MetaboDiff tutorial

Comparative non-targeted metabolomics comes of age through an increasing number of commercial vendors offering reproducible high-quality metabolomic data for translational researchers outside the mass spectrometry field. The MetaboDiff package aims to provide a low-level entry to differential metabolomic analysis by starting off with the table of relative metabolite quantifications provided by commercial vendors or core facilities.

# Getting started

The MetaboDiff R package requires R version 3.4.

## Installing dependencies

CRAN occasionally fails to compile the WGCNA package for Mac OS X. Hence, it is recommended to install the package before installing MetaboDiff:

install.packages("WGCNA")

If asked, install the package from source. Alternatively you might need to download the package from the [developer](https://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/Rpackages/WGCNA/) and install the package locally:

install.packages(path\_to\_file, repos = NULL, type="source")

If you encounter problems installing WGCNA, please refer to the [developer page](https://labs.genetics.ucla.edu/horvath/CoexpressionNetwork/Rpackages/WGCNA/). Please note that MetaboDiff can only be installed if WGCNA was successfully installed.

## Installing MetaboDiff

MetaboDiff can be installed via Github

library("devtools")  
install\_github("andreasmock/MetaboDiff")

and once installed loaded by

library("MetaboDiff")

# Part I: Data processing

## Reading data and annotation

### Example data

The example data is derived from a study by Priolo and colleagues in which they used the service of Metabolon® to compare the tissue metabolome of 40 prostate cancers with 16 normal prostate specimens[[1]](#footnote-28).

# load example data  
data = as.matrix(read.csv(system.file("extdata",   
 "Priolo\_data.csv",   
 package = "MetaboDiff")))

### Input files

The metabolomic data within MetaboDiff are stored as a MultiAssayExperiment class[[2]](#footnote-31). This framework enables the coordinated representation of multiple experiments on partially overlapping samples with associated metadata and integrated subsetting across experiments. In the context of metabolomic data analysis, multiple assays are needed to store raw data and imputed data which usually contain different number of metabolites due to missing values (see section on data imputation for more details).

The core components of the MultiAssayExperiment class are:

* ExperimentList - a slot of class ExperimentList containing data for each experimental assay. Within the ExperimentList slot, the metabolomic data are stored as  
  SummarizedExperiment objects consisting of:
  + assay - a matrix containing the relative metabolic measurements.
  + rowData - a dataframe containing the metabolite annotation.
* colData - a slot of class dataframe describing the sample metadata available across all experiments.
* sampleMap - a slot of class dataframe relating clinical data to experimental assay.
* metadata - a slot of class list. Within MetaboDiff, this slot contains a list of dataframes summarizing the results from the comparative data analysis.

Please refer to the MultiAssayExperiment vignette for more information.

#### Creation of assay object containing metabolomic data

assay = apply(data[4:nrow(data),8:ncol(data)],2,as.numeric)  
colnames(assay) = paste0(rep("sample",ncol(assay)),1:ncol(assay))  
rownames(assay) = paste0(rep("met",nrow(assay)),1:nrow(assay))

knitr::kable(assay[1:4,1:5])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | sample1 | sample2 | sample3 | sample4 | sample5 |
| met1 | 33964.73 | 117318.43 | 118856.90 | 78670.7 | 102565.94 |
| met2 | 18505.56 | 167585.32 | 59621.97 | 66220.4 | 74892.27 |
| met3 | NA | 42373.93 | 27141.21 | NA | 38390.78 |
| met4 | 61638.77 | 74595.78 | NA | NA | NA |

#### Creation of colData object containing sample metadata

colData = data.frame(id = colnames(data[,8:ncol(data)]),  
 tumor\_normal = as.vector(t(data[1,8:ncol(data)])),  
 row.names=paste0(rep("pat",ncol(assay)),1:ncol(assay)))

To showcase the full functionality of MetaboDiff, a second random group label "random\_gender" is generated.

#set seed for reproducibility   
set.seed(1)  
  
colData$random\_gender = sample(c("random\_male","random\_female"),size = nrow(colData),replace = TRUE)

knitr::kable(head(colData))

|  |  |  |  |
| --- | --- | --- | --- |
|  | id | tumor\_normal | random\_gender |
| pat1 | cp2 | N | random\_male |
| pat2 | cp7 | N | random\_male |
| pat3 | cp19 | N | random\_female |
| pat4 | cp26 | N | random\_female |
| pat5 | cp29 | N | random\_male |
| pat6 | cp32 | N | random\_female |

#### Creation of rowData object containing metabolite annotations and ids

The example data set comprises metabolite annotation provided by the commercial vendor:

rowData = as.data.frame(data[4:nrow(data),1:7])  
colnames(rowData) = data[3,1:7]  
colnames(rowData)[7] = "HMDB\_ID"  
rownames(rowData) = paste0(rep("met",nrow(assay)),1:nrow(assay))  
colnames(rowData)

## [1] "BIOCHEMICAL" "SUPER\_PATHWAY" "SUB\_PATHWAY" "METABOLON\_ID"   
## [5] "PLATFORM" "KEGG\_ID" "HMDB\_ID"

#### Annotation using Small Molecular Pathway Database

Alongside the metabolic measurements, Metabolon® provides metabolite annotation including so called super-pathways and sub-pathways. However, these annotations might very from vendor to vendor. Hence, the MetaboDiff package makes use of the Small Molecular Pathway Database (SMPDB) for metabolite annotation[[3]](#footnote-36). MetaboDiff supports all common metabolic ids as input for the annotation (HMDB, KEGG and ChEBI). In the example data, both the KEGG and the HMDB identifier are available. As the databases (HMDB, KEGG or ChEBI) cover unique annotations due to different standards in identifying and reporting metabolites[[4]](#footnote-37), the function get\_SMPDBanno queries the database using all three ids (if available) and joins all available information. The current SMPDB build used within MetaboDiff is version 2.0 and will be updated as new versions are released. Please refer to the development branch of MetaboDiff to work with the latest SMPDB annotation.

rowData = get\_SMPDBanno(rowData,  
 column\_kegg\_id=6,  
 column\_hmdb\_id=7,  
 column\_chebi\_id=NA)

### Creation of SummarizedExperiment and ExperimentList object

se = SummarizedExperiment(assays=assay,  
 rowData=rowData)  
experiment\_list = list(raw=se)  
experiment\_list

## $raw  
## class: SummarizedExperiment   
## dim: 307 86   
## metadata(0):  
## assays(1): ''  
## rownames(307): met1 met2 ... met306 met307  
## rowData names(22): BIOCHEMICAL SUPER\_PATHWAY ... SMPDB|InChI  
## SMPDB|InChI.Key  
## colnames(86): sample1 sample2 ... sample85 sample86  
## colData names(0):

### Creation of sampleMap object

sampleMap = data.frame(primary=rownames(colData),  
 colname=colnames(se))  
sampleMap\_list = listToMap(list(raw=sampleMap))

### Construction of MultiAssayExperiment object

met = MultiAssayExperiment(experiments = experiment\_list,  
 colData = colData,  
 sampleMap = sampleMap\_list)  
met

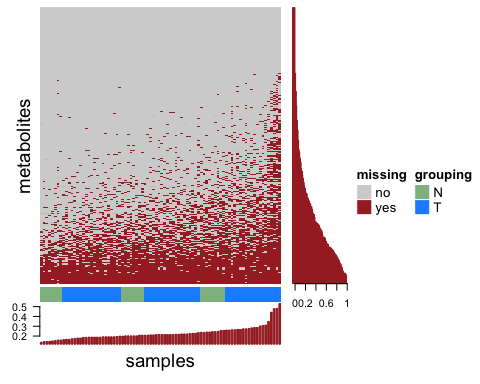
## A MultiAssayExperiment object of 1 listed  
## experiment with a user-defined name and respective class.   
## Containing an ExperimentList class object of length 1:   
## [1] raw: SummarizedExperiment with 307 rows and 86 columns   
## Features:   
## experiments() - obtain the ExperimentList instance   
## colData() - the primary/phenotype DataFrame   
## sampleMap() - the sample availability DataFrame   
## `$`, `[`, `[[` - extract colData columns, subset, or experiment   
## \*Format() - convert ExperimentList into a long or wide DataFrame   
## assays() - convert ExperimentList to a SimpleList of matrices

## Imputation of missing values

In contrast to other high-throughput technologies, missing values are common in quantitative metabolomic datasets.

The function na\_heatmap visualizes the missing metabolite measurements across the samples. The name of the column in colData for grouping and the label colors for the two groups need to be specified.

na\_heatmap(met,  
 group\_factor="tumor\_normal",  
 label\_colors=c("darkseagreen","dodgerblue"))



Missing metabolic measurements across the example data set. Missing measurements are visualized by a binary heatmap and barplots summarizing the fraction of missing measurement per metabolite and sample, respectively. In addition, the group label of interest (tumor (T) vs. normal (N)) is visualized. By default, metabolite measurements are only imputed if less than 40 percent are missing.

The example data supports the need for data imputation (figure @ref(fig:heat)). It could be shown that k-nearest neighbor imputation minimizes the effects on the normality and variance of the data as long as the number of missing data does not exceed 40%[[5]](#footnote-44).

The function knn\_impute adds the slot "impute" to the MultiAssayExperiment object that contains the imputed relative metabolite measurements for all metabolites with raw measurements in more than 60% of cases. We recommend a cutoff of 40% (i.e. 0.4). However the cutoff might be changed according to the discretion of the user.

met = knn\_impute(met,cutoff=0.4)  
met

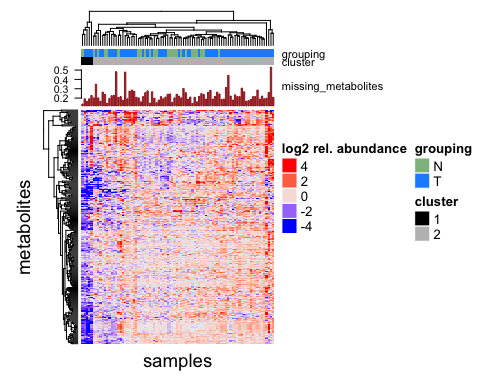
## A MultiAssayExperiment object of 2 listed  
## experiments with user-defined names and respective classes.   
## Containing an ExperimentList class object of length 2:   
## [1] raw: SummarizedExperiment with 307 rows and 86 columns   
## [2] imputed: SummarizedExperiment with 238 rows and 86 columns   
## Features:   
## experiments() - obtain the ExperimentList instance   
## colData() - the primary/phenotype DataFrame   
## sampleMap() - the sample availability DataFrame   
## `$`, `[`, `[[` - extract colData columns, subset, or experiment   
## \*Format() - convert ExperimentList into a long or wide DataFrame   
## assays() - convert ExperimentList to a SimpleList of matrices

As apparent form the summary description of the object met the imputed data matrix contains only 238 of the 307 original metabolites according the cutoff of 40% missing values.

## Removal of outliers

Before we normalize the data, we want to exclude outliers in the study set. To this end, the function outlier\_heatmap is provided. The sample annotation shows the number of missing metabolites per sample as a proxy of the impact of imputation on clustering. To showcase outliers, the hierarchical clustering tree is cut in 2 clusters (figure @ref(fig:outlier)).

outlier\_heatmap(met,  
 group\_factor="tumor\_normal",  
 label\_colors=c("darkseagreen","dodgerblue"))



Hierarchical clustering of metabolite measurements. The tree is cut in 2 clusters. The user has the choice to exclude one cluster which he thinks might represent outliers. To determine the effect of imputation, a barplot displaying the fraction of missing metabolites is shown in the column annotation of the heatmap.

The imputed data of the example study set displays a cluster of 5 samples (cluster 1) with in average lower relative metabolite measurements. Due to the lack of batch information, this cannot be investigated further at this time. To demonstrate how a cluster can be removed, we apply the function remove\_cluster to remove cluster 1:

met = remove\_cluster(met,cluster=1)  
met

## A MultiAssayExperiment object of 2 listed  
## experiments with user-defined names and respective classes.   
## Containing an ExperimentList class object of length 2:   
## [1] raw: SummarizedExperiment with 307 rows and 81 columns   
## [2] imputed: SummarizedExperiment with 238 rows and 81 columns   
## Features:   
## experiments() - obtain the ExperimentList instance   
## colData() - the primary/phenotype DataFrame   
## sampleMap() - the sample availability DataFrame   
## `$`, `[`, `[[` - extract colData columns, subset, or experiment   
## \*Format() - convert ExperimentList into a long or wide DataFrame   
## assays() - convert ExperimentList to a SimpleList of matrices

As displayed in the summary of the met object, the 5 samples of cluster 1 were successfully removed from the slots "raw" and "imputed".

## Normalization

Variance stabilizing normalization (vsn) is used to ensure that the variance remains nearly constant over the measured spectrum[[6]](#footnote-49).

met = normalize\_met(met)

## Summary of processed object

met

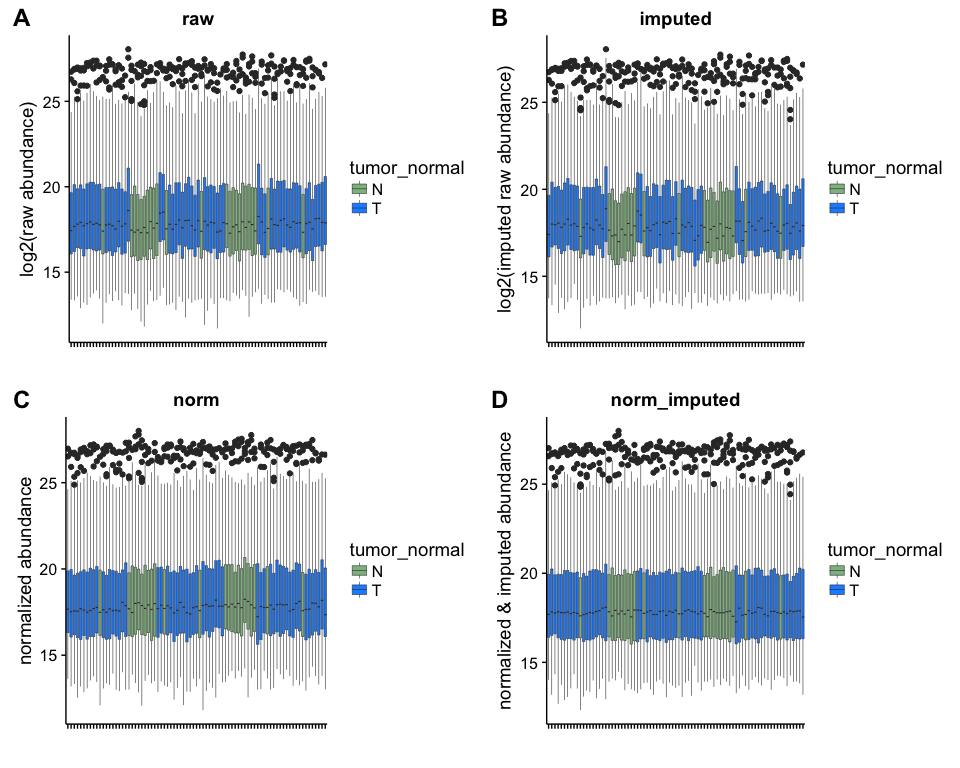
## A MultiAssayExperiment object of 4 listed  
## experiments with user-defined names and respective classes.   
## Containing an ExperimentList class object of length 4:   
## [1] raw: SummarizedExperiment with 307 rows and 81 columns   
## [2] imputed: SummarizedExperiment with 238 rows and 81 columns   
## [3] norm: SummarizedExperiment with 307 rows and 81 columns   
## [4] norm\_imputed: SummarizedExperiment with 238 rows and 81 columns   
## Features:   
## experiments() - obtain the ExperimentList instance   
## colData() - the primary/phenotype DataFrame   
## sampleMap() - the sample availability DataFrame   
## `$`, `[`, `[[` - extract colData columns, subset, or experiment   
## \*Format() - convert ExperimentList into a long or wide DataFrame   
## assays() - convert ExperimentList to a SimpleList of matrices

At this point the data processing is completed with the MultiAssayExperiment object containing 4 slots:

* raw - raw relative metabolic measurements as provided by company or core facility
* imputed - imputed relative metabolic measurements (k-nearest neighbor imputation)
* norm - normalized relative metabolic measurements (vsn)
* norm\_imputed - normalized and imputed relative metabolic measurements (vsn)

## Quality control of normalization

quality\_plot(met,  
 group\_factor="tumor\_normal",  
 label\_colors=c("darkseagreen","dodgerblue"))



Quality control plot. Boxplot displaying the distribution of (A) raw, (B) imputed (C) normalized and (D) imputed and normalized relative metabolite measurements for all samples of the study set. Boxplots are colored according to the grouping of interest, i.e. tumor vs. normal.

The quality control plots shows the distribution of raw and normalized metabolic measurements for every sample in the study set. As aimed, the distribution of measurements is comparable across the study set for the normalized and imputed data (figure @ref(fig:quality)D).

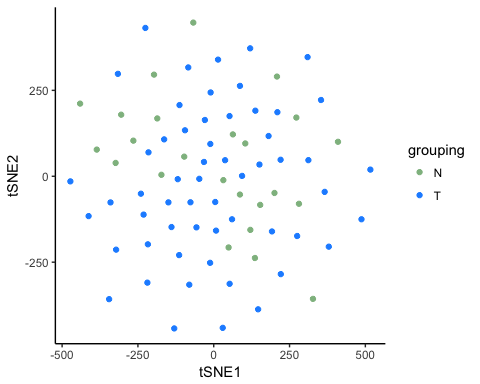
# Part II: Data analysis

Part II of MetaboDiff comprises a set of functions for comparative data analysis that are purpose-built for the preprocessed MultiAssayExperiment object. All statistics obtained in this part will be saved in the metadata slot of the object. Certain plotting functions will only be available once the correponding statistics has been run, e.g. you will only be able to plot the volcano plot after executing the function for differential analysis (diff\_test). Please note that the data analysis has been developed for metabolomic data sets of more than 100 metabolites. Hence, some functionality might not be available with smaller data sets < 50 metabolites.

## tSNE analysis

To explore the metabolomic profiles between samples in an unsupervised fashion, tSNE analysis is performed as a non-linear dimension reduction technique.

tsne\_plot(met,  
 group\_factor="tumor\_normal",  
 label\_colors=c("darkseagreen","dodgerblue"))



tSNE plot. Samples are colored according to the group factor.

A plot of the first two dimensions (figure @ref(fig:tSNE)) does not reveal a distinct difference in the metabolomic profiles between the normal and tumor samples.

## Hypothesis testing

Differential analysis for individual metabolites is performed using Student's T-Test. Correction for multiple testing is performed by independent hypothesis weighting (IHW[[7]](#footnote-57)) with variance as a covariate.

Hypothesis testing will be performed for the sample grouping "tumor\_normal", as well as for the randomly generated grouping "random\_gender".

met = diff\_test(met,  
 group\_factors = c("tumor\_normal","random\_gender"))

The results of the hypothesis testing is saved in the metadata slot of the MultiAssayExperiment object.

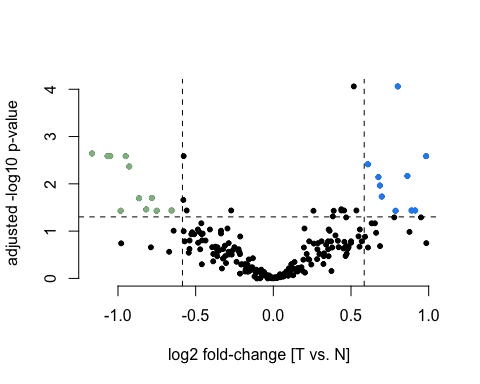
str(metadata(met), max.level=2)

## List of 2  
## $ ttest\_tumor\_normal :'data.frame': 238 obs. of 4 variables:  
## ..$ pval : num [1:238] 0.0206 0.7808 0.0832 0.0432 0.5859 ...  
## ..$ adj\_pval : num [1:238] 0.18 0.911 0.219 0.158 0.716 ...  
## ..$ fold\_change: num [1:238] 0.2872 0.0366 -0.3936 -0.5391 -0.1646 ...  
## ..$ var : num [1:238] 0.264 0.287 0.872 1.21 1.516 ...  
## $ ttest\_random\_gender:'data.frame': 238 obs. of 4 variables:  
## ..$ pval : num [1:238] 0.2318 0.8626 0.4048 0.0121 0.2111 ...  
## ..$ adj\_pval : num [1:238] 0.83 0.959 0.862 0.386 0.83 ...  
## ..$ fold\_change: num [1:238] 0.1372 0.0208 -0.1742 -0.607 -0.3438 ...  
## ..$ var : num [1:238] 0.264 0.287 0.872 1.21 1.516 ...

The columns of the result dataframe are the unadjusted p-value (pval), the adjusted p-value by independent hypothesis weighting (adj\_pval), the Fold-Change between groups (fold\_change) and the variance (var).

The comparative analysis can be visualized by means of a volcano plot (figure @ref(fig:volcano1)).

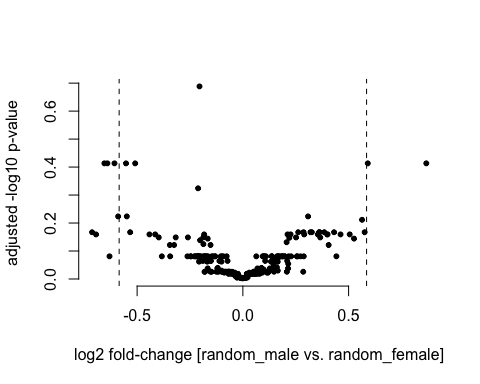
volcano\_plot(met,   
 group\_factor="tumor\_normal",  
 label\_colors=c("darkseagreen","dodgerblue"))



Volcano plot of comparison tumor vs normal. The dashed lines mark the significance thresholds adjusted p-value<0.05 and fold-change>1.5. Significant metabolites are colored.

As a sanity check, we also display the volcano plot for the random grouping "random\_gender" for which we do not expect the same number of significant metabolites.

volcano\_plot(met,   
 group\_factor="random\_gender",  
 label\_colors=c("brown","orange"))



Volcano plot of random comparison. The dashed lines mark the significance thresholds adjusted p-value<0.05 and fold-change>1.5.

As expected, no metabolite was significantly different between the randomly assigned grouping male vs. female after multiple testing with a cutoff of p-value<0.05 (figure @ref(fig:volcano2)).

## Pathway analysis

In this part of the analysis, we will exploit the Small Molecular Pathway Database (SMPDB) annotation to explore metabolic pathways and perform an enrichment analysis.

The annotations from the SMPDB have the prefix "SMPDB|". The remaining annotation was provided by the commercial vendor that generated the example data (i.e. Metabolon®).

colnames(rowData(met[["norm\_imputed"]]))

## [1] "BIOCHEMICAL" "SUPER\_PATHWAY"   
## [3] "SUB\_PATHWAY" "METABOLON\_ID"   
## [5] "PLATFORM" "KEGG\_ID"   
## [7] "HMDB\_ID" "SMPDB|SMPDB.ID"   
## [9] "SMPDB|Pathway.Name" "SMPDB|Pathway.Subject"  
## [11] "SMPDB|Metabolite.ID" "SMPDB|Metabolite.Name"  
## [13] "SMPDB|HMDB.ID" "SMPDB|KEGG.ID"   
## [15] "SMPDB|ChEBI.ID" "SMPDB|DrugBank.ID"   
## [17] "SMPDB|CAS" "SMPDB|Formula"   
## [19] "SMPDB|IUPAC" "SMPDB|SMILES"   
## [21] "SMPDB|InChI" "SMPDB|InChI.Key"

### Variance boxplots

For instance, this annotation can be used to explore the variance in pathways across the data set (figure @ref(fig:varbox)).

variance\_boxplot(met,  
 rowAnnotation = "SMPDB|Pathway.Name")



Pathway-wide distribution of normalized relative metabolite measurements. The pathway annotation from the Small Molecular Pathway Database (SMPDB) was used.

MetaboDiff refrains from 'classical' pathway enrichment analyses, as they are limited by the need to define a priori metabolite sets for evaluation and ignore the topology of interactions. Hence, MetaboDiff offers the analysis of metabolic correlation networks.

## Metabolic correlation network analysis

This section describes the generation and analysis of metabolic correlation networks. The workflow was adapted from the weighted gene co-expression analysis (WGCNA) proposed by Langfelder and Horvarth[[8]](#footnote-65).

### Construction dissimilarity matrix

The first step in constructing a metabolic correlation network is the creation of a dissimilarity matrix. Biweight midcorrelation was used as a similiarity measure as it is more robust to outliers than the absolute correlation coefficient[[9]](#footnote-68). This choice is important, as we do not expect metabolites to be correlated in all patients.

The core concept of the so called "weighted" correlation analysis by Langfelder and Horvarth is that instead of defining a "hard" threshold (e.g. an absolute correlation coefficient > 0.8) to decide whether a node to connected to another, the adjacency a is defined by raising the similarity s to a power beta ("soft" threshold):

Lastly, the dissimilarity measure w is defined by

For detailed rational of this approach, please see Zhang and Horvath[[10]](#footnote-70). For metabolic networks, we identified that a beta value of 3 was the lowest power for which the scale-free topology of the topology was met.

The function diss\_matrix creates the dissimilarity measure for the met objects and saves it in the metadata slot

met = diss\_matrix(met)  
str(metadata(met), max.level=1)

## List of 3  
## $ ttest\_tumor\_normal :'data.frame': 238 obs. of 4 variables:  
## $ ttest\_random\_gender:'data.frame': 238 obs. of 4 variables:  
## $ diss\_matrix : num [1:238, 1:238] 0 0.964 0.999 0.999 0.996 ...  
## ..- attr(\*, "dimnames")=List of 2

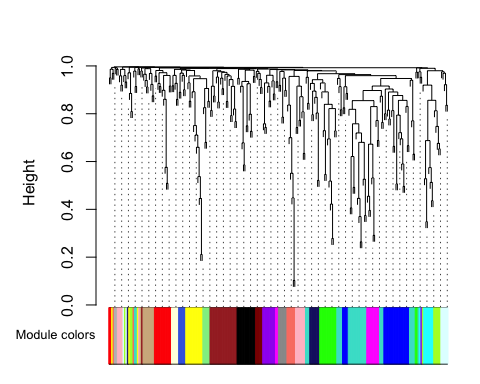
### Identification of metabolic correlation modules

To identify metabolic correlation modules, metabolites are next clustered based on the dissimilarity measure where branches of the dendrogram correspond to modules. Ultimately, modules are detected by applying a branch cutting method with a minimal module size of 5 metabolites. We employed the dynamic branch cut method developed by Langfelder and colleagues[[11]](#footnote-73), as constant height cutoffs exhibit suboptimal performance on complicated dendrograms. Figure @ref(fig:WGCNA) shows the hierarchical clustering and corresponding modules after branch cutting.

met = identify\_modules(met,   
 min\_module\_size=5)

## ..cutHeight not given, setting it to 0.991 ===> 99% of the (truncated) height range in dendro.  
## ..done.

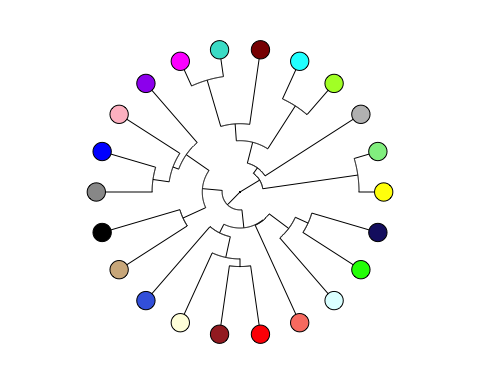
WGCNA::plotDendroAndColors(metadata(met)$tree,   
 metadata(met)$module\_color\_vector,   
 'Module colors',   
 dendroLabels = FALSE,   
 hang = 0.03,  
 addGuide = TRUE,   
 guideHang = 0.05, main='')



Hierarchical clustering of metabolites. The different colors represent the modules identified by the dynamic branch cutting method.

The relation between the identified metabolic correlation modules can be visualized by a dendrogram of their *eigengenes* (figure @ref(fig:eigengenes)). The module *eigengene* is defined as the first principal component of its expression matrix. It could be shown that the module *eigengene* is highly correlated with the gene that has the highest intramodular connectivity[[12]](#footnote-76).

par(mar=c(2,2,2,2))  
ape::plot.phylo(ape::as.phylo(metadata(met)$METree),  
 type = 'fan',  
 show.tip.label = FALSE,   
 main='')  
ape::tiplabels(frame = 'circle',  
 col='black',   
 text=rep('',length(unique(metadata(met)$modules))),   
 bg = WGCNA::labels2colors(0:21))



Hierarchy of metabolic correlation modules as revealed by the clustering of module eigengenes. Each node represents a metabolic correlation module.

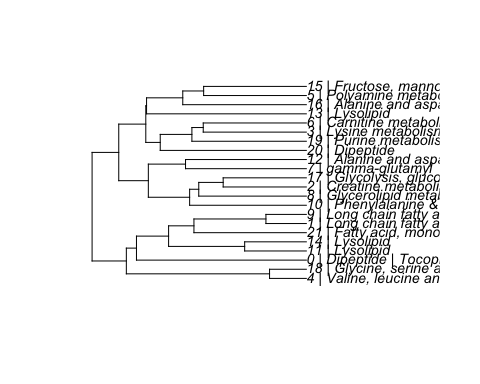
# number of metabolites per module  
table(metadata(met)$modules)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21   
## 2 30 22 20 15 14 14 13 12 11 10 9 9 8 8 8 7 6 5 5 5 5

### Name metabolic correlation modules

To enable a better interpretation of metabolic correlation modules, modules are named according to the most abundant pathway annotation in a module (figure @ref(fig:naming)).

# calculate module significance  
met = name\_modules(met,  
 pathway\_annotation = "SUB\_PATHWAY")  
  
# plot phylogram with names  
ape::plot.phylo(ape::as.phylo(metadata(met)$METree), cex=0.9)



Hierarchy of metabolic correlation modules. Modules are named according to the most abundant pathway annotation in a module. Module 0 comprises the two metabolites without a significant interaction.

### Relation of metabolic correlation modules to sample traits

An advantage of correlation network analysis is the possibility to integrate external information. At the lowest hierarchical level, *gene significance* (GS) measures can be defined as the statistical significance (i.e. p-value, ) between the -th node profile (metabolite) and the sample trait

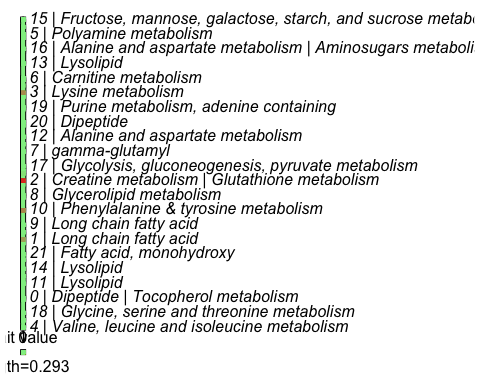
*Module significance* in turn can be determined as the average absolute gene significance measure. This conceptual framework can be adapted to any research question.

met = calculate\_MS(met,  
 group\_factors = c("tumor\_normal","random\_gender"))  
str(metadata(met), max.level=1)

## List of 13  
## $ ttest\_tumor\_normal :'data.frame': 238 obs. of 4 variables:  
## $ ttest\_random\_gender:'data.frame': 238 obs. of 4 variables:  
## $ diss\_matrix : num [1:238, 1:238] 0 0.964 0.999 0.999 0.996 ...  
## ..- attr(\*, "dimnames")=List of 2  
## $ tree :List of 7  
## ..- attr(\*, "class")= chr "hclust"  
## $ modules : num [1:238] 3 5 13 1 11 1 11 11 13 1 ...  
## $ module\_color\_vector: chr [1:238] "brown" "green" "salmon" "turquoise" ...  
## $ MEs :'data.frame': 81 obs. of 22 variables:  
## $ MEDiss : num [1:22, 1:22] 0 1.038 1.073 1.055 0.973 ...  
## ..- attr(\*, "dimnames")=List of 2  
## $ METree :List of 7  
## ..- attr(\*, "class")= chr "hclust"  
## $ module\_colors : chr [1:22] "yellow" "lightgreen" "grey" "greenyellow" ...  
## $ MM : num [1:238, 1:22] 0.32992 0.00722 0.09016 0.00808 0.12118 ...  
## ..- attr(\*, "dimnames")=List of 2  
## $ MS\_tumor\_normal :'data.frame': 22 obs. of 2 variables:  
## $ MS\_random\_gender :'data.frame': 22 obs. of 2 variables:

Figure @ref(fig:MSplot) shows that metabolic correlation module 2 (Creatine metabolism / Glutathione metabolism) was significantly associated with the tumor vs. normal comparison in the example data.

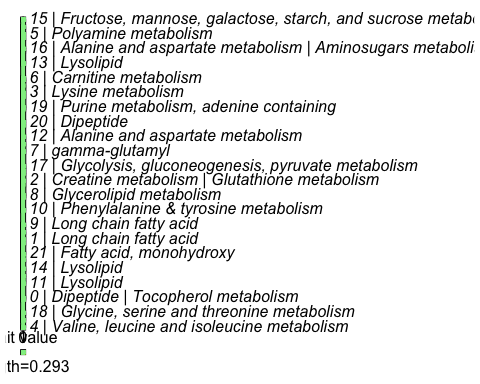
MS\_plot(met,  
 group\_factor="tumor\_normal")



Association of tumor vs normal trait with metabolic correlation modules. Module 2 (Creatine metabolism / Glutathione metabolism) showed a significant module significance.

In line with the volcano plots, the random grouping did not result in a significant association (figure @ref(fig:MSplot2))

MS\_plot(met,  
 group\_factor="random\_gender")



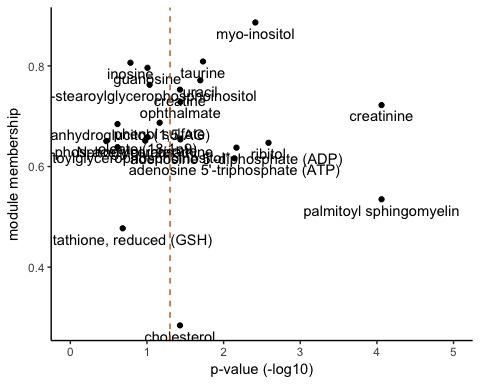
Association of random trait with metabolic correlation modules. No significant module significance can be observed.

### Explore module of interest

To explore the role of metabolites within a correlation module, Langfelder and Horvath suggest a 'fuzzy' measure of *module membership* defined as

where is the profile of gene and is the eigengene of module . Based on this definition, describes how closely related metabolite is to module . A meaningful visualization is consequently plotting the module membership over the p-value of the respective GS measure. As a third dimension, the dot-size is weighted according to the effect size, i.e. absolute fold-change between the grouping of interest.

MOI\_plot(met,  
 group\_factor="tumor\_normal",  
 MOI=2) + xlim(c(0,5))



Exploration of module of interest. Module membership is plotted over the -log10 p-value. The dashed line marks the significance threshold of adjusted p-value<0.05.

# Session information

sessionInfo()

## R version 3.4.0 (2017-04-21)  
## Platform: x86\_64-apple-darwin15.6.0 (64-bit)  
## Running under: macOS Sierra 10.12.6  
##   
## Matrix products: default  
## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/libBLAS.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_GB.UTF-8/en\_GB.UTF-8/en\_GB.UTF-8/C/en\_GB.UTF-8/en\_GB.UTF-8  
##   
## attached base packages:  
## [1] grid parallel stats4 stats graphics grDevices utils   
## [8] datasets methods base   
##   
## other attached packages:  
## [1] ape\_4.1 WGCNA\_1.60   
## [3] fastcluster\_1.1.22 dynamicTreeCut\_1.63-1   
## [5] IHW\_1.4.0 factoextra\_1.0.4   
## [7] cluster\_2.0.6 cowplot\_0.7.0   
## [9] genefilter\_1.58.1 RColorBrewer\_1.1-2   
## [11] HiveR\_0.2.55 vsn\_3.44.0   
## [13] impute\_1.50.1 ggfortify\_0.4.1   
## [15] dplyr\_0.7.2 purrr\_0.2.2.2   
## [17] readr\_1.1.1 tidyr\_0.6.3   
## [19] tibble\_1.3.3 tidyverse\_1.1.1   
## [21] gdata\_2.18.0 BiocStyle\_2.4.1   
## [23] rmarkdown\_1.6 MetaboDiff\_0.1.0   
## [25] devtools\_1.13.3 ggplot2\_2.2.1   
## [27] ComplexHeatmap\_1.14.0 MultiAssayExperiment\_1.2.1  
## [29] SummarizedExperiment\_1.6.3 DelayedArray\_0.2.4   
## [31] matrixStats\_0.52.2 Biobase\_2.37.2   
## [33] GenomicRanges\_1.28.2 GenomeInfoDb\_1.12.0   
## [35] IRanges\_2.10.2 S4Vectors\_0.14.3   
## [37] BiocGenerics\_0.22.0   
##   
## loaded via a namespace (and not attached):  
## [1] shinydashboard\_0.6.1 robust\_0.4-18   
## [3] RSQLite\_2.0 AnnotationDbi\_1.38.1   
## [5] htmlwidgets\_0.8 combinat\_0.0-8   
## [7] trimcluster\_0.1-2 munsell\_0.4.3   
## [9] animation\_2.5 codetools\_0.2-15   
## [11] preprocessCore\_1.38.1 withr\_1.0.2   
## [13] colorspace\_1.3-2 BiocInstaller\_1.27.2   
## [15] highr\_0.6 knitr\_1.16   
## [17] robustbase\_0.92-7 labeling\_0.3   
## [19] slam\_0.1-40 GenomeInfoDbData\_0.99.0  
## [21] lpsymphony\_1.4.1 mnormt\_1.5-5   
## [23] bit64\_0.9-7 rprojroot\_1.2   
## [25] coda\_0.19-1 clusterGeneration\_1.3.4  
## [27] diptest\_0.75-7 R6\_2.2.2   
## [29] doParallel\_1.0.10 flexmix\_2.3-14   
## [31] bitops\_1.0-6 assertthat\_0.2.0   
## [33] scales\_0.4.1 nnet\_7.3-12   
## [35] gtable\_0.2.0 affy\_1.54.0   
## [37] phangorn\_2.2.0 rlang\_0.1.1   
## [39] scatterplot3d\_0.3-40 GlobalOptions\_0.0.12   
## [41] splines\_3.4.0 lazyeval\_0.2.0   
## [43] acepack\_1.4.1 broom\_0.4.2   
## [45] checkmate\_1.8.2 yaml\_2.1.14   
## [47] reshape2\_1.4.2 modelr\_0.1.0   
## [49] backports\_1.1.0 httpuv\_1.3.3   
## [51] Hmisc\_4.0-3 tools\_3.4.0   
## [53] psych\_1.7.5 affyio\_1.46.0   
## [55] Rcpp\_0.12.12 plyr\_1.8.4   
## [57] base64enc\_0.1-3 zlibbioc\_1.22.0   
## [59] RCurl\_1.95-4.8 rpart\_4.1-11   
## [61] GetoptLong\_0.1.6 viridis\_0.4.0   
## [63] haven\_1.0.0 ggrepel\_0.6.5   
## [65] magrittr\_1.5 data.table\_1.10.4   
## [67] circlize\_0.4.0 mvtnorm\_1.0-6   
## [69] whisker\_0.3-2 hms\_0.3   
## [71] mime\_0.5 evaluate\_0.10   
## [73] xtable\_1.8-2 XML\_3.98-1.7   
## [75] jpeg\_0.1-8 mclust\_5.3   
## [77] readxl\_1.0.0 gridExtra\_2.2.1   
## [79] shape\_1.4.2 compiler\_3.4.0   
## [81] maps\_3.2.0 htmltools\_0.3.6   
## [83] pcaPP\_1.9-72 Formula\_1.2-1   
## [85] rrcov\_1.4-3 expm\_0.999-2   
## [87] lubridate\_1.6.0 DBI\_0.7   
## [89] MASS\_7.3-47 fpc\_2.1-10   
## [91] Matrix\_1.2-10 quadprog\_1.5-5   
## [93] igraph\_1.0.1 bindr\_0.1   
## [95] forcats\_0.2.0 pkgconfig\_2.0.1   
## [97] fit.models\_0.5-14 numDeriv\_2016.8-1   
## [99] foreign\_0.8-68 xml2\_1.1.1   
## [101] foreach\_1.4.3 annotate\_1.54.0   
## [103] XVector\_0.16.0 rvest\_0.3.2   
## [105] stringr\_1.2.0 digest\_0.6.12   
## [107] tsne\_0.1-3 phytools\_0.6-00   
## [109] msm\_1.6.4 fastmatch\_1.1-0   
## [111] cellranger\_1.1.0 htmlTable\_1.9   
## [113] dendextend\_1.5.2 kernlab\_0.9-25   
## [115] shiny\_1.0.3 gtools\_3.5.0   
## [117] modeltools\_0.2-21 rjson\_0.2.15   
## [119] nlme\_3.1-131 jsonlite\_1.5   
## [121] bindrcpp\_0.2 viridisLite\_0.2.0   
## [123] limma\_3.32.2 lattice\_0.20-35   
## [125] plotrix\_3.6-5 httr\_1.2.1   
## [127] DEoptimR\_1.0-8 survival\_2.41-3   
## [129] GO.db\_3.4.1 glue\_1.1.1   
## [131] fdrtool\_1.2.15 UpSetR\_1.3.3   
## [133] png\_0.1-7 prabclus\_2.2-6   
## [135] iterators\_1.0.8 bit\_1.1-12   
## [137] class\_7.3-14 stringi\_1.1.5   
## [139] blob\_1.1.0 latticeExtra\_0.6-28   
## [141] memoise\_1.1.0

1. Priolo, C., Pyne, S., Rose, J., Regan, E. R., Zadra, G., Photopoulos, C., et al. (2014). AKT1 and MYC Induce Distinctive Metabolic Fingerprints in Human Prostate Cancer. Cancer Research, 74(24), 7198–7204. <http://doi.org/10.1158/0008-5472.CAN-14-1490> [↑](#footnote-ref-28)
2. Sig M (2017). MultiAssayExperiment: Software for the integration of multi-omics experiments in Bioconductor. R package version 1.2.1 [↑](#footnote-ref-31)
3. Jewison T, Su Y, Disfany FM, et al. SMPDB 2.0: Big Improvements to the Small Molecule Pathway Database Nucleic Acids Res. 2014 Jan;42(Database issue):D478-84. [↑](#footnote-ref-36)
4. Thiele, I., Swainston, N., Fleming, R. M. T., Hoppe, A., Sahoo, S., Aurich, M. K., et al. (2013). A community-driven global reconstruction of human metabolism. Nature Biotechnology, 31(5), 419–425. <http://doi.org/10.1038/nbt.2488> [↑](#footnote-ref-37)
5. Armitage, E. G., Godzien, J., Alonso-Herranz, V., L pez-Gonz lvez, N., & Barbas, C. (2015). Missing value imputation strategies for metabolomics data. Electrophoresis, 36(24), 3050–3060. <http://doi.org/10.1002/elps.201500352> [↑](#footnote-ref-44)
6. Huber, W., Heydebreck, von, A., Sültmann, H., Poustka, A., & Vingron, M. (2002). Variance stabilization applied to microarray data calibration and to the quantification of differential expression. Bioinformatics, 18 Suppl 1, S96–104. [↑](#footnote-ref-49)
7. Ignatiadis, N., Klaus, B., Zaugg, J. B., & Huber, W. (2016). Data-driven hypothesis weighting increases detection power in genome-scale multiple testing. Nature Methods, 13(7), 577–580. <http://doi.org/10.1038/nmeth.3885> [↑](#footnote-ref-57)
8. Langfelder, P., & Horvath, S. (2008). WGCNA: an R package for weighted correlation network analysis. BMC Bioinformatics, 9, 559–559. <http://doi.org/10.1186/1471-2105-9-559> [↑](#footnote-ref-65)
9. Zheng, C.-H., Yuan, L., Sha, W., & Sun, Z.-L. (2014). Gene differential coexpression analysis based on biweight correlation and maximum clique. BMC Bioinformatics, 15 Suppl 15(Suppl 15), S3. <http://doi.org/10.1186/1471-2105-15-S15-S3> [↑](#footnote-ref-68)
10. Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. Statistical Applications in Genetics and Molecular Biology, 4(1), Article17. <http://doi.org/10.2202/1544-6115.1128> [↑](#footnote-ref-70)
11. Langfelder, P., Zhang, B., & Horvath, S. (2008). Defining clusters from a hierarchical cluster tree: the Dynamic Tree Cut package for R. Bioinformatics, 24(5), 719–720. <http://doi.org/10.1093/bioinformatics/btm563>. [↑](#footnote-ref-73)
12. Horvath, S., & Dong, J. (2008). Geometric Interpretation of Gene Coexpression Network Analysis. PLoS Computational Biology (PLOSCB) 4(8), 4(8), e1000117–e1000117. <http://doi.org/10.1371/journal.pcbi.1000117> [↑](#footnote-ref-76)