look like this:



As you have noticed, this task requires several steps that are repeated. Here is a screenshot of how the Galaxy workflow would look like (you can find it under "Shared Data" → "Published Workflows" → "Summary plots for X and autosomal genes" where we have constructed it with the example histone marks from the Data Library. Be aware that running this workflow will take up quite some computation timing, but it won't require much input from your part - so start if before you go off for lunch ;))

If you're not sure how to use the published workflow, please read this entry.



Galaxy-specific questions

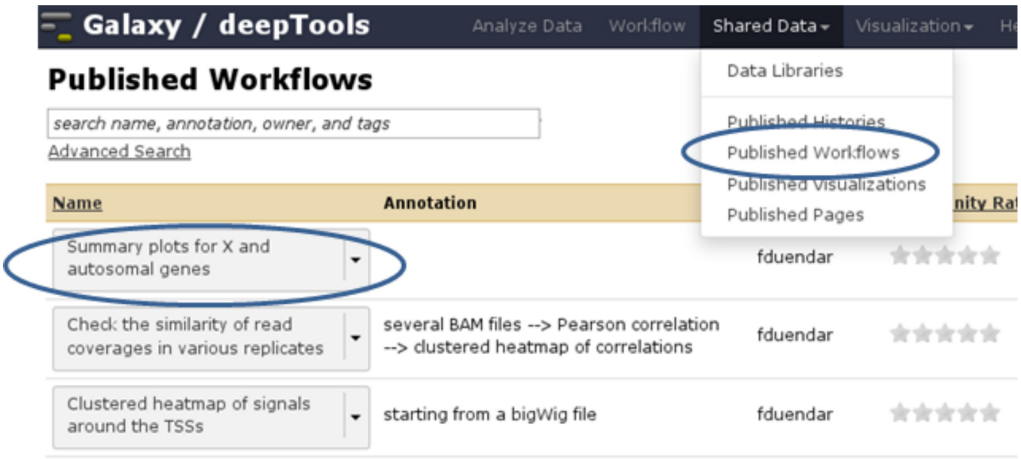
I've reached my quota - what can I do to save some space?

1. make sure that all the data sets you deleted are **permanently** eliminated from our disks: go to the history option button and select "Purge deleted data sets", then hit the "refresh" button on top of your history panel
2. download all data sets for which you've completed the analysis, then remove the data sets (click on the "x" and then make sure they're purged (see above)

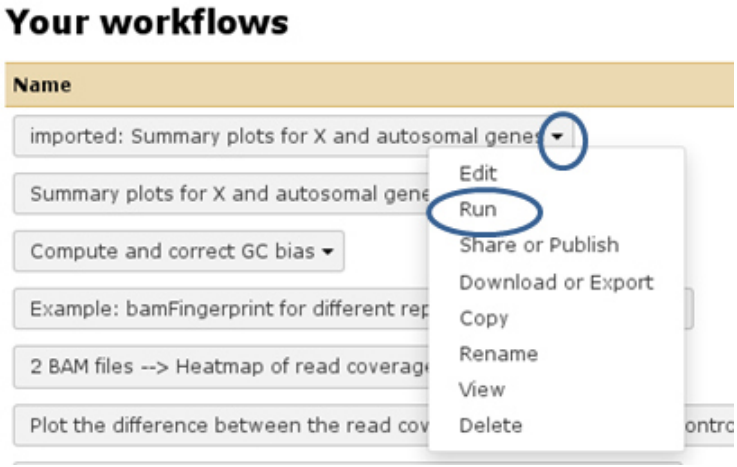
How can I use a published workflow?

You **must register** if you want to use the workflows within deepTools Galaxy. ("User" → "Register" - all you have to supply is an email address)

You can find workflows that are public or specifically shared with you by another user via "Shared Data" → "Published Workflows". Click on the triangle next to the workflow you're interested in and select "import".



A green box should appear, there you select "start using this workflow" which should lead you to your own workflow menu (that you can always access via the top menu "Workflow"). Here, you should now see a workflow labeled "imported: ....". If you want to use the workflow right away, click on the triangle and select "Run". The workflow should now be available within the Galaxy main data frame and should be waiting for your input.



What's the best way to integrate the deepTools results with other downstream analyses (outside of Galaxy)

- you can save all the data tables underlying every image produced by deepTools, i.e. if you would like to plot the average profiles in a different way, you could download the corresponding data (after ticking the profiler option at "advanced output options") and import them into R, Excel, GraphPadPrism etc.

How can I determine basic parameters of a BAM file, such as the number of reads, read length, duplication rate and average DNA fragment length?

Eventhough MACS is meant to do peak calling for you, it also outputs a number of useful information such as those listed above. Simply run MACS on the BAM file that you would like to gain the information for and check the .xls file from the MACS output. It will list:

- tag length = read length
- duplication rate
- number of tags = number of reads
- d = distance = average DNA fragment size

General deepTools-related questions

How can I test a tool with little computation time?

- when you're playing around with the tools to see what kinds of results they will produce: limit the operation to one chromosome only to **save computation time**! ("advanced output options" → "Region of the genome to limit the operation to")

When should I exclude regions from computeGCbias?

In general, we recommend that you should only correct for GC bias (using computeGCbias followed by correctGCbias) if you observe that the majority of the genome (the region between 30-60%) is continuously GC-biased **and** you want to compare this sample with another sample that is not GC-biased.

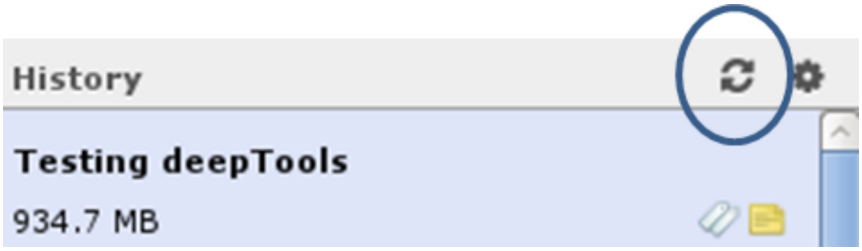
Sometimes, a certain GC bias is expected, for example for ChIP samples of H3K4me3 in mammalian samples where GC-rich promoters are expected to be enriched. To not confound the GC bias caused by the library preparation with the inherent, expected GC bias, we incorporated the possibility to supply a file of regions to computeGCbias that will be excluded from the GC bias calculation. This file should typically contain those regions that one expects to be significantly enriched per se. This way, the computeGCbias will focus on background regions.

Does it speed up the computation if I limit bamCorrelate to one chromosome, but keep the same numbers and sizes of sampling bins?

Yes. However, the way bamCorrelate (and all the other deepTools handle the option "limit the computation to a specific region" is as follows: first, the *entire* genome represented in the BAM file will be regarded and sampled, *then* all the regions or sampled bins that do not overlap with the region indicated by the user will be discarded. This means that if you wanted 10,000 bins to be sampled and you focus on, let's say, chromosome 2, the final computation will not be performed on the whole set of 10,000 bins, but only on those bins that overlap with chromosome 2.

Copying from one history to another doesn't work for me - the data set simply doesn't show up in the target history!

Once you've copied a data set from one history to another, check two things: * do you see the destination history in your history panel, i.e. does the title of the current history panel match the name of the destination history you selected in the main frame? * hit the refresh button



The heatmap I generated looks very "coarse", I would like a much more fine-grained image.

- decrease the **bin size** when generating the matrix using computeMatrix
  - go to "advanced options" → "Length, in base pairs, of the non-overlapping bin for averaging the score over the regions length" → define a smaller value, e.g. 50 or 25 bp
- make sure, however, that you used a sufficiently small bin size when calculating the bigWig file, though (if

generated with deepTools, you can check the option "bin size")

How can I change the automatic labels of the clusters in a kmeans clustered heatmap?

Each cluster will get its own box, exactly the same way as different groups of regions. Therefore, you can use the same option to define the labels of the final heatmap: Heatmapper → "Advanced output options" → "Labels for the regions plotted in the heatmap".

If you indicated 3 clusters for kmeans clustering, enter here: C1, C2, C3 → instead of the full default label ("cluster 1"), the heatmap will be labeled with the abbreviations.

How do I calculate the effective genome size for an organism that's not in your list?

This is something you will have to find a solution outside of deepTools at the moment. We suggest to run faCount from UCSC tools. If you used multi-read alignment (e.g. with bowtie2), then you can use that tool to report the total number of bases as well as the number of unmapped bp, indicated by 'N'. The effective genome size is the total number of reads minus the number of 'N'.