flowCyBar - Analyze flow cytometric data using gate information

Schumann, J., Koch, C., Günther S., Fetzer, I., Müller, S.
Helmholtz Centre for Environmental Research - UFZ
Department of Environmental Microbiology

May 1, 2024

Abstract

CyBar means cytometric barcoding and is a tool for evaluation of cytometric datasets by using gate information. On the first hand it was developed for evaluation of microbial cytometric data but it can also be used for any other cytometric dataset. CyBar is used to record cell abundances per gate by creating a gate template that is valid for all samples that are part of an investigation. Thus, cell abundance variation per gate over time or per treatment is provided. As a result, the gate based population or community alterations are quantified and visualized in heatmaps. In addition, functional maps can be designed that indicate certain functions of cells in respective gates due to abiotic parameters that are recorded in parallel to the cytometric measurements. Also interactions of cells between gates can be suggested. The procedure allows a segregated and/or function-dependent identification of subpopulations/-communities of interest e.g. for cell sorting. The CyBar is useful to identify dynamics in a cell system, to find factors that shape a population or community and to make sort decisions easier.

1 Introduction

The background of the CyBar tool is published in [Günther et al., 2012] and [Koch et al., 2013b]. The basic idea is to identify all subpopulations or subcommunities in cytometrically analyzed samples of a finalized test series. As a result a gate template is created. This template provides cell abundance information per gate over time or per treatment. Following, the CyBar provides the overall cell abundance information per gate as a box plot. Since cell abundances per gate can differ by tenfold and more a normalization procedure was introduced. Thus, variation in cell abundance is demonstrated per gate independently of their respective real share of the population or community. The resulting heatmap provides, therefore, both gate based cell abundance and variation over time or treatment of a population or community. Furthermore, a tool is provided to indicate biotic and abiotic background that causes variations in population or community structures. Using Spearman's rank-order correlation coefficient (r_S) the impact of abiotic and biotic parameters on cell abundances per gate is approximated. As a result a functional map is created where influences of the various parameters are profiled. Abiotic parameters can be pH values or concentrations of N- or C-classes but also weather data. Biotic parameters are subpopulations or subcommunities that were detected in the sample. Here, multidimensional data evaluation can be included [Günther et al., 2012]. In this way huge cytometric and corresponding chemical or further related datasets can be combined and interpreted. Subpopulations or subcommunities of interest can be sorted due to their functional role in the population or community [Koch et al., 2013a]. An overview on possible applications in ecosystems microbiology is given in [Koch et al., 2014]. It needs to be stated that CyBar requires datasets that are highly calibrated by using e.g. fluorescent beads. Every shift in the adjustment of a cytometer will be discovered using CyBar and will have an influence on the outcome. The application of CyBar is unlimited. The strength of the tool lies in the deep segregated insight into population and community dynamics, the suggestions that are given towards the functional background of the measured individuals, and the possibility to use this information for intentional sorting. In microbiology, the tool is especially valuable as data can be recorded every few minutes and thus follow generation times and therefore altering of community structures in a very fast way.

2 Overview

Normalization of measured cell numbers After the cytometric measurement, separation of cells of interest from the absolute event counts by setting a parent gate, and creation of the gate template the data consists of relative cell numbers for each gate. For more details see [Koch et al., 2013b]. These cell numbers have to be normalized to ensure the comparability of gates with highly different (low and high) cell abundances for further investigations.

Barcode and Boxplot of the measured relative/normalized cell numbers The barcode describing community structure and variation is created using the normalized cell numbers for each gate and measurement. The cell abundance distribution between gates is shown in a color key. In addition, gates that vary in similar directions can be clustered by a dendrogram. Furthermore, the abundance variation of the relative cell numbers per gate can be shown as a vertical boxplot.

NMDS plot of the measured relative/normalized cell numbers The distance between samples based on their cytometrically measured cell abundance information is visualized using nonmetric multidimensional scaling (NMDS). This step can be done using the relative cell numbers or the normalized cell numbers.

Correlation analysis of relative cell numbers and abiotic parameters The application investigates the correlation between the cell number variations and changes in abiotic parameter values. Respective positive and negative correlations are visualized as a heatmap. In this step the measured relative cell numbers and the measured values of the abiotic parameters are used.

3 Reading the data

The data always have to be read in R. One dataset should be a tab-separated text file. Use one row per sample and one column per e.g. gate or abiotic parameter. Use the first column for the first gate/abiotic parameter and the first line as header. The names of the gates/abiotic parameters may not contain any whitespace. If the values contain commas they must be expressed as ".". Missing values or NA's are allowed (except for the NMDS plot) but should be expressed as "NA" or empty fields in the table. The structural requirements on the data and the way of reading them (see example below) are the same in every further section. In R change the working directory to the path where the data are located and read the data. All backslashes "\" may be changed to slashes "/" when using Windows.

```
> # Type the path into the quotes
> setwd("")
> # or in Windows use
> setwd(choose.dir(getwd(), "Choose the folder containing the data"))
> # Type the filename (including extension) the quotes
> # Example of reading the file Cell_number_sample.txt
> Cell_number_sample<-read.table("Cell_number_sample.txt",header=TRUE,sep="\t")
```

4 Functions

This section gives an overview of the functions that are included in the package. The datasets that are attached to the package serve as examples for demonstration and the influence of different parameter settings on the results. Unless noted otherwise these datasets are taken from [Koch et al., 2013b]. First of all the package has to be loaded in R.

```
> # Load package
> library(flowCyBar)
```

4.1 normalize

This step should be done to ensure the comparability of gates with highly different (low and high) cell abundances. The normalization can be performed by two different methods. In this example the attached dataset `Cell_number_sample` is used for demonstration. It contains sample names and the percentages of cells per gate instead of the relative measured cell numbers. Percentages of cells per gate are calculated by dividing the hundredfold of the relative cell number of one gate by the sum of all cells within the parent gate. This is done for each sample.

The content of this dataset without sample names is shown in Table 1 (excerpt). Nonetheless, instead of using the percentages also the relative measured cell numbers per gate can be used.

```
> # Load dataset
> data(Cell_number_sample)
> # Show data
> Cell_number_sample[,-1]
```

	G1	G2	G3	...	G30
1	12.75	3.86	3.26	...	3.99
2	22.44	1.91	2.84	...	0.91
3	15.51	3.59	3.42	...	1.94
4	20.99	6.31	5.52	...	1.79
5	2.15	1.74	26.74	...	1.19
6	6.62	2.77	15.30	...	1.20
7	5.38	3.88	17.89	...	1.23
8	5.44	3.73	12.12	...	1.41
9	5.06	3.10	9.10	...	1.33
10	5.20	3.03	9.25	...	1.33

Table 1: Percentages of all cell numbers

Two methods of normalization can be chosen. By choosing the default method (`"mean"`) the individual relative/percental cell number of each gate is normalized by dividing the relative/percental value by the mean of all relative/percental cell numbers for that gate. The averages are displayed on the console by default, too. For better reading use 2 digits.

```
> # Normalize data, print 2 digits
> normalize(Cell_number_sample[,-1],digits=2)
```

The averages and normalized values of each gate are printed to the R console. The normalized values look as follows (excerpt):

	G1	G2	G3	...	G30
1	1.26	1.14	0.31	...	2.44
2	2.21	0.56	0.27	...	0.56
3	1.53	1.06	0.32	...	1.19
4	2.07	1.86	0.52	...	1.10
5	0.21	0.51	2.54	...	0.73
6	0.65	0.82	1.45	...	0.74
7	0.53	1.14	1.70	...	0.75
8	0.54	1.10	1.15	...	0.86
9	0.50	0.91	0.86	...	0.81
10	0.51	0.89	0.88	...	0.81

Table 2: Normalized cell numbers after using the default method `"mean"`

By choosing the method `"first"` the individual relative/percental cell number of each gate is normalized by dividing the relative/percental value by the first relative/percental cell number of that gate. This method should be used to investigate changes within the gates ensuing from explicit starting conditions.

```
> # Use method "first"
> normalize(Cell_number_sample[,-1],method="first",digits=2)
```

The normalized values looks different now (see Table 3). Note that the first value of each gate is 1.00 which means that this is the starting value and all other values are calculated relative to this value.

	G1	G2	G3	...	G30
1	1.00	1.00	1.00	...	1.00
2	1.76	0.49	0.87	...	0.23
3	1.22	0.93	1.05	...	0.49
4	1.65	1.63	1.69	...	0.45
5	0.17	0.45	8.20	...	0.30
6	0.52	0.72	4.69	...	0.30
7	0.42	1.01	5.49	...	0.31
8	0.43	0.97	3.72	...	0.35
9	0.40	0.80	2.79	...	0.33
10	0.41	0.78	2.84	...	0.33

Table 3: Normalized cell numbers after using the method `"first"`

In the previous examples the first column of the dataset `Cell_number_sample` was omitted due to the fact that names can not be normalized. The content of this dataset actually looks as follows (excerpt):

```
> # Show data
> Cell_number_sample
```

	sample	G1	G2	G3	...	G30
1	S1	12.75	3.86	3.26	...	3.99
2	S2	22.44	1.91	2.84	...	0.91
3	S3	15.51	3.59	3.42	...	1.94
4	S4	20.99	6.31	5.52	...	1.79
5	S5	2.15	1.74	26.74	...	1.19
6	S6	6.62	2.77	15.30	...	1.20
7	S7	5.38	3.88	17.89	...	1.23
8	S8	5.44	3.73	12.12	...	1.41
9	S9	5.06	3.10	9.10	...	1.33
10	S10	5.20	3.03	9.25	...	1.33

Table 4: Table with sample names and percental cell numbers

To avoid errors the columns used for normalization were selected manually by using box brackets. Within the brackets the space in front of the comma defines the rows and the space behind the comma defines the columns to be selected. If one space is empty all rows/columns are selected. If the first and the second columns are used for e.g. sample names or other additional information the normalization should start with the third column.

The values can now be written to a new file using the `write.table` function of R. For more details type "?write.table" to the R console.

```
> # Save normalized values
> normalized<-normalize(Cell_number_sample[,-1],digits=2)
> # Write a new file containing the normalized values
> write.table(normalized,file="normalized.txt",row.names=TRUE,
+ quote=FALSE,sep="\t")
```

4.2 cybar_plot

To reveal the cell number variation within each gate or to follow it from explicit starting conditions a barcode based on the normalized values is used. In addition, the measured relative or percental values are used to visualize the distribution of the cell numbers for each gate as a boxplot. The attached dataset `Cell_number_sample` is used for demonstration.

4.2.1 Investigate cell number variation

To investigate the cell number variation within each gate the normalized values created by using the method `"mean"` should be used (see section 4.1 Table 2). The dataset `Cell_number_sample` contains percental cell numbers as shown in Table 1.

Figure 1 shows the pictures using default parameters. The spreading in the color key within the barcode is well balanced in this example, as evidenced by the color histogram in the top left corner. The orientation of the data labels in the boxplot is horizontal by default. The order of the boxes is by default the same as in the dataset but unlike the order within the barcode.

```
> # Load dataset
> data(Cell_number_sample)
> Normalized_mean<-normalize(Cell_number_sample[,-1],digits=2)
> Normalized_mean<-data.frame(data.matrix(Normalized_mean))
> # Plot with default parameters
> cybar_plot(Normalized_mean,Cell_number_sample[,-1])
```

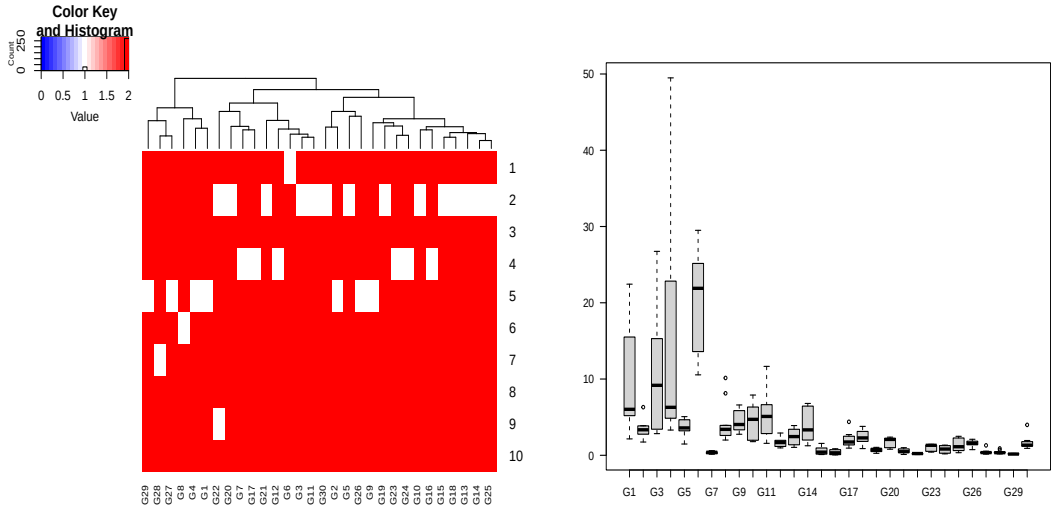

Figure 1: Barcode and boxplot using default parameters

4.2.2 Investigation ensuing from explicit starting conditions

To investigate the cell number variation ensuing from explicit starting conditions the normalized values created by using the method `"first"` should be used (see section 4.1 Table 3). Figure 2 shows the pictures using default parameters. Now, the spreading in the color key is not well balanced.

```
> Normalized_first<-normalize(Cell_number_sample[,-1],method="first",digits=2)
> Normalized_first<-data.frame(data.matrix(Normalized_first))
> # Plot with default parameters
> cybar_plot(Normalized_first,Cell_number_sample[,-1])
```

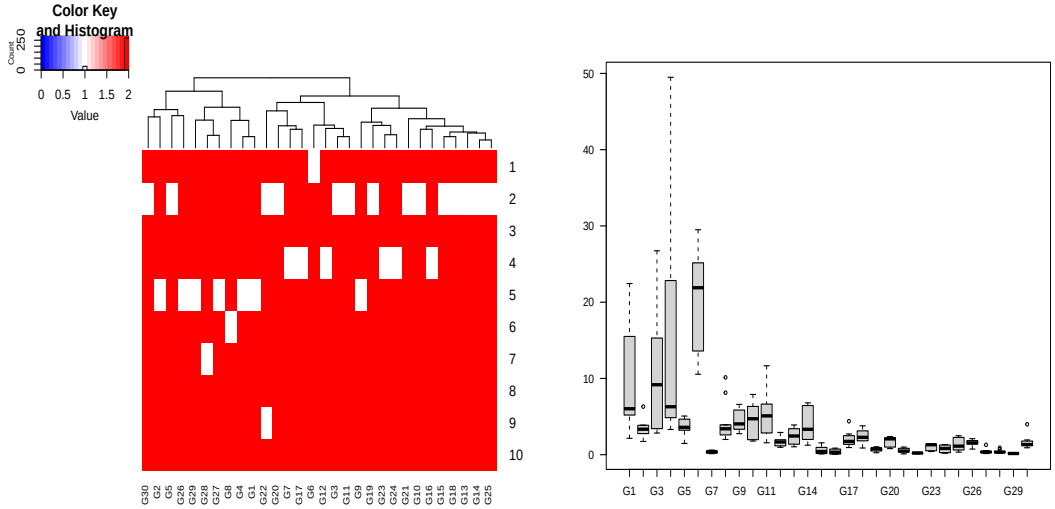

Figure 2: Barcode and boxplot using default parameters, spreading in the color key is not well balanced (see histogram)

For visualizing the barcode of the normalized values it is very important to look at the range of the data. If the range is set to large/close (see Figure 2) or simply wrong the barcode will not give useful information. The spreading of the histogram within the color key should be should be well balanced. The range of the data can be changed using the corresponding parameters of the function. In this example most of the values are ranging from 0 and 3.5 so the end of the range needs to be adjusted. The beginning of the range is set to 0 by default so it is not necessary to change it.

```
> # Plot with changed range
> cybar_plot(x=Normalized_first,to=3.5)
```

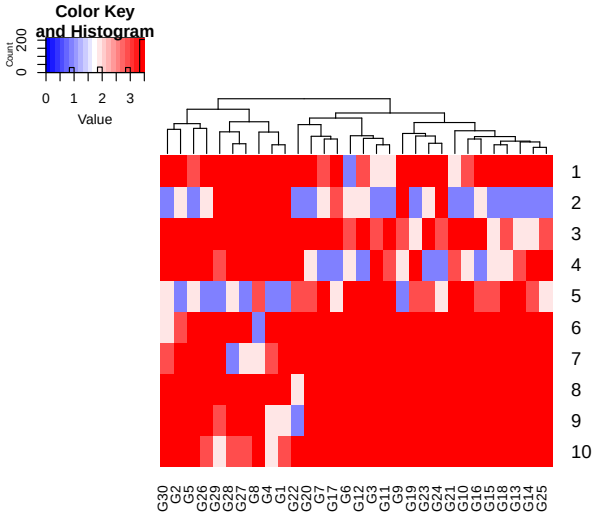

Figure 3: Range of the data is set correct

4.2.3 Improve visualization

There are several parameters having an influence on the results and being useful for improving the visualization. By looking at the boxplots of Figure 1 and Figure 2 not all data labels are visible due to the horizontal orientation. For better visualization the parameter used for the orientation can be changed to `"vert"`. The order of the boxes is the same as within the dataset but unlike the order within the barcode. The parameter used for the order of the boxes can be changed to `"sim"` which means that the boxes are ordered according to the similarity as calculated in the heatmap.

```
> # Plot with different parameters
> cybar_plot(Normalized_mean,Cell_number_sample[,-1],labels="vert",order="sim")
```

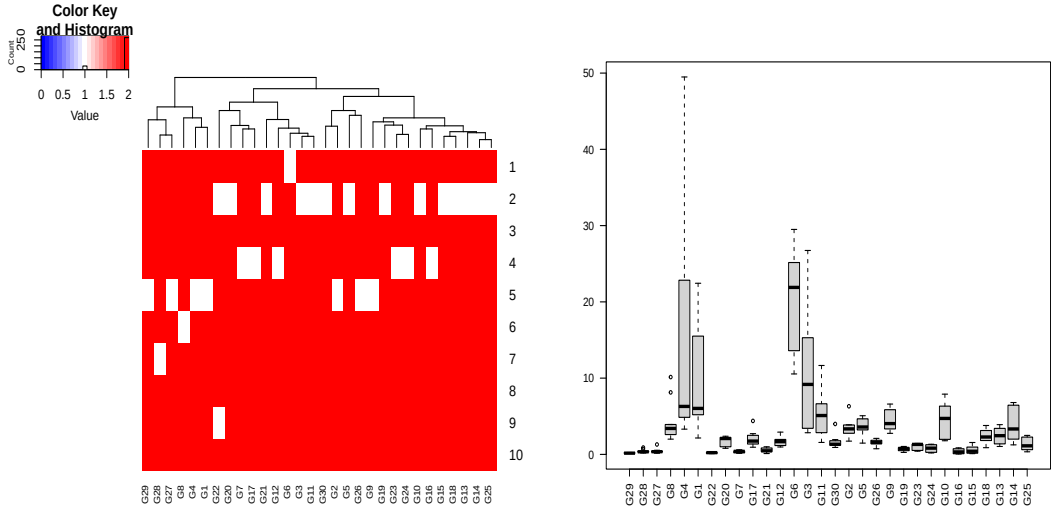

Figure 4: Vertical data labels and boxes ordered according to the similarity in the barcode, using values normalized by method `"mean"`

```
> # Plot with different parameters
> cybar_plot(Normalized_first,Cell_number_sample[,-1],to=3.5,labels="vert",order="sim")
```

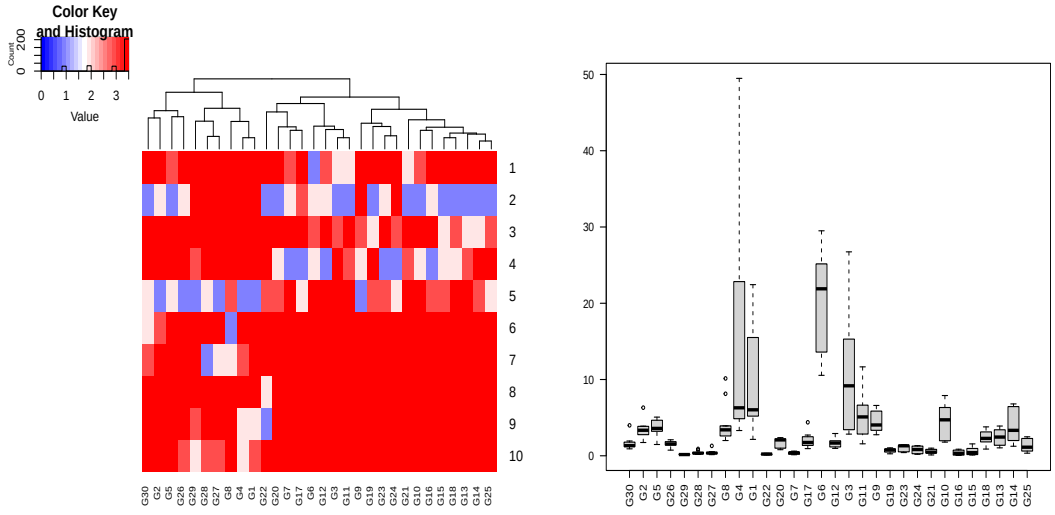

Figure 5: Using values normalized by method `"first"`, same changes as described in Figure 4

In Figure 4 the rows of the barcode are arranged according to the dataset. This order can be changed according to the similarity of the samples. In this case it is possible to add another dendrogram to the plot. Of course, the number of gradiations of the color code can be changed. In addition, a title can be added to the plot. The boxplot is not created in this example.

```
> # Barcode with ordered rows
> cybar_plot(x=Normalized_mean,Rowv=TRUE,grad=15,dendrogram="both",
+ barmain="Normalized cell numbers")
```

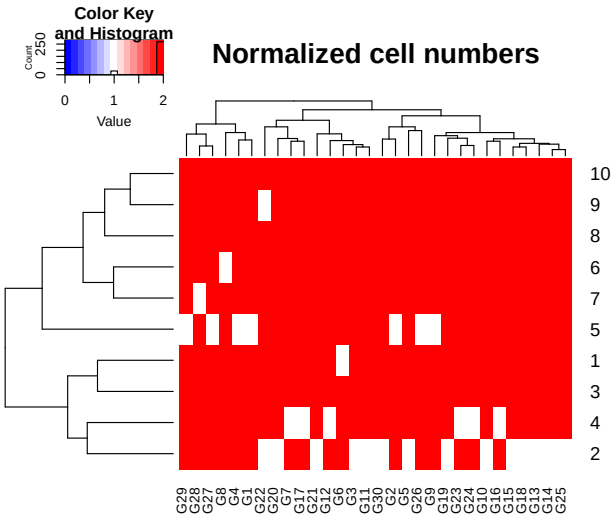

Figure 6: A dendrogram for the rows is added, less gradiations are used and a title is added to the plot

If the original data include e.g. sample names (see Table 4) these names can be added to the barcode instead of the row numbers by changing the row names of the normalized data. The number and the order of the rows have to be identical in both datasets. The columns are selected by using box brackets. The dataset `Cell_number_sample` (see Table 4) is used in this example, again.

```
> # Change row names
> rownames(Normalized_mean)<-Cell_number_sample[,1]
> # Plot
> cybar_plot(Normalized_mean,Cell_number_sample[,-1],
+ barmain="Normalized cell numbers",labels="vert",order="sim",boxmain="Cell numbers")
```

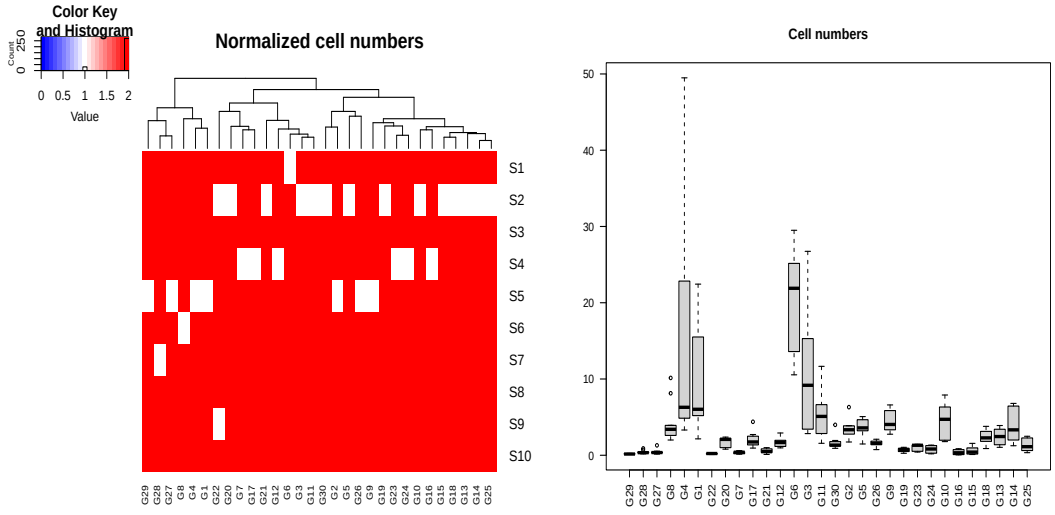

Figure 7: Sample names used as row names, titles added to the plots

The result is the same as shown in Figure 4 except that the row names within the barcode are changed to the sample names.

4.3 nmds

NMDS (nonmetric multidimensional scaling) is a mathematical technique to visualize the distances between the samples. The distances within the plot are used to be in accordance with the dis-/similarities of the respective samples. The NMDS plot can be created using the measured relative cell numbers, the percental cell numbers or the normalized cell numbers, depending on their respective values. Figure 8 and Figure 9 will show a possible difference. At first, the percental cell numbers contained in the dataset `Cell_number_sample` (see Table 1) are used for demonstration. By default the plotting symbols are black circles. The position of the text is to the right of the data points.

```
> # NMDS plot of percental cell numbers
> nmds(Cell_number_sample[,-1])
```

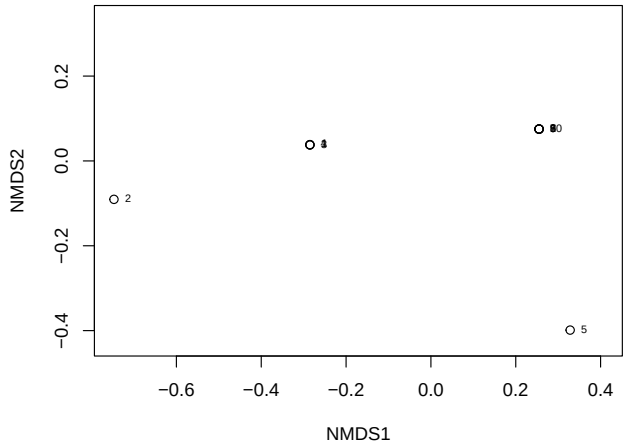

Figure 8: NMDS plot of percental cell numbers using default parameters

The NMDS plot looks distorted and is not very useful for evaluation. One possible reasons for that is the existence of high abundance and low abundance gates. In this case it is better to use the normalized values. Therefore, the normalized values as shown in Table 2 are used in the next example. The NMDS plot looks substantially better and should be preferred for evaluation.

```
> Normalized_mean<-normalize(Cell_number_sample[,-1],digits=2)
> Normalized_mean<-data.frame(data.matrix(Normalized_mean))
> # NMDS plot of normalized cel numbers
> nmds(Normalized_mean)
```

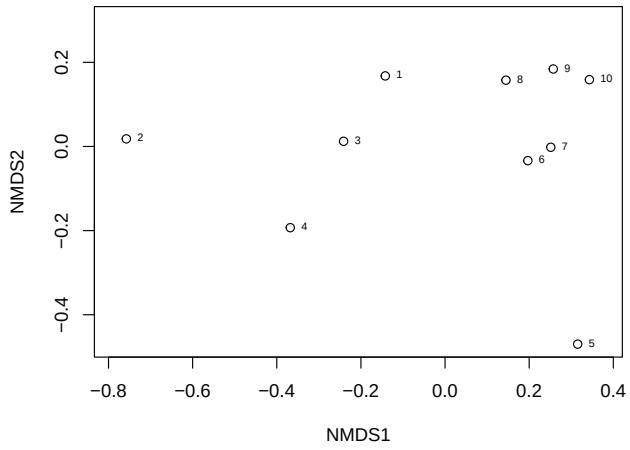

Figure 9: NMDS plot of normalized cell numbers using default parameters

Instead of the default parameters the plotting symbols can be changed e.g. to red triangles with lines connecting the data points. The text can be set above the data points and a title can be added to the plot.

```
> # NMDS plot with different parameters as described
> nmds(Normalized_mean,main="NMDS plot of
+ normalized cell numbers",type="b",pos=3,pch=2,col="red")
```

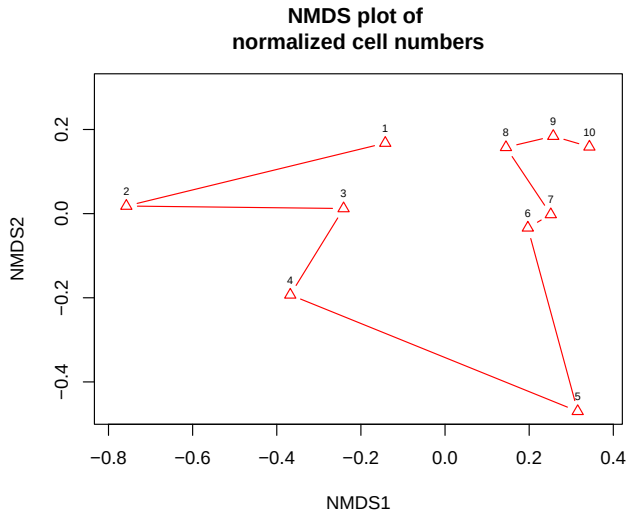

Figure 10: NMDS plot using red triangles, lines, text above the data points and a title

The samples can be classified into groups manually. Therefore, every sample has to be assigned to one group. In this example 10 samples are assigned to three different groups using the data frame `groups`. The data frame looks as follows:

```
> # Create data frame with group assignments
> groups<-data.frame("groups"=c(1,1,1,1,2,3,3,3,3,3))
```

	groups
1	1
2	1
3	1
4	1
5	2
6	3
7	3
8	3
9	3
10	3

Table 5: Group assignments

This data frame has ten entries and describes the assignment of every sample to one group (three groups overall in this example). Every sample is assigned to one value (1-25) so the same value means the same group. Up to 25 groups are possible (as only 25 plotting symbols are available). It is also possible to read a text file containing as many lines as samples whereas every line contains one value that assigns the sample to the respective group. The samples can be grouped by e.g. treatment, day of sampling or personal preferences.

```
> # NMDS plot with groups
> nmds(Normalized_mean,group=groups,main="NMDS plot of
+ normalized cell numbers")
```

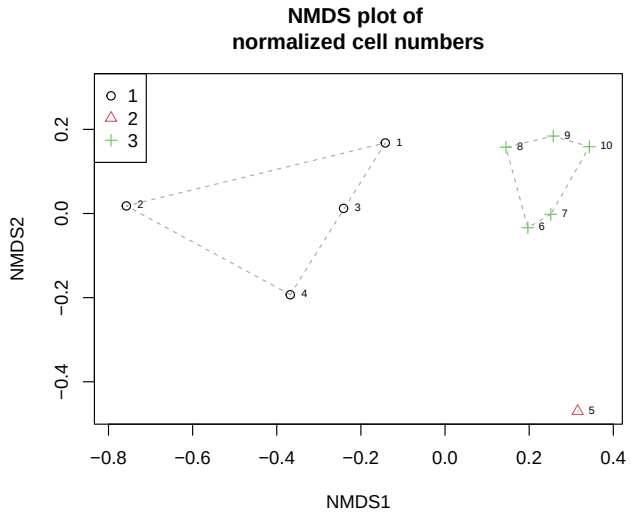

Figure 11: NMDS plot with groups

In addition, abiotic parameters/gate information can be added to the NMDS plot suggesting the differences found within the plot. In this example the attached dataset `Abiotic_data_sample` is used for demonstration. Due to the fact that normalized values were preferably used for demonstration in this section this dataset contains normalized values of each gate (method=`"mean"`) and abiotic data for every sample. The content without sample names looks as follows (excerpt):

```
> # Load dataset
> data(Abiotic_data_sample)
> # Show data
> Abiotic_data_sample[,-1]
```

	G1	...	G30	delta_B1	delta_B2	...	COD	...	redox_pot
1	1.26	...	2.44	-0.08	-0.04	...	431	...	638.8
2	2.21	...	0.56	-0.02	0.01	...	543	...	607.4
3	1.53	...	1.19	0.12	0.19	...	948	...	663.0
4	2.07	...	1.10	0.10	0.23	...	1710	...	657.6
5	0.21	...	0.73	0.28	0.47	...	1420	...	661.0
6	0.65	...	0.74	0.51	0.67	...	1890	...	686.6
7	0.53	...	0.75	0.76	0.94	...	1210	...	677.2
8	0.54	...	0.86	0.87	1.11	...	1140	...	669.0
9	0.50	...	0.81	0.97	1.25	...	1240	...	682.2
10	0.51	...	0.81	1.11	1.45	...	1060	...	683.6

Table 6: Table with normalized values and abiotic parameters

If the percental cell numbers are used for the NMDS plot the table looks like shown in Table 8 (see section 4.4).

The significance level of the gates/abiotic parameters is set to 0.05 and the color used for plotting is set to `"magenta"` by default.

```
> # NMDS plot with groups and abiotic parameters
> nmds(Normalized_mean,group=groups,main="NMDS plot of
+ normalized cell numbers",abiotic=Abiotic_data_sample[,-1],verbose=TRUE)
```

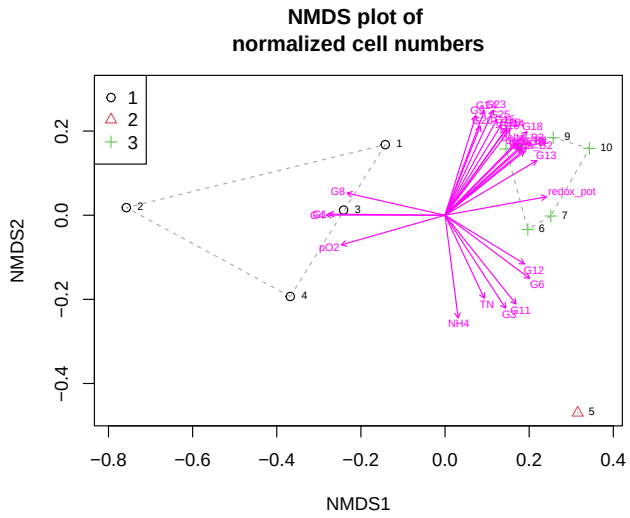

Figure 12: NMDS plot with groups, abiotic parameters and gate information

The abiotic parameters/gate information that are added to the plot can be selected manually using box brackets as described in section 4.1. This may be useful if there is a high number of abiotic parameters/gate information but not all are of interest. In this example only the most significant (p=0.05) abiotic parameters (columns 32-44) are added to the plot. The function `ncol()` basically returns the number of columns so typing 44 instead would also work.

```
> # NMDS plot of selected abiotic parameters
> nmds(Normalized_mean,group=groups,main="NMDS plot of
+ normalized cell numbers",abiotic=Abiotic_data_sample[,32:ncol(Abiotic_data_sample)],
+ verbose=TRUE)
```

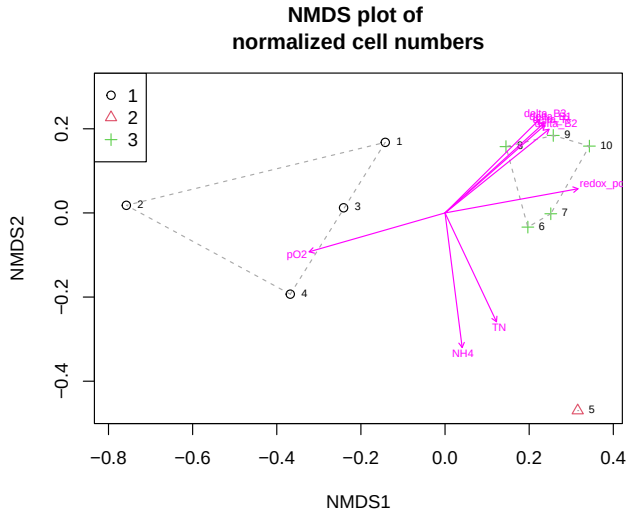

Figure 13: NMDS plot with selected abiotic parameters

In the next example only the most significant (p=0.05) gate information (columns 2-31) are added to the plot.

```
> # NMDS plot with selected gates
> nmds(Normalized_mean,group=groups,main="NMDS plot of
+ normalized cell numbers",abiotic=Abiotic_data_sample[,2:31],verbose=TRUE)
```

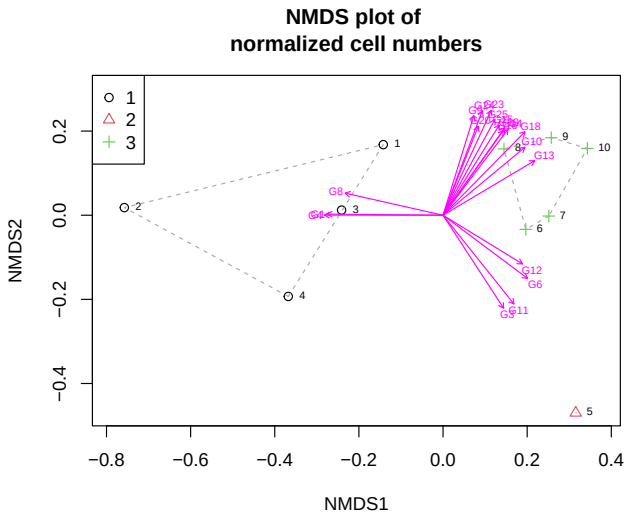

Figure 14: NMDS plot with selected gate information

Now, the legend is placed to the top right corner of the plot, the color of the arrows is changed to `"darkred"` and the maximum p-value is set to 0.01. All abiotic parameters/gate information less or equal the maximum p-value are added to the plot. Only 15 parameters (former: 26, see Figure 12) are plotted due to changing the maximum p-value. The number of parameters may change due the results of the NMDS plot are a little bit different everytime the function is called.

```
> # NMDS plot
> nmds(Normalized_mean,group=groups,main="NMDS plot of
+ normalized cell numbers",legend_pos="topright",abiotic=Abiotic_data_sample[,-1],
+ p.max=0.01,col_abiotic="darkred",verbose=TRUE)
```

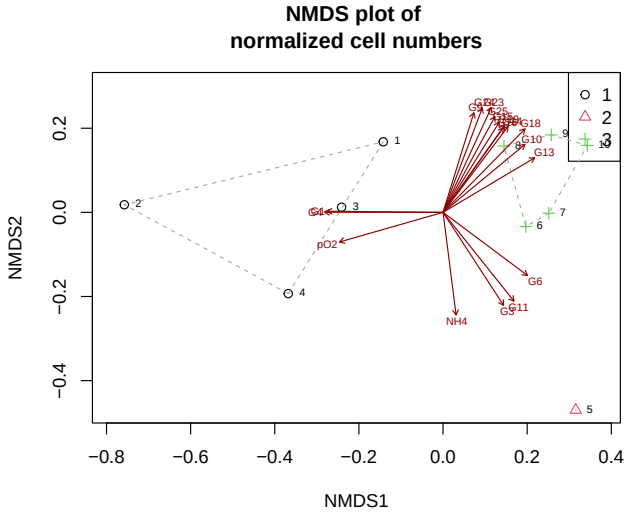

Figure 15: NMDS plot with changed maximum p-value

It may be useful to plot the sample names instead of the row names as text of the data points. Since the attached dataset `Abiotic_data_sample` also contains the sample names in the right order the row names of the data frame containing the normalized values are changed according to the first column of `Abiotic_data_sample`.

```
> # Load datasets
> data(Abiotic_data_sample)
> # Show data
> Abiotic_data_sample
```

The content of this dataset actually looks as follows (excerpt):

	sample	G1	...	G30	delta_B1	delta_B2	...	COD	...	redox_pot
1	S1	1.26	...	2.44	-0.08	-0.04	...	431	...	638.8
2	S2	2.21	...	0.56	-0.02	0.01	...	543	...	607.4
3	S3	1.53	...	1.19	0.12	0.19	...	948	...	663.0
4	S4	2.07	...	1.10	0.10	0.23	...	1710	...	657.6
5	S5	0.21	...	0.73	0.28	0.47	...	1420	...	661.0
6	S6	0.65	...	0.74	0.51	0.67	...	1890	...	686.6
7	S7	0.53	...	0.75	0.76	0.94	...	1210	...	677.2
8	S8	0.54	...	0.86	0.87	1.11	...	1140	...	669.0
9	S9	0.50	...	0.81	0.97	1.25	...	1240	...	682.2
10	S10	0.51	...	0.81	1.11	1.45	...	1060	...	683.6

Table 7: Abiotic data with sample names

```
> # Change row names
> rownames(Normalized_mean)<-Abiotic_data_sample[,1]
> # NMDS plot with sample names
> nmds(Normalized_mean,group=groups,main="NMDS plot of
+ normalized cell numbers",legend_pos="topright",abiotic=Abiotic_data_sample[,-1],
+ p.max=0.01,col_abiotic="darkred",verbose=TRUE)
```

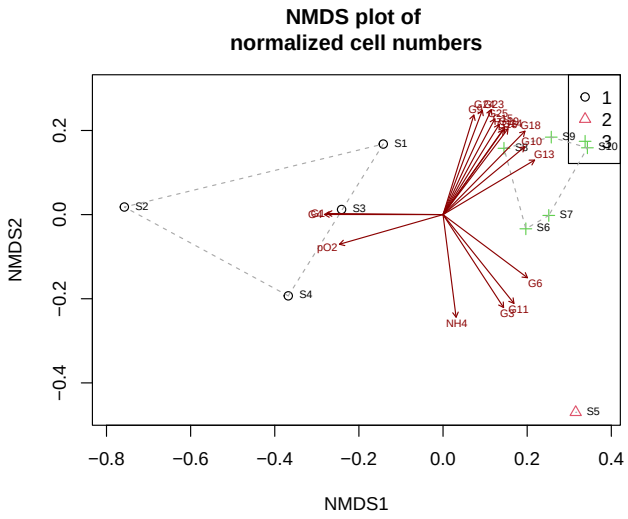

Figure 16: NMDS plot with sample names as text of the data points

4.4 correlation

The relationships between the measured relative or percental cell numbers and the abiotic parameters are investigated using correlation analysis. It is recommended to use the measured relative or percental cell numbers for this procedure instead of the normalized cell numbers. The attached dataset `Corr_data_sample` is used for demonstration in the following section.

```
> # Load dataset
> data(Corr_data_sample)
> # Show correlation values
> Corr_data_sample[,-1]
```

The content without sample names looks as follows (excerpt):

	G1	...	G30	delta_B1	delta_B2	...	COD	...	redox_pot
1	12.75	...	3.99	-0.08	-0.04	...	431	...	638.8
2	22.44	...	0.91	-0.02	0.01	...	543	...	607.4
3	15.51	...	1.94	0.12	0.19	...	948	...	663.0
4	20.99	...	1.79	0.10	0.23	...	1710	...	657.6
5	2.51	...	1.19	0.28	0.47	...	1420	...	661.0
6	6.62	...	1.20	0.51	0.67	...	1890	...	686.6
7	5.38	...	1.23	0.76	0.94	...	1210	...	677.2
8	5.44	...	1.41	0.87	1.11	...	1140	...	669.0
9	5.06	...	1.33	0.97	1.25	...	1240	...	682.2
10	5.20	...	1.33	1.11	1.45	...	1060	...	683.6

Table 8: Table with correlation values

By default Spearman's rank-order correlation coefficient is used for calculating the correlation values and their p-values, respectively [Günther et al., 2012]. The rows and columns are ordered according to similarity. In addition dendrograms are shown to reveal clusters within the data. For visualizing the correlation values a color key with 21 gradiations is used ranging from blue (strong negative correlation) to red (strong positive correlation). Neutral correlations are shown in white.

```
> # Run correlation analysis
> correlation(Corr_data_sample[,-1])
```

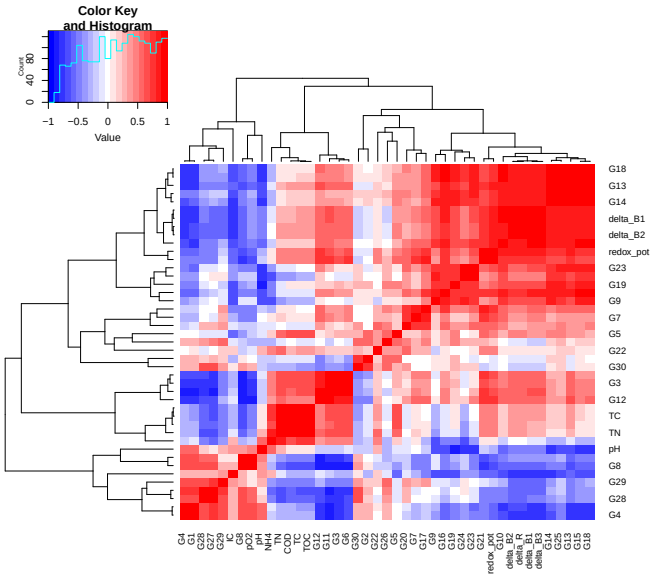

Figure 17: Heatmap of correlation values using default parameters

The color key can be changed using different color palettes of R or defining an own colorRampPalette (see Details of this function within the package). The dendrograms can be omitted and a title can be added to the plot.

```
> # Plot with different parameters as described
> correlation(Corr_data_sample[,-1],colkey=topo.colors(16),dendrogram="column")
```

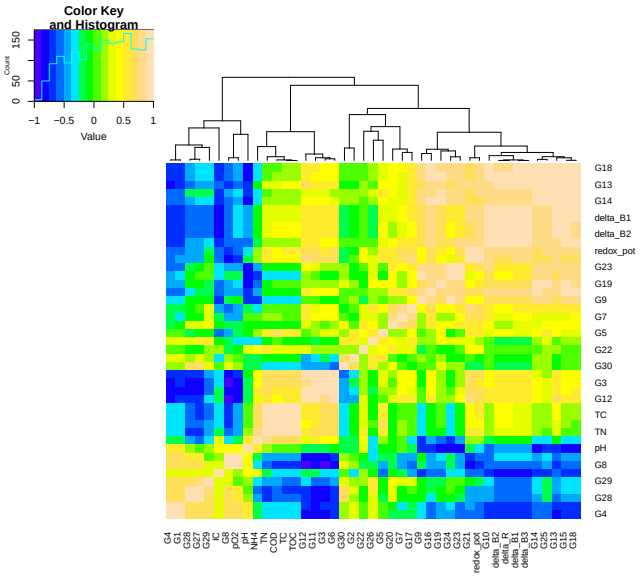

Figure 18: Using topographic color palette, omit row dendrogram

```
> # Plot
> correlation(Corr_data_sample[,-1],colkey=cm.colors(15),dendrogram="row")
```

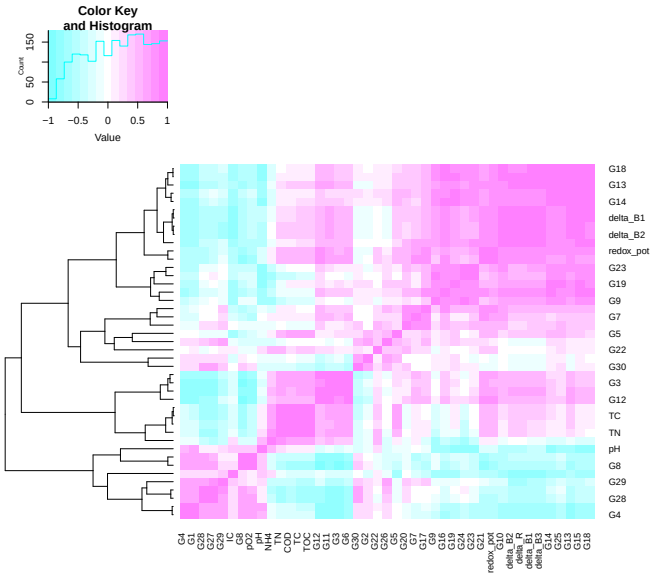

Figure 19: Using color palette `cm`, omit column dendrogram

```
> # Plot
> correlation(Corr_data_sample[,-1],colkey=heat.colors(11),dendrogram="none",
+ main="Correlation of Corr_data_sample")
```

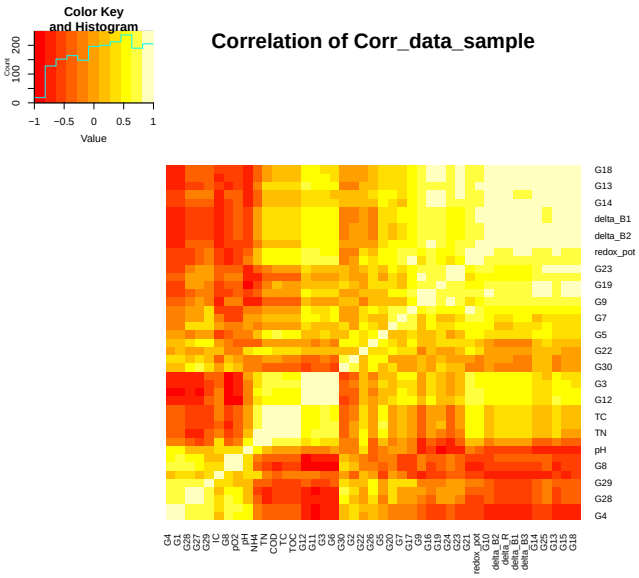

Figure 20: Using color palette **heat**, omit dendrograms and add a title

Sometimes it is useful to arrange the gates/abiotic parameters according to the order used in the table, not according to similarity. This can be done for the rows, the columns or both.The corresponding dendrograms are not shown in this case as they would not be reasonable.

```
> # Plot
> correlation(Corr_data_sample[,-1],Rowv=FALSE,Colv=FALSE,dendrogram="none")
```

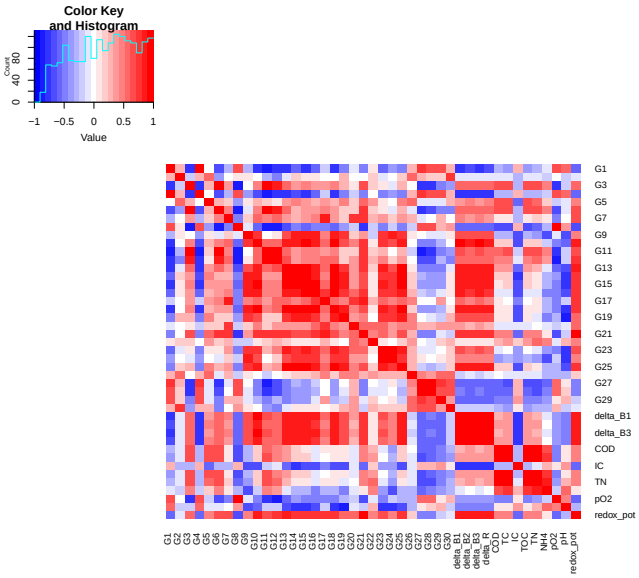

Figure 21: Rows and columns ordered according to the table

The columns used for the correlation analysis can be selected manually. This is important if e.g. one column is used for sample names or if the correlation of specific gates/abiotic parameters is of particular interest. The selection is again realized using box brackets. For demonstration the attached dataset `Corr_data_sample` is used, again.

The content actually looks as follows (excerpt):

```
> # Load dataset
> data(Corr_data_sample)
> # Show data
> Corr_data_sample
```

	sample	G1	...	G30	delta_B1	delta_B2	...	COD	...	redox_pot
1	S1	12.75	...	3.99	-0.08	-0.04	...	431	...	638.8
2	S2	22.44	...	0.91	-0.02	0.01	...	543	...	607.4
3	S3	15.51	...	1.94	0.12	0.19	...	948	...	663.0
4	S4	20.99	...	1.79	0.10	0.23	...	1710	...	657.6
5	S5	2.51	...	1.19	0.28	0.47	...	1420	...	661.0
6	S6	6.62	...	1.20	0.51	0.67	...	1890	...	686.6
7	S7	5.38	...	1.23	0.76	0.94	...	1210	...	677.2
8	S8	5.44	...	1.41	0.87	1.11	...	1140	...	669.0
9	S9	5.06	...	1.33	0.97	1.25	...	1240	...	682.2
10	S10	5.20	...	1.33	1.11	1.45	...	1060	...	683.6

Table 9: Correlation values with sample names

As the correlation can only be computed using numerical values the first column was omitted in the previous examples. To select specific columns used for correlation analysis the simplest way is to count the columns. In this example the colums 2-11 (G1-G10), the columns 32-34 (delta_B1, delta_B2, delta_B3) and the last three columns (pO2,pH,redox_pot) are used for the calculation.

```
> # Select columns manually using concatenation
> correlation(Corr_data_sample[,c(2:11,32:34,42:ncol(Corr_data_sample))])
```

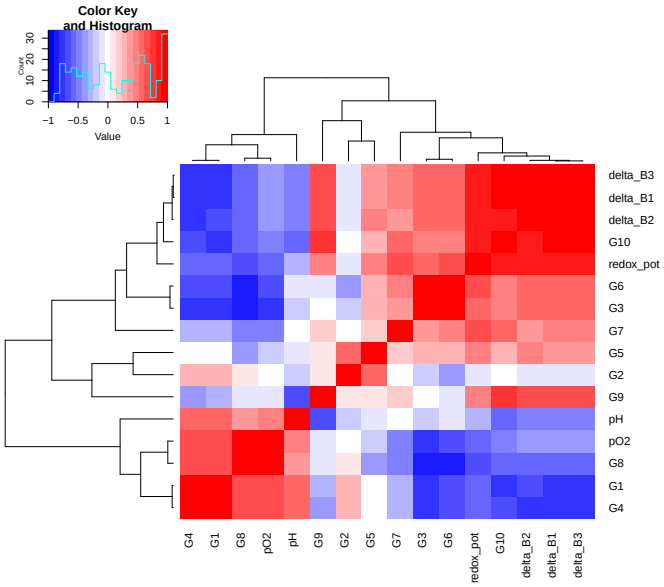

Figure 22: Correlation of selected gates/abiotic parameters

It is possible to select the columns by their names, too. This could be helpful if there is a very high number of colums. Instead of counting the columns as done in the last example they are now selected be their names. The result is the same as shown in Figure 22.

```
> # Select columns manually using concatenation of their names
> correlation(Corr_data_sample[,c("G1","G2","G3","G4","G5","G6","G7",
+ "G8","G9","G10","delta_B1","delta_B2","delta_B3","pO2","pH","redox_pot")])
```

Sometimes it is useful to export a table with the calculated estimates or the p-values that are used for creating the heatmap. By setting `verbose=TRUE` the estimates, the p-values and additional information on the heatmap are printed to the R console. A list of these outputs can be saved in a new variable (`cor` in this example).

```
> # Save list of results in new variable
> cor<-correlation(Corr_data_sample[,-1],verbose=TRUE)
```

The entries of the list can be accessed manually and saved as a new file. Typing `cor["est"]` returns the estimates, `cor["pvals"]` returns the p-values and `cor["heat"]` returns additional information on the heatmap.

```
> # Show estimates
> cor["est"]
> # Show p-values
> cor["pvals"]
```

The entries can now be written in a new file.

```
> # Write a new file containing the estimates
> write.table(cor["est"],file="estimates.txt",row.names=TRUE,
+ quote=FALSE,sep="\t")
```

References

[Günther et al., 2012] Günther, S., Koch, C., Hübschmann, T., Röske, I., Müller, R. A., Bley, T., Harms, H., and Müller, S. (2012). Correlation of community dynamics and process parameters as a tool for the prediction of the stability of wastewater treatment. *Environmental Science & Technology*, 46(1):84–92.

[Koch et al., 2013a] Koch, C., Fetzer, I., Schmidt, T., Harms, H., and Müller, S. (2013a). Monitoring functions in managed microbial systems by cytometric bar coding. *Environmental Science & Technology*, 47(3):1753–1760.

[Koch et al., 2013b] Koch, C., Günther, S., Desta, A. F., Hübschmann, T., and Müller, S. (2013b). Cytometric fingerprinting for analysing microbial intra-community structure variation and identifying sub-community function. *Nature Protocols*, 8(1):190–202.

[Koch et al., 2014] Koch, C., Harms, H., and Müller, S. (2014). Dynamics in the microbial cytome - single cell analytics in natural systems. *Current Opinion in Biotechnology*, 27:134–141.