

Two-Tier Mapper] **Two-Tier Mapper: a user-independent clustering method for global gene expression analysis based on topology**

***Rachel Jeitziner***

UPBRI& ISREC  
EPFL Lausanne

## Contents

1	Introduction . . . . .	1
2	Prepare the data . . . . .	4
3	TTMap part1: Adjustment of the control group (ctrl_adj) 4	
4	TTMap part1: Hyperrectangle deviation assessment (hda) . . . . .	5
5	TTMap part2: Global-to-local Mapper (gtlmap) . . . . .	6
6	TTMap: Finding the significant genes (sgn_genes) . . . . .	7
7	Conclusion . . . . .	7

## 1 Introduction

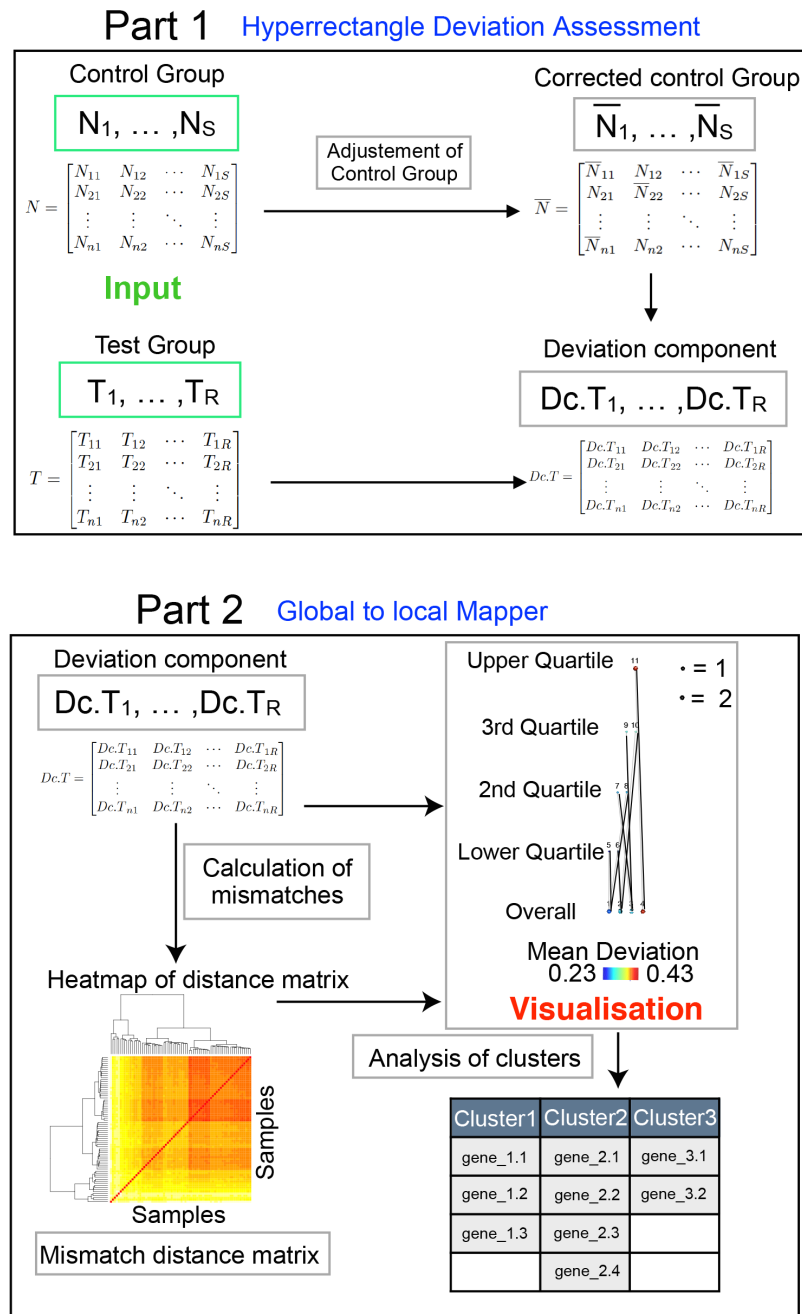
---

We developed a new user-independent analytical framework, called *Two-Tier Mapper (TTMap)*. This tool is separated into two parts. TTMap consists of two separated and independent parts : 1. Hyperrectangle Deviation assessment

[

(HDA) and 2. Global-to-Local Mapper (GtLMap), where the first step establishes properties of the control group and removes outliers in order to calculate the deviation of each vector in the test group from the corrected control group. The second step uses the traditional Mapper algorithm [1] with a two-tier cover and a special distance. This topological tool detects both global and local differences in the patterns of deviations and thereby captures the structure of the test group. The samples are clustered according to the shape of their deviation (do they both deviate positively, negatively or are they as the control). To still keep on the information about the amount of deviation, one separates the data into 4 clusters according to a function measuring the amount of deviation. These represent then the second tier. Each cluster is colored by the extent of the deviation. A list of the differentially expressed genes is also provided. The functions and methods presented on this *vignette* provide explanation on how to use TMap, by default and what can be changed by the user.

[



**Figure 1:** Schematic overview of TMap. The inputs (green) are given by two gene expression matrices, the control (N) and the test group (T), rows represent genes and columns samples. In Part 1, TMap adjusts the control group for outlier values ( $\bar{N}_*$ ), feature by feature. It calculates deviation from this corrected control group for individual samples in the test group ( $Dc.T_*$ ). In Part 2, TMap computes a similarity measure, the mismatch distance (represented as a heatmap) using the deviation components. The Mapper [?] algorithm is used with a two-tier cover to generate a visual representation of the clustering creating a network of global clusters (Overall) and local clusters (1st, 2nd, 3rd, 4th quartile of a filter function). It takes as inputs the mismatch distance and the deviation components.

[

## 2 Prepare the data

---

Upload the file(s) to compare in R. Log transform and subselect them. Use *make\_matrices* to create the needed files for the first function of TMap since it generates the control and the test matrix in the right format. As an example, we use the airway data set available at Bioconductor.

```
> library(airway)
> data(airway)
> airway <- airway[rowSums(assay(airway))>80,]
> assay(airway) <- log(assay(airway)+1,2)
> experiment <- TMap::make_matrices(airway, seq_len(4), seq_len(4)+4,
+ NAME = rownames(airway), CLID = rownames(airway))
```

This function can directly be used on a normalised count table from RNA- seq precising what are the columns of the control group (in col\_ctrl) and what are the columns in the test group (in col\_test) .

## 3 TMap part1: Adjustment of the control group (ctrl\_adj)

---

The first part of the method checks if the control and the test matrices have the same row-names, and if not the method subselects the common rows. It outputs the files with the common rows subselected (with the extension mesh). It then calculates the corrected control matrix, which removes outliers and replaces them by a chosen method (given by a function with input the matrix with NAs where there is an outlier and should return a matrix without NAs), or by the median of the other values by default. The inputs can even be given by the CTRL and TEST variables of the list given by the output of *make\_matrices* or by imputed control and test matrices in pcl format (see [2]). The name of the control group and the project name need to be inputed as well as the working directory, in which the output files will be created. A value for what to consider as an outlier (called e) can be imputed or use the data-driven default value given by the method. If there are any batch effects to consider, they can be imputed using the variable B, which is a vector of numbers representing the batches. Last, the parameter *P* is a value which will remove the genes that have a higher percentage than *P* of outlier values.

```
> E=1
> Pw=1.1
> Bw=0
> TMAP_part1prime <- TMap::control_adjustment(normal.pcl = experiment$CTRL,
+ tumor.pcl = experiment$TEST,
+ normalname = "The_healthy_controls", dataname = "The_effect_of_cancer",
+ org.directory = getwd(), e=E,P=Pw,B=Bw);
```

This outputs:

- A file with the number of outliers per sample (Dataname followed by the number of the batch followed by na\_numbers\_per\_col.txt)

[

- A file with the number of outliers per row (Dataname followed by the number of the batch followed by `na_numbers_per_col.txt`)
- A picture of the distribution of the mean against variance for each gene, before (Dataname followed by `_mean_vs_variance.pdf`) and
- after correction of outliers (Dataname followed by `_mean_vs_variance_after_correction.pdf`).

The corrected control matrix is output in the next step. A possible output after this first step is shown in figure 2



**Figure 2:** `barplotSignifSignatures`: Plot of mean against variance per gene.

## 4 TMap part1: Hyperrectangle deviation assessment (hda)

This part consists in calculating deviation components from a hyperrectangle. This enables the calculation in the third function (`ttmap_part2_gtlmap`) of the shape of deviation. One parameter  $k$  is given by if all the vectors of the control group should be kept or if only the the top  $k$ -dimensional principal component approximation of the control matrix should be kept using the singular value decomposition (as in [2]). The default is to keep all the vectors.

```
> TMAP_part1_hda <- TMap::hyperrectangle_deviation_assessment(x =
+ TMAP_part1prime,k=dim(TMAP_part1prime$Normal.mat)[2],
+ dataname = "The_effect_of_cancer", normalname = "The_healthy_controls");
> head(TMAP_part1_hda$Dc.Dmat)
```

	SRR1039516.Dis	SRR1039517.Dis	SRR1039520.Dis	SRR1039521.Dis
ENSG00000000003	2.5977598	3.540946	-5.017074	-2.8395353
ENSG000000000419	5.4810747	5.552852	-3.522795	-0.5227945
ENSG000000000457	2.8373989	5.467910	-3.973458	-2.6042244
ENSG000000000460	2.9880457	4.234311	-2.688056	-0.7705182
ENSG000000000971	0.9162157	4.453708	-4.687548	-2.0848835
ENSG00000001036	1.2333603	-2.783987	-4.568258	-2.3631438

The outputs of this step are the following.

- The corrected control matrix, calculated at the first step is given in `The_healthy_controls.NormalModel.pcl`, with a possible trimming of columns if  $k$  is different than the number of columns in the corrected matrix.
- The deviation component of each test sample is written in `The_effect_of_cancer.Tdis.pcl`. An example of the deviation component is found in the previous script by writing `head(TMAP_part1_hda$Dc.Dmat)`
- The normal component of each test sample is written in `The_effect_of_cancer.Tnorm.pcl`.

[

The two values of this function are the deviation component matrix and the overall deviation (calculated by summing in absolute values the deviation components).

## 5 TMap part2: Global-to-local Mapper (gtlmap)

The third part corresponds to the Global-to-local Mapper part. One starts with an annotation file of our samples, in order to annotate the obtained clusters. In this example here we just copied several times the column names. This annotation file needs to have as rownames the columns of the test samples followed by ".Dis". We then calculate the distance matrix between the samples using the *generate\_mismatch\_distance* function, which uses a cutoff parameter  $\alpha$  in order to decide what is considered as noise. Any other distance matrix can be computed here and used for the next step. Then, we calculate and output the clusters using *tmap\_part2\_gtlmap*, which needs as inputs the values of *tmap\_part1\_ctrl\_adj*, *tmap\_part1\_hda*. The default parameter uses all the genes to calculate the overall deviation, but if a subset should be selected (only one pathway for example), it can be imputed here. *tmap\_part2\_gtlmap* then calculates using *calcul\_e* a parameter of closeness using the data, in order to know what distance is "close" enough to clusters samples together. The parameter *n* determines which column of metadata should be chosen for the output files. Two more parameters of convenience, if *ad* is set to something different than 0 (the default) then the clusters on the output picture will not be annotated and if *bd* is different than 0 (default), the output will be without outliers of the test data set. After the picture has been adjusted to what one wants to see one can save it using the *rgl.postscript* function.

```
> library(rgl)
> ALPHA <- 1
> annot <- c(paste(colnames(experiment$TEST[, -seq_len(3)]), "Dis", sep=".")
+ ,paste(colnames(experiment$CTRL[, -seq_len(3)]), "Dis", sep="."))
> annot <- cbind(annot, annot)
> rownames(annot) <- annot[, 1]
> dd5_sgn_only <- TMap::generate_mismatch_distance(TTMAP_part1_hda,
+ select=rownames(TTMAP_part1_hda$Dc.Dmat), alpha = ALPHA)
> TTMAP_part2_gtlmap <-
+ TMap::tmap(TTMAP_part1_hda, TTMAP_part1_hda$m,
+ select=rownames(TTMAP_part1_hda$Dc.Dmat),
+ annot, e= TMap::calcul_e(dd5_sgn_only, 0.95, TTMAP_part1prime, 1),
+ filename="first_comparison",
+ n=1, dd=dd5_sgn_only)

[1] "e_map = 0.377016505761445"
[1] "e_map = 0.377016505761445"
[1] "e_map = 0.377016505761445"
[1] "e_map = 0.377016505761445"
[1] "e_map = 0.377016505761445"

> rgl.postscript("first_output.pdf", "pdf")
```

[

## 6 TMap: Finding the significant genes (sgn\_genes)

---

This last function analyses the different clusters for significant features. It outputs a file per level (one for overall, called all, one for the lower quartile, called low, one for the second quartile, called mid1, the third, mid2, and the higher quartile, called high). In each of them one file per cluster is given, with the list of significant genes linked to the cluster. Relaxed is a parameter permitting to select as a match one sample that would be 0 for the deviation component, while the others deviate in the same shape.

```
> TMap::tmap_sgn_genes(TMAP_part2_gtlmap,  
+ TMAP_part1_hda, TMAP_part1prime,  
+ annot, n = 2, a = ALPHA,  
+ filename = "first_trial", annot = TMAP_part1prime$tag.pcl, col = "NAME",  
+ path = getwd(), Relaxed = 0)
```

## 7 Conclusion

---

Two-Tier Mapper (TMap) is a topology-based clustering tool, which is user-friendly and reliable. The algorithm first provides an overall clustering, in an unbiased manner, since all the parameters are defined in a data-driven manner or by reliable default parameters. This method enables a refined view on the composition of the clusters by delineating how clusters differ locally and how the local clusters relate to the global structure of the dataset. The output is a visual interpretation of the data given by a colored graph that is easy to interpret, which describes the shape of the data according to the chosen distance.

## References

- [1] P. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3, 2013.
- [2] Monica Nicolau, Arnold Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Science*, 108(17):7265–7270, 2011.