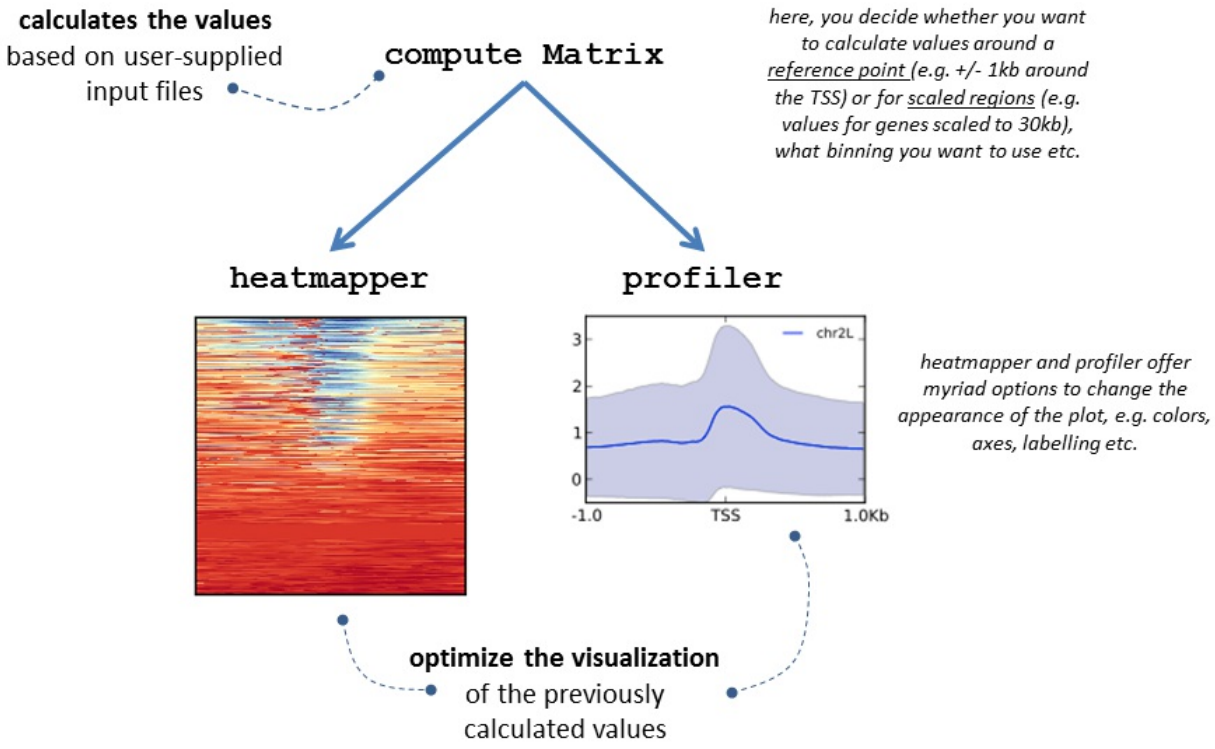
# Visualization

The modules for visualizing scores contained in bigWig files are separated into 1 tool that calculates the values (*computeMatrix*) and 2 tools that contain many, many options to fine-tune the plots (*heatmapper* and *profiler*). In other words: computeMatrix generates the values that are the basis for heatmapper and profiler.




## 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.

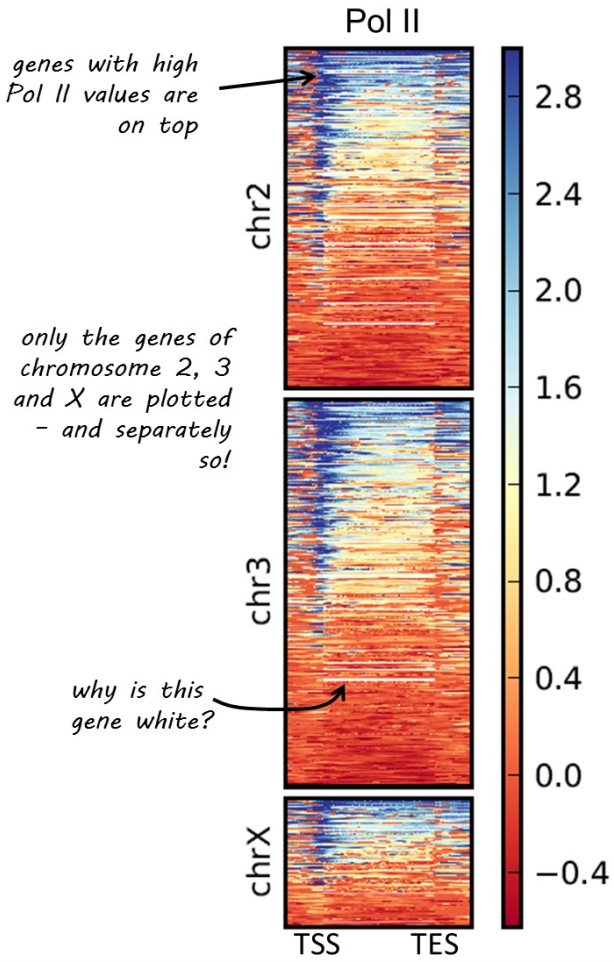
## profiler

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.

### 1st example: Heatmap with all genes scaled to the one size and user-specified groups of genes




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.

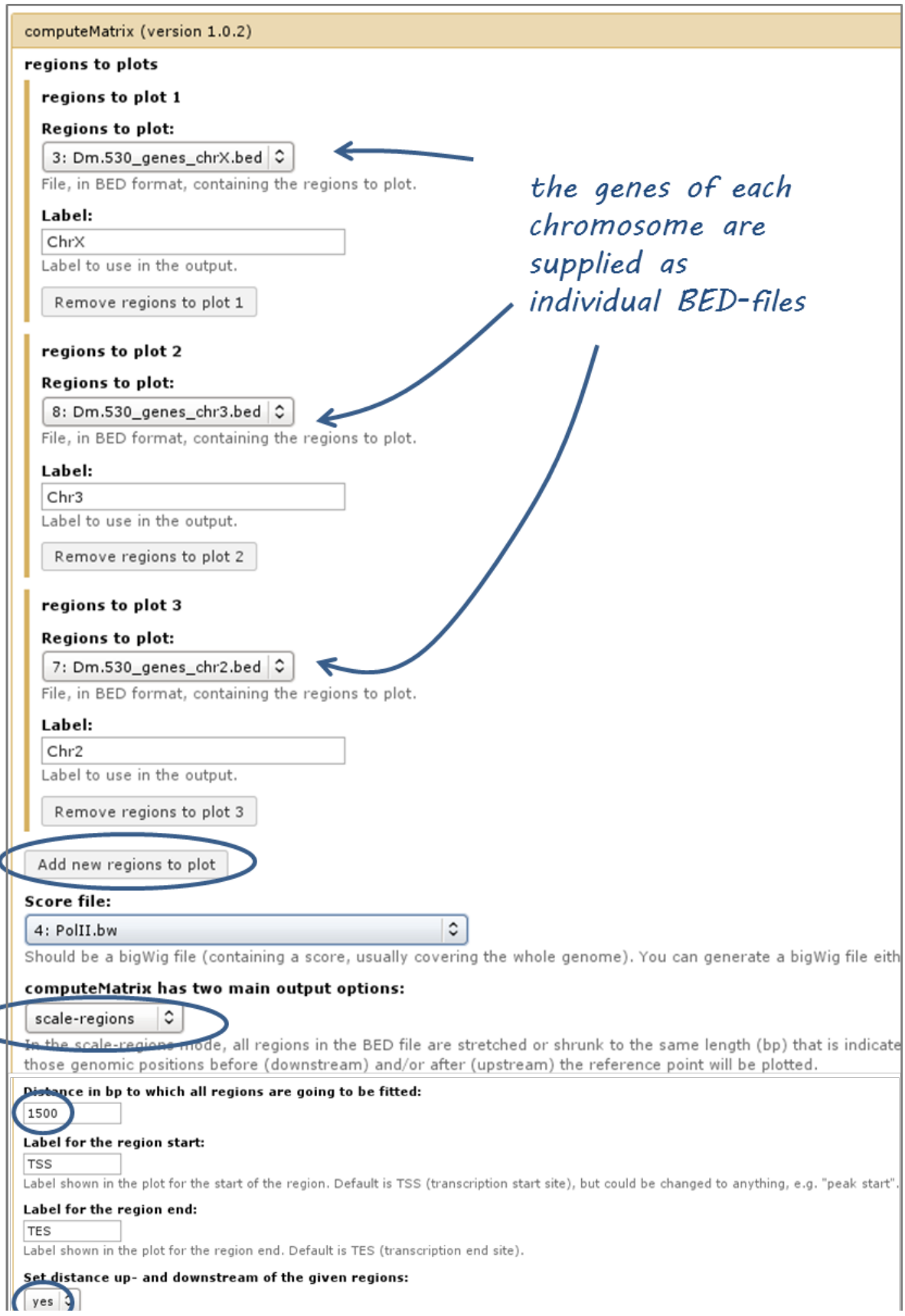
The plot was produced with the following commands:

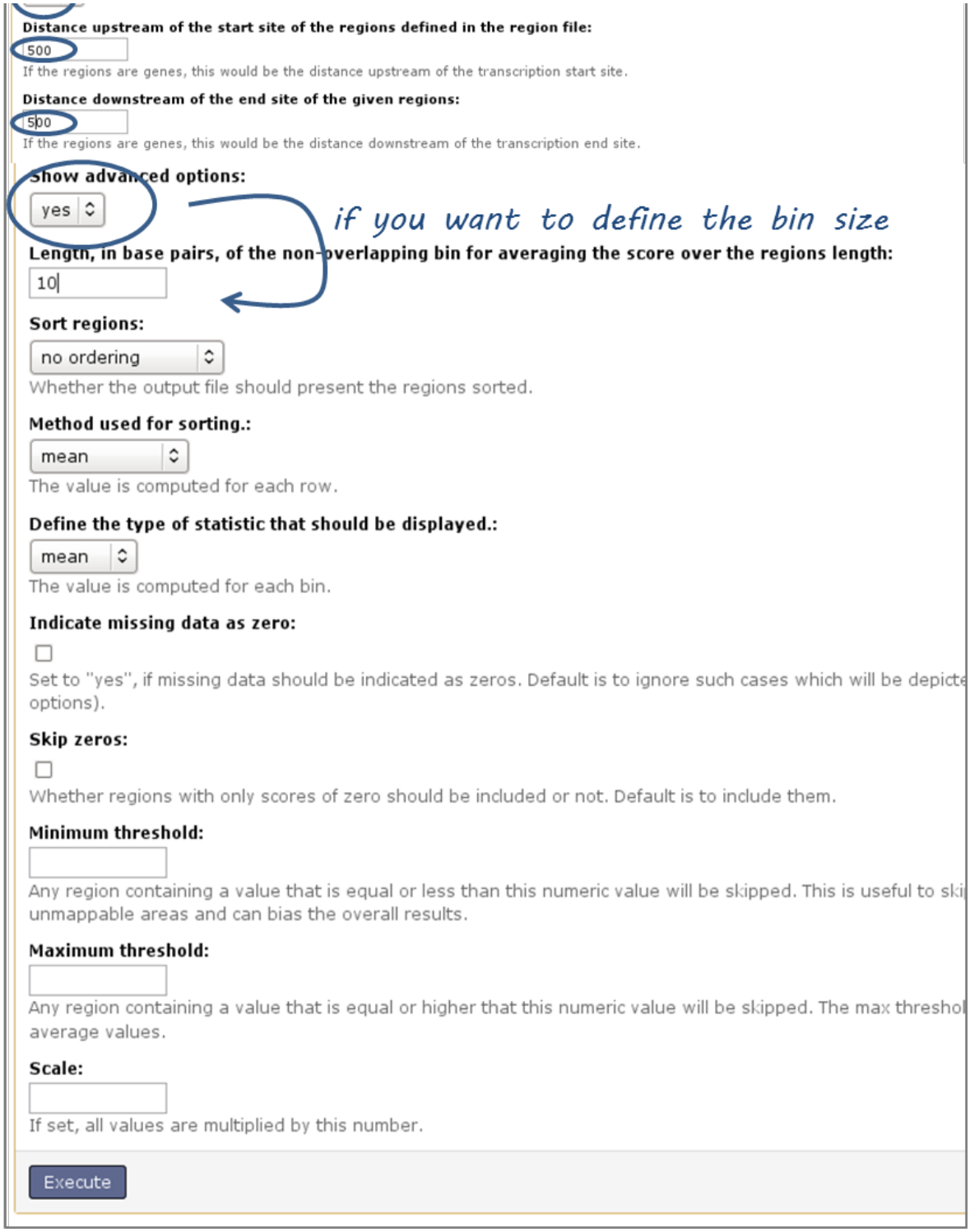
```
$ /deepTools-1.5.2/bin/computeMatrix scale-regions --regionsFileName Dm.genes.indChromLabeled.bed \
--scoreFileName PolII.bw --beforeRegionStartLength 500 --afterRegionStartLength 500 \
--regionBodyLength 1500 --binSize 10 \
--outFileName PolII_matrix_scaledGenes --sortRegions no

$ /deepTools-1.5.2/bin/heatmapper --matrixFile PolII_matrix_scaledGenes \
```

```
--outFileName PolII_indChr_scaledGenes.pdf \
--plotTitle "Pol II" --whatToShow "heatmap and colorbar"
```

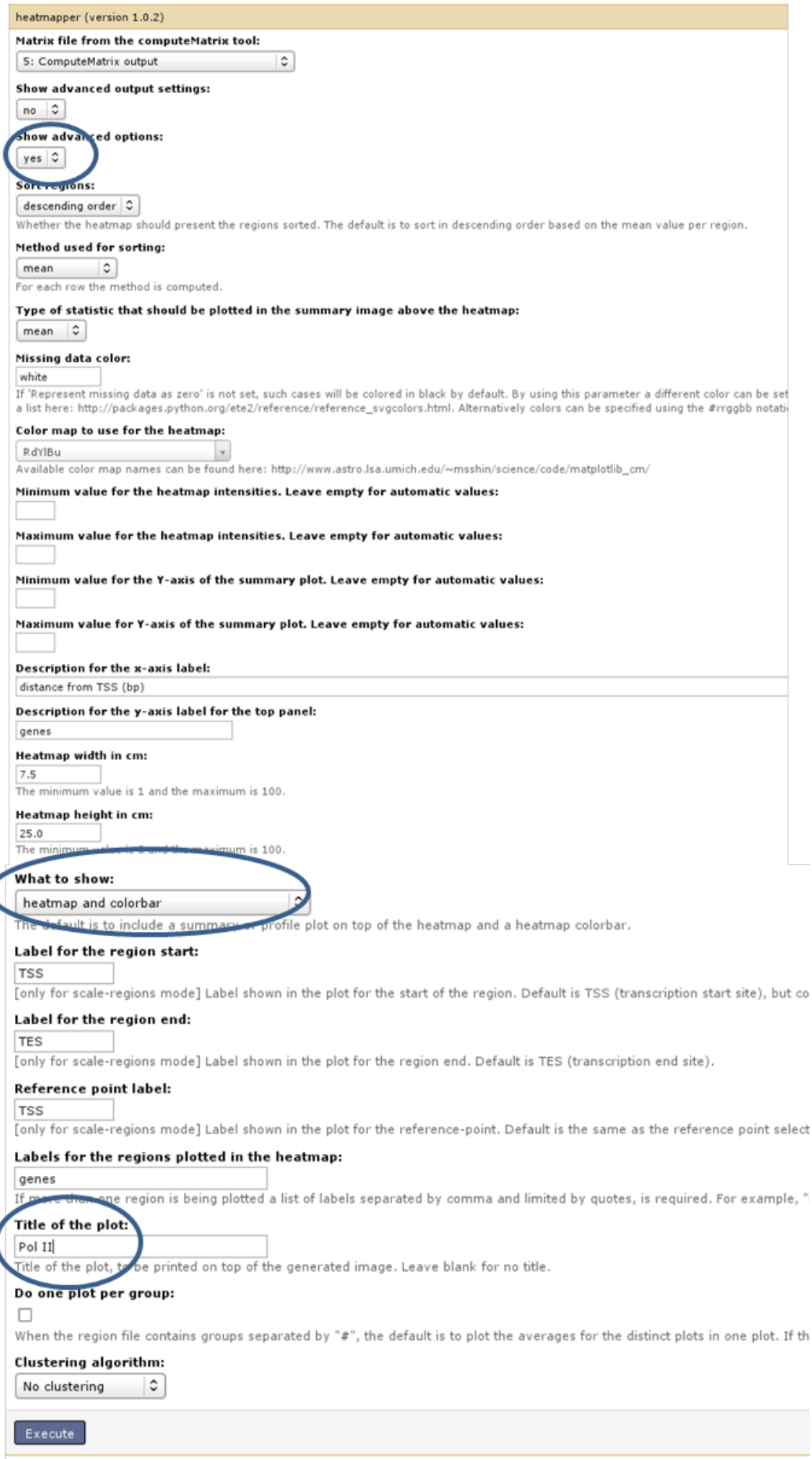
This is what you would have to select to achieve the same result within Galaxy (pay attention to the fact that you will have to use two tools, computeMatrix and heatmapper):

**computeMatrix**




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.
Show advanced options:
yes
if you want to define the bin size
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 thresho
average values.
Scale:
If set, all values are multiplied by this number.
Execute

**heatmapper**




**The 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 of *Drosophila melanogaster* genes by using the "Filter" tool ("Filter and Sort" --> "Filter") on the entries in the first column 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).