

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?
- * Copying from one history to another doesn't work for me - the data set simply doesn't show up in the target history!
- How can I use a published workflow within deepTools Galaxy?
- I would like to use one of your workflows - not in the deepTools Galaxy, but in the local Galaxy instance provided by my institute. Is that possible?
- 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

- How does deepTools handle data from paired-end sequencing?
- I just want to try out a tool, how can I optimize the computation time?
- Does it speed up the computation if I limit bamCorrelate to one chromosome, but keep the same numbers and sizes of sampling bins?
- When should I exclude regions from computeGCbias?

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?
- Where can I download the 2bit genome files required for computeGCbias?

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 (we marked 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 have uploaded your files into a Galaxy instance with a deepTools installation, e.g. [deepTools Galaxy](#).

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.

bamCoverage (version 1.0.2)

BAM file:
37: IMR90_H3K27ac_SRX012496.bam

The BAM file must be sorted.

Length of the average fragment size:
150

Reads will be extended to this length unless they're mapped to a genomic region which causes them to be extended to match the fragment length. If the reads are mapped to multiple genomic regions, they will be extended to 1x". The formula to normalize using the sequencing depth is genomeSize/(number of mapped reads * fragment length). *NOTE*: If the BAM files contain mated and unmated paired-end reads, unmated reads will be extended to match the fragment length.

Indicate the average DNA fragment size of your sample.

Bin size in bp:
50

very small bins only make sense with very deeply sequenced data

The genome will be divided into bins of the specified length. For each bin the overlapping number of fragments (or reads) will be reported. If only half a fragment overlaps, this fraction will be reported.

Scaling/normalization method:
Normalize coverage to 1x

Genome size:
2451960000

Enter the genome size to normalize the reads counts. Sequencing depth is defined as the total number of mapped reads * fragment length / effective genome size. To use this option, the effective genome size has to be given. Common values are: mm9: 2150570000, hg19:2451960000, dm3:121400000 and ce10:93260000.

Coverage file format:
bigwig

Show advanced options:
no

Execute

compare the size to the BAM file's size!

105: bamCoverage on data

37

6.2 MB

format: bigwig, database: hg19

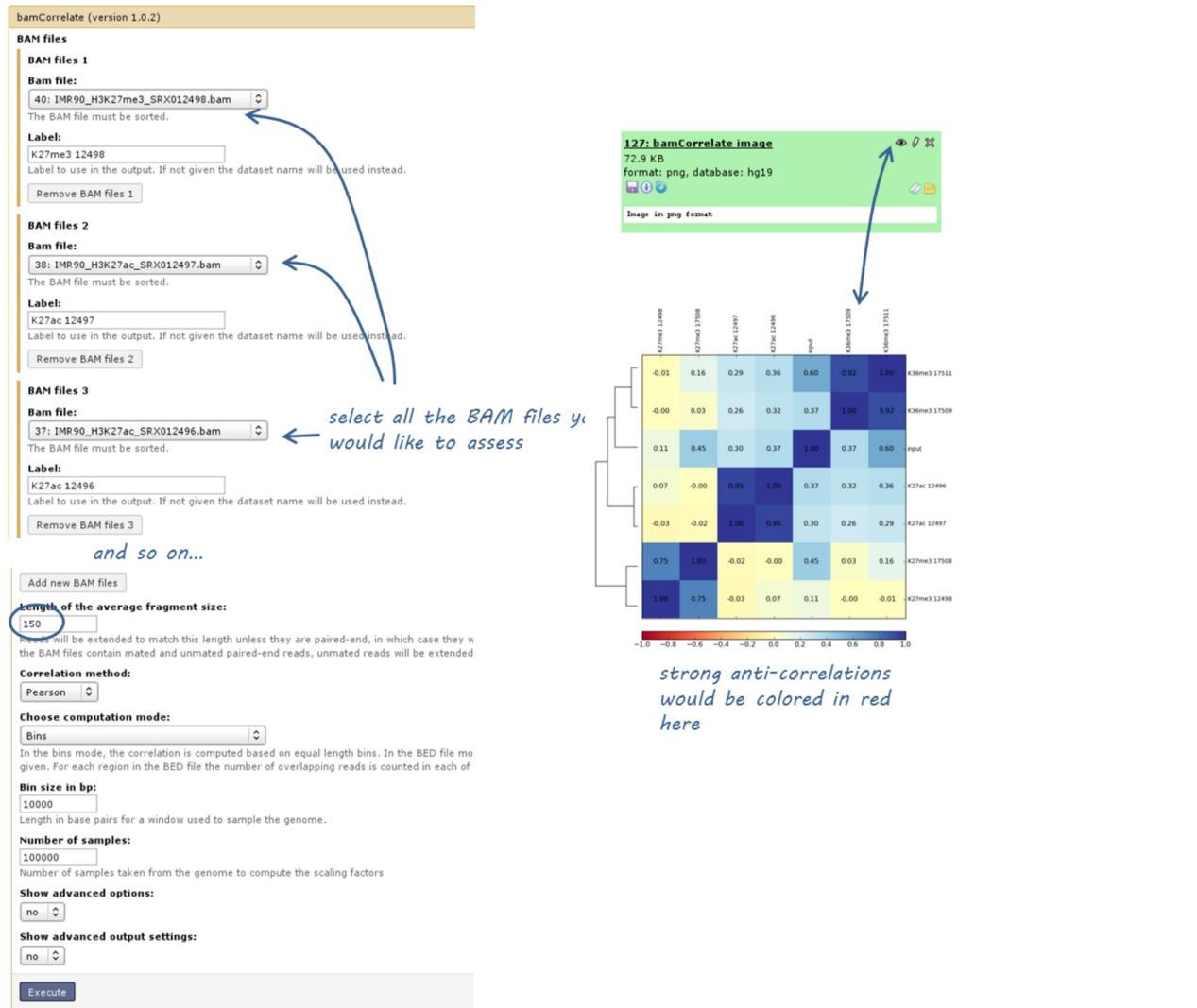
display at UCSC main test
display in IGB Local Web

Binary UCSC BigWig file

can be downloaded and easily uploaded into IGV browser

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)

The screenshot shows the Galaxy web interface for the `computeGCbias` tool. The tool version is 1.0.2. The configuration includes:

- BAM file:** 44: IMR90_Input_SRX017548.bam (selected)
- Reference genome:** locally cached (hg19)
- Using reference genome:** Human (Homo sapiens): hg19 (selected)
- Effective genome size:** hg19 (selected)
- Fragment length used for the sequencing:** 150 (indicated fragment length)
- Show advanced options:** no
- GC bias plot:** Image in pdf format (selected)

The tool's log output shows:

```
S9: computeGCbias (version 1.0.2)
20.7 KB
format: pdf; database: hg19
reads_per_bp: 0.00651585140051 genome_size:
3137161264 total_reads: 15976607 filter_out: None 2bit:
(database: hg19; filter_out: 0.2bit extra_sampling_file: None
min_reads: 0.0 bam: local_baminput.bam max_reads:
2720 computing frequencies
pdf file/plots
```

The output plots include:

- GC-bias Plot:** A histogram showing 'reads per regions of 300 bp' versus 'absolute read numbers' across different GC fractions. The distribution is relatively flat.
- ratio obs/exp:** A line plot showing the 'log2ratio observed/expected' versus 'GC fraction'. The ratio fluctuates around zero, indicating a 'slightly biased sample'.

- 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)

correctGCBias (version 1.0.2)

Output of computeGCBias:

58: computeGCBias on data 44

BAM file:
44: IMR90_Input_SRX017548.bam

This should be same file that was used for computeGCBias. The BAM file must be sorted.

Reference genome:
locally cached

Using reference genome:
Human (Homo sapiens): hg19

If your genome of interest is not listed, contact the Galaxy team

Effective genome size:
hg19

The effective genome size is the portion of the genome that is mappable. Large fractions of the genome should be discarded. Also, if repetitive regions were not included in the mapping of reads, they should be adjusted accordingly. See Table 2 of <http://www.plosone.org/article/info%3Adoi%2F10.1371/journal.pone.0027111> and http://www.nature.com/nbt/journal/v27/n1/fig_tab/nbt.1518_T1.html for several effective genome sizes.

File format of the output:
bam

Show advanced options:
no

Execute

103: correctGCBias on data 44 and data 58

format: bam, database: hg19
 applying correction genome partition size for
 multiprocessing: 61388754 using region None Sam for
 chr1 0 61388754 MainProcess, processing 415589
 (6445.2 per sec) reads @ chr1:0-61388754 duplicated
 reads removed 19848 of 415589 (4.78) /dev/shm
 /tmpBluXC

display at UCSC main test
 display at Ensembl Current
 display with IGV web current local
 display in IGB Local Web

Binary bam alignments file

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

bamCompare (version 1.0.2)

Treatment BAM file:
42: IMR90_H3K36me3_SRX017509_4.bam (my "treatment" sample
(ChIP sample in this case)
The BAM file must be sorted.

BAM file:
44: IMR90_Input_SRX017548.bam (my control sample)
The BAM file must be sorted.

Length of the average fragment size:
150

Reads will be extended to match this length unless they are paired-end, in which case they will be extended to match the fragment length. If this value is set to the read length or smaller, the read will not be extended. *Warning* the fragment length affects the normalization to 1x (see "normalize coverage to 1x"). The formula to normalize using the sequencing depth is genomeSize/(number of mapped reads * fragment length). *NOTE*: If the BAM files contain mated and unmated paired-end reads, unmated reads will be extended to match the fragment length.

Bin size in bp:
50 the smaller the bin, the bigger the output file
The genome will be divided into bins (also called tiles) of the specified length. For each bin the overlapping number of fragments (or reads) will be reported. If only half a fragment overlaps, this fraction will be reported.

Method to use for scaling the largest sample to the smallest:
read count to account for differences in sequencing depth between the 2 samples - choose your favorite option

How to compare the two files:
compute log2 of the number of reads ratio

Coverage file format:
bigwig many different options possible

Show advanced options:
yes

Smooth values using the following length (in bp):
150

The smooth length defines a window, larger than the bin size, to average the number of reads. For example, if the bin size is set to 20 bp and the smooth length is set to 60 bp, then, for each bin size the average of it and its left and right neighbors is considered. Any value smaller than the bin size will be ignored and no smoothing will be applied.

Region of the genome to limit the operation to:
chr2 just for us, to test the tool and save computation
This is useful when testing parameters to reduce the computing time. The format is chr:start:end, for example "chr10" or "chr10:456700:891000" time; leave empty if you want the whole genome

Do not extend paired ends:

Ignore duplicates:
 define which reads should be included for the read count

Minimum mapping quality (e.g. BOWTIE2 measures):
10

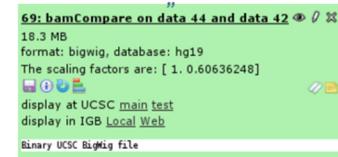
If set, only reads that have a mapping quality score higher than the given value are considered. *Note* Bowtie's Mapping quality is related to uniqueness: the higher the score, the more unique is a read. A mapping quality defined by Bowtie of 10 or less indicates that there is at least a 1 in 10 chance that the read truly originated elsewhere.

Treat missing data as zero:

This parameter determines if missing data should be treated as zeros. If unchecked, missing data will be ignored and not included in the output file. Missing data is defined as those regions for which both BAM files have 0 reads.

Execute

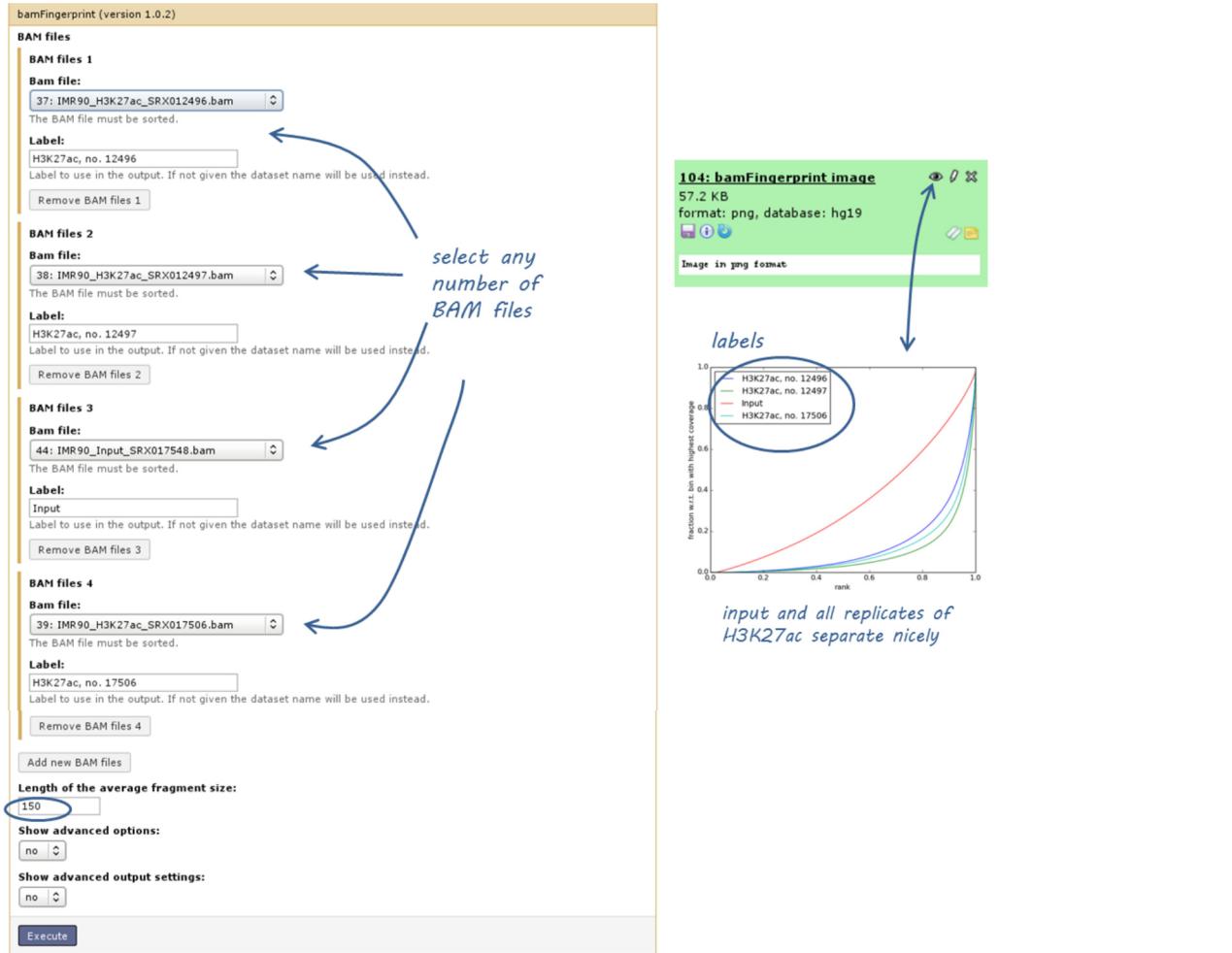
should give meaningful name, e.g.
"log2ratio_ChIP_input_H3K27ac.bw"



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)

Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections between tracks, and to retrieve DNA sequence covered by a track. For help in using this application see [Using the Table Browser](#) for a description of the controls in this form, the [User's Guide](#) for general information and sample queries, and the OpenHelix Table Browser [tutorial](#) for a narrated presentation of the software features and usage. For more complex queries, you may want to use [Galaxy](#) or our [public MySQL server](#). To examine the biological function of your set through annotation enrichments, send the data to [GREAT](#). Refer to the [Credits](#) page for the list of contributors and usage restrictions associated with these data. All tables can be downloaded in their entirety from the [Sequence and Annotation Downloads](#) page.

clade: Mammal genome: Human assembly: Feb. 2009 (GRCh37/hg19)

group: Genes and Gene Prediction Tracks track: RefSeq Genes add custom tracks

track hubs

table: refGene describe table schema

region: genome ENCODE Pilot regions position: chr21:33031597-33041570 lookup
define regions

identifiers (names/accessions): paste list upload list

filter: create

intersection: create

correlation: create

output format: all fields from selected table Send output to Galaxy GREAT

output file: (leave blank to keep output in browser)

file type returned: plain text gzip compressed

get output summary/statistics

To reset all user cart settings (including custom tracks), [click here](#).

Z2: UCSC Main on Human: refGene							
48,120 regions							
format: bed, database: hg19							
display at UCSC main test							
display in IGB Local Web							
display at Ensembl Current							
display at RViewer main							
1	Chrom	2	Start	3	End	4	Name
chr1	65595024	67219769	IM_012251	0	+	67000041	672005
chr1	1612192	185151	185122	185160	185153	204516	205511
chr1	40548526	50485626	IM_012705	0	-	40559844	504855
chr1	16767166	16708564	IM_010050	0	+	16767156	167085
chr1	21546712	21585555	IM_052550	0	+	21547850	215655
chr1	16767166	16708564	IM_081145270	0	+	16767156	167085

- 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)

computeMatrix (version 1.0.2)

regions to plots

regions to plot 1

Regions to plot:

72: UCSC Main on Human: refGene (genome)
Filter by BED format, containing the regions to plot.

genes we previously obtained through UCSC

Label:
Genes
Label to use in the output.

Add new regions to plot

Score file:
105: bamCoverage on 37

seq- depth- normalized file from previous step

computeMatrix has two main output options:
reference-point: beginning of region (e.g. TSS)

In the scale-regions mode, all regions in the BED file are stretched or shrunk to the same length (bp) that is indicated by the user. Reference-point refers to a position within the BED regions (e.g. start of region). In the reference-point mode only those genomic positions before (downstream) and/or after (upstream) the reference point will be plotted.

The reference point for the plotting:
beginning of region (e.g. TSS)

Discard any values after the region end:
This is useful to visualize the region end when not using the scale-regions mode and when the reference-point is set to the TSS.

Distance upstream of the start site of the regions defined in the region file:
2000
If the regions are genes, this would be the distance upstream of the transcription start site.

Distance downstream of the end site of the given regions:
4000
If the regions are genes, this would be the distance downstream of the transcription end site.

Show advanced output settings:
no

Show advanced options:
no

Execute

106: computeMatrix on data 72 and data 105: Matrix							
647.9 KB							
format: bgzip, database: hg19							
Warning: 94.16% of regions are not associated to a score in the given BAM file. Check that the chromosome names from the BED file are consistent with the chromosome names in the given BAM file and that both files refer to the same species							
binary data							

warning is raised because the bamCoverage we used contained reads for chr2 only, while the gene file contained all genes

- 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:

Show advanced options:

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 'allow 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_color.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.iza.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.

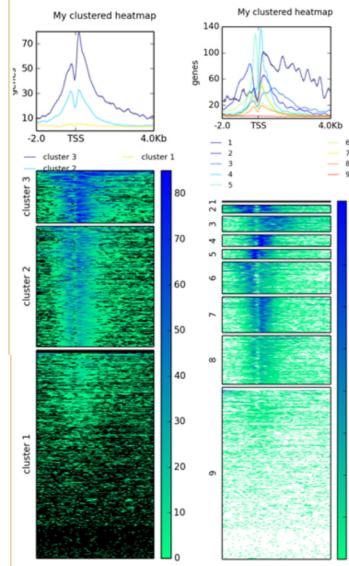
Do one plot per group:

When the region file contains groups separated by "#", the default is to plot the average 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 of regions. (default: None).

Execute



only two entries differed in heatmapper between these two plots:

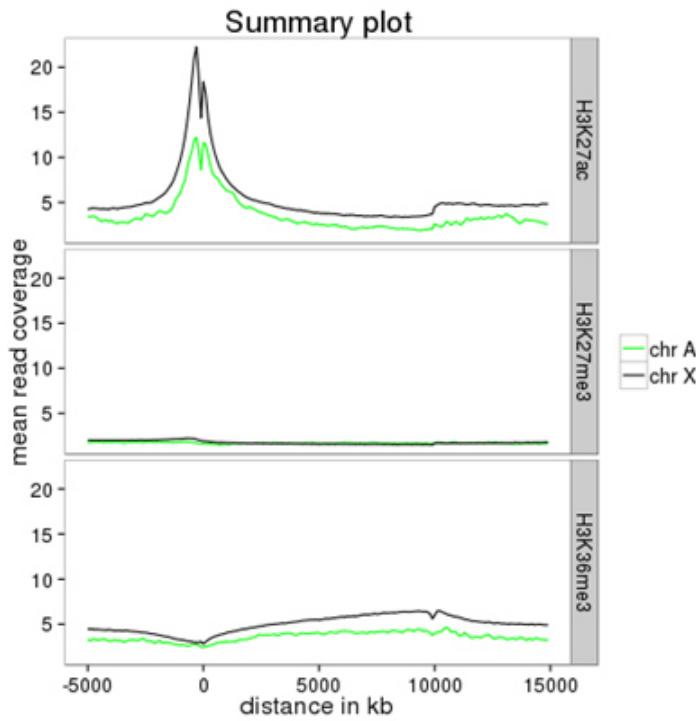
- color for missing data (black vs. white)
- 3 vs. 9 clusters

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?

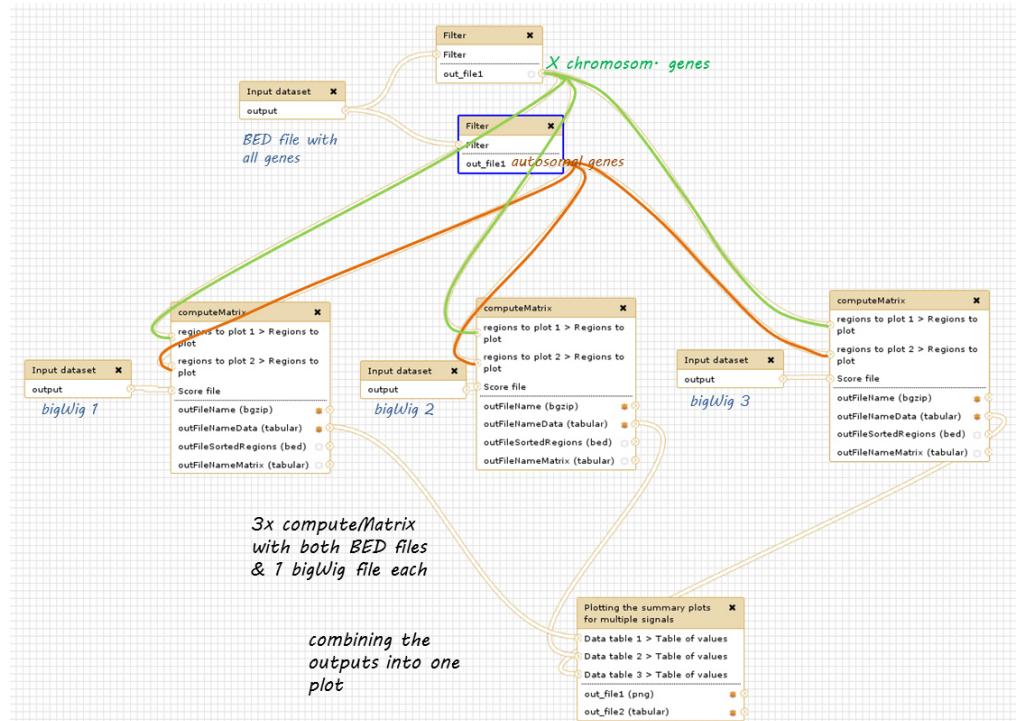
- you need two **BED files**: one with X-chromosomal and one with autosomal genes (1)
 - you can download a full list of genes via "Get Data" → "UCSC main table browser" → group:"Genes and Gene Predictions" → tracks: (e.g.) "RefSeqGenes" → send to Galaxy
 - then filter the full list twice using the tool "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](#) or go to the central [Galaxy learning page](#) full of tutorials.



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)

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



How can I use a published workflow?

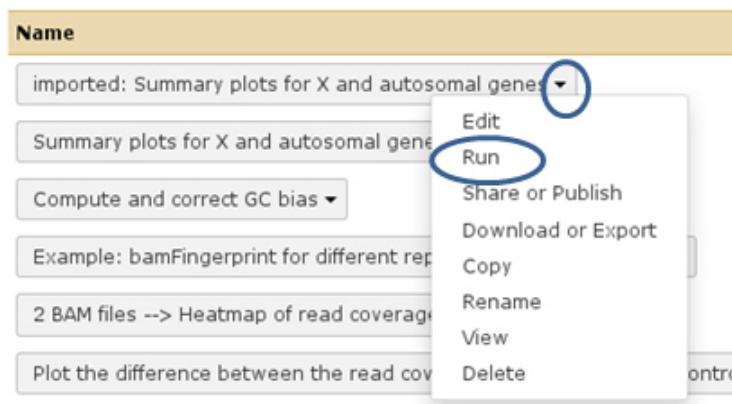
You **must register** if you want to use the workflows within [deepTools Galaxy](#). ("User" → "Register" - all you have to supply is an email address)

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 screenshot of the Galaxy interface. At the top, there's a navigation bar with tabs for "Galaxy / deepTools", "Analyze Data", "Workflow", "Shared Data", "Visualization", and "Help". The "Workflow" tab is active. A dropdown menu is open under "Shared Data", showing options: "Data Libraries", "Published Histories", "Published Workflows" (which is highlighted and circled with a blue line), "Published Visualizations", and "Published Pages". Below this, there's a table titled "Published Workflows". The first row shows a workflow named "Summary plots for X and autosomal genes" by user "fduendar" with a 5-star rating. The second row shows a workflow named "Check: the similarity of read coverages in various replicates" by user "fduendar" with a 5-star rating. The third row shows a workflow named "Clustered heatmap of signals around the TSSs" by user "fduendar" with a 5-star rating. Each workflow entry has a small green box with a triangle icon next to it, indicating it can be imported.

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.

Your workflows



I would like to use one of your workflows - not in the deepTools Galaxy, but in the local Galaxy instance provided by my institute. Is that possible?

Yes, it is possible. The only requirement is that your local Galaxy has a recent installation of deepTools.

Go to the workflows, click on the ones you're interested in and go to "Download". This will save the workflows into .ga files on your computer. Now go to your local Galaxy installation and login. Go to the workflow menu and select "import workflow" (top right hand corner of the page). Click on "Browse" and select the saved workflow. If you have the same tool versions installed in your local Galaxy, these workflows should work right away.

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 does deepTools handle data from paired-end sequencing?

Generally, all the modules working with BAM files (*bamCorrelate*, *bamCoverage*, *bamCompare*, *bamFingerprint*, *computeGCbias*) recognize paired-end sequencing data. You can enforce to ignore the fragment length based on the mate pairs using the option `_doNotExtendPairedEnds`

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")

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.

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.

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'.

Where can I download the 2bit genome files required for *computeGCbias*?

The 2bit files of most genomes can be found [here](#). Search for the .2bit ending. Otherwise, **fasta files can be converted to 2bit** using the UCSC programm faToTwoBit (available for different platforms from [here](#)

deepTools are developed by the Bioinformatics Group at the Max-Planck-Institute of Immunobiology and Epigenetics, Freiburg.