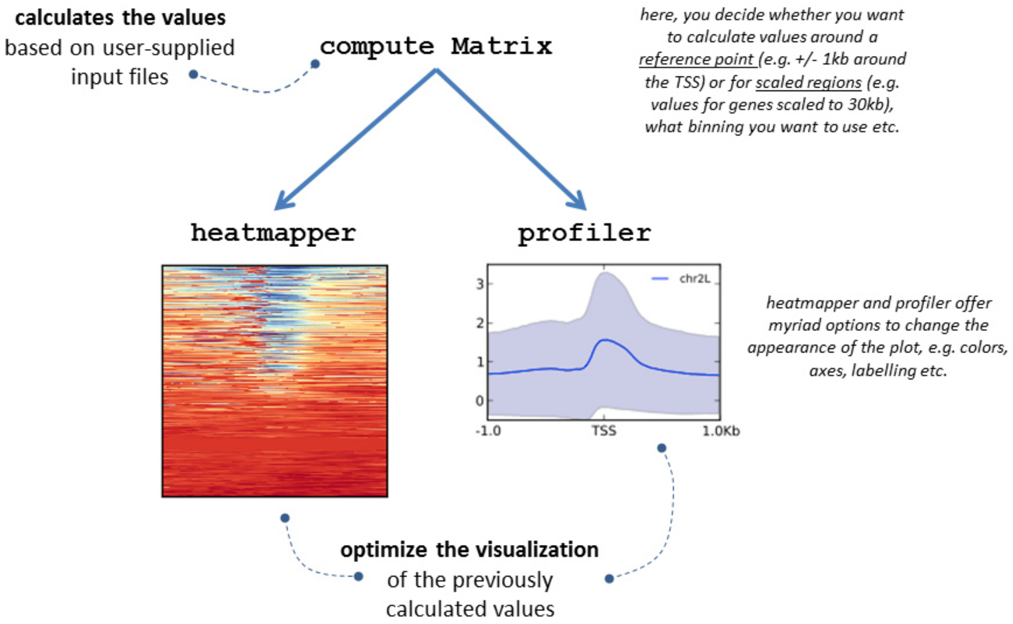
Visualization

The modules for visualizing scores contained in bigWig files are separated into a tool that calculates the values (*computeMatrix*) and two tools that contain many, many options to fine-tune the plot (*heatmapper* and *profiler*).



calculates the values
based on user-supplied input files
compute Matrix
here, you decide whether you want to calculate values around a reference point (e.g. +/- 1kb around the TSS) or for scaled regions (e.g. values for genes scaled to 30kb), what binning you want to use etc.
heatmapper
profiler
chr2L
3
2
1
0
-1.0
TSS
1.0Kb
heatmapper and profiler offer myriad options to change the appearance of the plot, e.g. colors, axes, labelling etc.
optimize the visualization
of the previously calculated values

Table of content

computeMatrix

This tool summarizes and prepares an intermediary file containing scores associated with genomic regions that can be used afterwards to plot a heatmap or a profile.

Genomic regions can really be anything - genes, parts of genes, ChIP-seq peaks, favorite genome regions... as long as you provide a proper file in BED or INTERVAL format. This tool can also be used to filter and sort regions according to their score.

As indicated in the plot above, computeMatrix can be run with either one of the two modes: **scaled regions** or **reference point**.

Please see the example figures down below for explanations of parameters and options.

Output files

- **obligatory**: zipped matrix of values to be used with *heatmapper* and/or *profiler*
- **optional** (can also be generated with heatmapper or profiler in case you forgot to produce them in the beginning):
    - BED-file of the regions sorted according to the calculated values
    - list of average values per genomic bin
    - matrix of values per genomic bin per genomic interval

heatmapper

The heatmapper depicts values extracted from the bigWig file for each genomic region individually. It requires the output from computeMatrix and most of its options are related to tweeking the visualization only. The values calculated by computeMatrix are not changed.

Definitely check the example at the bottom of the page to get a feeling for how many things you can tune.

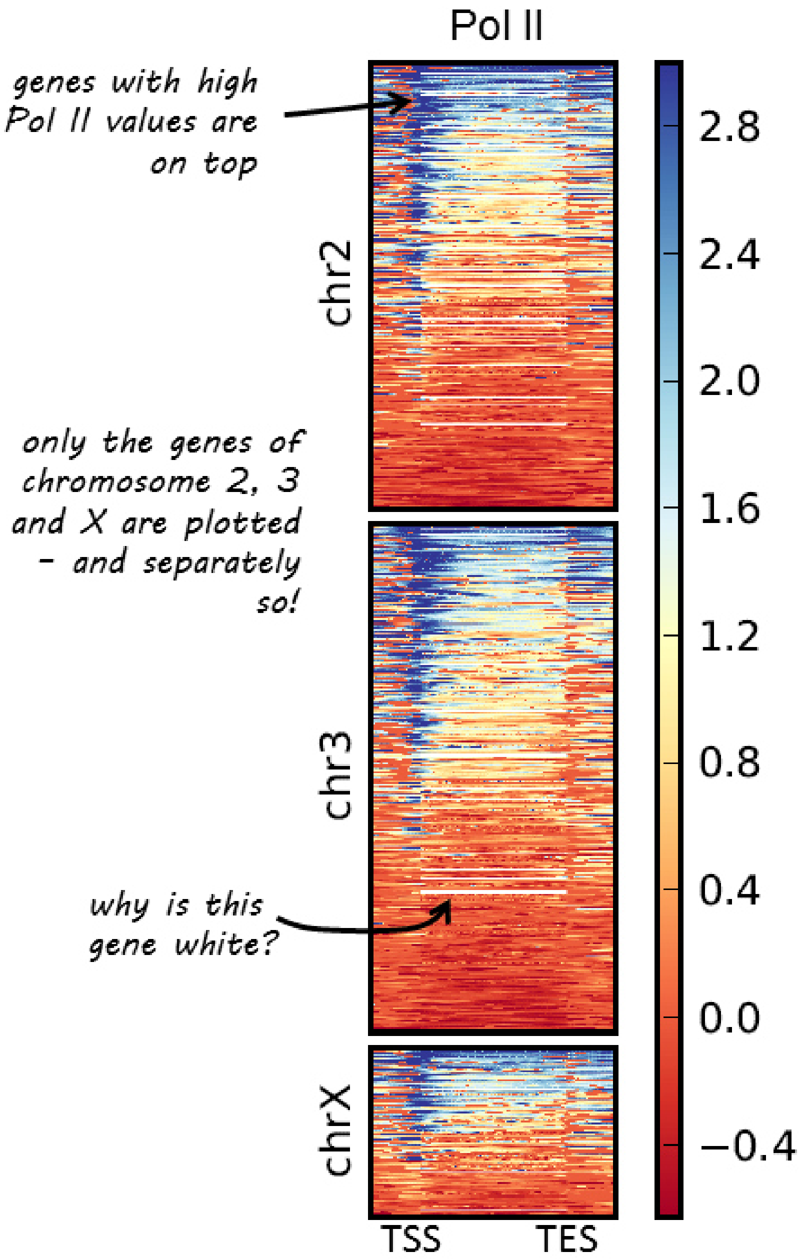
profiler

This tool plots the average enrichments over all genomic regions supplied to computeMarix. It is a very useful complement to the heatmapper, especially in cases when you want to compare the scores for many different groups. Like heatmapper, profiler does not change the values that were compute by computeMatrix, but you can choose between many different ways to color and display the plots.

Example figures

Here you see a typical, not too pretty example of a heatmap. We will use this example to explain several features of computeMatrix and heatmapper, so do take a closer look.

Heatmap with all genes scaled to the one size and user-specified groups of genes



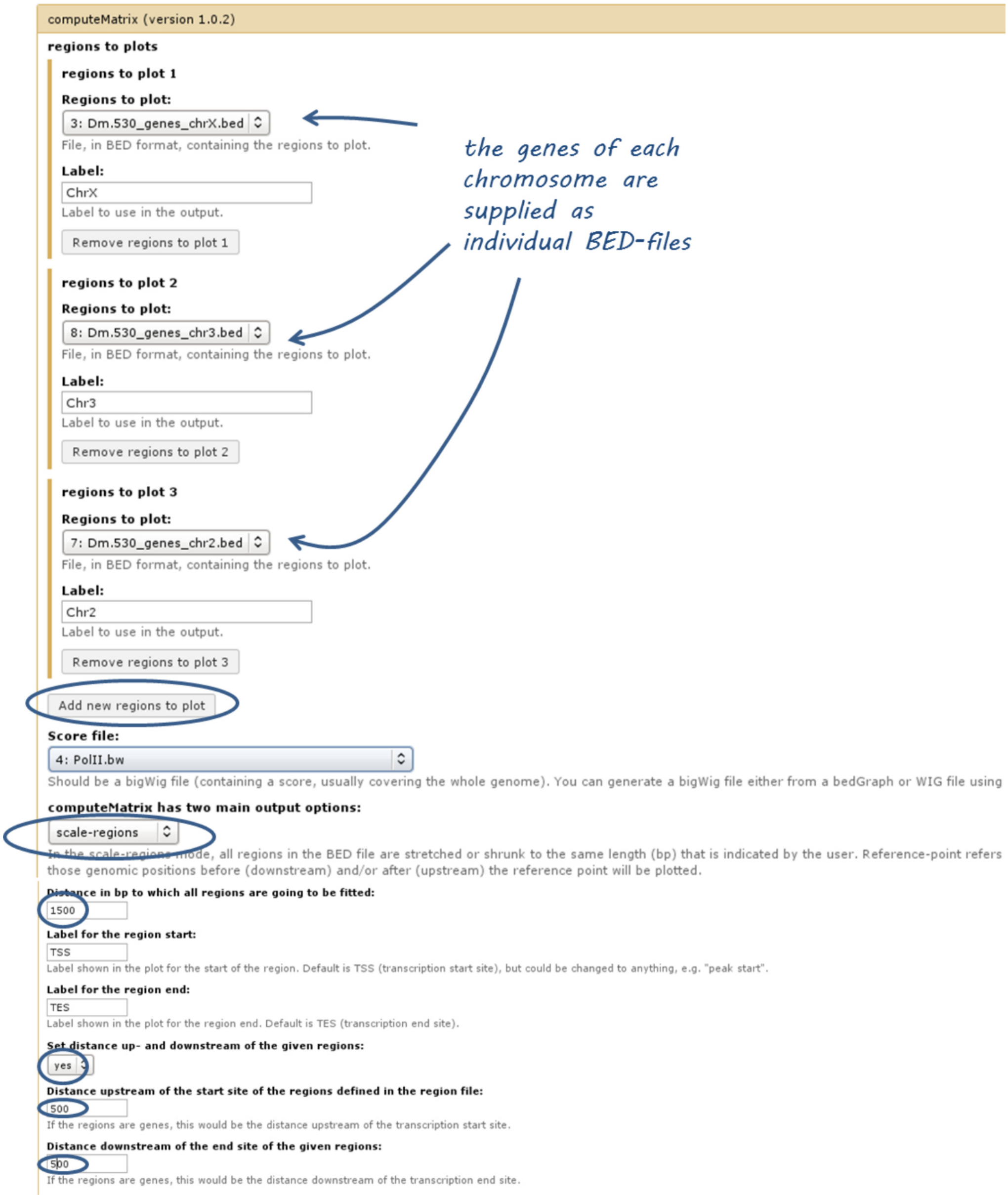
The plot was produced with the following commands:

$ /deepTools-1.5.2/bin/computeMatrix scale-regions --regionsFileName Dm.genes.indChromLabeled.bed --scoreFileName PolII.bw --beforeRegionStartLength 500 --afterRegionStartLength 500 --regionBodyLength 1500 --binSize 10 --outFileName PolII_matrix_scaledGenes --sortRegions no $ /deepTools-1.5.2/bin/heatmapper --matrixFile PolII_matrix_scaledGenes --outFileName PolII_indChr_scaledGenes.pdf --plotTitle "Pol II" --whatToShow "heatmap only"

As you can see, all genes have been scaled to the same size and the (mean) values per bin size (10 bp) are colored accordingly. In addition to the gene bodies, we added 500 bp up- and down-stream of the genes.

This is what you would have to select to achieve the same result within Galaxy:

computeMatrix



computeMatrix (version 1.0.2)

regions to plots

regions to plot 1

Regions to plot: 3: Dm.530_genes_chrX.bed

File, in BED format, containing the regions to plot.

Label: ChrX

Label to use in the output.

Remove regions to plot 1

regions to plot 2

Regions to plot: 8: Dm.530_genes_chr3.bed

File, in BED format, containing the regions to plot.

Label: Chr3

Label to use in the output.

Remove regions to plot 2

regions to plot 3

Regions to plot: 7: Dm.530_genes_chr2.bed

File, in BED format, containing the regions to plot.

Label: Chr2

Label to use in the output.

Remove regions to plot 3

Add new regions to plot

Score file: 4: PolII.bw

Should be a bigWig file (containing a score, usually covering the whole genome). You can generate a bigWig file either from a bedGraph or WIG file using

computeMatrix has two main output options: scale-regions

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 those genomic positions before (downstream) and/or after (upstream) the reference point will be plotted.

Distance in bp to which all regions are going to be fitted: 1500

Label for the region start: TSS

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

Label shown in the plot for the region end. Default is TES (transcription end site).

Set distance up- and downstream of the given regions: yes

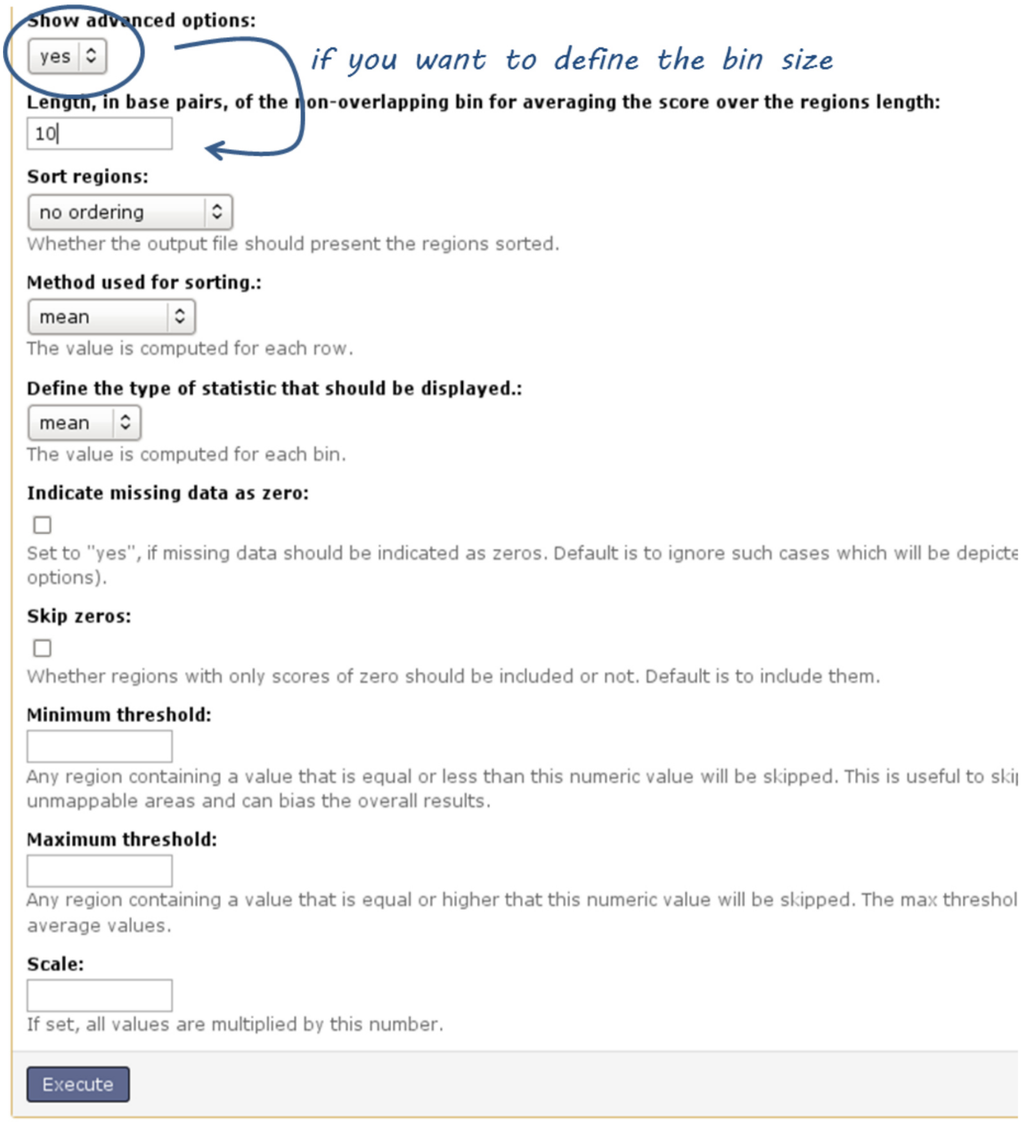
Distance upstream of the start site of the regions defined in the region file: 500

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: 500

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

the genes of each chromosome are supplied as individual BED-files



if you want to define the bin size

Show advanced options:

yes

Length, in base pairs, of the non-overlapping bin for averaging the score over the regions length:

10

Sort regions:

no ordering

Whether the output file should present the regions sorted.

Method used for sorting.:

mean

The value is computed for each row.

Define the type of statistic that should be displayed.:

mean

The value is computed for each bin.

Indicate missing data as zero:

□

Set to "yes", if missing data should be indicated as zeros. Default is to ignore such cases which will be depicte
options).

Skip zeros:

□

Whether regions with only scores of zero should be included or not. Default is to include them.

Minimum threshold:

Any region containing a value that is equal or less than this numeric value will be skipped. This is useful to ski
unmappable areas and can bias the overall results.

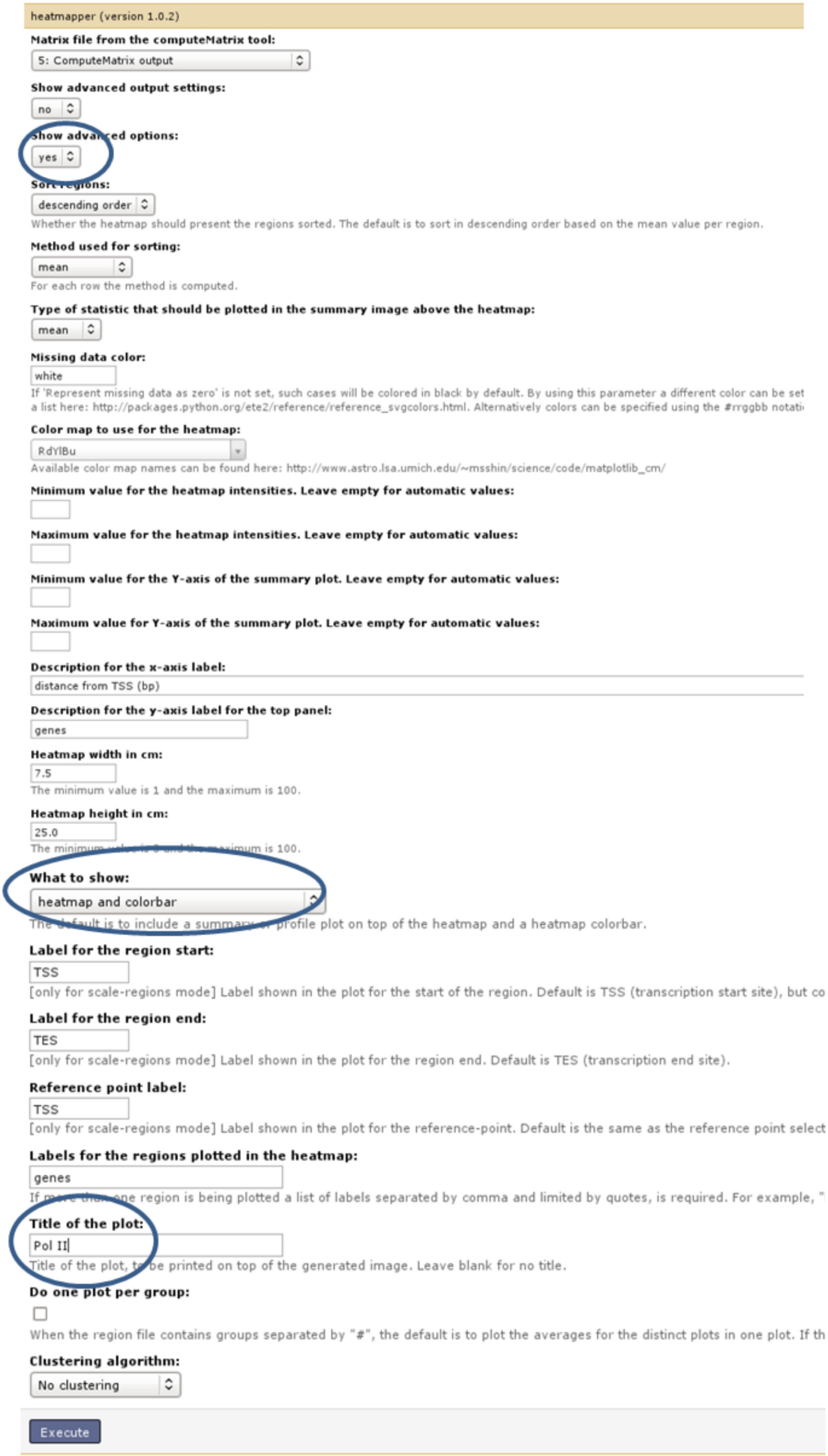
Maximum threshold:

Any region containing a value that is equal or higher that this numeric value will be skipped. The max threshol
average values.

Scale:

If set, all values are multiplied by this number.

heatmapper



heatmapper (version 1.0.2)

Matrix file from the computeMatrix tool:
5: ComputeMatrix output

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:
white
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 list here: http://packages.python.org/ete2/reference/reference_svgcolors.html. Alternatively colors can be specified using the #rrggbb notati

Color map to use for the heatmap:
RdYlBu
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:
25.0
The minimum value is 3 and the maximum is 100.

What to show:
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 co

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 select

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, "

Title of the plot:
Pol II
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 th

Clustering algorithm:
No clustering

main difference between computeMatrix usage on the command line and Galaxy: the input of the regions file (BED)

Note that we supplied just *one* BED-file via the command line whereas in Galaxy we indicated three different files (one per chromosome).

On the command line, the program expects a BED file where different groups of genomic regions are concatenated into one file, where the beginning of each group should be indicated by "#group name". The BED-file that was used here, contained 3 such lines and could be prepared as follows:

```
$ grep ^chr2 AllGenes.bed > Dm.genes.indChromLabeled.bed
$ echo "#chr2" >> Dm.genes.indChromLabeled.bed
$ grep ^chr3 AllGenes.bed >> Dm.genes.indChromLabeled.bed
$ echo "#chr3" >> Dm.genes.indChromLabeled.bed
$ grep ^chrX AllGenes.bed >> Dm.genes.indChromLabeled.bed
$ echo "#chrX" >> Dm.genes.indChromLabeled.bed
```

In Galaxy, you can simply generate three different data sets starting from a whole genome list by using the "Filter" tool three times:

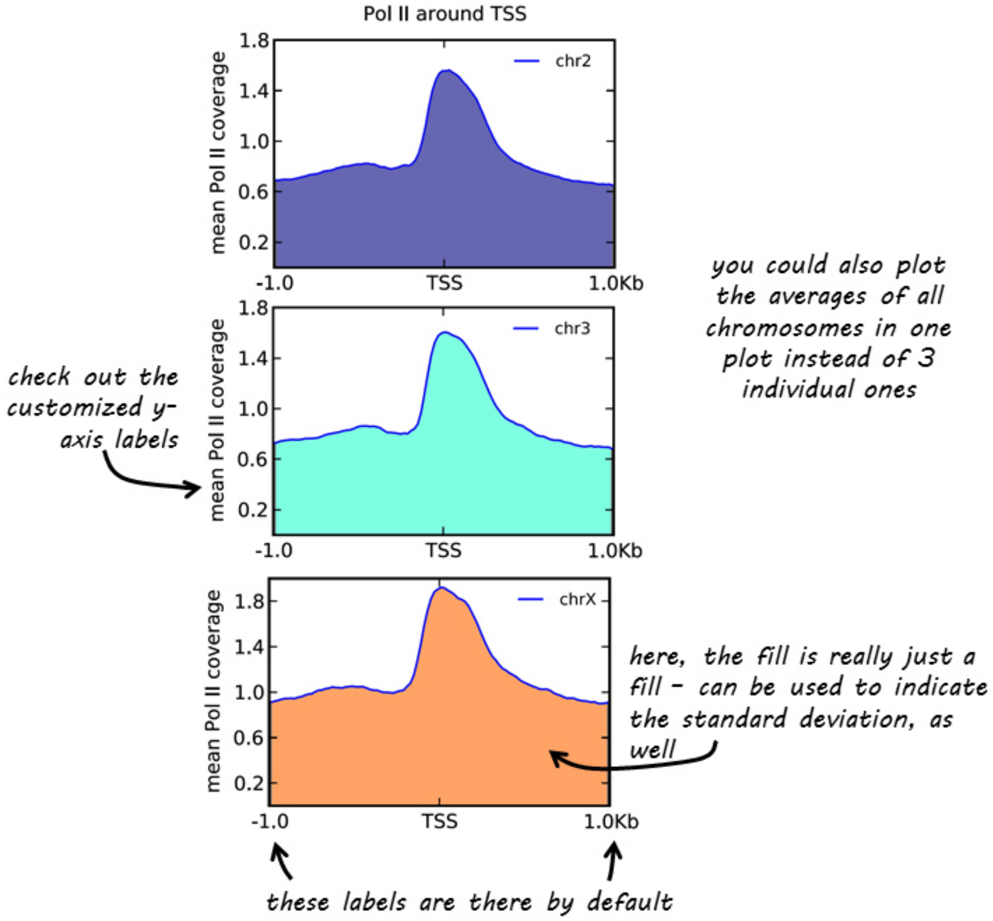
1. c1=="chr2" --> Dm.genes.chr2.bed
2. c1=="chr3" --> Dm.genes.chr3.bed
3. c1=="chrX" --> Dm.genes.chrX.bed

Important parameters for optimizing the visualization

1. **sorting of the regions**: The default of heatmapper is to sort the values in descending order. You can change that to ascending, no sorting at all or according to the size of the region (Using the `--sort` option on the command line or advanced options in Galaxy). We strongly recommend to leave the sorting option at "no sorting" for the initial computeMatrix step.
2. **coloring**: The default coloring by heatmapper is done using the python color map "RdYlBu", but this can be changed (--colorMap on the command line, advanced options within Galaxy).
3. **dealing with missing data**: You have certainly noticed that some gene bodies are depicted as white lines within the otherwise colorful mass of genes. Those regions are due to genes that, for whatever reason, did not have any read coverage in the bigWig file. There are several ways to handle these cases:
    - **--skipZeros** this is useful when your data actually has a quite nice coverage, but there are 2 or 3 regions where you deliberately filtered out reads or you don't expect any coverage (e.g. hardly mapable regions). This will only work if the entire region does not contain a single value.
    - **--missingDataAsZero** this option allows computeMatrix do interpret missing data points as zeroes. Be aware of the changes to the average values that this might cause.
    - **--missingDataColor** this is in case you have very sparse data or were missing values make sense (e.g. when plotting methylated CpGs - half the genome should have no value). This option then allows you to pick out your favorite color for those regions. The default is black (was white when the above shown image was produced).

Summary plots

Here's the **profiler** plot corresponding to the heatmap above. There's one major difference though - do you spot it?



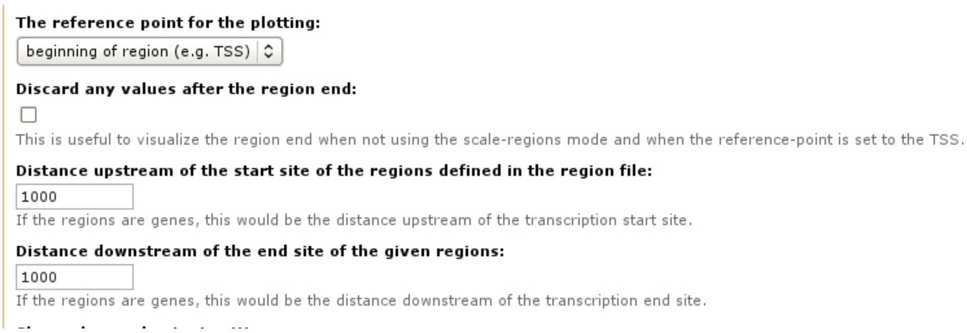
We used the same BED file(s) as for the heatmap, hence the 3 different groups (1 per chromosome). However, this time we used computeMatrix not with *scale-regions* but with *reference-point* mode.

```
$ /deepTools-1.5.2/bin/computeMatrix reference-point --referencePoint TSS --regionsFileName
Dm.genes.indChromLabeled.bed --scoreFileName PolII.bw --beforeRegionStartLength 1000
-afterRegionStartLength 1000 --binSize 10 --outFileName PolII_matrix_indChr_refPoint --missingDataAsZero
--sortRegions no $ /deepTools-1.5.2/bin/profiler --matrixFile PolII_matrix_indChr_refPoint --outFileName
profile_PolII_indChr_refPoint.pdf --plotType fill --startLabel "TSS" --plotTitle "Pol II around TSS" --yAxisLabel
"mean Pol II coverage" --onePlotPerGroup
```

When you compare the profiler commands with the heatmapper commands, you also notice that we made use of many more labeling options here, e.g. `--yAxisLabel` and a more specific title via `-T`

This is how you would have obtained this plot in Galaxy (only the part that's *different* from the above shown command for the scale-regions version is shown):

computeMatrix



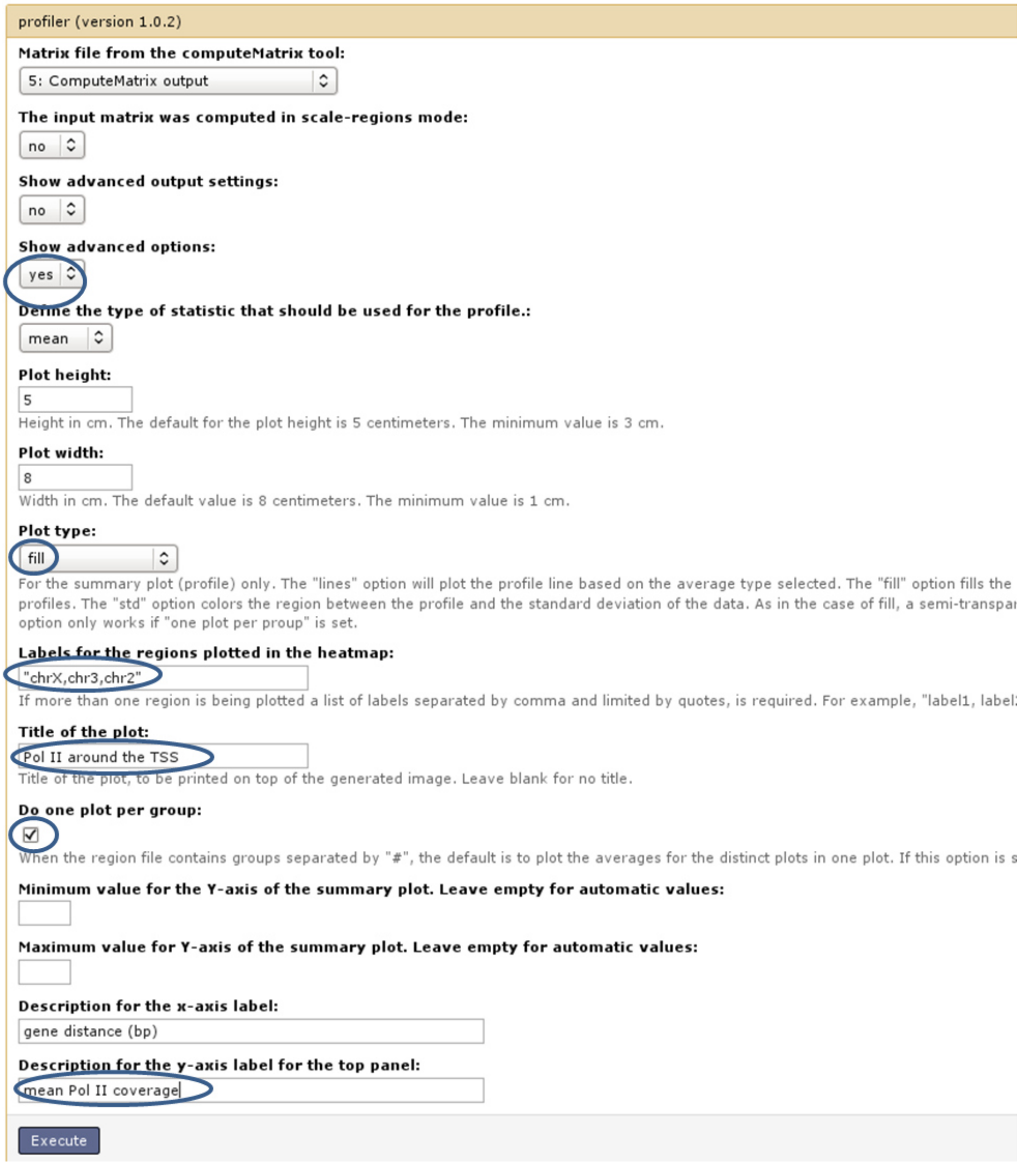
The reference point for the plotting:
beginning of region (e.g. TSS)

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:
1000
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:
1000
If the regions are genes, this would be the distance downstream of the transcription end site.

profiler



profiler (version 1.0.2)

Matrix file from the computeMatrix tool:
5: ComputeMatrix output

The input matrix was computed in scale-regions mode:
no

Show advanced output settings:
no

Show advanced options:
[✓ yes]

Define the type of statistic that should be used for the profile.:
mean

Plot height:
5
Height in cm. The default for the plot height is 5 centimeters. The minimum value is 3 cm.

Plot width:
8
Width in cm. The default value is 8 centimeters. The minimum value is 1 cm.

Plot type:
[✓ fill]
For the summary plot (profile) only. The "lines" option will plot the profile line based on the average type selected. The "fill" option fills the profiles. The "std" option colors the region between the profile and the standard deviation of the data. As in the case of fill, a semi-transpar option only works if "one plot per proup" is set.

Labels for the regions plotted in the heatmap:
[✓ "chrX,chr3,chr2"]
If more than one region is being plotted a list of labels separated by comma and limited by quotes, is required. For example, "label1, label

Title of the plot:
[✓ Pol II around the TSS]
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 s

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:

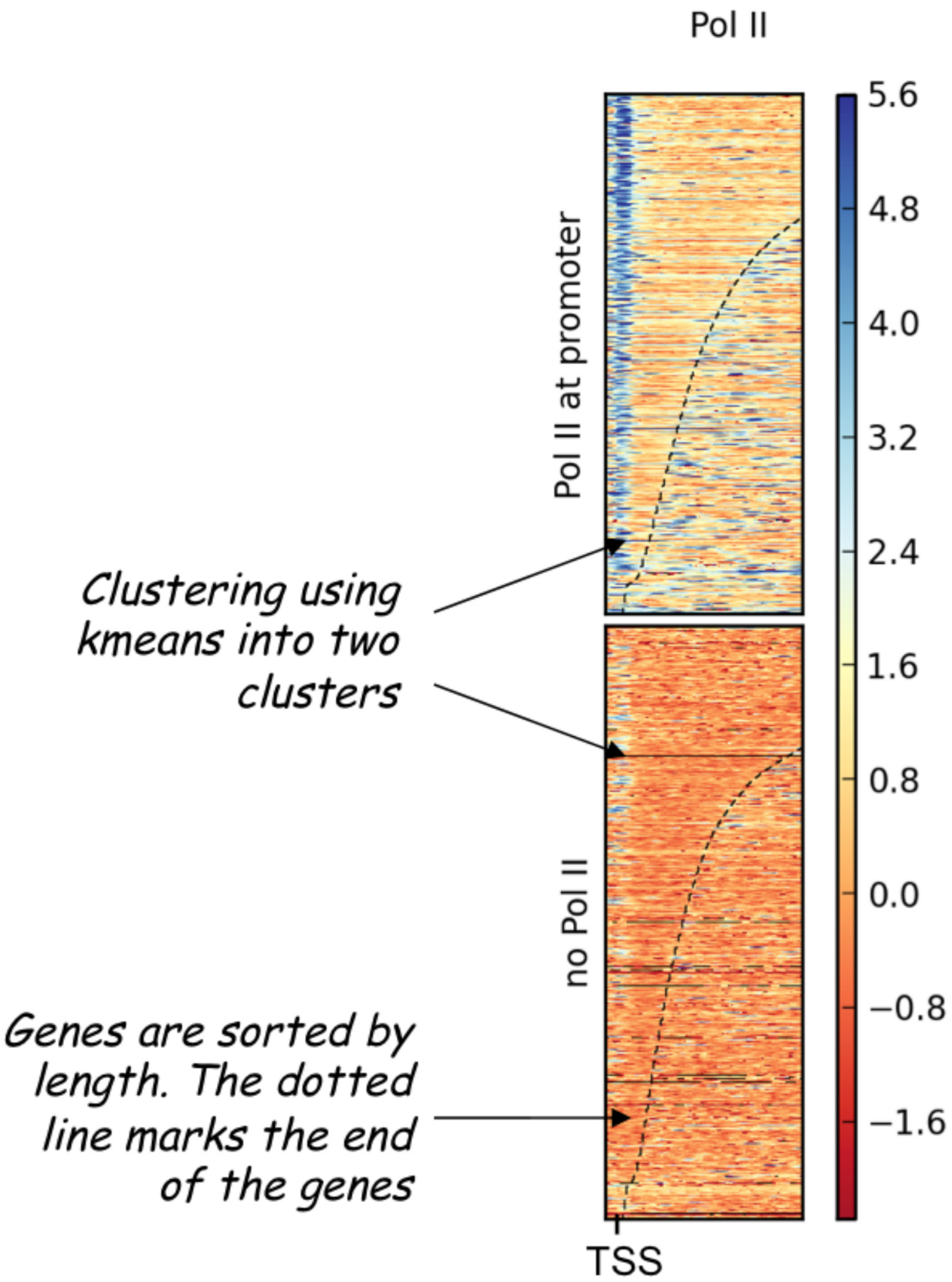
Description for the x-axis label:
gene distance (bp)

Description for the y-axis label for the top panel:
[✓ mean Pol II coverage]

Heatmap with all genes scaled to the one size and kmeans clustering

Instead of supplying groups of regions on your own, you can use the clustering function of heatmapper to get a first impression whether the signal of your experiment can be easily clustered into two or more groups of similar signal distribution.

Have a look at this example with two clusters:



The plot was produced with the following commands:

$ /deepTools-1.5.2/bin/computeMatrix reference-point -regionsFilenName Dm.genes.indChromLabeled.bed --scoreFileName PolII.bw --beforeRegionStartLength 500 --afterRegionStartLength 500 --binSize 50 --outFileName PolII_matrix_TSS $ /deepTools-1.5.2/bin/heatmapper --matrixFile PolII_matrix_TSS --kmeans 2 --outFileName PolII_two_clusters.pdf --plotTitle "Pol II" --sortUsing region_length --whatToShow "heatmap only"

When the `--kmeans` option is chosen and more than 0 clusters are specified, heatmapper will run the k-means clustering algorithm. In this example *Drosophila m.* genes were divided into two clusters separating those genes with Pol II at the promoter region (top) from those genes without Poll II at the promoter (bottom). Please note that the clustering will only work if the initial BED-file used with computeMatrix contained only *one* group of genes (i.e. all genes, without any hash tags separating them)

The genes belonging to each cluster can be obtained by via `--outFileSortedRegions` on the command line and "advanced output options in Galaxy". On the command line, this will result in a BED file where the groups are separated by a hash tag. In Galaxy, you will obtain individual data sets per cluster.

To have a better control on the clustering it is recommended to load the matrix raw data into **specialized software like cluster3 or R**. You can obtain the matrix via the option `--outFileNameMatrix` on the command line and by the "advanced output options" in Galaxy. The order of the rows is the same as in the output of the `--outFileSortedRegions` BED file.

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