Masterarbeit
von Philipp Angerer
in Bioinformatik

Subpopulation Identification and Analysis in Multivariate Data

LUDWIG - MAXIMILIANS - UNIVERSITÄT
TECHNISCHE UNIVERSITÄT MÜNCHEN

Institute of Computational Biology
Helmholtz Zentrum München

Masterarbeit
in Bioinformatik

Subpopulation Identification and Analysis in Multivariate Data

Philipp Angerer

Aufgabensteller:	Prof. Dr. Dr. Fabian Theis
Betreuer:	Justin Feigelman, M. Sc.; Dr. Carsten Marr
Abgabedatum:	15. April 2014

Ich versichere, dass ich diese Masterarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

11. April 2014 ____________________________
Philipp Angerer

Acknowledgement

I would to thank Prof. Fabian Theis for giving me the opportunity to write this work in the welcoming environment of the ICB, where I got a glimpse at all the exciting projects being worked on there.

My heartfelt thanks go to my supervisors Justin Feigelman and Carsten Marr for providing answers to my questions in person and per mail, for investing the time and patience to meet me weekly, and give feedback as well as encouragement for my work.

Special thanks go to Jan Krumsiek, who had answers to my questions about all things Metabolomics by mailing from all over the world, as well as Kiki Do, who always offered to talk to me over some coffee.

Finally, I want to express my gratitude towards my friends for dragging me away on weekends, my wonderful girlfriend Tina for her support and patience, as well as my mother Evi for support and faciliation of paperwork.

Abstract

Correlations provide a measure of the interdependence of variable pairs in multivariate data, and thus are useful for the analysis of relations in transcriptional, metabolic, and regulatory biological data. However, data from cell populations can contain behaviorally heterogeneous subpopulations that render correlation analysis misrepresentative of the true regulation when behaving differently to the population as a whole.

Knowledge about heterogeneous subpopulations exposes those different modes of behavior in all kinds of biological interactomes and proves to be important for the inference of interaction networks and correlation analysis.

However, correlation analyses in biological data mostly do not consider the possibility of heterogeneous subpopulations, and even if they do, their identification and analysis have no established procedures except the heuristic of thresholding according to the level of one variable, which divides the population into two subpopulations. This method has several problems including a sensibility to outliers and the lack of both a concept for overlapping subpopulations and of information about the robustness of subpopulation correlations.

This work presents a novel interactive framework for visualizing correlations within subpopulations utilizing Multiresolution Correlation Analysis (MCA), a tool for exploratory subpopulation analysis in multivariate data. Using it, it is possible to detect and analyze such heterogeneities, as well as find subpopulations with notably different correlation behavior.

This work provides an overview of statistical groundwork for correlation analysis and several correlation types. It explains the principles of the MCA visualization technique, provides a guide for interpreting the resulting plots and an algorithm to filter out interesting plots.

Furthermore the architecture and usage of an R library and a web application are described, both developed in scope of this work. They complement each other in that the latter provides interactive and intuitive access to the functionality of the former, which in turn can be used to for automation.

Finally, this work presents the results of an application of those tools to two metabolic data sets, where connections to existing biological knowledge are found, and the application to a transcriptomic study, the conclusions of which could be confirmed and extended. This way it is shown that MCA is an useful tool for the inspection of correlations in cell subpopulations, and is thus able to provide better insight than conventional methods.

Kurzzusammenfassung

Korrelationen stellen ein Maß für den Zusammenhang von Variablenpaaren eines multivariaten Datensatzes dar, und sind daher nützlich für die Analyse von transkriptionellen, metabolischen, und regulatorischen Daten. Jedoch können Zellpopulationsdaten aus Subpopulationen heterogenen Verhaltens zusammengesetzt sein, für welche eine Korrelationsanalyse der Gesamtpopulation nicht repräsentativ ist.

Wissen über die Zusammensetzung dieser Subpopulationen zeigt ihre verschiedenen Verhaltensweisen auf und erweist sich daher als wichtig für sowohl die Inferenz von Interaktionsnetzwerken als auch Korrelationsanalyse.

Dennoch erkennen die wenigsten Korrelationsanalysen biologischer Daten die mögliche Existenz derartiger Subpopulationen an, und selbst wenn, ist die einzige etablierte Methode um die Gesamtpopulation in Subpopulationen aufzuteilen eine Zweitelung anhand eines Schwellwertes. Diese Methode hat verschiedene Probleme, welche eine Sensibilität gegenüber Ausreißern einschließen, sowie den Mangel der Möglichkeit, überlappende Subpopulationen oder Robustheit gefundener Subpopulationen darzustellen.

Diese Arbeit präsentiert ein neuartiges Framework, mit dessen Hilfe Korrelationen innerhalb von Subpopulationen visualisiert werden können. Dazu nutzt es Mehrfach-aufgelöste Korrelationsanalyse (MCA), ein Werkzeug zur explorativen Subpopulationsanalyse in multivariaten Daten. Auf diese Weise ist es möglich, obige Heterogenitäten aufzuspüren, zu analysieren, und Subpopulationen mit deutlich abweichendem Korrelationsverhalten zu identifizieren.

Diese Arbeit zeigt einen Überblick des statistischen Fundaments der Korrelationsanalyse, sowie verschiedener Korrelationstypen. Sie erklärt die Prinzipien der MCA-Visualisierungstechnik, und stellt einen Leitfaden zur Interpretation der resultierenden Diagramme zur Verfügung, sowie einen Filter-Algorithmus für dieselben, welcher interessante Diagramme erkennt.

Weiterhin werden Architektur und Nutzung einer R-Programmbibliothek und einer Web-Anwendung beschrieben, die im Rahmen dieser Arbeit entstanden sind. Die Programme ergänzen einander, indem zweiteres intuitiven und interaktiven Zugang zur Funktionalität des ersteren zur Verfügung stellt, welches wiederum dazu genutzt werden kann, MCA zu automatisieren.

Schließlich präsentiert diese Arbeit die Ergebnisse einer Anwendung besagter Programme auf zwei metabolische Datensätze, in welchen Verbindungen zu existiendem Wissen gefunden werden, und Anwendung auf eine transkriptionelle Studie, deren Ergebnisse bestätigt und erweitert werden konnten. Auf diese Weise wird gezeigt, dass MCA ein nützliches Werkzeug für die Analyse von Korrelationen in Zellpopulationen ist, und dabei tiefere Erkenntnisse als existierende Methoden ermöglicht.

Contents

1 Introduction

Systems biology is based on the idea that all kinds of biological systems can only be understood as a whole instead of trying to do research for each part individually. The reason why we are still able to make sense of all of those highly complex biological systems is their modularity. This modularity is hierarchical and reaches from the molecular level over gene regulation up to division of labor between cells, organs, and organisms (Lorenz et al., 2011 and Vemuri and Aristidou, 2005). This and the systems approach are not contradictory – only when considering the interactions between modules, they can be fully understood.

In bacteria as well as eukaryotes, this heterogeneity through specialization is of advantage, and while the former are already able to amalgamate into heterogeneous biofilms, it becomes a permanent part of the eukaryotic organism through cell differentiation additionally to regular heterogeneity (Crespi, 2001). Differentiation itself is preceded by reduced expression of genes maintaining pluripotence state (Trott et al., 2012). Regardless of differentiation, cells are therefore heterogeneous in another example of modularity, gene regulation (see **Figure 1.1**). Cells with different metabolic and catabolic roles also have different gene activation patterns (Reik, 2007).

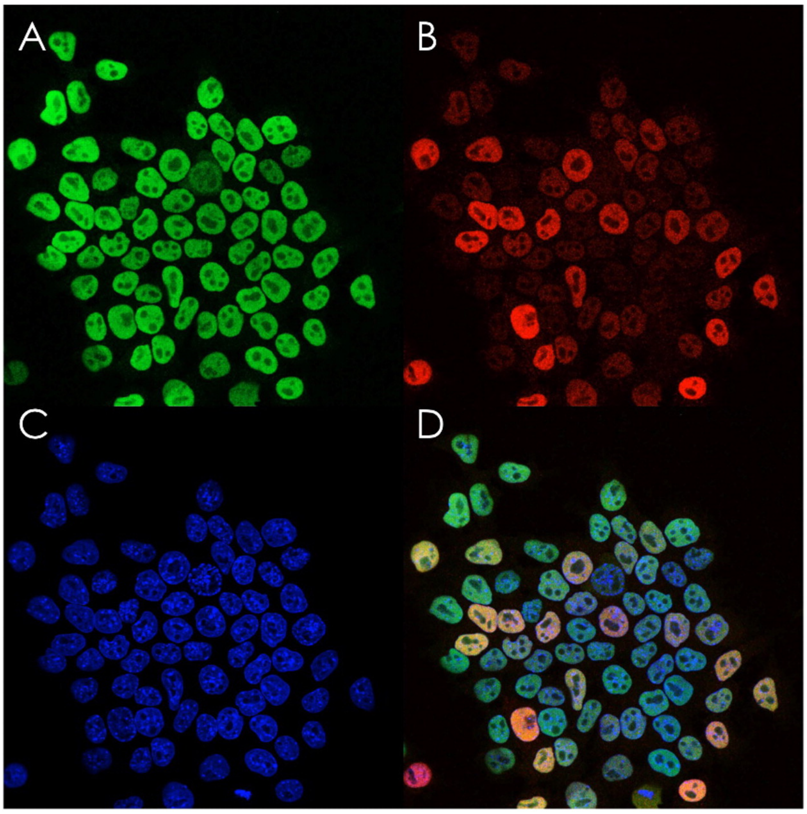


Figure 1.1 Immunoflourescence staining in mouse embryonic stem cells as example of heterogeneity. **A)** Oct4, **B)** Nanog **C)** DAPI, and **D)** an overlay. Nanog can be seen to be heterogeneously expressed. (Figure taken with permission from Roeder and Radtke (2009))

The heterogeneity of gene expression in cell populations can therefore be attributed to a combination of cell-intrinsic regulation and extrinsic influences such as intracellular organization, availability of nutrients, or presence of toxins. It can be maintained even through cell division where epigenetic changes and concentrations of metabolites and gene products are passed on to daughters of dividing cells.

Cell population heterogeneities like this can be found in pluripotent stem cell populations (Narsinh et al., 2011 and Huang et al., 2005), metabolomics (Amantonico et al., 2010), and transcriptomics (Tang et al., 2009). This becomes apparent with emerging technologies capable of single-cell analysis such as transcription analysis like quantitative single-cell PCR and flow cytometry, next-generation sequencing methods like mRNA-Seq (Tang et al., 2009), protein analysis methods like mass cytometry, and mass spectrometry (MS) methods like electrospray MS or secondary ion MS (Amantonico et al., 2010). Advances in the area of single-cell imaging and tracking also allow for subpopulation analysis in live cells.

Motivation

Common to all fields based on mentioned single-cell analyses is the dawning realization that the ensemble system state inferred by averaging data taken from the whole population is not indicative of the real multitude of states the subpopulations reside in. For example, a regulatory subsystem that either keeps a gene product at a very low level or a very high level via feedback loops is not aptly represented by the calculated medium level resulting from averaging the measurements of multiple cells with either high or low level.

An important field for integration of subpopulation information is the modeling of the system as a network, be it pathways, gene regulatory networks (GRNs) or other dependency graphs. Represented this way, interactions between subsystems are most easily understandable while also being assessable by computational methods using graph statistics like connectedness, or by overlaying and extracting parts of the network (Krumsiek et al., 2012).

One approach to create networks like this are Gaussian Graphical Models (GGMs), which are based on partial correlations. Partial correlations themselves are a correlation statistic for data sets that aims to reduce or remove the effects that indirect associations between variables have, while retaining direct associations (Baba et al., 2004). GGMs are networks created by removing all edges with insignificant partial correlations, optimally leaving only direct associations between the respective network nodes, be it genes or metabolites (Schäfer and Strimmer, 2005 and Krumsiek et al., 2012).

Inference of networks able to faithfully represent associations between genes is complicated by having to rely on data that is not divided into heterogeneous subpopulations, which creates the demand for subpopulation identification (Trott et al., 2012).

Common approaches to the inference of subpopulations include mostly space- or density-based clustering algorithms such as EM and DBSCAN. There are also correlation-based clustering algorithms available that cluster data points according to their distance to regression lines (Böhm et al., 2004). All of those clustering algorithms share the property of being dependent of some sort of separation in the data, be it space or density. Even correlation-based approaches are not able to identify strongly intersecting subpopulations, and highly sensitive to any separation in the data.

Another criterion to define subpopulations depends on the level of one factor, which is common in metabolomics, where marker metabolites for diseases are sought (Krumsiek et al., 2012), as well as in transcriptomics, where concentrations of transcripts are able to represent system states (Trott et al., 2012).

Current approaches to alleviate the problem of finding this kind of heterogeneous subpopulations rely on the manual division of levels into a high and low segment (Trott et al., 2012). This can be problematic because the threshold for division has to be carefully selected, subpopulations that overlap cannot be described, outliers influence the resulting values greatly, and the found subpopulations may be unstable to small changes.

Overview

This work presents an approach to analyze data using Multiresolution Correlation Analysis (MCA), which allows a more detailed analysis of correlations within possible subpopulations in multivariate data by visualizing them for all possible segments of a binned sorting variable domain. It is therefore also dependent on one factor, but able to represent overlapping subpopulations and leaving out data. MCA visualizes the correlations of those subpopulations in a way that variations in similar subpopulations give insight into correlation estimation confidence, subpopulation robustness, and outlier sensibility, which allows for identification of interesting subpopulations.

Chapter 2 presents the statistical foundation of MCA and how they are related, starting with regression and continuing over Pearson's and Spearman's correlation coefficients to partial correlation. The part ends with confidence assessment using confidence intervals and p-values.

Chapter 3 presents the operating principle of the MCA visualization method based on subdivision of a sorting variable quantile and a scoring of the resulting plots which is able to extract potentially interesting variable pairs from high-dimensional data sets.

Chapter 4 explains the capabilities and usage of an R library implementing those methods, as well as a web application utilizing this library to provide an interactive interface to all described methods. The web application unifies all presented concepts and methods behind an easy to use interface and puts selected subpopulations in a context to the whole population. It is available internally on the servers of the Helmholtz Zentrum München, Institute of Computational Biology (ICB).

Chapter 5 shows the application of the scoring to two metabolic data sets, in order to generate hypotheses concerning differentially regulated subpopulations and test them using existing biological literature. Also, it shows a third application to a transcriptomic data set, where conclusions about previously-described subpopulations from a published study were analyzed and extended.

2 Statistical background: Correlation analysis

Statistical analysis is often based on relations between the existing variables, examples including gene regulatory networks and metabolic profiles. They also intend to give information about whole populations. However, due to only a limited number of measurements being possible, they generally have to be estimated from a sample such as the measurements in a data set. Those observations each contain a number of realizations of the variables in that data set.

The available measurements are used as sample from which to estimate statistics describing those relations. A basic relation between pairs of variables is the correlation coefficient, which describes how the value of one variable influences that of another – or the other way round, as correlations are undirected (Böhm et al., 2004).

A significant positive correlation between two variables usually means that there is a common factor regulating both or they have another relation, including more indirections. Negative correlations imply the same, only for downregulation. Similarly it applies to metabolites, where the presence of one metabolite might provoke a biological response that depletes or produces other metabolites to fuel a reaction.

Correlations are statistical measures in the interval $[-1, 1]$, with the two extrema being perfect negative and positive correlation, respectively, and 0 being no correlation of the two compared variables.

In this section, formulas refer to X, which is a representation of an example data set with p variables and n observations. Let $X = [X_1, ..., X_p]^{\mathrm{T}} \in \mathbb{R}^{p \times n}$ be a matrix containing p vectors X_i which each represent a sample of n realizations of variable i. Thus $X_i = (X_{i1}, ..., X_{in})\ \forall i \in \{1, ..., p\}$.

2.1 Regression and correlation

Some correlation types are related to regression, the act of describing the relation of a subset of variables with a simple formula. There are multiple such correlation types, for example both Pearson's product-moment correlation coefficient and Spearman's rank correlation coefficient.

Pearson's correlation coefficient is for example based on linear regression, which means the act of describing the relation between variables using the line that best describes all data points. This line can be estimated by finding the linear function with the minimal sum of squared distances ε to all points, also called the least squares estimator (Seber and Lee, 2012). A regression line has the function $y = \alpha + \beta x$. Given two variables i and j from the data set, α and β are the solution for

$$\underset{\alpha,\beta}{\operatorname{argmin}} \left[\sum_{k=1}^{n} \hat{\varepsilon}_k^2 = \sum_{k=1}^{n} (X_{ik} - \alpha - \beta X_{jk})^2 \right] \tag{2.1}$$

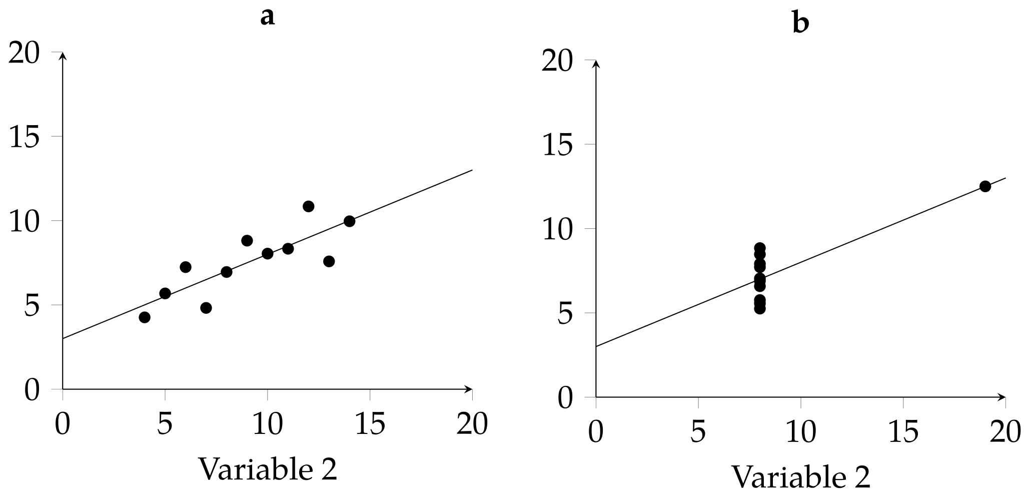


Figure 2.1 Scatter plots of data sets for two variables with linear regression lines. The Pearson correlation coefficient of both is 0.816, despite their different shape, while the Spearman coefficient is 0.818 for **a** and 0.50 for **b**. (Data from „Anscombe's Quartet", (Anscombe, 1973))

This optimization problem has the following solution:

$$\hat{\beta} = \frac{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j)}{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)^2} \tag{2.2}$$

$$\hat{\alpha} = \bar{X}_j - \hat{\beta}\bar{X}_i \tag{2.3}$$

A colinearity of both variables, i. e. all data points being on the regression line and the sum of squares being zero is a perfect Pearson correlation of -1 or 1, with positive correlation meaning the regression line is sloped upwards and negative meaning it is sloped downwards. Zero correlation indicates a non-elongated shape of the data cloud, such as a radially symmetric shape (Pearson, 1895).

The relation of Pearson correlation and linear regression extends to the sensitivity against outliers, while certain other correlationtypes do not share this problem to the same degree (see **Figure 2.1**).

2.2 Pearson correlation estimation

The most common way to calculate correlation is Pearson's product-moment correlation coefficient (Pevsner, 2013), which will be described in the following section. It relies on the definitions of sample covariance and variance, which already appeared in the least squares estimator for linear regression.

The sample covariance $\mathrm{Cov}(X_i, X_j) = \hat{\Sigma}_{ij}$, which is an estimator for the true population covariance can be calculated from X:

$$\hat{\Sigma}_{ij} = \mathrm{Cov}(X_i, X_j) = \frac{1}{n-1}\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j) \tag{2.4}$$

with $\bar{X}_i$ being the mean value of each sample X_i of variable i. The sample variance is derived from it: $\mathrm{Var}(X_i) = \mathrm{Cov}(X_i, X_i)$

The correlation matrix ρ is obtained by normalizing the covariance matrix with the respective standard deviations. The sample estimator r_{ij} for the correlation coefficient is written as

$$r_{ij} = \hat{\rho}_{ij} = \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathrm{Var}(X_i)}\sqrt{\mathrm{Var}(X_j)}} = \frac{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)(X_{jk} - \bar{X}_j)}{\sqrt{\sum_{k=1}^{n}(X_{ik} - \bar{X}_i)^2}\sqrt{\sum_{k=1}^{n}(X_{jk} - \bar{X}_j)^2}}. \tag{2.5}$$

Using this definition, another way to solve the linear regression formula for β is possible, with s being the sample standard deviation of a variable:

$$\hat{\beta} = \frac{\mathrm{Cov}(X_i, X_j)}{\mathrm{Var}(X_i)} = r_{ij}\frac{s_{X_i}}{s_{X_j}} \tag{2.6}$$

Pearson correlation has a few weaknesses and limitations. For example, the metric is not robust against outliers, a requirement fulfilled by other correlation statistics such as Spearman's rank correlation coefficient and others (Rousselet and Pernet, 2012). A limitation in some use cases is that it does not distinguish between direct and indirect inter-variable relations, which are better represented by partial correlations.

2.3 Spearman's rank correlation coefficient

The estimation of Spearman's rank correlation coefficient simply involves ranking the correlated samples and estimating Pearson's correlation of the ranked samples (Spearman, 1904). For example, the ranking of the samples $(0.64, 0.1, 0.13)$ is $(3, 1, 2)$, even while the distances between the sample values vary.

The Spearman coefficient describes how monotonic the association of two variables is. So two variables have a perfect Spearman correlation coefficient of 1 if each variable is a monotonically increasing function of the other, with the same applying to a perfect negative coefficient for a monotonically decreasing relation (Spearman, 1904).

This coefficient is more robust against outliers and approximately similar to Pearson's coefficient when the data is shaped roughly elliptically, as it is the case with normally distributed data. If however a real linear relation between variables is to be assessed, Pearson correlation represents this property best (Žežula, 2009).

2.4 Partial correlations

Apart from problems like outlier sensitivity, there are properties of normal correlation coefficients that are undesirable for analysis, such as the lack of distinction between direct and indirect correlation. Partial correlations eliminate those indirect effects, leaving only the direct portions of correlations.

Applied to the earlier example of Gene Regulatory Networks, the Pearson correlation between two gene products might be high while no direct regulatory connection exists between them, and they are regulated by a common third gene instead. Provided the partial correlation between both is estimated factoring in the third gene, partial correlations between it and the first two will be high, while the removal of the indirection from the correlation between the first two leads to a lower value (see **Figure 2.2**).

Due to this property, partial correlations are for example useful to illustrate and process correlation networks such as Gaussian Graphical Models (GGMs), which are sparse networks containing only those edges where partial correlations between variables is non-zero.

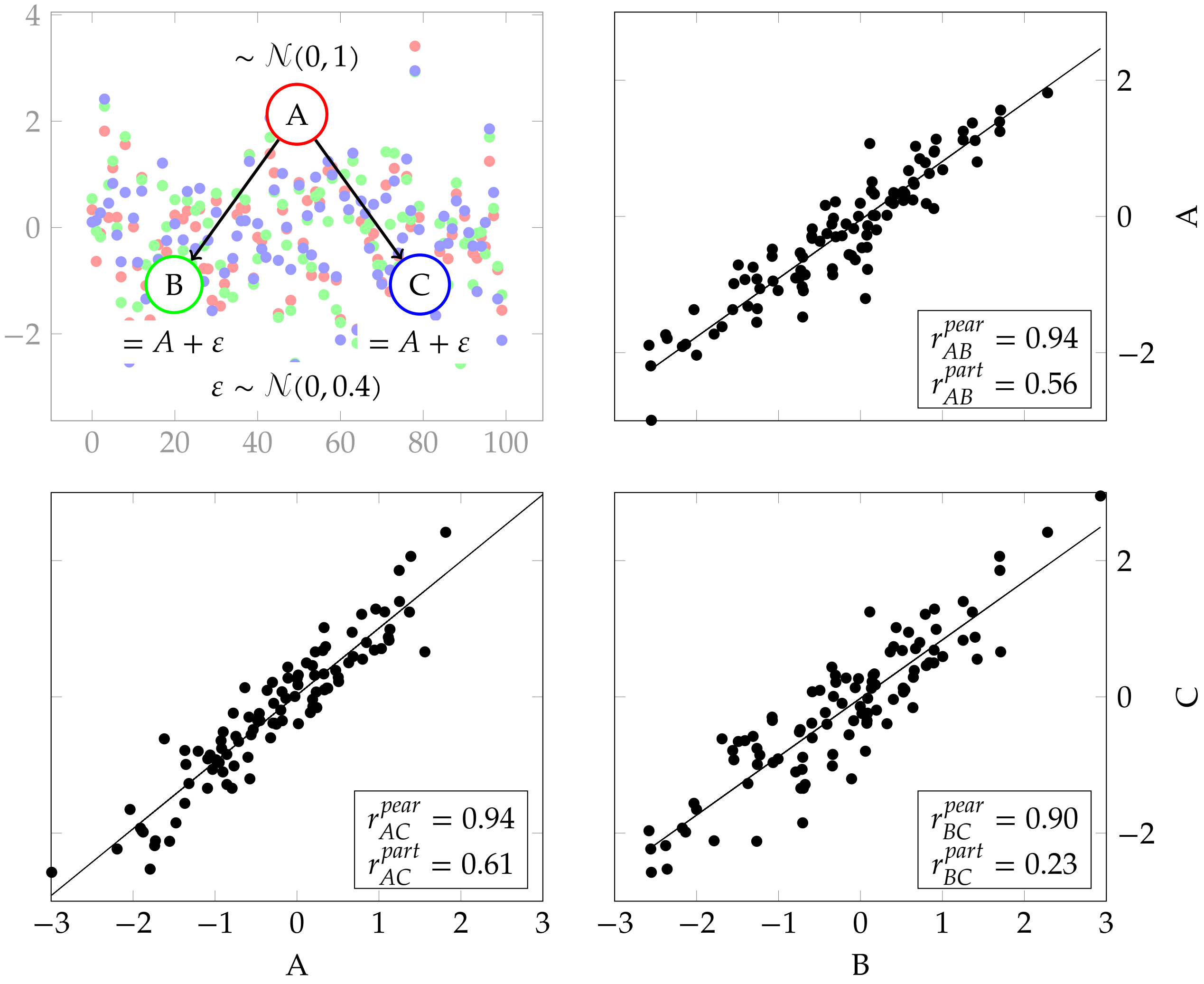


Figure 2.2 Regular Pearson correlation is unable to distringuish between direct and indirect effects: Both B and C were generated by adding noise to A, and are thus only indirectly correlated, but the Pearson correlation coefficient reflects this as only a slight decrease, while the partial correlation between B and C is much lower than between A and each. Partial correlations are usually lower than non-partial ones while staying significant.

The term "partial correlations" applies to both limited-order and full partial correlations. Limited – or ith – order partial correlations are correlation coefficients with the effect of i other variables removed, and are easier to obtain the smaller the i is (de la

Fuente et al., 2004). Full partial correlations are more similar to the true direct relations between variables, but are also more difficult to obtain in situations where "small n, large p" situations are prevalent (Schäfer and Strimmer, 2005). This is common in biological fields, especially in metabolomics, where the number of metabolites measured can be an order of magnitude higher than the number of study participants.

Formulas for i^{th} order partial correlations can be derived from the $i - 1^{\text{st}}$ order one, with zeroth order being the Pearson correlation coefficient described in **Section 2.2**. With $xy.z$ meaning the removal of z's effect and $xy.zq$ meaning the removal of both z's and q's effects, as well as r as generic denominator for sample correlations, r_{xy} can be extended for partial correlations:

$$\begin{aligned}\text{first-order: } r_{xy.z} &= \frac{r_{xy} - r_{xz}r_{yz}}{(1 - r_{xz}^2)(1 - r_{yz}^2)} \\ \text{second-order: } r_{xy.zq} &= \frac{r_{xy.z} - r_{xq.z}r_{yq.z}}{(1 - r_{xq.z}^2)(1 - r_{yq.z}^2)} \\ &\dots\end{aligned}$$

2.4.1 Partial correlation estimation using matrix inversion

While limited-order partial correlations prove to be useful for analysis (de la Fuente et al., 2004), complete partial correlations, while more difficult to obtain, offer better potential for interpretation (Schäfer and Strimmer, 2005).

The full partial correlation matrix ρ can be estimated from the covariance matrix Σ using matrix inversion, provided it itself is well-conditioned, and estimated very accurately. This can only be the case in a $n \geq p$ situation, i. e. when more observations than variables exist in the data set, but a $n \geq p$ situation does not imply a well-conditioned covariance matrix (Schäfer and Strimmer, 2005). If all those predicates are given, an estimator for the partial correlation matrix can be derived from the precision matrix $\Omega = (\omega_{ij})$, which is the inverse of the covariance matrix $\Omega = \Sigma^{-1}$. Given an estimated covariance matrix, the partial correlation is also an estimate

$$r_{ij}^{part} = \hat{\rho}_{ij} = -\frac{\hat{\omega}_{ij}}{\hat{\omega}_{ii}\hat{\omega}_{jj}} \tag{2.7}$$

2.4.2 Partial correlation estimation using GeneNet's shrinkage estimator

A well-behaved covariance matrix, the requirement of the matrix inversion method for partial correlation estimation, cannot always be easily obtained. A more reliable way to estimate those full-order partial correlations is provided by the shrinkage estimator implemented in GeneNet (Schäfer and Strimmer, 2005). GeneNet is the result of an effort to find the best-performing estimator for partial correlations in an $n < p$ situation.

This is the essential advantage gained by using GeneNet, while it is also able to outperform the simple estimator in $n \geq p$ situations. Without the ability to estimate partial correlations in a $n < p$, subpopulation analysis like the one provided by this work would be only possible for situations with enough observations that even small subpopulations containing few bins still have more observations than variables.

GeneNet's shrinkage estimator works by estimating a covariance matrix in a way that is guaranteed to yield a positive definite result while being fast and reliable. The chosen method is a shrinkage estimator that reduces covariances while fulfilling those constraints.

In $n < p$ situations, only the unbiased empirical covariance matrix can be derived, which however, especially in those situations, is a bad estimator. It is both able to become non-invertible (singular) and lose its positive definiteness, making it unfit for deriving an estimation for the partial correlation matrix from it. GeneNet employs this unbiased empirical covariance matrix as full model U, as well as a submodel T, and a shrinking factor $\lambda \in [0, 1]$ which mixes both.

The principle of GeneNet is to select a submodel T as estimator for the whole data set, which is easy to estimate, yet has significant bias, since it only contains a subset of all variables of the data set. Then this submodel is mixed with U into U^* by fitting a shrinkage factor λ that describes the ratio of mixing.

$$U^* = \lambda T + (1 - \lambda)U \tag{2.8}$$

The target T is selected from a range of suitable candidates like the identity matrix. This way, starting from an unbiased estimation U and shrinking the covariances towards the target T, U^* is always improved, and other than U guaranteed to meet the constraints of having to be invertible (non-singular) and positive definite. After the shrinkage step, that covariance matrix U is tested for its ability to describe the input data using an error loss function based on the Frobenius norm, a distance between the true and estimated covariance matrix. This can be generalized by using multiple targets and shrinkage factors (Schäfer and Strimmer, 2005).

2.5 Significance assessment

Due to the random nature of the drawn sample, i. e. the measured datapoints, there is a possibility that a statistic on that sample was obtained by chance. This probability is called the p-value, and the hypothesis H_0 is that the data is not correlated at all. For correlations, the p-value is the probability that a correlation R, which is at least as strong as the estimated one, is estimated by chance under H_0:

$$p = P\left[|R| \geq r\right], \text{with } \mathrm{H}_0 \text{ being true} \tag{2.9}$$

The p-value of a correlation can be calculated as the probability that the correlation is at least as extreme under H_0 as the one estimated. In case of a two-tailed test, this means the area under the probability distribution from the estimated correlation r (see **Figure 2.3**).

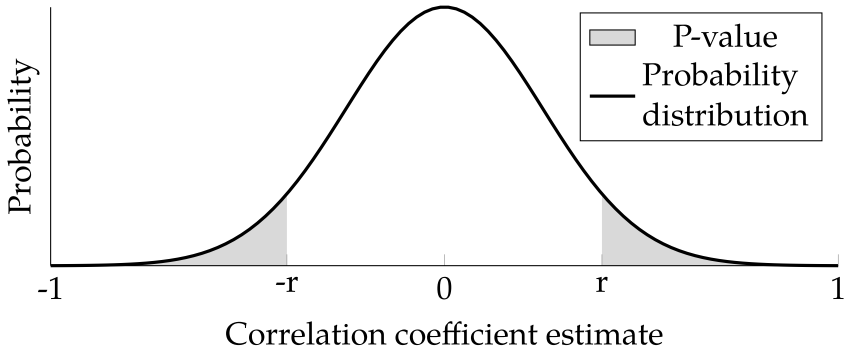


Figure 2.3 When estimating correlations, the probability distribution for the possible values can be derived from the sample size. The area under its curve is 1. Using it, the two-tailed p-value can be calculated from the area to both sides of a correlation estimate r and $-r$.

2.5.1 Pearson correlation

It is based on the concept of the confidence interval, which denotes the limits around from the estimate in which the real statistic lies with a certain confidence c. Using Fisher's r to z transformation, a value z can be constructed that is suitable to construct those interval limits $z_\pm$ (Fan and Thompson, 2001),

$$z = \frac{1}{2}\ln\left(\frac{1+r^{pear}}{1-r^{pear}}\right) \tag{2.10}$$

$$z_\pm = z \pm c \cdot \sqrt{(n-3)^{-1}} \tag{2.11}$$

whereupon the limits are converted back to the numerical space r resides in:

$$r_\pm = \frac{e^{2z_\pm}-1}{e^{2z_\pm}+1} \tag{2.12}$$

The p-value for Pearson's correlation coefficient can be calculated using the probability density function Φ_σ of the normal distribution with standard deviation $\sigma = \sqrt{(n-3)^{-1}}$ (Pevsner, 2013):

$$\Phi_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2.13}$$

$$p = 2 \cdot \Phi_\sigma(z) \tag{2.14}$$

2.5.2 Partial correlations

The p-value for the matrix-based estimation can be calculated using a t-statistic of the correlation and deriving the two-tailed p-value from it as described in **Figure 2.3** (Levy and Narula, 1978). The t statistic is distributed according to Student's t-distribution which is dependent of the $n - 2 - p$ degrees of freedom.

$$t = r_{ij}^{part} \sqrt{\frac{n - 2 - p}{1 - (r_{ij}^{part})^2}}, \text{ with } t \sim \text{Student's t-distribution} \tag{2.15}$$

GeneNet is a sophisticated tool in which many parts are selected from a number of choices, such as model selection of the shrinking target, the option to shrink towards multiple simultaneous targets, the choice of an error loss function and more. Utilizing those, it is able to improve the accuracy of the resulting partial correlations. The specifics of confidence assessment for GeneNet-estimated partial correlations is another part of the project covered in Strimmer (2008), a full description of which is beyond the scope of the present work.

3 Multiresolution correlation analysis

Facing data from single-cell sources, a scientist often encounters heterogeneities within the cell populations which can provide great insight once found. A common example for heterogeneity is gene expression dynamics, which often contain multiple stable modes in which genetically identical cells can exist, and can be switched between when a certain condition is met. Cells residing in one of those modes can be caused by the presence of one or multiple gene products, such as for example pluripotency markers, and be reversible or irreversible (Huang et al., 2005 and Narsinh et al., 2011). The same is true for metabolic markers: Organisms with certain concentrations of disease markers have different metabolic states (Amantonico et al., 2010).

In order to detect heterogeneities in data, the extraction and comparison of subpopulations is necessary. Because the real subpopulations are a property of the heterogeneous biological processes responsible for the data, they are often mixed stochastically and therefore impossible to fully separate. However, methods like clustering algorithms exist that identify approximate subpopulations in the data (Böhm et al., 2004).

An effective subpopulation extraction method reveals the inherent heterogeneity of heterogeneous data (see **Figure 3.1**). The method presented in this chapter is able to visualize and extract heterogeneities based on the level of one variable in the data set.

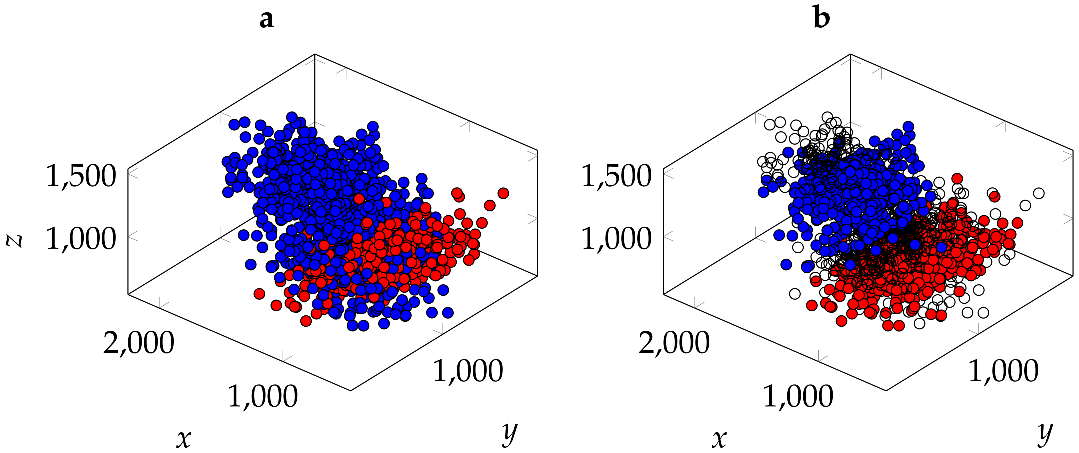


Figure 3.1 Correlation analyis detects heterogeneous structures. **a)** The original components, which were created by different processes, and have a different correlation structure, yet overlap. **b)** Subpopulations that were found using the tool developed in scope of this work which are able to reflect the two-componential nature of the data.

3.1 MCA plots

Multiresolution correlation analysis (MCA) plots are a way to visualize sample correlations between two variables as a diagnostic for the identification of subpopulations, as defined by the distribution of a sorting variable (Feigelman et al., 2014).

An MCA plot is defined by its granulatity, its sorting variable, and the two correlated variables. The sorting variable is used for the MCA plot's coordinate system, and the granularity is defined by the bin number and determines the MCA plot's resolution. The variables determine which correlations are plotted on the resulting grid. Therefore, for one sorting variable and bin number, one MCA plot for each variable combination exists.

By arranging the sorting variable according to its quantile, a range of a certain length taken from the sorting variable quantile domain is guaranteed to contain a certain number of data points, which would not be the case when using ranges of the sorting variable's domain itself. For example, a sorting variable with 500 observations is guaranteed to have 250 below its 50th quantile, even if 499 are above the mean value.

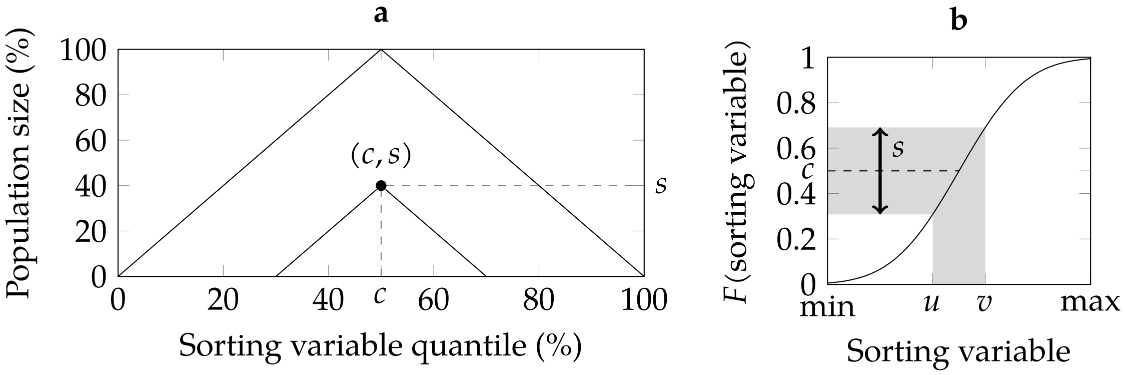


Figure 3.2 Subpopulations according to the sorting variable quantile. **a)** The meaning of the coordinates (c, s). The example point has a populations size $s = 40\%$ and a center $c = 50\%$, which results in a population including the observations from the 30% to the 70% quantile. **b)** The empirical cumulative distribution function F maps a sorting variable range $[u, v]$ to those quantiles.

Mathematically, the empirical distribution function F, computed from and applied to the sorting variable has a range of $[0, 1]$. MCA plots are 2D charts using this quantile range as coordinates for plotted points.

Each quantile range and therefore subpopulation can be defined by a point (c, s) that encloses it by being a distance $s \in [0, 1]$ above its center $c \in [0, 1]$ (see **Figure 3.2**). The coordinates of such a point can also be called the relative population size s and the quantile center c.

By plotting all correlation estimates $r_{c,s}$ for a specific variable pair at the respective coordinates (c, s) for all populations and with a certain granularity, the triangle below the tip is filled. This tip $r_{0.5,1}$ represents the full quantile range, so the whole population. This chart is able to represent the estimated correlation for each subpopulation of a chosen granularity and sorting variable.

The granularity defines how highly resolved the corresponding chart is. It corresponds to a number of bins in which the data can be divided using equidistant quantiles. While the bins are defined by only one sorting variable quantile range, they are a sample of all variables. Using those bins, a correlation can be calculated for each point of the chosen granularity by combining a range of bins instead of selecting data points anew by recomputing their sorting variable quantile.

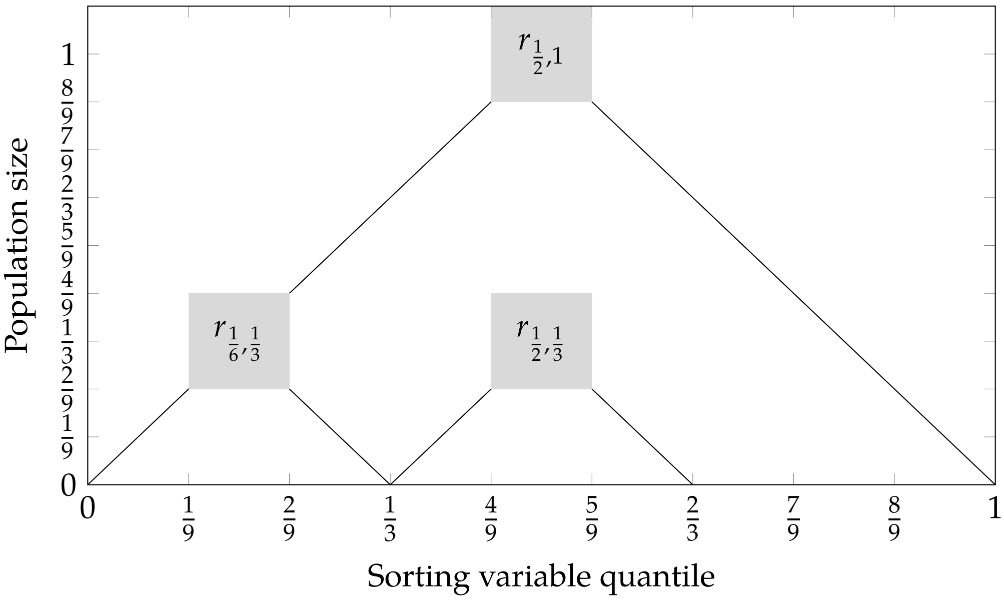


Figure 3.3 With a bin size of $b = 9$, the correlation estimated for the first three bins, and therefore the first third of the population, is represented by the term $r_{\frac{1}{6},\frac{1}{3}}$, which can be visualized as a square colored according to to the population's correlation estimate. The correlation of a subpopulation of the same size spanning the middle third quantile range is $r_{\frac{1}{2},\frac{1}{3}}$, and the correlation for the whole population is $r_{\frac{1}{2},1}$.

Dividing the sorting variable range into a number of bins b like this and inverting the empirical distribution function F maps each bin to an almost equal number of observations (see **Figure 3.2**). The upper limit of b can be specified by a minimum possible sample size of the correlation estimation method, or defaults to the number of observations.

Therefore, the size of samples to estimate the correlation from can be guaranteed to have a lower bound when taking one or multiple bins as sample, as described above regarding quantile ranges, which is useful for correlation estimation methods requiring such a minimum size.

Let b be the number of bins and $q = \frac{1}{b}$ the relative bin size, the value $r_{(i-0.5)q,q}$ $\forall i \in 1, ..., b$ represents the sample correlation estimated for the i^{th} single bin, with observations between the sorting variable quantiles $(i-1)q$ and iq. Higher population sizes include more bins (see **Figure 3.3**).

Using that chart to show a heatmap of all subpopulations' partial or regular correlations allows to spot locally different regions that indicate correlation behavior different from the whole (see **Figure 3.4**). Further characteristics and statistics for subpopulations can also be incorporated in or displayed as MCA plot, which is done in **Chapter 3.2**.

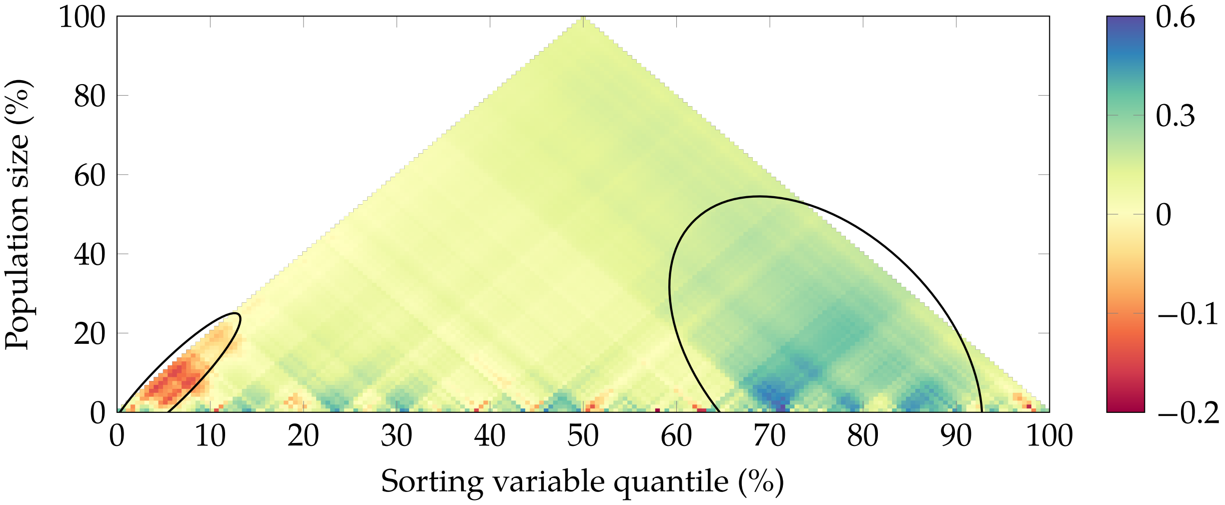


Figure 3.4 Partial correlation MCA example with 201 bins and a spectral color map, with two bigger, differently correlated regions of subpopulations marked.

3.2 A scoring system for subpopulations

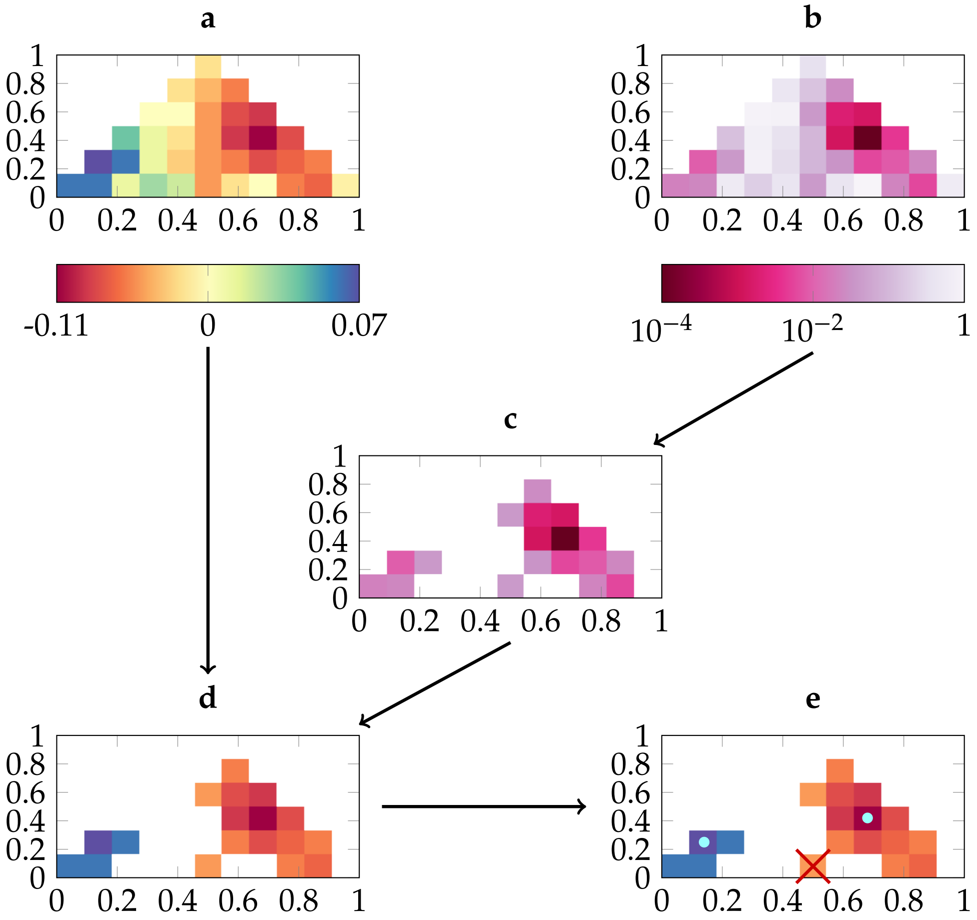


Figure 3.5 Correlations are masked when their p-value is above a certain threshold. **a)** Correlations. **b)** Their p-values. **c)** P-values > 0.05 are masked **d)** Applying the same mask to the correlations forms regions. **e)** Small regions are removed (red cross) and the local score maxima are extracted from the remaining regions (cyan dots).

MCA plots can incorporate more information than correlation alone. For example the p-values of each subpopulation's correlation estimate can be either displayed as triangle like a MCA plot themselves (see **Figure 3.5b**), or used as a thresholding of the corresponding correlation plot, hiding subpopulation points that were not significantly correlated (see **Figure 3.5d**).

P-values are calculated from correlation value and sample size like explained in **Chapter 2.5**. Since the p-value is dependent on population size and correlation, using p-values as score eliminates those small subpopulations which show stronger correlations than bigger ones due to chance.

In order to identify such subpopulations with sufficiently high confidence, a function of the p-values is used as a score with the inherent ability to eliminate low-confidence values. A low p-value can mean either that it is not significant because there is no real correlation, or because the population size was too small to make confident estimations. For each subpopulation of size s and center c with a correlation estimate $r_{c,s}$ and corresponding p-value $p_{c,s}$, the score is defined as

$$score_{c,s} = \begin{cases} 0 & \text{, if } p_{c,s} > threshold \\ -\log_{10}\left(p_{c,s}\right) & \text{else} \end{cases} \tag{3.1}$$

The score-masked MCA plot yields several connected regions in the chart, separated by other regions with p-values exceeding the significance threshold. The bigger such a region, the more robust is the correlation within, since its size means that smaller changes in subpopulation size and quantile do not change the correlation behavior.

Conversely, small or interrupted regions mean that correlation behavior is not stable and can be changed by inclusion or removal of few data points. Similarly, outliers can easily be detected when the inclusion of single bins has great impact on the correlation of subpopulations.

Each of those stable regions contains a score maximum that is a local maximum in the scoring chart. In bigger regions, those maxima define subpopulations that are both big and highly correlated (see **Figure 3.5e**). As previously mentioned, this applies due to the definition of the p-values the score is based on. Due to their robustness, those maxima define subpopulations that are representative for similar subpopulations (see **Figure 3.1**).

Based on the subpopulation sizes of those maxima, the region sizes around them, and other features, filtering criteria for whole MCA plots can be specified.

3.3 A scoring system for MCA plots

Although MCA plots give an overview of all subpopulations' correlations, manual assessment is only feasible for a certain number of combinations, depending on time constraints and patience while skimming plots.

The number of MCA plots possible for a certain set of variables corresponds to the number of 2-combinations between those variables. This number is calculable using the second binomial coefficient. If no suitable sorting variable was found in advance, all plots for all sorting variables have to be calculated, which multiplies this number by the number of variables p:

$$\#combinations_{sorting} = \binom{p}{2} \tag{3.2}$$

$$\#combinations = p\binom{p}{2} \tag{3.3}$$

Those formulas grow in $\mathcal{O}(p^2)$ and $\mathcal{O}(p^3)$, respectively, depending on the number of variables in the input data set. This motivates the automatic elimination of MCA plots – and therefore variable combinations – which are definitely not interesting.

A minimal definition of uninteresting plots can be generated: Plots that are uniformly positively, or uniformly negatively correlated, or those for which only a small number of subpopulations were predicted to be correlated differently than the rest with low confidence (see the regions of $\leq 5\%$ population size in **Figure 3.4**).

As described in **Chapter 3.2**, MCA plots masked using a score threshold yield coherent regions of similar populations (see also **Figure 3.5**). Two subpopulations are considered as being together in a region if they are left, right, above or below another, while both are within threshold, or if both are connected transitively with this criterion. Let $I \subset \mathbb{Z}^2$ be the set of two-dimensional points in the image raster that fall below the threshold. Then $\circledcirc \subseteq I \times I$ is the relation between two such points a and b, so that

$$a \circledcirc b \Leftarrow \quad \forall a = (a_1, a_2), b = (b_1, b_2), c \in I : \begin{cases} a = b & \text{or} & (3.4) \\ a_1 = b_1 \wedge a_2 = b_2 \pm 1 & \text{or} & (3.5) \\ a_2 = b_2 \wedge a_1 = b_1 \pm 1 & \text{or} & (3.6) \\ a \circledcirc c \wedge c \circledcirc b & . & (3.7) \end{cases}$$

Then a region is defined as a point sets within which $\circledcirc$ applies, and the set of regions in an MCA plot is the set of all sets possible according to this definition.

In order to extract potentially interesting plots automatically, several filtering criteria can be applied to each MCA plot's set of regions, a process able to narrow down the number of potentially interesting plots drastically.

A set of criteria able to spot the corresponding kind of potentially interesting MCA plots is the presence of at least two large subpopulations with differing correlation sign in one plot. For example in **Figure 3.5e**, three regions are visible, one of which is pruned for being too small. Due to two big regions with differing correlation signs remaining, the plot is considered interesting. The maximum size to be pruned is a free parameter dependent only on plot granularity.

This algorithm is applied to three data sets in **Chapter 5**.

4 Implementation

In order to do statistical analysis in biology, there are generally three major interaction levels: Programming language libraries, spreadsheet applications, and specialized interfaces for a certain type of analysis. MCA encompasses a novel visualization technique not available in spreadsheet applications, which therefore requires a reutilization of drawing and calculation code to be of use. This and the error-prone nature of spreadsheets (Powell et al., 2008) necessitates a specialized interface or programming library based approach, despite the widespread use of spreadsheets in biological research (Topaloglou et al., 2004).

In order to make MCA available for researchers with and without programming experience, a combined approach was adopted where a specialized front-end was developed based on a documented, independently usable back-end library. This library for the programming language R is capable of performing the automatic part of Multiresolution Correlation Analysis by plotting MCA plots and identifying interesting ones as described in **Chapter 3**. The front-end, a web application, automates this procedure and provides interactive access to the plots of one sorting variable at a time.

4.1 R package

R is an open source programming language with focus on statistics and data processing. It has many statistical functions and plotting facilities readily available in its standard library, as well as an extensive repository of packaged libraries extending the standard library. Those include a partial correlation estimation method suited for small subpopulations, the shrinkage estimator provided by GeneNet.

The R package developed for this work contains a library providing algorithms and data structures for Multiresolution Correlation Analysis. This includes methods to create the matrices holding the MCA analysis data, as well as functionality to generate MCA plots and scatter plots of subpopulations (see **Chapter 3.1**). Finally, it provides algorithms for identifying potentially interesting plots (see **Chapter 3.3**).

In the Appendix, a full API documentation can be found describing the specific methods implemented in the R library.

MCA plots are stored in a structure – the MCA object – containing input parameters such as data, bin number, and correlation method, as well as two recursive lists of MCA plots. Those lists contain an MCA plot for each pair of variables in the data once. This has the advantage of reducing the memory footprint compared to storing the MCA plots twice for the inverted variable order. The MCA plots themselves and their p-values are stored as matrices in two recursive lists like this.

4.1.1 MCA plot creation

Creating a set of MCA plots with partial correlations involves estimating the partial correlations of all pairs of variables for each subpopulation. Since calculating the partial correlation between one pair of variables involves removing the influence of all others, as described in **Chapter 2.4**; the calculation of all partial correlations is done simultaneously. Since only creating one MCA plot would not reduce the overall run time, all plots for one sorting variable are created simultaneously.

As described in **Chapter 3.1**, for each possible subpopulation all correlations between pairs of variables are obtained by applying the GeneNet shrinkage estimator to the values of the current subpopulation. Processing one subpopulation therefore fills one point in each plot of the chosen sorting variable. After all subpopulations are analyzed, all MCA plots are filled and available. This process is described in **Listing 4.1**.

```
def create_mca(DATA, p, b):
    height ← ceil(b ÷ 2)
    offset ← 1 if is_odd(b) else 2
    # MCA_X and MCA_P ∈ $\mathbb{R}^{p \times p \times b \times height}$
    MCA_X ← [p×p×b×height]
    MCA_P ← [p×p×b×height]
    for c in 1 to b:
        h_max ← c if c ≤ height else height + offset - c
        for h in 1 to h_max:
            # X and P ∈ $\mathbb{R}^{p \times p}$
            subpop = extract_subpopulation(DATA, c, h)
            X ← estimate_partial_correlations(subpop)
            P ← fdrtool_pvalues(X)
            for v1 in 1 to p:
                for v2 in 1 to p:
                    MCA_X[v1, v2, c, h] ← X[v1, v2]
                    MCA_P[v1, v2, c, h] ← P[v1, v2]
    return MCA_X, MCA_P
```

Listing 4.1 Pseudocode for MCA creation with `DATA` being the data set, `p` the number of variables in it, `b` the wanted number of bins, `estimate_pcors` the shrinkage estimator from GeneNet (Schäfer and Strimmer, 2005), and `fdrtool_pvalues` the p-value calculation method from Strimmer (2008).

MCA plots for non-partial correlations do not have to be created this way, since only partial correlations are dependent on other variables than the two the correlation is estimated for, therefore the correlation for each subpopulation and plot can be estimated individually.

A minimal session to create a MCA object sorted by a variable called “var1” from a data file named “data.dat”, and assigning it to the name “mca.variable1” looks as follows:

```
library(mca)
data <- read.table('data.dat')
```

```
mca.variable1 <- mca(data, 'var1')
```

4.1.2 Plotting

Visualizing MCA requires the computed MCA object, which contains the common properties of all plots, and a pair of variables to be plotted. Other parameters can be specified to override defaults, like for example the color map translating partial correlations to colors, the p-value threshold, and axis labels.

Plotting an MCA plot with default settings for the variable combination “var2” and “var3” from the MCA object created earlier looks as follows:

```
plot(mca.variable1, 'var2', 'var3')
```

and plotting all variable combinations containing “var2” with the strict significance threshold of 0.001 and a red-black-green colormap is done like this:

```
plotMCAs(mca.variable1, 'var2', 0.001, palette = 'rbg')
```

4.1.3 Filtering interesting MCA plots

Alternatively, all plots of an MCA object can be visualized and optionally saved to disk, or interesting plots can be extracted and visualized together with their scoring plots and scatter plots of the subpopulations with the highest scores.

Multiple functions on different abstraction levels exist in the R library to automate or manually execute the steps of extracting maxima, plotting and listing MCA plots whose criteria match the ones described in **Chapter 3.3**. The full API documentation can be found in the Appendix.

Finally, the filtering for interesting plots sorts the found results by maximum distance between correlation maxima, a criterion that favors plots with strong, differential correlation. Since the filtering only allows plots where at least one positive and one negatively correlated score maximum is available, this score can be defined as the distance between the correlations of the most positive and the most negative subpopulation:

$$c_{\min}, h_{\min} = \operatorname*{argmin}_{c,h} score_{c,h} \tag{4.1}$$

$$c_{\max}, h_{\max} = \operatorname*{argmax}_{c,h} score_{c,h} \tag{4.2}$$

$$dist = r_{c_{\max},h_{\max}} - r_{c_{\min},h_{\min}} \tag{4.3}$$

Finding and plotting potentially interesting variable combinations, as well as their scoring charts and scatter plots, and saving them to the directory “mca-variable1” looks as follows:

```
interesting <- findInteresting(mca.variable1, base.dir='mca-variable1/')
head(interesting)
```

This code snippet generates a series of plots sorted by the maximal difference of the subpopulation scores, and saves them to the specified directory on disk. The top ten results are also displayed on screen.

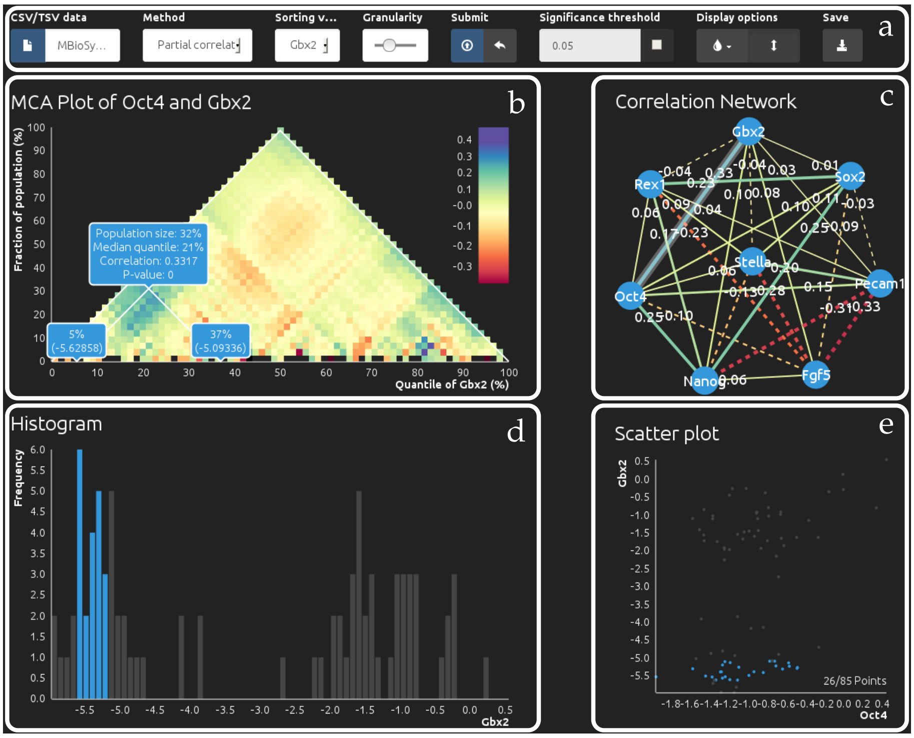
Creating and plotting the subpopulation scores including found maxima manually for only one known variable combination is also possible:

```
scores <- populationScores(mca.variable1, 'var2', 'var3')
plot(scores)
```

4.2 Web application

The web application offers a fully interactive interface to all functionality available in the R library, except from automatability. It is intended for biological researchers with little or no programming experience who want to assess subpopulations in their data using only a mouse or touch driven interface.

It provides a server application that can be run locally by more programming-experienced researchers or remotely by an institute or hosting company. The front-end is realized as single interactive web page. The server is available for local installation on the attached CD and available on request from ICB, where it is also deployed on an internal server.



CSV/TSV data: MBioSy...
Method: Partial correlat
Sorting v...: Gbx2
Significance threshold: 0.05

Figure 4.1 Screenshot of the web application. **a)** Parameters for MCA creation and display options. **b)** The current MCA plot. **c)** The graph used to select variable combinations. **d)** The sorting variable histogram. **e)** The scatter plot of the selected variables.

The web application consists of three structural parts: A back-end consisting of a WebSocket server, an R script allowing the server to interface with the R library described above, and a front-end implemented as a web page client implemented using HTML5, SVG and JavaScript (see **Figure 4.1**).

4.2.1 Communication Architecture

The communication between server and client happens via standard HTTP requests for assets like style sheets and the page itself, as well as WebSocket technology for client-server communication. WebSockets are full-duplex channels between server and client that use a persistent TCP socket connection to exchange messages (Pimentel and Nickerson, 2012). Once the page is loaded, a WebSocket connection is established and used for transmitting all computation requests from the front-end to the back-end, as well as returning their progress and results to the front-end.

The constant serial connection makes it feasible to show a real-time progress status as seen **Figure 4.2** of server side computations for the client, which would be far less immediate and more error-prone if HTTP requests were used. This is the case because with WebSockets, multiple messages can be sent from the server at any time, while HTTP requests only allow one server response per request, and therefore have to resort to a polling solution with answers arriving at time points decoupled from the order in which they were sent (Pimentel and Nickerson, 2012). This would also cause traffic overhead due to request headers being repeatedly exchanged.

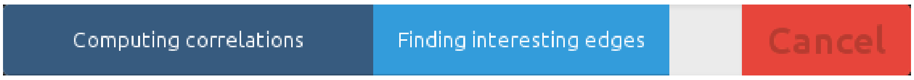


Figure 4.2 Real-time progress bar able to give exact feedback about the server-side state

The server-side implementation is written in the Python programming language and can be used as coupled HTTP/WebSocket server able to serve static files as well as WebSocket connections, or a standalone WebSocket server used in tandem with an arbitrary static file server configured to dispatch WebSocket connections to it. Its implementation uses the Autobahn | Python library (Tavendo GmbH, 2014) as well as the AsyncIO library recently included into Python's standard library. This allows for multiple parallel clients running, cancelling and monitoring the progress of their own computations independently from each other.

Requests for Multiresolution Correlation Analyses carry the data set, correlation method, sorting variable, and bin number, from which a unique identifier is derived. Computed MCAs are stored in a persistent cache database using this identifier, so that they do not have to be reproduced for the same data set and parameters when those are requested again by the user. The database is a lightweight SQLite container with a configurable path, chosen for portability. On demand, the database connection logic can be easily replaced with a database server connection, since the queries and insertions are standard SQL. Old cache entries can be removed via SQL using the stored time stamps for each entry.

Once a new request not available in the cache is encountered, and due to R's nonexistent capability for inter-process communication, a sub-process is started to pass the parameters received via WebSocket to the R library and translate the progress messages, as well as the final MCA result to the JSON format. Communication happens via JSON because the WebSocket specification includes compression of the otherwise verbose serialization format, and JSON is the fastest possible format for JavaScript to parse (Nurseitov et al., 2009).

4.2.2 Client architecture

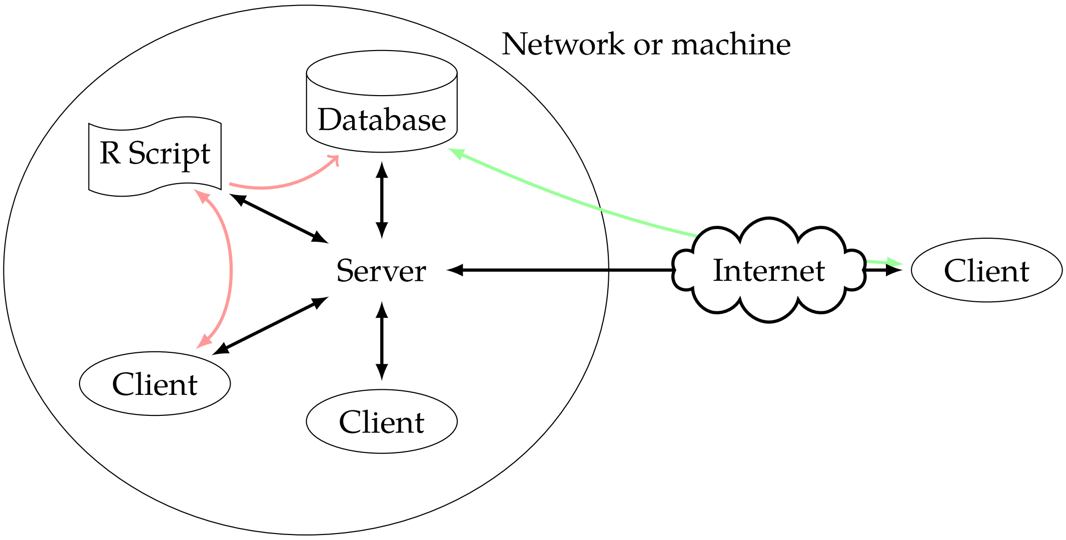
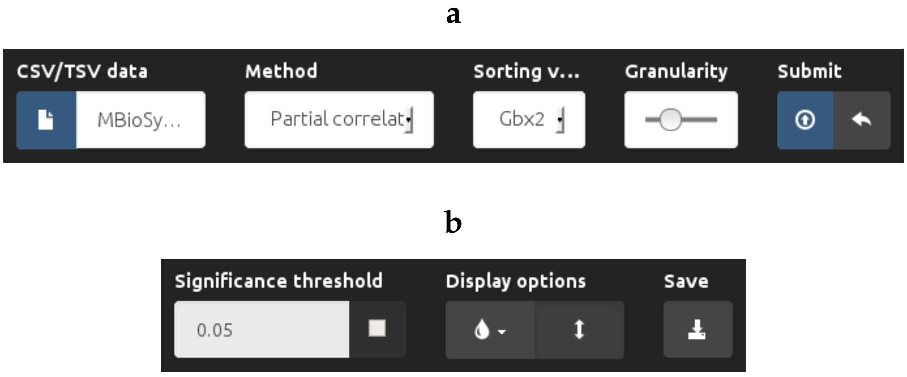


Figure 4.3 Interactions between the back-end and front-end. The server is able to interact with multiple clients transparently, independently if they are located on the same machine or network, or through the internet. Parameters get queried in the database cache and a MCA is fetched there if available (green arrow). If the right MCA is not available, the parameters get passed to the R script wrapping the MCA library, whereupon the result is sent back to the client (red double arrow) and the cache (red single headed arrow).

The webpage client is built to leverage modern web technologies and browser capabilities including the aforementioned WebSocket protocol, the HTML5 canvas, CSS3 and SVG. It heavily relies on the D3.js library (Bostock, 2014) for binding data to document elements, which enables high-performance visualizations and charts, and the Bootstrap 3 CSS framework for responsive visual design on all device form factors from handheld-size to widescreen displays.

4.2.3 Visualizations

Up to four interactive visualizations can be active, depending on the parameters specified by the user: A correlation network graph showing all overall correlations between variable pairs for the currently active data set and sorting variable, a histogram of the sorting variable value distribution, the MCA plot between the currently selected



CSV/TSV data: MBioSy…

Method: Partial correlat

Sorting v...: Gbx2

Granularity

Submit

Significance threshold: 0.05

Display options

Save

Figure 4.4 **a)** Parameters for the MCA calculation from left to right: Data set, correlation method, sorting variable, bin number, as well as buttons to submit or revert the parameters. **b)** Display options, such as toggleable significance threshold, color map selection and color bar scaling, as well as a button to save the MCA plot as SVG image.

correlated variables and the sorting variable, and a scatter plot between the currently selected variables (see **Figures 4.5a, b, c, and d**).

Apart from the parameters to be submitted to the server, visualization options are available that apply to MCA plot and graph (see **Figure 4.5a**): A colorblind-friendly set of color maps known as ColorBrewer (Brewer, 1994), the option to automatically scale the color range to minimum and maximum correlation available, and a toggleable significance threshold for the MCA plot.

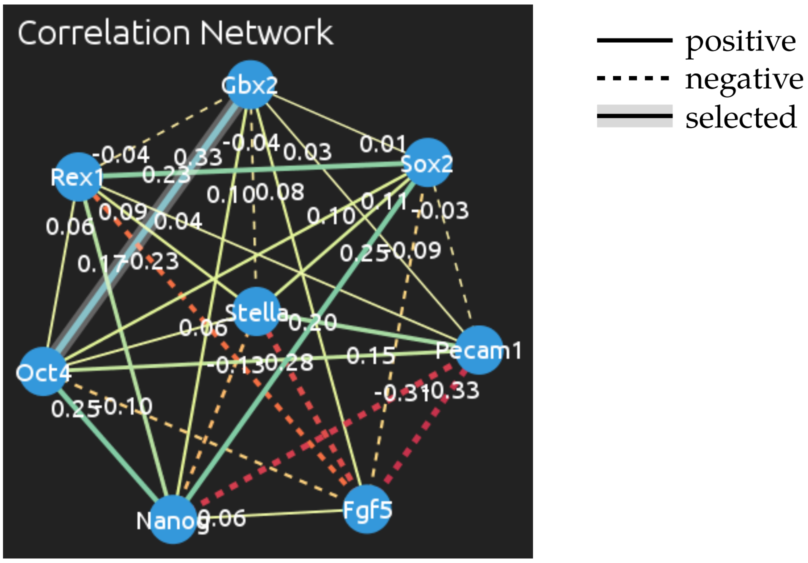


Figure 4.5a Variable combination graph with the color map scaled to the values available in the MCA plot. Edges with negative correlation are additionally drawn dashed, and the selected edge has a grey aura.

The graph is manipulable for easier assessment of the correlations of all variable combinations which are shown both numerically and visualized as edge thickness and color (see **Figure 4.5a**). Another important function of the graph is to provide an interface to select edges and therefore variable pairs to display as MCA and scatter plot. Alternatively to the graph, a table of variable combinations is displayed, filtered for interestingness and ranked by the maximum correlation distance, as described in **Chapter 4.1.3**. If the variables in the data set are more than ten, the set of all variable combinations to be displayed as graph is considered to be too big for being displayed as a graph, and only the table is available.

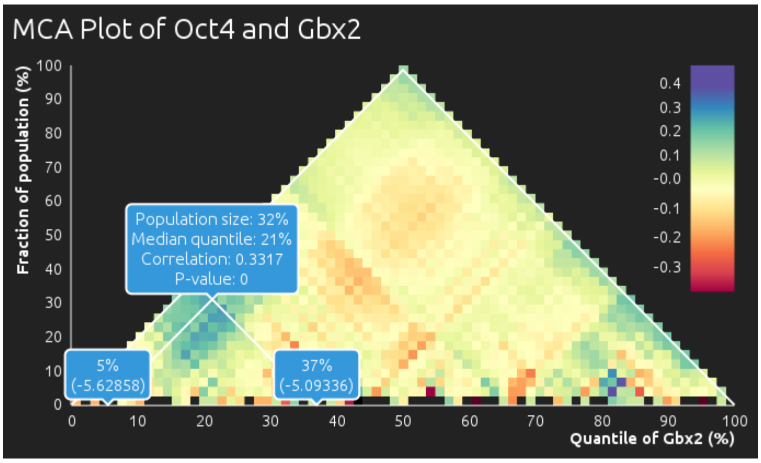


Figure 4.5b **(a)** MCA plot, with the color map scaled to the available correlation range using the respective display parameter.

The MCA plot (shown in **Figure 3**) is realized using both a HTML5 canvas and SVG in order to leverage the performance advantages of both technologies (Smus, 2009). The visualization responds to the chosen display options and allows introspection of subpopulations by hovering the mouse over them. When hovered, lines indicate the subpopulations and bins included in the selected subpopulations, and three tool tips indicate information about it, such as size, median percentile, correlation and p-value, as well as start and end percentile, and the corresponding sorting variable values.

When selecting a subpopulation by hovering above the respective plot point, the corresponding bars and points on both histogram and scatter plot, get highlighted as well, as seen in **Figures 4.5c**. Also, the correlation of the selected subpopulation for all variable combinations gets shown on the graph edges.

The histogram, when hovered, also shows a tool tip of the range of sorting variable values in the hovered bar, as well as the number of values in that bar, i. e. its height. The histogram serves to show how range binning differs from quantile binning and from what values exactly a hovered subpopulation is composed.

The scatter plot shows the number of observations, as well as the relation of the two selected variables by plotting the values of both against each other for each observation in a 2D coordinate system. It also shows the number of observations and, if applicable, of observations in the currently selected subpopulation. The highlighted subpopulation on the scatter plot visualizes its correlation.

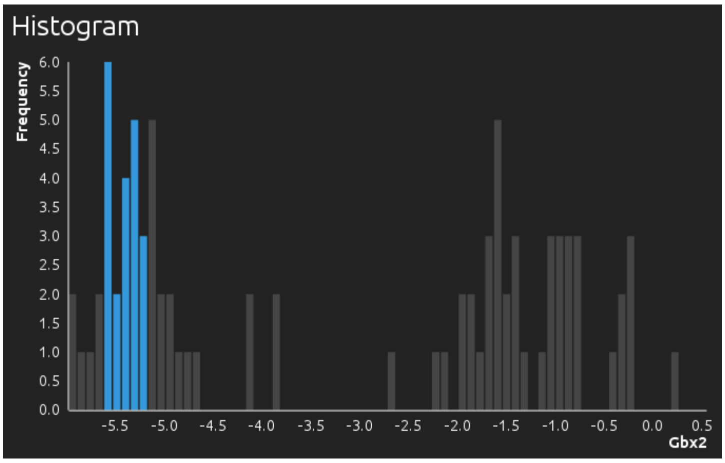


Figure 4.5c Histogram with bars highlighted blue that correspond to the subpopulation selected in **Figure 3**.

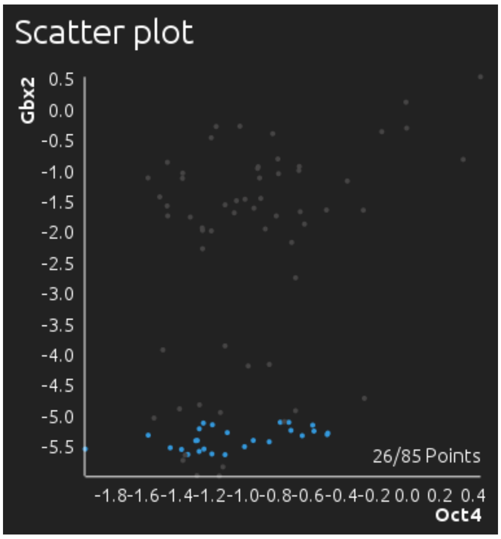


Figure 4.5d Scatter plot with the subpopulation selected in **Figure 3** highlighted blue. The relatively strong correlation of the subpopulation is reflected in its elongated shape of the highlighted section in the scatter plot.

4.2.4 Client usage

Due to the dependencies in visualization from parameters and variable pair, a session involves the following interaction levels:

- The user selects a data file to upload, a sorting variable, a correlation method, and a granularity that determines the number of bins. The data file is allowed to be in common text delimiter based formats like CSV or TSV with header (see **Figure 4.6**).

  The maximum available granulatity is defined by a pragmatic bin number maximum calculated from the number of variables p:

$$(1000/p) \cdot p^{0.3} \tag{4.4}$$

a

```
A,B,C
3.13,4.18,0.2
1.5e-10,4,5.4
```

b

A	B	C
3.13	4.18	0.2
1.5e-10	4	5.4

Figure 4.6 Example CSV file. **a)** Standard comma separated values format. **b)** Table data structure with a variable name per column derived from it.

With the default granularity being a third of this maximum. The default correlation method is partial correlation.

- A two-part progress bar shows the current state of the MCA computation and extraction of interesting plots.
- After computation, a network graph is shown with nodes for all variables in the data and edges displaying the correlations for each pair of variables. If the data set has too many variables to sensibly display as a graph, a table of interesting variable combinations is shown instead.
- A histogram showing the density of the sorting variable is shown.
- Once a variable combination is selected by clicking a graph edge or table row, the MCA plot of the corresponding variable pair is shown, as well as a scatter plot of the variables.

Options

The MCA plot and graph share the same color map, which can be scaled to display maximal contrast for the available correlations in the plot. The p-value thresholding can also be adjusted and disabled.

By moving the mouse over the MCA plot, information about the subpopulation selected this way is shown, such as population size and median quantile, as well as correlation and p-value. The data in the subpopulation is highlighted on the scatter plot and histogram, and, if available, the graph is updated to show the correlations of other variable combinations for that subpopulation.

Finally, the user can download the current MCA plot as image file.

5 Application

In order to demonstrate the usability of MCA for hypothesis generation in data sets and its advantage over simply subpopulation extraction using a threshold, MCA is applied to three data sets of different content and size.

5.1 KORA (*Ko*operative Gesundheitsforschung in der *R*egion *A*ugsburg)

KORA is an observational study on health and lifestyle related data. It provides a research platform that can be accessed by international research groups after signing a project agreement. In the course of the initial KORA plan, four series of "S" surveys were carried out every five years (Peters, 2012). The protocol of each survey set included examination of a different representative sample of Augsburg residents in the age of 25 to 74. Those were then repeatedly tested in follow-up "F" studies as long as the participants were available.

The last and ongoing phase of KORA is the evaluation of the data resulting from each series of initial and follow-up studies. Included in the data gathered by KORA, depending on the survey could be lung and skin examinations, balancing ability, ECGs and ultrasonic bone density assessment (Peters, 2012).

The study providing the data on which the following MCA is based is F4, the follow-up study of the survey series S4, the last initial study, which lasted from 1999 to 2001. Of its 4261 initial baseline examination participants between the ages of 25 to 74 years, 3080 participants between 32 and 77 years were still available for the follow-up study F4 in the years 2006 to 2008 (Meisinger et al., 2010).

Application

For MCA, a version of the F4 data cleaned for missing values was obtained from Jan Krumsiek. This data set contains 1765 observations of 355 metabolites and the additional variables gender, body mass index (BMI), age, as well as high- and low-density lipoprotein; HDL and LDL. Consequently, this amounted to 360 variables and therefore 64 620 possible combinations and MCA plots.

As an example application, the interesting plot extraction algorithm described in **Chapter 3.3** was applied to Pearson correlation MCA plots. Sets of MCA plots were generated for each of the non-metabolite variables as sorting variable, and for Pearson as well as partial correlations. Pearson correlations are easy to interpret and the resulting sets of plots showed no obvious differences to the ones of the partial correlations, and were therefore chosen for further analysis.

a

Score	Variable 1	Variable 2
0.35	heptanoate70	pyridoxate
0.33	ldl	urate
0.32	alphatocopherol	urate
0.32	caproate60	pyridoxate
0.32	alphahydroxyisovalerate	valerate

b

Score	Variable 1	Variable 2
0.32	ADpSGEGDFXAEGGGVR	butyrylcarnitine
0.31	indolepropionate	linoleate182n6
0.3	bilirubinZZ	cysteineglutathionedisulfide
0.3	methionine	threitol
0.3	dehydroisoandrosteronesulfateDHEAS	tyrosine

c

Score	Variable 1	Variable 2
0.41	acetylphosphate	arabitol
0.33	alanine	mannose
0.31	cortisol	pcresolsulfate
0.3	serotonin5HT	valerate
0.3	arachidonate204n6	phenylalanine

d

Score	Variable 1	Variable 2
0.36	pcresolsulfate	phenolsulfate
0.29	alphahydroxyisovalerate	uridine
0.28	betaine	caffeine
0.27	eicosapentaenoateEPA205n3	glutamate
0.26	ADpSGEGDFXAEGGGVR	cortisone

Table 5.1 The 5 MCA plots in KORA with the maximum correlation distance, as described in **Chapter 4.1.3** for each of the sorting variables **a)** age, **b)** BMI, **c)** glucose, and **s)** HDL.

For each sorting variable, 162 to 371 plots with potentially interesting properties were found. The found Pearson correlation MCA plots were ranked by maximum correlation distance between local score maxima. Among those plots were 120 to 284 combinations with unknown metabolites. Filtering out those and all variable combinations containing the sorting variable itself left 23 to 81 potentially interesting plots per sorting variable. The top 5 of those plots, according to the maximum correlation distance described in **Chapter 4.1.3**, are listed in **Table 5.1**.

To illustrate the different subpopulations, the two top-scoring MCA plots for age, BMI and HDL are also replicated in **Figure 5.1**.

For some of those combinations, connections to the literature could be made. These combinations include the second-top scoring age MCA plot (**Figure 5.1b**), which shows negative correlations between LDL and urate for older study participants and positive ones for younger. Indeed, serum urate excretion declines with age, while urate is antioxidant for LDL at high concentrations, but pro-oxidant at low concentrations (Stiburkova and Bleyer, 2012 and Filipe et al., 2002). Increasing urate concentrations therefore stop oxidizing LDL and starts antioxidizing it, which could explain the inverting correlation.

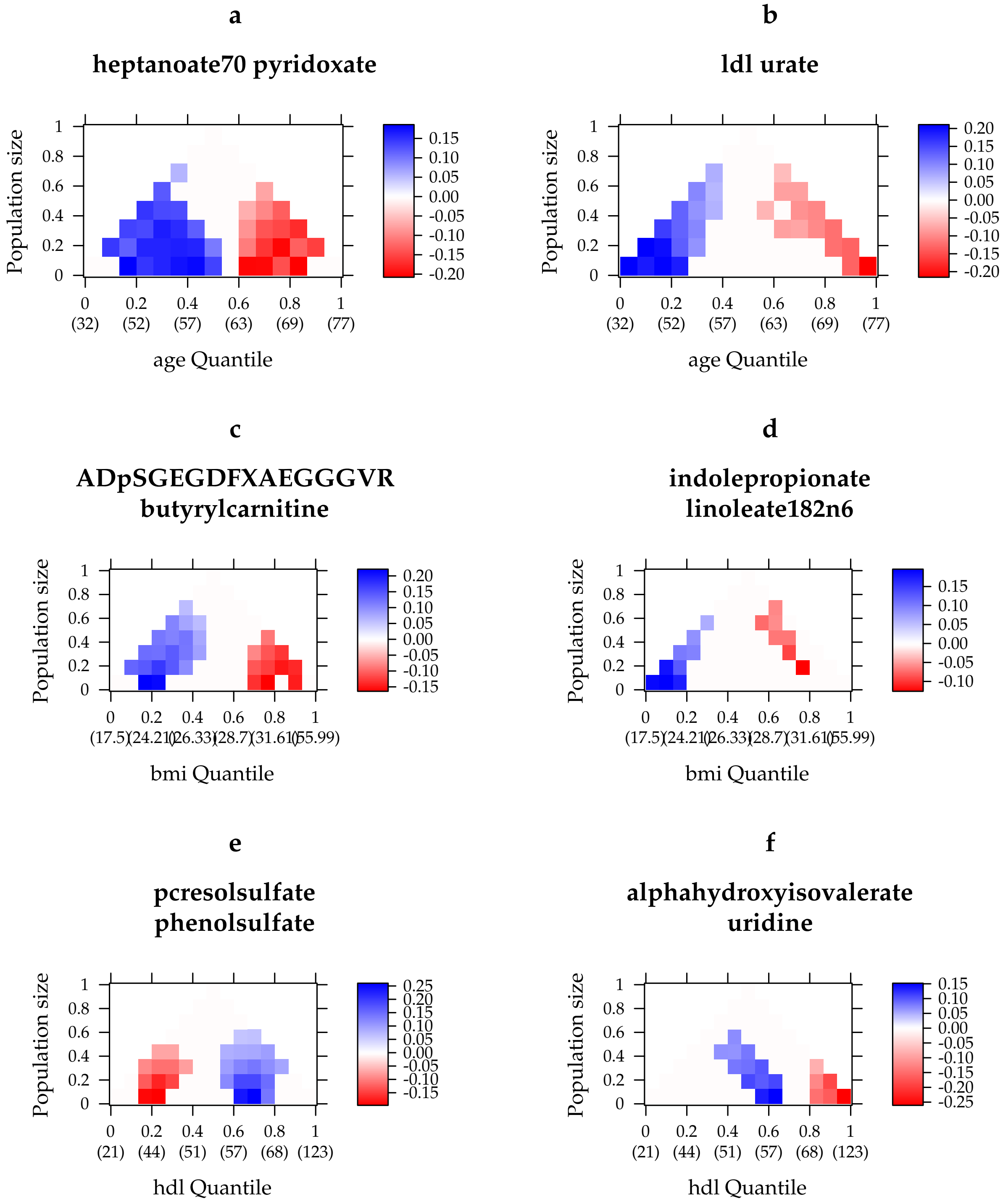


Figure 5.1 Potentially interesting KORA variable combinations. Examples of the top scoring Pearson correlation MCA plots for each of the variables **ab)** age, **cd)** BMI, and **ef)** HDL. The real, non-quantile values are displayed in braces along the x axes.

Others have obvious connections like the top scoring MCA plot for HDL (**Figure 5.1e**) where phenol sulfate and 4-methyl-phenol sulfate (another name for p-cresol sulfate) only differ by methylation. This modification might occur depending on HDL level or a process responsible for it (see **Figure 5.2**).

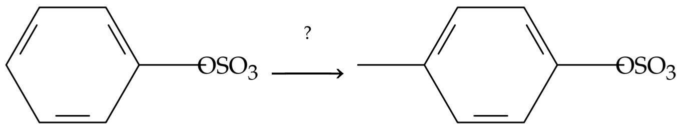


Figure 5.2 Methylation of phenol sulfate to p-cresol sulfate. The reactants are positively correlated for low HDL concentrations and negatively for high ones in the KORA data set.

5.2 Qatar Metabolomics Study on Diabetes (QMDiab)

QMDiab is a 2012 study from the Dermatology Department of Hamad Medical Corporation in Doha, Qatar. The incentive was Qatar's high prevalence of type II diabetes mellitus, where the country ranked #21 worldwide in 2013 (International Diabetes Federation, 2014).

a

Score	Variable 1	Variable 2
0.15	estrone-3-sulfate	nicotinamide
0.12	cortisol	indoleacetylglutamine
0.12	adenosine-5-monophosphate (AMP)	BMI
0.11	dihydroorotate	scyllo-inositol
0.11	cystine	N-acetylphenylalanine

b

Score	Variable 1	Variable 2
0.12	beta-sitosterol	bisphenol-A-monosulfate
0.12	dihydroorotate	indolepropionate
0.11	isovalerate	pantothenate
0.11	indoleacetylglutamine	undecanoate (11-0)
0.11	glycolate (hydroxyacetate)	stearamide

c

Score	Variable 1	Variable 2
0.13	estrone-3-sulfate	indoleacetate
0.13	allantoin	LDL
0.12	estrone-3-sulfate	glycolate (hydroxyacetate)
0.12	methyl-4-hydroxybenzoate	pro-hydroxy-pro
0.12	homostachydrine-	N6-carbamoylthreonyladenosine

Table 5.2 The 5 top scoring MCA plots in QMDiab for each of the sorting variables age, BMI, and glucose.

The study measured metabolites in 369 individuals within the age of 17 to 81. The metabolites were measured in the three body fluids non-fasting blood plasma, urine, and saliva. In the time from February to June 2012, 1107 samples were taken from the participants, comprising 1563 metabolites including amino acids, peptides, carbohydrates and lipids, as well as age, gender, ethnicity, weight, height, body mass index (BMI) and personal history of diabetes type II (Do, 2013).

The samples were analyzed by the three companies Metabolon Inc., Chenomx Inc., and Biocrates Life Sciences AG. The respective companies utilized liquid/gas chromatography with mass spectrometry injections, targeted profiling using nuclear magnetic resonance (NMR), and multiple reaction monitoring (MRM) (Do, 2013).

The study found that all variables of ethnicity, gender and smoking had a strong effect on a diabetes risk factor, advanced glycation end products. So were women, Arabs, Filipinos, and smokers more strongly affected than men, south Asians, and non- or irregular smokers (Do, 2013).

Application

For an example MCA application, plasma was selected of the three fluids due to it being the most complete set, in which 758 of the 1563 total metabolites were measured. The data was minimum-imputed, which describes a method of cleaning it for missing values using the minimum concentration for a variable as described in Do (2013). This resulted in a data set of 636 metabolites, of which many were unknown, as well as the variables age, gender, BMI, HDL, LDL, and logarithmized glucose concentration from the urine data.

MCA plots were again created and ranked as described in **Chapter 5.1**, only for partial correlations. Similarly to the KORA data set, the QMDiab data set and therefore the potentially interesting MCA plots contained many unknown metabolites. After running the extraction algorithm for interesting plots, the number of MCA plots per sorting variable were reduced from around 2500 to around 310 by eliminating the combinations where at least one of both variables was an unknown metabolite.

Searching for the participating variables in the literature provides some insight to the subpopulations found like this. For example, the correlation between cortisol and the acetylated glutamine derivative indoleacetylglutamine seems to be negative in very old participants, and not significantly correlated in the young, while being positively correlated in people around the age of 50 (see **Figure 5.3b**). A rise in plasma cortisol concentration, which can be the result of prolonged bodily exercise, stimulates glutamine release (Gleeson, 2008). Also, plasma cortisol levels increase with age in obese subjects (Chalew et al., 1993), which, together with the earlier mentioned diabetes mellitus tendency in the region, might explain the later negative correlation as an adaption effect.

Another example is the relation between allantoin and low desity lipoprotein (LDL). Apparently, these variables are negatively correlated only if the urine glucose level is low, and positively or not significantly correlated for higher glucose levels (see **Figure 5.3f**). Uraic acid, also named urate, is a precursor to allantoin.

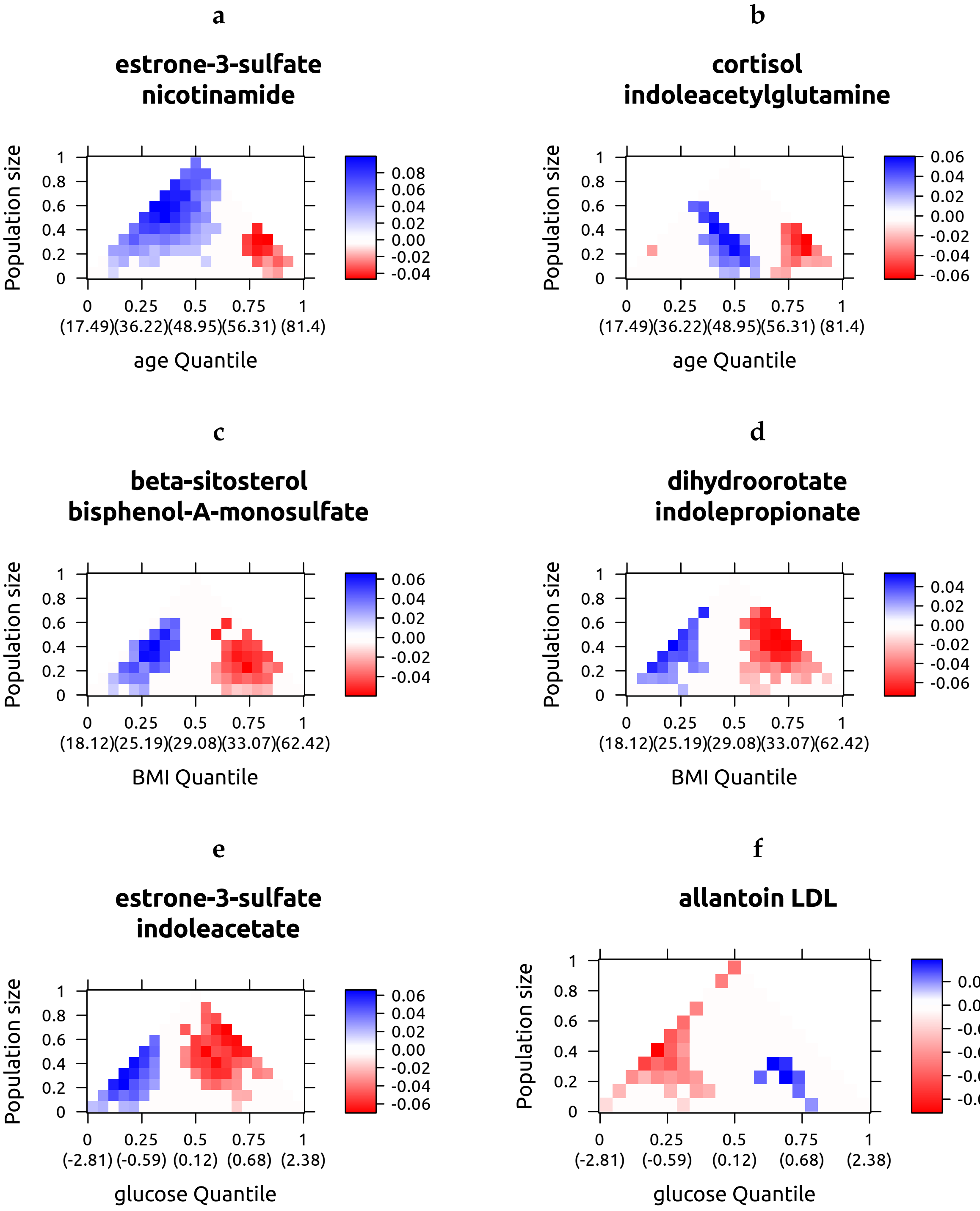


Figure 5.3 Potentially interesting QMDiab variable combinations. Examples of the top scoring partial correlation MCA plots for each of the variables **ab)** age, **cd)** BMI, and **ef)** glucose

The urate/allantoin ratio is an indicator for oxidative stress, since urate reacts with oxidants, which is a step in the reaction from urate to allantoin (Mikami et al., 2000). This reaction with oxidants leads to an antioxidant effect of low density lipoprotein under high urate concentrations (Kopprasch et al., 2000).

Also, uraic acid levels are lower in people with diabetes mellitus (Cook et al., 1986), which is consistent with the high urine glucose levels and high oxidative stress levels found in diabetes type II patients (King and Loeken, 2004). Combined, this confirms that non-diabetics with low urine glucose levels have less oxidative stress, less allantoin, more uraic acid and more unoxigenized LDL, with the situation reversed for type II diabetics.

5.3 Ensemble networks in embryonic stem cell populations

Apart from metabolic data that contains a multitude of variable combinations that have to be filtered computationally, MCA is also suited for manual examination of whole data sets with a surmountable amount of MCA plots or a goal-oriented approach of testing existing hypotheses.

Trott et al. (2012) published an analysis of heterogeneity in subpopulations of mouse embryonic stem cells (mESCs). They used transcript level data from 83 mESCs by Hayashi et al. (2008). The cells were grown in a medium promoting pluripotency and the mRNA transcript numbers of 9 transcription factors were measured using RT-PCR. Those transcription factors contained amongst others Sox2, Oct4, and Nanog, a known triad of pluripotency related transcription factors called the SON genes (Trott et al., 2012). Using principal component analysis, they were able to see a separation between three subpopulations of cells with different expression behavior depending on Nanog and Fgf5 levels: Low Nanog and Fgf5$^+$, high Nanog and Fgf5$^-$, as well as low Nanog and Fgf5$^-$.

They proposed that the heterogeneity in the data was representative for multiple subnetworks, active in each of the subpopulations, of the gene regulatory network. Argumenting from their reconstructed subnetworks, they proposed that Rex1 expression levels are indicative of the cell state.

They calculated Pearson correlations for two sets of subpopulations defined by Fgf5 and Nanog levels by choosing a threshold between low levels and high levels of the data. For those subpopulations they found different correlation behavior of some variable combinations (see **Figure 5.4a and b**).

Application

MCA is able to display all subpopulation categorizations based on one variable at once, which provides greater insight than simply choosing a high/low threshold (see **Figure 5.4c and d**).

The Pearson correlations of Gbx2 and Sox2, as well as Pecam1 and Stella can be confirmed to be significantly positive for high Fgf5 (varying from above the 50th percentile to above the 62th) and insignificant for lower Fgf5.

But also, using MCA plots, it can be seen for Gbx2 and Sox2 that the overall positive correlation comes from certain bins around the 80th pertencile casting streaks across

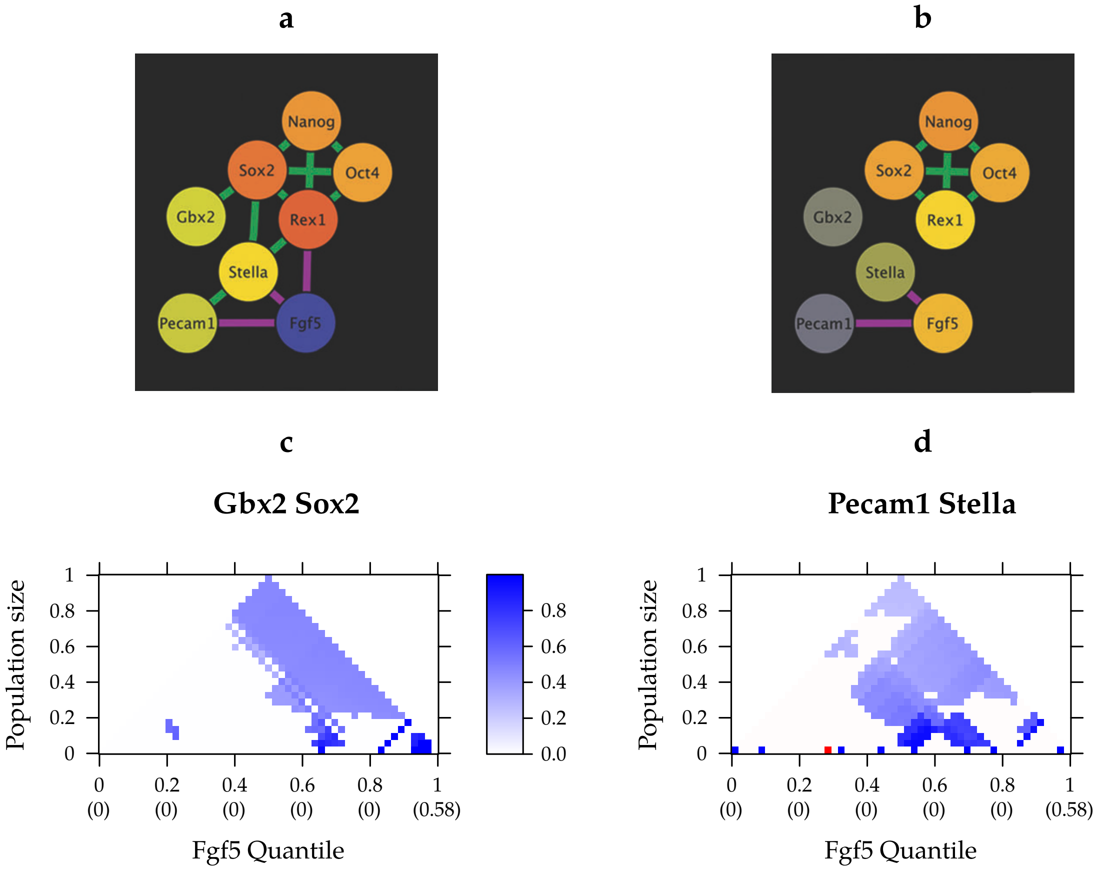


Figure 5.4 Subpopulations of mouse embryonic stem cells based on Fgf5 level are differently collelated. **ab)** Trott et al. (2012) found different correlations for subpopulations depending on low (**a**) and high Fgf5 (**b**). Edges designate significantly correlated variable pairs, with purple being negative, and green ones being positive correlations. **cd)** The MCA plots confirm those results, while providing more insight by displaying many subpopulations instead of just 2. (Graphs **a** and **b** taken from Trott et al. (2012))

the plot. If those bins are assumed outliers and only considering subpopulations containing lower or and higher Fgf5, not much significant plot area is left.

Also, considering the large robust area in the plot for Pecam1 and Stella, which is also robustly significant for subpopulations including only medium Fgf5, and MCA plot allows finer-grained information and therefore an improved conclusion over the proposal derived from the graph and made using only two subpopulations that "Pecam1 and Stella are insignificantly correlated in low-Fgf5 subpopulations".

Using MCA plots, one is easily able to identify subpopulations like the ones found by Trott et al. (2012). This process, however, is more interactive, intuitive and versatile than finding a threshold like they did, as looking at a chart is sufficient to recognize subpopulations of any size and location, and other local phenomena like outliers do not go undetected.

6 Conclusion

In this work, a method for the assessment a kind of subpopulations based on the level of one variable was presented that allows to see correlation structures of subpopulations and therefore identify a set of interesting subpopulations which may overlap and leave out data points.

This Multiresolution Correlation Analysis is able to visualize correlations between two variables for all subpopulations defined by a sorting variable in one novel chart type, the MCA plot. This MCA plot is based on a sorting variable quantile, as well as two coordinates derived from it: Subpopulation size and median quantile.

Because the manual assessment of MCA plots is only feasible if the number of potentially interesting plots is small enough, their use would be constrained to data sets where either the number of variables is small enough to assess all plots, or the potential for interestingness is known. This would only be the case when reducing the plot number by excluding certain variables from the set of which plots are generated – a likely candidate is for example the sorting variable – for example when testing a theory.

Therefore, a heuristic criterion to eliminate uninteresting plots was found and implemented, which is based on extracting significant regions from each plot, each corresponding to a stable, significantly correlated subpopulation, and comparing the MCA plots using those regions' features. These features include correlation distance between the subpopulations, number of subpopulations and corresponding region size, a measure related to correlation robustness.

The methods were implemented in R and exposed in an interactive web application that will be available for public use on the ICB's homepage[1], which includes other statistics and visualizations in addition to MCA plots that react to interactive input such as subpopulation and variable pair selection per mouse. Those include a scatter plot visualizing correlations and a histogram that shows the sorting variable distribution in depencence of the selected subpopulation.

Finally, the algorithm to extract interesting MCA plots was applied to two data sets, after which some of the top-scoring combinations could be found plausible from literature. MCA was also performed on a third data set, where the results of the accompanying paper could be confirmed and extended upon.

Results

Despite the relevance of subpopulations in single cell data, methods for their identification were found to be lacking. Clustering algorithms are unsuited to find overlapping subpopulations based on correlation, and using a single thresholding of one variable level to divide all data into two subpopulations is unable to find overlapping subpopulations as well. The latter also proved to be sensitive to local effects that cannot be

[1] http://helmholtz-muenchen.de/icb/

described by a single threshold, such as outliers and randomly found fragile subpopulations.

MCA was shown to be an effective tool able to visualize and partially automate subpopulation identification in biological data. It proved to be useful in the correlation analysis of pairs of variables by uncovering subpopulations with interesting correlation behavior, and showed information regarding correlation robustness of those subpopulations, and outliers.

The algorithm for the extraction of potentially interesting plots was able to automatically find plausible heterogeneous correlation behavior for metabolic variable combinations, which was not possible with existing tools before MCA.

MCA has been made usable through the web application for biological researchers without programming experience, and additionally through a well-documented R package for computational biologists. Both are simple to access for users of the web frontend or the R library, respectively.

Outlook

Looking forward, the integration of MCA in a subpopulation analyis workflow will be of advantage. Extension of the method for integration of found robust subpopulations into heterogeneity-sensitive network inference might be an interesting project.

While working on this thesis, I implemented another method related to correlation-based subpopulations, that is a hierarchical clustering algorithm which merges existing EM clusters according to internal correlation. Its goal is to find clusters with different locations and sizes, but similar correlation behavior. It works by initializing a separate correlation model per cluster and iteratively re-calculating a smaller set of correlation models as long as fitting it to all clusters improves its BIC score consisting of fitting correctness and model parameter number.

It was not included into this thesis due to its different nature and reliance on a good clustering, a requirement that, as mentioned earlier, is not fulfillable in data with overlapping subpopulations. It nevertheless shows promise that can be expanded upon in cases where precise, yet possibly incomplete clustering is possible, like it is done by the 4C algorithm from Böhm et al. (2004).

7 References

Amantonico, A., Urban, P. L. and Zenobi, R. (2010). Analytical techniques for single-cell metabolomics: state of the art and trends. *Analytical and bioanalytical chemistry, 398*(6), 2493–2504.

Anscombe, F. J. (1973). Graphs in statistical analysis. *The American Statistician, 27*(1), 17–21.

Baba, K., Shibata, R. and Sibuya, M. (2004). Partial correlation and conditional correlation as measures of conditional independence. *Australian & New Zealand Journal of Statistics, 46*(4), 657–664.

Bostock, M.. D3.js – data driven documents `http://d3js.org/`

Brewer, C. A. (1994). Color use guidelines for mapping and visualization. *Visualization in modern cartography, 2*, 123–148.

Böhm, C., Kailing, K., Kröger, P. and Zimek, A. (2004). Computing clusters of correlation connected objects. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* SIGMOD '04, pages 455–466. New York, NY, USA.

Chalew, S. A., Zadik, Z., Lozano, R. A., Nelson, J. C. and Kowarski, A. A. (1993). Plasma cortisol levels increase with age in obese subjects. *Obesity research, 1*(3), 199–202.

Cook, D. G., Shaper, A., Thelle, D. and Whitehead, T. (1986). Serum uric acid, serum glucose and diabetes: relationships in a population study.. *Postgraduate medical journal, 62*(733), 1001–1006.

Crespi, B. J. (2001). The evolution of social behavior in microorganisms. *Trends in ecology & evolution, 16*(4), 178–183.

de la Fuente, A., Bing, N., Hoeschele, I. and Mendes, P. (2004). Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics, 20*(18), 3565-3574.

Do, K. T. (2013). Metabolomic analysis of multiple bodyfluids in the qatar metabolomics study of diabetes. Master's thesis, München, Deutschland: Institute of Computational Biology, Helmholtz Zentrum.

Fan, X. and Thompson, B. (2001). Confidence intervals for effect sizes confidence intervals about score reliability coefficients, please: An epm guidelines editorial. *Educational and Psychological Measurement, 61*(4), 517–531.

Feigelman, J., Theis, F. J. and Marr, C. (2014). MCA: Multiresolution correlation analysis, a tool for subpopulation identification in single-cell gene expression data. *BMC bioinformatics*.

Filipe, P., Haigle, J., Freitas, J., Fernandes, A. and Mazière, J.-C. et al. (2002). Anti- and pro-oxidant effects of urate in copper-induced low-density lipoprotein oxidation. *European Journal of Biochemistry, 269*(22), 5474–5483.

Gleeson, M. (2008). Dosing and efficacy of glutamine supplementation in human exercise and sport training. *The Journal of nutrition, 138*(10), 2045S–2049S.

Hayashi, K., Lopes, S. M., Tang, F. and Surani, M. A. (2008). Dynamic equilibrium and heterogeneity of mouse pluripotent stem cells with distinct functional and epigenetic states. *Cell stem cell, 3*(4), 391–401.

Huang, S., Eichler, G., Bar-Yam, Y. and Ingber, D. E. (2005). Cell fates as high-dimensional attractor states of a complex gene regulatory network. *Physical review letters, 94*(12), 128701.

International Diabetes Federation. IDF diabetes atlas - sixth edition `http://www.idf.org/atlasmap/atlasmap`

King, G. L. and Loeken, M. R. (2004). Hyperglycemia-induced oxidative stress in diabetic complications. *Histochemistry and cell biology, 122*(4), 333–338.

Kopprasch, S., Richter, K., Leonhardt, W., Pietzsch, J. and Grler, J. (2000). Urate attenuates oxidation of native low-density lipoprotein by hypochlorite and the subsequent lipoprotein-induced respiratory burst activities of polymorphonuclear leukocytes. *Molecular and cellular biochemistry, 206*(1-2), 51–56.

Krumsiek, J., Suhre, K., Evans, A. M., Mitchell, M. W. and Mohney, R. P. et al. (2012). Mining the unknown: A systems approach to metabolite identification combining genetic and metabolic information. *PLoS Genet, 8*(10), e1003005.

Levy, K. J. and Narula, S. C. (1978). Testing hypotheses concerning partial correlations: Some methods and discussion. *International Statistical Review/Revue Internationale de Statistique*, pp. 215–218.

Lorenz, D. M., Jeng, A. and Deem, M. W. (2011). The emergence of modularity in biological systems. *Physics of life reviews, 8*(2), 129–160.

Meisinger, C., Strassburger, K., Heier, M., Thorand, B. and Baumeister, S. E. et al. (2010). Prevalence of undiagnosed diabetes and impaired glucose regulation in 35-59-year-old individuals in southern germany: the kora f4 study.. *Diabet Med, 27*(3), 360–362.

Mikami, T., Kita, K., Tomita, S., Qu, G.-J. and Tasaki, Y. et al. (2000). Is allantoin in serum and urine a useful indicator of exercise-induced oxidative stress in humans?. *Free radical research, 32*(3), 235–244.

Narsinh, K. H., Sun, N., Sanchez-Freire, V., Lee, A. S. and Almeida, P. et al. (2011). Single cell transcriptional profiling reveals heterogeneity of human induced pluripotent stem cells. *The Journal of clinical investigation, 121*(3), 1217.

Nurseitov, N., Paulson, M., Reynolds, R. and Izurieta, C. (2009). Comparison of json and xml data interchange formats: A case study.. *Caine, 9*, 157–162.

Otto, M., Jacob, McDonald, T., Mazovetskiy, G. and Rebert, C. et al.. Bootstrap `http://smus.com/canvas-vs-svg-performance/`

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London, 58*(347-352), 240–242.

Peters, P. D. A.. KORA – kooperative gesundheitsforschung in der region augsburg `http://www.helmholtz-muenchen.de/en/kora-en/about-kora/index.html`

Pevsner, J. (2013). *Bioinformatics and Functional Genomics*. Wiley.

Pimentel, V. and Nickerson, B. G. (2012). Communicating and displaying real-time data with websocket. *Internet Computing, IEEE, 16*(4), 45–53.

Powell, S. G., Baker, K. R. and Lawson, B. (2008). A critical review of the literature on spreadsheet errors. *Decision Support Systems, 46*(1), 128–138.

Reik, W. (2007). Stability and flexibility of epigenetic gene regulation in mammalian development. *Nature, 447*(7143), 425–432.

Roeder, I. and Radtke, F. (2009). Stem cell biology meets systems biology. *Development, 136*(21), 3525–3530.

Rousselet, G. A. and Pernet, C. R. (2012). Improving standards in brain-behaviour correlation analyses. *Frontiers in Human Neuroscience, 6*(119).

Schäfer, J. and Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology, The Berkeley Electronic Press, 4*(1).

Seber, G. A. and Lee, A. J. (2012). *Linear regression analysis*, volume 936. John Wiley & Sons.

Shani, U. (1980). Filling regions in binary raster images: A graph-theoretic approach. In *ACM SIGGRAPH Computer Graphics*, pages 321–327.ACM

Smus, B.. Performance of canvas versus svg `http://smus.com/canvas-vs-svg-performance/`

Spearman, C. (1904). The proof and measurement of association between two things. *American journal of Psychology, 15*(1), 72–101.

Stiburkova, B. and Bleyer, A. J. (2012). Changes in serum urate and urate excretion with age. *Advances in chronic kidney disease, 19*(6), 372–376.

Strimmer, K. (2008). A unified approach to false discovery rate estimation. *BMC Bioinformatics, 9*(1), 303.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E. and Lee, C. et al. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature methods, 6*(5), 377–382.

Tavendo GmbH. Autobahn|python `http://autobahn.ws/python/`

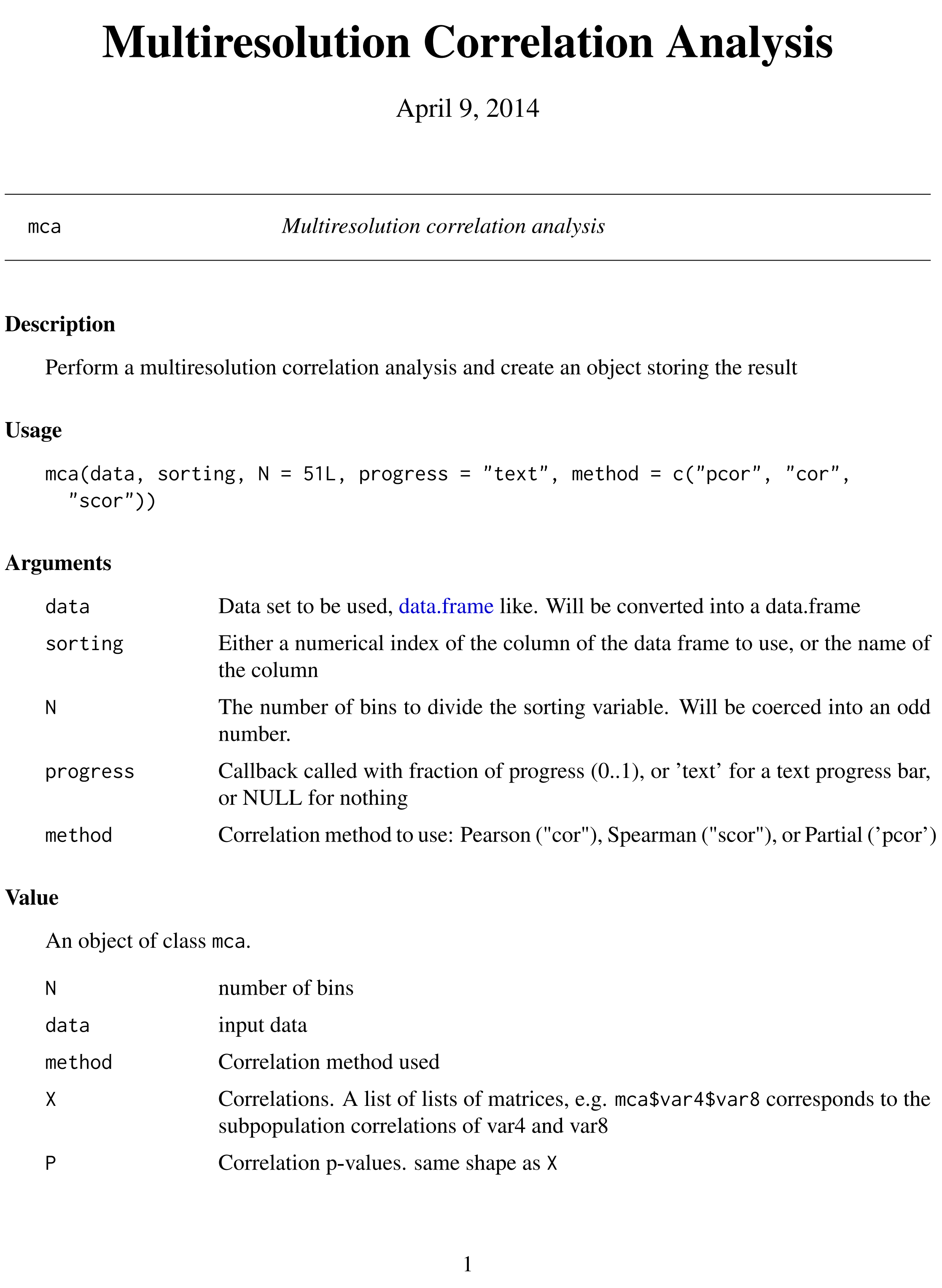
Topaloglou, T., Davidson, S. B., Jagadish, H., Markowitz, V. M. and Steeg, E. W. et al. (2004). Biological data management: Research, practice and opportunities. In *VLDB*, pages 1233–1236.

Trott, J., Hayashi, K., Surani, A., Babu, M. M. and Martinez-Arias, A. (2012). Dissecting ensemble networks in es cell populations reveals micro-heterogeneity underlying pluripotency. *Mol. BioSyst., 8*, 744-752.

Vemuri, G. N. and Aristidou, A. A. (2005). Metabolic engineering in the-omics era: elucidating and modulating regulatory networks. *Microbiology and Molecular Biology Reviews, 69*(2), 197–216.

Žežula, I. (2009). On multivariate gaussian copulas. *Journal of Statistical Planning and Inference, 139*(11), 3942–3946.

8 Appendix



Multiresolution Correlation Analysis

April 9, 2014

`mca` *Multiresolution correlation analysis*

Description

Perform a multiresolution correlation analysis and create an object storing the result

Usage

```
mca(data, sorting, N = 51L, progress = "text", method = c("pcor", "cor",
  "scor"))
```

Arguments

`data` Data set to be used, data.frame like. Will be converted into a data.frame

`sorting` Either a numerical index of the column of the data frame to use, or the name of the column

`N` The number of bins to divide the sorting variable. Will be coerced into an odd number.

`progress` Callback called with fraction of progress (0..1), or 'text' for a text progress bar, or NULL for nothing

`method` Correlation method to use: Pearson ("cor"), Spearman ("scor"), or Partial ('pcor')

Value

An object of class `mca`.

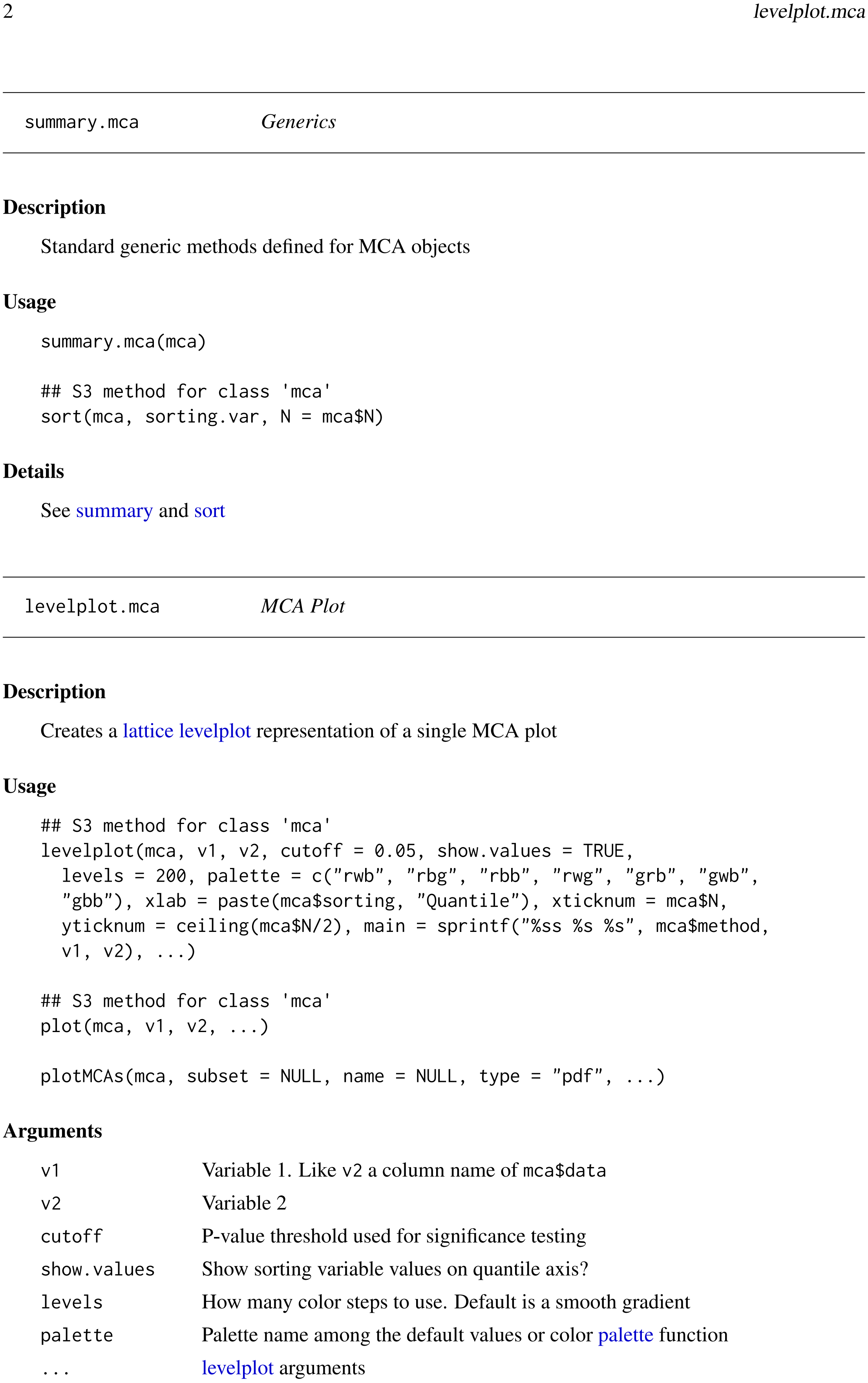
`N` number of bins

`data` input data

`method` Correlation method used

`X` Correlations. A list of lists of matrices, e.g. `mca$var4$var8` corresponds to the subpopulation correlations of var4 and var8

`P` Correlation p-values. same shape as `X`



`summary.mca` *Generics*

Description

Standard generic methods defined for MCA objects

Usage

```
summary.mca(mca)

## S3 method for class 'mca'
sort(mca, sorting.var, N = mca$N)
```

Details

See summary and sort

`levelplot.mca` *MCA Plot*

Description

Creates a lattice levelplot representation of a single MCA plot

Usage

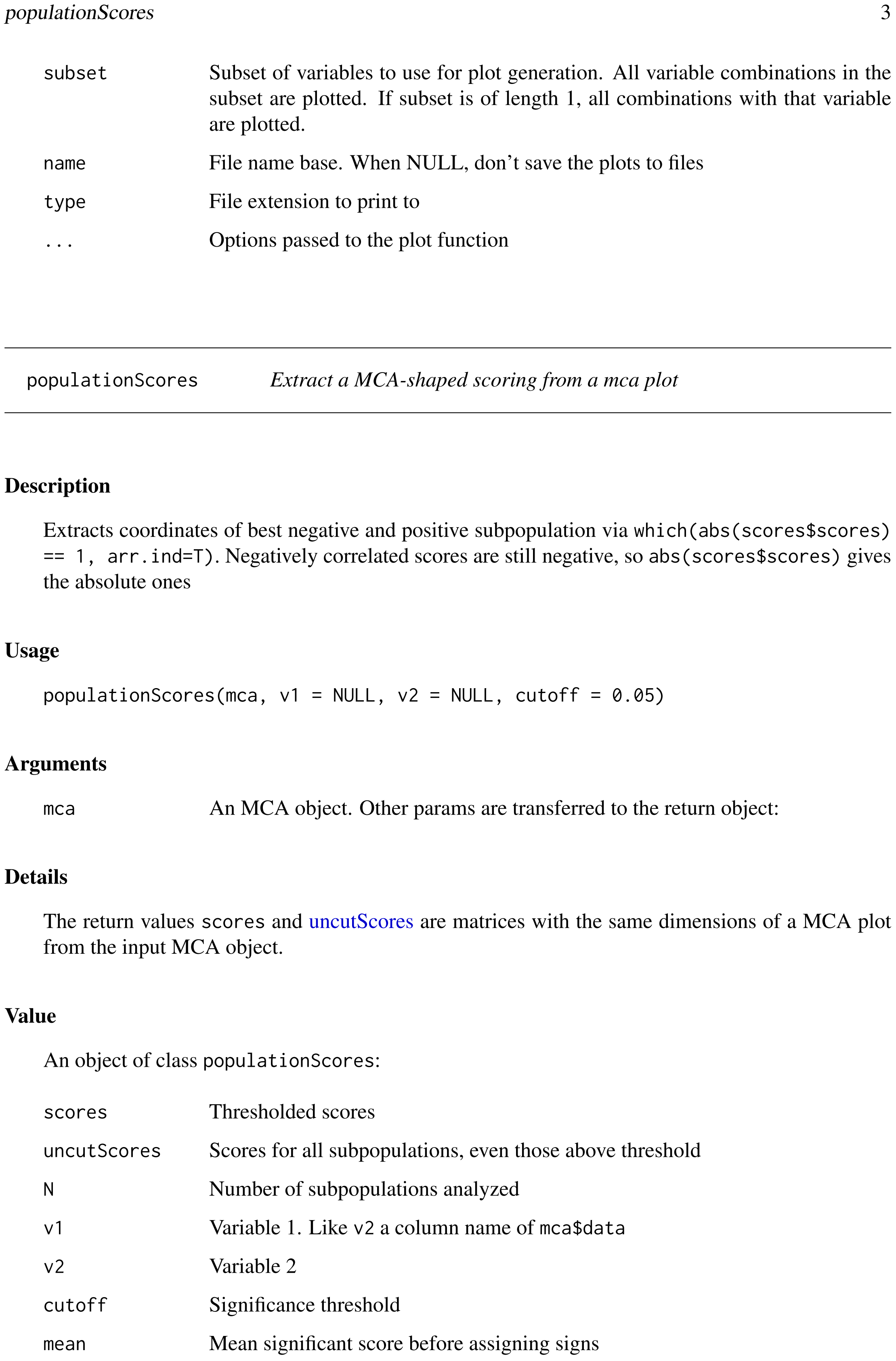
```
## S3 method for class 'mca'
levelplot(mca, v1, v2, cutoff = 0.05, show.values = TRUE,
  levels = 200, palette = c("rwb", "rbg", "rbb", "rwg", "grb", "gwb",
  "gbb"), xlab = paste(mca$sorting, "Quantile"), xticknum = mca$N,
  yticknum = ceiling(mca$N/2), main = sprintf("%ss %s %s", mca$method,
  v1, v2), ...)

## S3 method for class 'mca'
plot(mca, v1, v2, ...)

plotMCAs(mca, subset = NULL, name = NULL, type = "pdf", ...)
```

Arguments

`v1`	Variable 1. Like `v2` a column name of `mca$data`
`v2`	Variable 2
`cutoff`	P-value threshold used for significance testing
`show.values`	Show sorting variable values on quantile axis?
`levels`	How many color steps to use. Default is a smooth gradient
`palette`	Palette name among the default values or color palette function
`...`	levelplot arguments



`subset`	Subset of variables to use for plot generation. All variable combinations in the subset are plotted. If subset is of length 1, all combinations with that variable are plotted.
`name`	File name base. When NULL, don't save the plots to files
`type`	File extension to print to
`...`	Options passed to the plot function

`populationScores` *Extract a MCA-shaped scoring from a mca plot*

Description

Extracts coordinates of best negative and positive subpopulation via `which(abs(scores$scores) == 1, arr.ind=T)`. Negatively correlated scores are still negative, so `abs(scores$scores)` gives the absolute ones

Usage

```
populationScores(mca, v1 = NULL, v2 = NULL, cutoff = 0.05)
```

Arguments

`mca`	An MCA object. Other params are transferred to the return object:

Details

The return values `scores` and uncutScores are matrices with the same dimensions of a MCA plot from the input MCA object.

Value

An object of class `populationScores`:

`scores`	Thresholded scores
`uncutScores`	Scores for all subpopulations, even those above threshold
`N`	Number of subpopulations analyzed
`v1`	Variable 1. Like `v2` a column name of `mca$data`
`v2`	Variable 2
`cutoff`	Significance threshold
`mean`	Mean significant score before assigning signs

`levelplot.populationScores`

Population score plot

Description

Plots population scores and their local maxima, as found by extractPeaks.

Usage

```
## S3 method for class 'populationScores'
levelplot(scores, levels = 200L,
  draw.cutoff = TRUE, draw.uncut = TRUE, big = NULL, palette = "gwb",
  main = paste("Scores", scores$v1, scores$v2), ...)

plot.populationScores(scores, ...)

plotAllPopulationScores(mca, name = NULL, type = "pdf", ...,
  score.args = list())
```

Arguments

`scores`	An object of class populationScores
`levels`	How many color steps to use. Default is a smooth gradient
`draw.cutoff`	Draw line where cutoff is located
`draw.uncut`	Draw all scores instead of just the ones remaining after thresholding
`big`	See extractPeaks
`palette`	Palette name among the default values or color palette function. For allowed palette anmes, see levelplot.mca
`...`	Options passed to levelplot
`name`	If present, write files with this base name
`type`	File extension to print to
`score.args`	List of arguments to pass to populationScores

Details

The default is to plot all scores and draw a line where the threshold lies.

`plotAllPopulationScores` generates and plots the scores for all variable combinations of `mca`.

quantseq *Used to create sequences of quantile steps, mostly until slightly above .5*

Description

Used to create sequences of quantile steps, mostly until slightly above .5

Usage

```
quantseq(n, end = 0.5)
```

dataIndex *Subpopulation extraction*

Description

Logical array used to extract a subpopulation from a MCA object

Usage

```
dataIndex(mca, from, to)
```

Arguments

`mca`	MCA object
`from`	Sorting variable quantile at which the subpopulation starts
`to`	Quantile at where it stops

extractPeaks *Extract local score maxima of thresholded scoring regions*

Description

Extracts multiple score peaks by segmenting the score-image and finding the maximum in each of the biggest segments

Usage

```
extractPeaks(scores, big = NULL, fuzzy = 0.02)
```

Arguments

`scores` `populationScores` object or score matrix

`big` Function applied to the list of region sizes to determine which regions are big enough to be considered for a maximum. Takes a vector of integers and returns a logical vector of the same size. Defaults to `function(sizes) sizes > sqrt(max(sizes))`

`fuzzy` How far away from a subpopulation triangle (the whole plot is seen as width 1, height .5) is still considered “inside” a subpopulation. Used to prune subpopulations slightly outside of bigger ones which got split into multiple segments (separated by insignificant areas).

Details

Only “big” sized regions are considered, because they contain more stable subpopulations that re robust against adding and removing a few bins.

`centered.seq` *0-Centered sequence*

Description

Gives a sequence along the range of values in x, with as many below as above 0. Cuts the divergent palette accordingly

Usage

```
centered.seq(x, levels, palette)
```

Arguments

`x` Vector of values

`levels` Number of levels. Forced to be odd

`palette` Color palette

`cor.with.p` *Pairwise correlation*

Description

Estimates correlations for all variable pairs in x

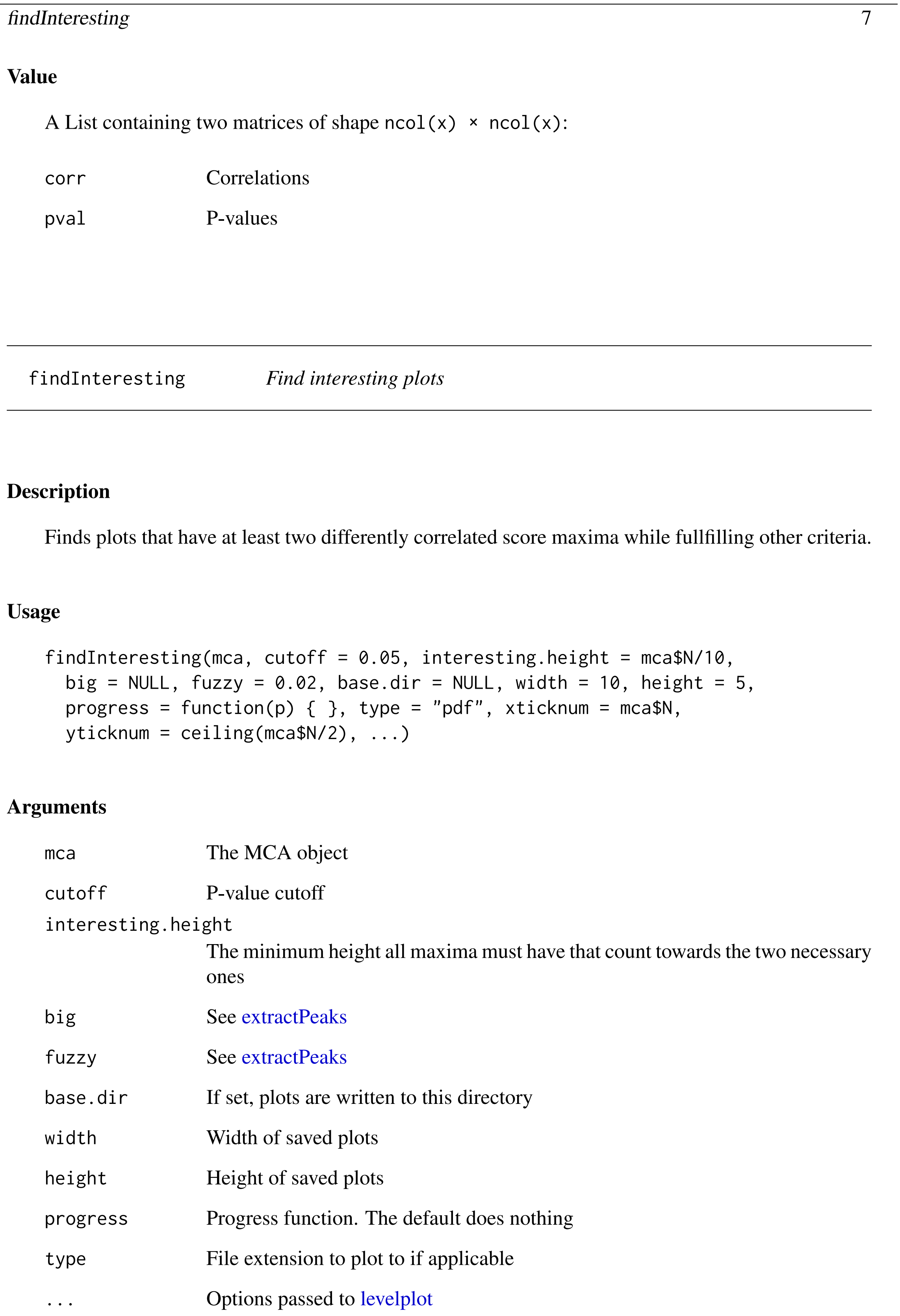
Usage

```
cor.with.p(x, method = c("pearson", "kendall", "spearman"))
```

Arguments

`x` Matrix or data.frame

`method` Correlation method



findInteresting 7

Value

A List containing two matrices of shape `ncol(x) × ncol(x)`:

`corr`	Correlations
`pval`	P-values

`findInteresting` *Find interesting plots*

Description

Finds plots that have at least two differently correlated score maxima while fullfilling other criteria.

Usage

```
findInteresting(mca, cutoff = 0.05, interesting.height = mca$N/10,
  big = NULL, fuzzy = 0.02, base.dir = NULL, width = 10, height = 5,
  progress = function(p) { }, type = "pdf", xticknum = mca$N,
  yticknum = ceiling(mca$N/2), ...)
```

Arguments

`mca`	The MCA object
`cutoff`	P-value cutoff
`interesting.height`	The minimum height all maxima must have that count towards the two necessary ones
`big`	See extractPeaks
`fuzzy`	See extractPeaks
`base.dir`	If set, plots are written to this directory
`width`	Width of saved plots
`height`	Height of saved plots
`progress`	Progress function. The default does nothing
`type`	File extension to plot to if applicable
`...`	Options passed to levelplot

`shrunk.pvals` *P-values for shrunk partial correlations*

Description

Calculates p-values for partial correlations estimated using pcor.shrink

Usage

```
shrunk.pvals(pcors)
```

Arguments

`pcors`	Partial correlation matrix generated using ggm.estimate.pcor or pcor.shrink

Details

A version of GeneNet's ggm.test.edges that is silent, much faster, and outputs a matrix instead of a sparse list of top edges.

Value

p-values for input partial correlations

`subpopulationPlot` *Subpopulation plot*

Description

Plots a scatter plot for a number of subpopulations specified using a table of subpopulation coordinates.

Usage

```
subpopulationPlot(mca, maxima, v1, v2, colors = c("#0000FF", "#FF0000"), ...)
```

Arguments

`mca`	MCA object
`maxima`	data.frame with $height and $center attributes, as generated by extractPeaks.
`v1`	Variable 1. Like `v2` a column name of `mca$data`
`v2`	Variable 2
`colors`	Colors to override default scatter group colors