Introduction to the *MoDentify* R package

Kieu Trinh Do[1], David Rasp[1], Gabi Kastenmueller[2,3], Karsten Suhre[4], Jan Krumsiek[1,2,5]

[1] Institute of Computational Biology, Helmholtz-Zentrum Muenchen, Neuherberg, Germany [2] German Center for Diabetes Research (DZD), Neuherberg, Germany [3] Institute of Bioinformatics and systems biology, Helmholtz-Zentrum Muenchen, Neuherberg, Germany [4] Department of Physiology and Biophysics, Weill Cornell Medical College in Qatar, Education, City, Doha, Qatar [5] Institute for Computational Biomedicine, Englander Institute for Precision Medicine, Department of Physiology and Biophysics, Weill Cornell Medicine, New York, USA

2018-01-18

Table of Contents

Introduction to *MoDentify*

Phenotype associations in large-scale, heterogeneous metabolomics data sets can be expected to be substantially complex, spanning functional modules. Functional modules are commonly defined as groups of correlating entities that are functionally coordinated, coregulated, or generally driven by a common biological process (Mitra et al. 2013). Moreover, phenotypes will associate with metabolic modules at different scales, ranging from global associations spanning entire pathways or even sets of pathways, to localized associations with only few metabolites (K. T. Do et al. 2017). *MoDentify* is an algorithm for the identification of modules at different layers of resolution (both at the fine-grained metabolite level and the more global pathway-level).

Given a network, a phenotype variable, a scoring function, and a *seed* (starting) node, a greedy search algorithm identifies an optimal module by score maximization. This module is determined by extending candidate modules along its network edges, until no further score improvement can be achieved (see Module identification).

Network inference

MoDentify searches for phenotype associated modules based on a correlation network. Depending on the chosen resolution level, networks are either inferred using metabolite-metabolite correlations or pathway-pathway correlations. To this end, Gaussian graphical models (GGMs) are estimated using the *GeneNet* R package. GGMs are based on partial correlations, which represent associations between two variables corrected for all remaining variables in multivariate Gaussian distributions (Krumsiek et al. 2011). Required covariates such as age, gender, or body mass index (BMI) can be included into the model. Nodes correspond to the variables of the data set (metabolites, or pathways), and edges between nodes are considered if both Pearson correlations and partial correlations are statistically significant with a chosen significance threshold (e.g. at α=0.05) after multiple testing correction.

Pathway representation

The pathway network is inferred by calculating a representative variable for each pathway, thereby creating a new data set with pathway representation values for each observations. *MoDentify* provides two approaches for pathway representation:

1) *eigenmetabolite* approach: For each pathway a principal component analysis (PCA) or Singular Value Decomposition (SVD) is performed after scaling all variables to a mean of 0 and a variance of 1. The first principal component - also termed *eigenmetabolite* - is used as a representative value for the entire set of variables in the pathway (Langfelder and Horvath 2007). These *eigenmetabolites* are then subjected to the network inference procedure.

2) *average* approach: All variables are scaled such that the mean is 0 and the variance is 1. Subsequently, the pathway representative is calculated as the average of all variable values in the given pathway (average z-score).

If pathways can be assumed to be homogeneous (by sharing common biological or biochemical properties), the *eigenmetabolite* approach can be used. If homogeneity cannot be guaranteed, the average approach might be a better choice. In general, it is helpful to explore the explained variances of the *eigenmetabolites* (See Explained variances). This new data set finally serves as input for the estimation of the pathway GGM.

Module representation and scoring

Based on the network, a greedy approach is used to search for modules by score maximization. The score of a candidate module M is obtained from the multivariable linear regression model:

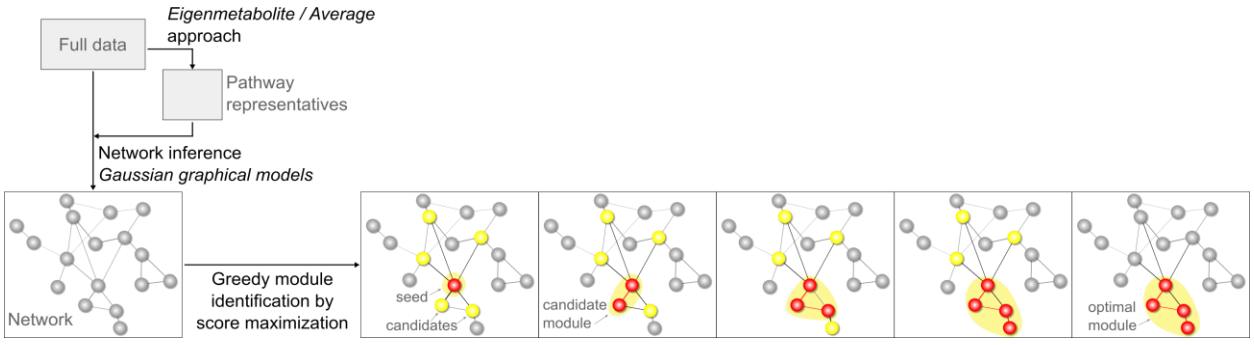
$$R_M = \beta_{M,0} + \beta_{M,1} \times P + \sum_{i=1}^{|C|} (\beta_{M,i+1} \times c_i) + \epsilon_M$$

where R_M is the module representative value, $\beta_{M,0}$ is the intercept, $\beta_{M,i}$ is the regression coefficient for the respective independent variable, P is the phenotype of interest, C is the set of covariates c_i, and ϵ_M is a normally distributed error term. The module score is then defined as the negative logarithmized p-value of the coefficient $\beta_{M,1}$, which represents the magnitude of phenotype association. Notably, the score of a single component equals its negative logarithmized p-value from a univariate analysis.

R_M, the module representative can either be defined by the *eigenmetabolite* approach, or by the average approach as described in section Pathway representation. If modules at pathway level should be identified and M consists of multiple pathways, then R_M is calculated using all metabolites in all pathways independent of pathway assignment. Notably, for module representatives we recommend to use the average approach since modules can be highly heterogeneous.

Module identification

Given the scoring function and a seed node, a greedy search procedure is performed to identify an *optimal module*. The algorithm starts with a *seed node* as *candidate module*. The score of the *candidate module* that contains only the seed node is obtained from the univariate differential analysis. In each iteration, the neighborhood of the *candidate module* is identified. Each neighbor (*candidate*) is added to the *candidate module* and the score of the extended *candidate module* is calculated with the scoring function as defined in section Module representation and scoring. The *candidate* node leading to the highest score improvement is added to the *candidate module* if the module score is higher than the score of each of its single components. The algorithm terminates if no further score improvements can be made, and the *optimal module* is returned. In an optional consolidation step, overlapping *optimal modules*, e.g. from different *seed nodes*, can be combined into one module, which is re-scored by the scoring function. To assess the significance of the modules, a conservative multiple testing correction procedure was used by correcting for the total number of nodes in the underlying network.



Prerequisites and installation

The following software packages are required to run *MoDentify*:

1. R version 3.3.1 or newer

2. R-packages:

- CRAN: *data.table* (>= 1.10.4)
- CRAN: *igraph* (>= 1.1.2)
- CRAN: *GeneNet* (>= 1.2.13)
- CRAN: *Hmisc* (>= 4.0-3)
- CRAN: *ggplot2* (>= 2.2.1)
- CRAN: *Rdpack*
- Bioconductor: *metabolomics* (>= 0.1.4)

3. For visualization of the identified modules in the underlying network, the following software is required:

- Bioconductor: *RCytoscape* (>= 1.24.1)
- Cytoscape version 2.8.3, available at http://www.cytoscape.org/download_old_versions.html
- The *CytoscapeRPC* plugin for Cytoscape. Installation instructions can be found here: https://wiki.nbic.nl/index.php/CytoscapeRPC_install. CytoscapeRPC has to be activated (Plugins > CytoscapeRPC > Activate CytoscapeRPC) and opened on port 9000.

To install *MoDentify*, download the package from Github (https://github.com/krumsiek/MoDentify). Change the working directory in R to the folder where the downloaded file is stored, and install *MoDentify* via R package *devtools*:

```
library(devtools)
install("MoDentify",build_vignettes = TRUE)
```

Open the Vignette with:

```
browseVignettes("MoDentify")
```

Load *MoDentify* with:

```
library(MoDentify)
```

Example data set

In the following, we show an application of *MoDentify* on preprocessed metabolomics data for blood, urine, and saliva samples from the Qatar Metabolomics Study on Diabetes (QMDiab), a type 2 diabetes case-control cohort published in several previous publications (K. T. Do et al. 2015, M. J. Mook-Kanamori et al. (2013), Suhre et al. (2017), Yousri et al. (2015)). The study was conducted in 2012 at the Dermatology Department of Hamad Medical Corporation and the Weill Cornell Medical College in Doha, Qatar. Untargeted metabolomics measurements (by Metabolon, Inc.) were available for 190 diabetes patients and 184 non-diabetics of Arab and Asian ethnicities aged 17-81 years. After preprocessing and combining data for all three fluids, 1524 (501 plasma, 734 urine, and 289 saliva) metabolites measured for 310 sindividuals were available.

Raw and preprocessed metabolomics data for blood, urine, and saliva, metabolite pathway annotations, and phenotype information on age, gender, body mass index, and type 2 diabetes status are available at the figshare repository via the following link https://doi.org/10.6084/m9.figshare.5904022. The preprocessed data, stored in the three data.tables `qmdiab.data`, `qmdiab.annos`, and `qmdiab.phenos`, is part of *MoDentify* and will be available after installation.

`qmdiab.data` is a data.frame with metabolites in columns and observations in rows. Each entry corresponds to the preprocessed relative ion counts for the respective metabolite in the respective sample measured by Metabolon, Inc. All variable labels contain the prefixes "P::", "U::", or "S::" indicating metabolites measured in plasma, urine, and saliva, respectively.

```
# QMDiab metabolomics data
knitr::kable(qmdiab.data[1:4,1:3])
```

	P::1,11-Undecanedicarboxylic acid	P::1,2-dipalmitoylglycerol	P::1,2-propanediol
QMDiab222	0.5174648	0.0558522	-0.2218105
QMDiab113	1.4007276	0.0558522	-0.0157088
QMDiab29	0.0023790	0.0102016	0.0056830
QMDiab243	2.0366493	-0.6981049	-0.4401081

`qmdiab.annos` is a data.frame with metabolites as rows and annotations as columns. We used *a priori* pathway annotations from Metabolon, Inc. which assigns each metabolite to one "sub-pathway" representing metabolic pathways and biochemical subclasses (e.g. Branched-chain amino acids), and to one "super-pathway" representing the general chemical or functional class (e.g. Amino acids).

```
# QMDiab annotations
knitr::kable(qmdiab.annos[1:4,2:5])
```

BIOCHEMICAL	SUPER_PATHWAY	SUB_PATHWAY	COMP_ID
1,11-Undecanedicarboxylic acid	P::Lipid	P::Fatty acid, dicarboxylate	43027
1,2-dipalmitoylglycerol	P::Lipid	P::Diacylglycerol	11953
1,2-propanediol	P::Lipid	P::Ketone bodies	38002
1,3,7-trimethylurate	P::Xenobiotics	P::Xanthine metabolism	34404

Finally, `qmdiab.phenos` is a data.frame containing information about age, gender, body mass index, and type 2 diabetes status for each study participant.

```
# QMDiab phenotypes
knitr::kable(qmdiab.phenos[1:3,])
```

AGE	GENDER	BMI	T2D
34.50513	0	25.01021	0
47.06639	1	28.36776	0
55.49076	1	29.70564	0

Alternatively, the three data.tables can be generated from the excel file `QMDiab_metabolomics_Preprocessed.xlsx` stored at the Dryad Data Repository:

```
qmdiab <- get.qmdiab.data("QMDiab_metabolomics_Preprocessed.xlsx")
```

Module identification at metabolite level

Network inference

```
met.graph <- generate.network(data = qmdiab.data,
                              covars = qmdiab.phenos,
                              annotations =
                              qmdiab.annos[,c("name","SUPER_PATHWAY","SUB_PATHWAY")],
                              correlation.type = "partial",
                              alpha = 0.05,
                              correction.method = "bonferroni")
```

- `data` is a data.table or data.frame with variables in columns and observations in rows.
- `covars` are the variables that should be included as covariates for correlation estimation.
- `annotations` is a data.table or data.frame with the column "name" containing the unique names of the variables, and optinally many annotations.
- `correlation.type` is the type of correlation to be estimated. It can be either `"pearson"` or `"partial"`.
- `alpha` is the significance threshold (α) to determine significant partial correlations.
- `correction.method` is the multiple testing correction method for the correlations (`"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"holm"`, `"hochberg"`, `"hommel"`, or `"none"` from the function `p.adjust` of package `stats`).

The inferred network is an igraph object with 1524 nodes and 2012 edges.

```
summary(met.graph)
```

```
## IGRAPH 2e076c3 UN-- 1524 2012 --
## + attr: name (v/c), SUPER_PATHWAY (v/c), SUB_PATHWAY (v/c), label
## | (v/c), cor (e/n), p (e/n)
```

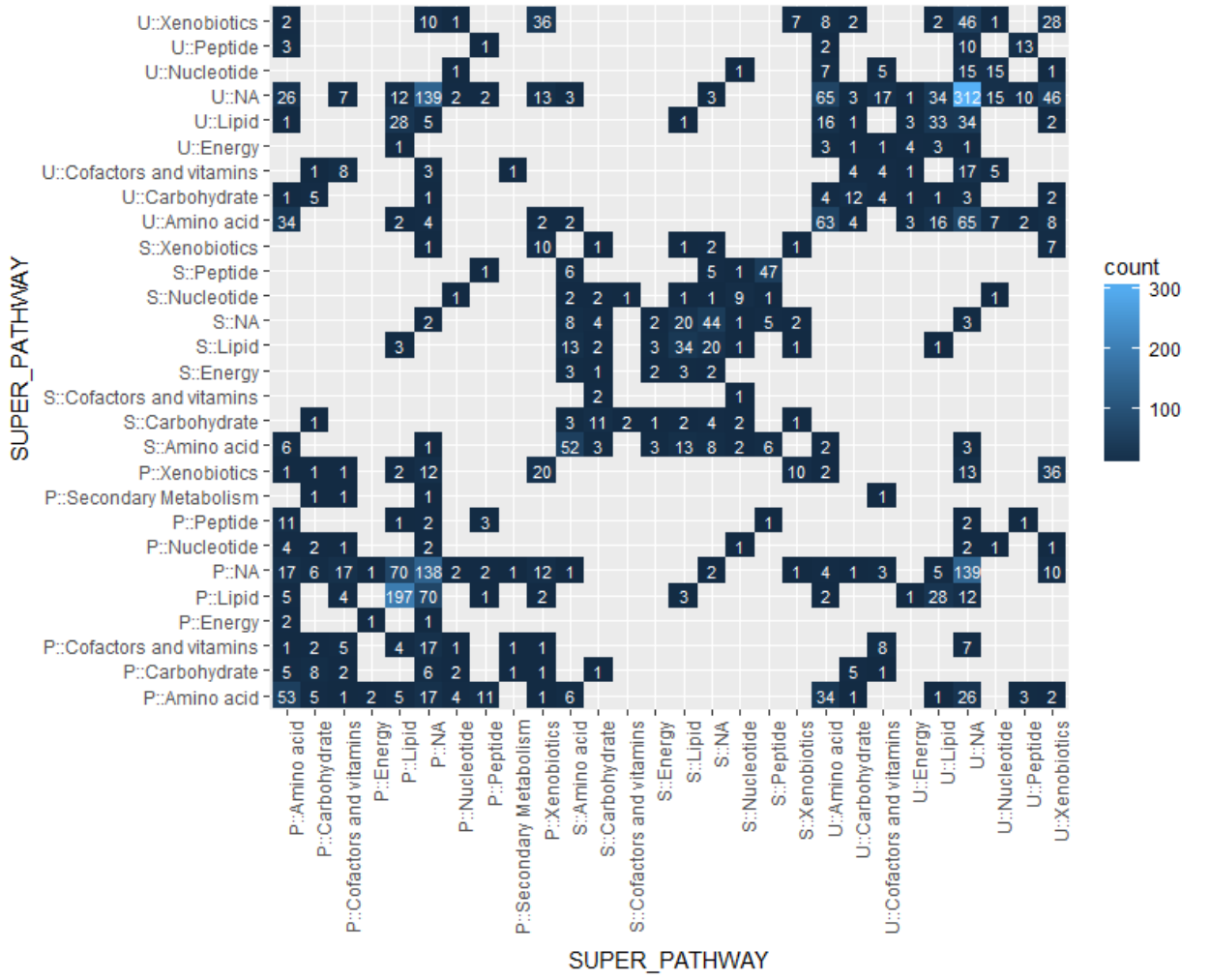
Alternatively, a pre-existing network can be used for module identification. Note that all nodes in the network must be available as variables in the input data.

```
net.DT <- fread(input = "QMDiab_flatfiles/qmdiab.network.txt", header = TRUE)


annotations <- fread(input = "QMDiab_flatfiles/qmdiab.annotations.txt",
                     header = TRUE)

library(igraph)
met.graph <- graph_from_data_frame(d = net.DT,
                                   directed = FALSE,
                                   vertices = annotations)
```

Edges between variable groups

In metabolomics, it has frequently been observed that metabolites from the same pathways tend to correlate with each other, while there are only few correlations across pathways (K. T. Do et al. 2015, Krumsiek et al. (2011), Mitra et al. (2013)). This trend can be explored in the inferred network by investigating the number of edges within and between pathways:

```
gg.edges.between.attributes(met.graph, "SUPER_PATHWAY", rm.unknown = TRUE)
```



Module identification

```
modules.summary <- identify.modules(graph = met.graph,
                                    data = qmdiab.data,
                                    annotations = qmdiab.annos,
                                    covars = qmdiab.phenos[,1:3],
                                    phenotype = qmdiab.phenos$T2D,
                                    alpha = 0.05,
```

```
                                    correction.method ="bonferroni",
                                    representative.method = "average")
```

- `graph` is an igraph object loaded from an external source or inferred within *MoDentify*.
- `data` is a data.table or data.frame with variables in columns and observations in rows.
- `annotations` is a data.table or data.frame with the column "name" containing the unique names of the variables, and optinally many annotations.
- `covars` are the variables that should be included in the scoring function as covariates.
- `phenotype` is the phenotype of interest coded as a vector with the same observations as in `data`.
- `alpha` is the significance threshold (α), which is used to determine significant phenotype associations and to correct for multiple testing.
- `correction.method` is the multiple testing correction method to be used (`"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"holm"`, `"hochberg"`, `"hommel"`, or `"none"` from the function **`p.adjust`** of package `stats`).
- `representative.method` is either `"eigenmetabolite"` for the *eigenmetabolite* approach, or `"average"` for the *average* approach to calculate the pathway representative.
- The output `modules.summary` consists of four data.tables: `modules.summary$modules` contains the module scores and effect sizes. `modules.summary$nodes` and `modules.summary$seeds` contain the score for each node and the assignment to module IDs in `modules.summary$modules`. `modules.summary$cache` contains all *candidate modules*, their scores, and the frequency of access by the algorithm.

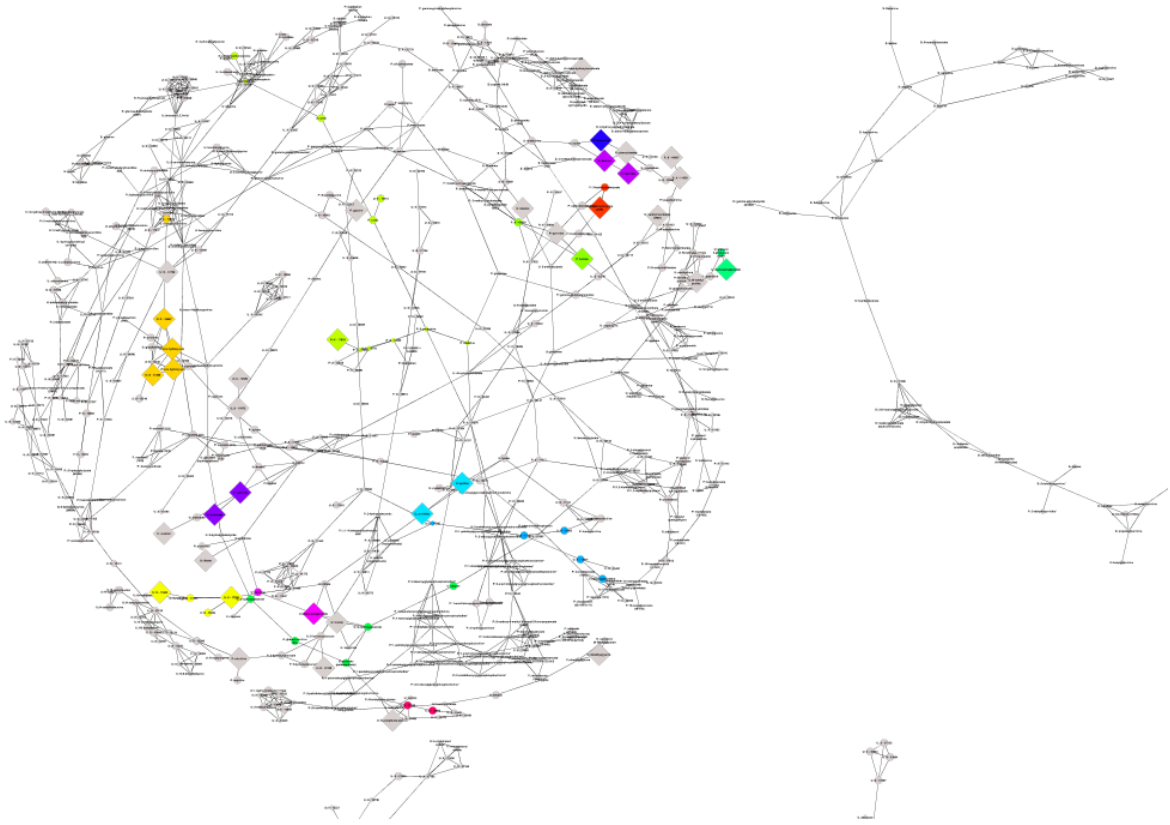
Modules can be exported to a file using:

```
export.modules(modules.summary, "QMDiab_modules.txt")
```

Network and module visualization

```
#- remember to open Cytoscape and activate the CytoscapeRPC plugin
draw.modules(graph = met.graph,
             title = "QMDiab_modules",
             summary = modules.summary,
             save.image = FALSE,
             modules.to.draw = NULL)
```

- `graph` is an igraph object loaded from an external source or inferred by *MoDentify*.
- `title` is the title for the Cytoscape session.
- `summary` is the output from **`identify.modules`**.
- `save.image` is a logical parameter specifying whether *png* files of the modules should be saved.
- `modules.to.draw` contains the module IDs to be drawn. If this parameter is set to `NULL` all modules will be visualized.



Module identification at pathway level

Network inference

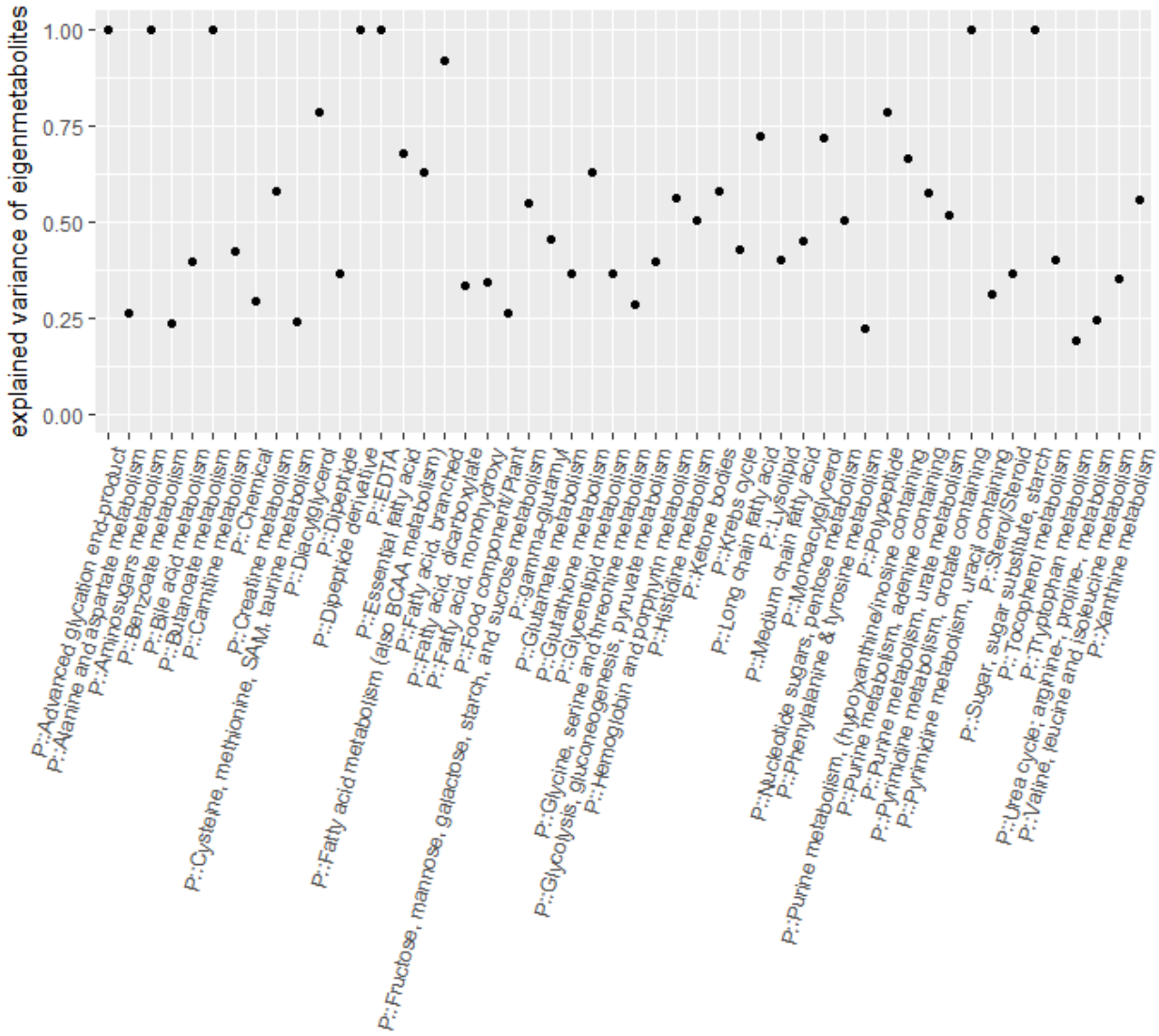
```
pathway.graph <- generate.pathways.network(data = qmdiab.data,
                                           annotations = qmdiab.annos,
                                           level = "SUB_PATHWAY",
                                           correlation.type = "partial",
                                           alpha = 0.05,
                                           correction.method = "bonferroni",
                                           rm.unknown = TRUE,
                                           representative.method = "eigenmetabolite")
```

- `data` is a data.table or data.frame with variables in columns and observations in rows.
- `annotations` is a data.table or data.frame with the column "name" containing the unique names of the variables, and optinally many annotations.
- `correlation.type` is the type of correlation to be estimated. It can be either `"pearson"` or `"partial"`.
- `level` is the resolution level. The parameter should be a string of the name of the column containing the pathway assignments to be used. (For metabolite level, it is set to `NULL`)
- `alpha` is the significance threshold (α), which is used to determine significant correlations.
- `correction.method` is the multiple testing correction method for the correlations (`"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"holm"`, `"hochberg"`, `"hommel"`, or `"none"` from the function **`p.adjust`** of package `stats`).
- `rm.unknown` is `TRUE` if variables with no pathway assignments should be removed. These variables should be labeled with "Unknown" or "NA" in the corresponding pathway column.
- `representative.method` is either `"eigenmetabolite"` for the *eigenmetabolite* approach, or `"average"` for the *average* approach to calculate the pathway representative.

Explained variances of *eigenmetabolites*

Eigenmetabolites are calculated as the first principal components of a PCA. The explained variances can be plotted with:

```
# plot all pathways
gg.explained.variance(pathway.graph)

# plot the first 50 pathways
gg.explained.variance(pathway.graph, 1:50)
```



Module identification

```
pathway.modules <- identify.modules(graph = pathway.graph$network,
                                    data = qmdiab.data,
                                    covars = qmdiab.phenos[,1:3],
                                    phenotype = qmdiab.phenos$T2D,
                                    level = pathway.graph$level,
                                    annotations = qmdiab.annos,
                                    alpha = 0.05,
                                    correction.method ="bonferroni",
                                    representative.method = "average")
```

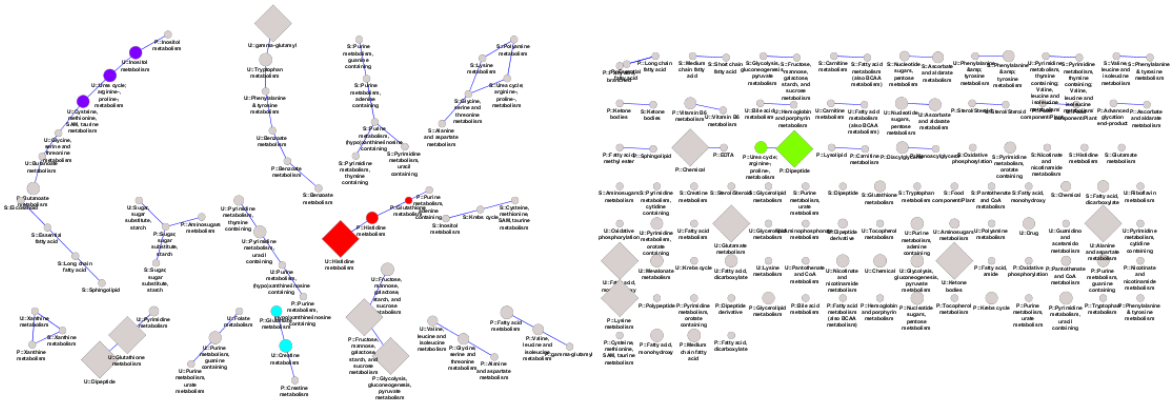
- `graph` is an igraph object loaded from an external source or inferred within *MoDentify*.

- `data` is a data.table or data.frame with variables in columns and observations in rows.
- `covars` are the variables that should be included in the scoring function as covariates.
- `phenotype` is the phenotype of interest coded as a vector with observations as in `data`.
- `level` is the resolution level. The parameter should be a string of the name of the column containing the pathway assignments to be used. (For metabolite level, it is set to `NULL`)
- `annotations` is a data.table or data.frame, which must contain the columns "name" and the column name stored in `level`.
- `alpha` is the significance threshold (α), which is used to determine significant phenotype associations.
- `correction.method` is the multiple testing correction method to be used (`"bonferroni"`, `"BH"`, `"BY"`, `"fdr"`, `"holm"`, `"hochberg"`, `"hommel"`, or `"none"` from the function **`p.adjust`** of package `stats`).
- `representative.method` is either `"eigenmetabolite"` for the *eigenmetabolite* approach, or `"average"` for the *average* approach to calculate the module representative.

Network and module visualization

```
draw.modules(graph = pathway.graph$network,
             title = "QMDiab_modules_pw",
             summary = pathway.modules,
             save.image = FALSE,
             modules.to.draw = NULL)
```

- `graph` is an igraph object loaded from an external source or inferred by *MoDentify*.
- `title` is the title for the Cytoscape session.
- `summary` is the output from **`identify.modules`**.
- `save.image` is a logical parameter specifying whether *png* files of the modules should be saved.
- `modules.to.draw` contains the module IDs to be drawn. If this parameter is set to `NULL` all modules will be visualized.



How to cite

To cite *MoDentify*, please cite the following publication:

XXX Application Note. Will be soon updated.

References

Do, Kieu Trinh, Gabi Kastenmüller, Dennis O. Mook-Kanamori, Noha A. Yousri, Fabian J. Theis, Karsten Suhre, and Jan Krumsiek. 2015. "Network-Based Approach for Analyzing Intra- and Interfluid Metabolite Associations in Human Blood, Urine, and Saliva." *Journal of Proteome Research* 14 (2): 1183–94. doi:10.1021/pr501130a.

Do, Kieu Trinh, Maik Pietzner, David JNP Rasp, Nele Friedrich, Matthias Nauck, Thomas Kocher, Karsten Suhre, Dennis O. Mook-Kanamori, Gabi Kastenmüller, and Jan Krumsiek. 2017. "Phenotype-driven identification of modules in a hierarchical map of multifluid metabolic correlations." *Npj Systems Biology and Applications* 3 (1): 28. doi:10.1038/s41540-017-0029-9.

Krumsiek, Jan, Karsten Suhre, Thomas Illig, Jerzy Adamski, and Fabian J Theis. 2011. "Gaussian graphical modeling reconstructs pathway reactions from high-throughput metabolomics data." *BMC Systems Biology* 5 (1): 21. doi:10.1186/1752-0509-5-21.

Langfelder, Peter, and Steve Horvath. 2007. "Eigengene networks for studying the relationships between co-expression modules." *BMC Systems Biology* 1 (1): 54. doi:10.1186/1752-0509-1-54.

Mitra, Koyel, Anne-Ruxandra Carvunis, Sanath Kumar Ramesh, and Trey Ideker. 2013. "Integrative Approaches for Finding Modular Structure in Biological Networks." *Nature Reviews. Genetics* 14 (10): 719–32. doi:10.1038/nrg3552.

Mook-Kanamori, Marjonneke J, Mohammed M El-Din Selim, Ahmed H Takiddin, Hala Al-Homsi, Khoulood A S Al-Mahmoud, Amina Al-Obaidli, Mahmoud A Zirie, et al. 2013. "Ethnic and Gender Differences in Advanced Glycation End Products Measured by Skin Auto-Fluorescence." *Dermato-Endocrinology* 5 (2): 325–30. doi:10.4161/derm.26046.

Suhre, Karsten, Matthias Arnold, Aditya Mukund Bhagwat, Richard J. Cotton, Rudolf Engelke, Johannes Raffler, Hina Sarwath, et al. 2017. "Connecting Genetic Risk to Disease End Points Through the Human Blood Plasma Proteome." *Nature Communications* 8 (February): 14357. doi:10.1038/ncomms14357.

Yousri, Noha A., Dennis O. Mook-Kanamori, Mohammed M. El-Din Selim, Ahmed H. Takiddin, Hala Al-Homsi, Khoulood A. S. Al-Mahmoud, Edward D. Karoly, et al. 2015. "A Systems View of Type 2 Diabetes-Associated Metabolic Perturbations in Saliva, Blood and Urine at Different Timescales of Glycaemic Control." *Diabetologia* 58 (8): 1855–67. doi:10.1007/s00125-015-3636-2.