Statistical integration of diverse omics data

Bouhaddani, S. el

Author: Bouhaddani, S.
Title: Statistical integration of diverse omics data
Issue Date: 2020-06-02

3

Integrating omics datasets with the OmicsPLS package

3.1 Abstract

With the exponential growth in available biomedical data, there is a need for data integration methods that can extract information about relationships between the data sets. However, these data sets might have very different characteristics. For interpretable results, data-specific variation needs to be quantified. For this task, Two-way Orthogonal Partial Least Squares (O2PLS) has been proposed. To facilitate application and development of the methodology, free and open-source software is required. However, this is not the case with O2PLS.

We introduce **OmicsPLS**, an open-source implementation of the O2PLS method in R. It can handle both low- and high-dimensional datasets efficiently. Generic methods for inspecting and visualizing results are implemented. Both a standard and faster alternative cross-validation methods are available to determine the number of components. A simulation study shows good performance of OmicsPLS compared to alternatives, in terms of accuracy and CPU runtime. We demonstrate OmicsPLS by integrating genetic and glycomic data.

We propose the OmicsPLS R package: a free and open-source implementation of O2PLS for statistical data integration. OmicsPLS is available at `cran.r-project.org/package=OmicsPLS` and can be installed in R via `install.packages("OmicsPLS")`.

3.2 Introduction

With rapid advances in technology, several levels of biological variation can be measured. Consequently, multiple omics data sets are available on the same set of subjects. For a better understanding of the underlying biological systems, these data should be analyzed simultaneously [12].

Several data integration methods have been developed that estimate joint parts while ignoring some of the data-specific characteristics. For example, Partial Least Squares (PLS) [21] maximizes the covariance to calculate joint principal components. Canonical Correlation Analysis (CCA) [3] considers correlation rather than covariance. Several other methods perform analysis on a concatenated version of the data sets, such as Simultaneous Component Analysis (SCA) [17]. For many data integration methods, open source software packages are available [10]. In particular, the mixOmics R package implements several variants of PCA, PLS and CCA [13].

Omics data sets might be heterogeneous in that they typically differ in data-specific characteristics, such as size, scale, distribution and experimental error. This hampers the estimation of joint parts between these data. For correct interpretation of data integration results, data-specific variation should be modeled [16]. This variation captures information specific to each data set. Furthermore, it can distort interpretation of the estimated joint part [14]. Therefore, we consider approaches that estimate both joint and specific components. Such approaches include Two-Way Orthogonal PLS (O2PLS) [14], JIVE [9] and DISCO-SCA [18]. O2PLS considers two data sets and decomposes both in a joint, specific and residual part. The joint parts are calculated by maximizing the covariation between the two data sets, while correcting for data-specific variation. JIVE uses iterative PCA on the concatenation of multiple datasets to alternately find joint and data-specific parts. DISCO-SCA performs SCA and rotates the solution to obtain joint and specific components for each data set.

In the JIVE and DISCO-SCA approach, the joint and specific components are constrained to be orthogonal to each other. Moreover, they assume that the data sets share exactly the same joint latent variables. O2PLS only imposes orthogonality of the components within each part and assumes correlated joint latent variables for each data set. Therefore, we expect a better performance of O2PLS in complex situations.

O2PLS is implemented within the software package SIMCA [15], which is closed-source and commercial. Unavailability of source code hampers developing and extending the methodology. No free and open source alternative implementing O2PLS is available to the best of our knowledge. Therefore, we propose OmicsPLS, a free and open-source R software package to decompose two datasets into joint and specific parts. With regard to the other methods, DISCO-SCA [5] is available only from the commercial computing environment MATLAB, whereas JIVE is freely available in the r.jive package [11]. Therefore, we compare OmicsPLS to r.jive.

Our aim is to provide easy access to both the method and visualization tools and to facilitate the development of more advanced methodology. The rest of the article is organized as follows. First, we discuss the implementation of OmicsPLS in detail. Second, the OmicsPLS package is illustrated using genetic and glycan data from a Croatian population cohort. We also apply JIVE to these data. Motivated by the data analysis, we conduct a simulation study to compare OmicsPLS to r.jive in terms of estimation accuracy, execution time and robustness against the presence of data-specific characteristics. Finally, we discuss future extensions of OmicsPLS.

3.3 Implementation

3.3.1 O2PLS Model

Let the observed data be collected in a matrix $X = [x_1, \dots, x_p]$ $(N \times p)$ and a matrix $Y = [y_1, \dots, y_q]$ $(N \times q)$. Here, N denotes the number of subjects, and p and q denote the number of variables in X and Y, respectively. The O2PLS method decomposes X and Y in two joint, specific and residual parts. The dimension of the joint part is given by n, the dimension of each specific part is given by n_X and n_Y, respectively. The joint parts consist of matrices T, U (both $N \times n$), W $(p \times n)$ and C $(q \times n)$. The matrices T and U are referred to as joint scores or joint latent components, and the matrices W and C are referred to as joint loadings or joint principal components. These joint parts represent the statistical overlap between X and Y. The specific parts consist of matrices $T_{Y\perp}$ $(N \times n_X)$, $U_{X\perp}$ $(N \times n_Y)$, $P_{Y\perp}$ $(p \times n_X)$ and $P_{X\perp}$ $(q \times n_Y)$. These matrices are referred to as specific scores and loadings, respectively. The residual parts are denoted by E $(N \times p)$ and F $(N \times q)$. Then, the O2PLS decomposition is

$$\begin{array}{ccccccccc} X & = & TW^{\mathrm{T}} & + & T_{Y\perp}P_{Y\perp}^{\mathrm{T}} & + & E & , \\ \underbrace{Y}_{Data} & = & \underbrace{UC^{\mathrm{T}}}_{Joint} & + & \underbrace{U_{X\perp}P_{X\perp}^{\mathrm{T}}}_{Specific} & + & \underbrace{F}_{Residuals} & . \end{array}$$

Each row of X and Y contains measurements on *the same* subject. Throughout the paper, it is assumed that the columns of X and Y are centered around zero. The relationship between T and U is given by the linear model $U = TB_T + H$ or $T = UB_U + H'$. Here, B_T and B_U are square matrices of size n, representing regression coefficients for the two models. The particular choice of the model does not affect the estimates, as the O2PLS algorithm is symmetric in X and Y.

Note that, in PLS, only a joint and a residual part is considered for each data set. Any data-specific variation is absorbed by these two parts. This makes interpretation of PLS results more difficult, as the estimated loadings may be biased and the correlation between the joint scores typically seem weaker. O2PLS restricts the joint loadings W and C and the specific scores $T_{Y\perp}$ and $U_{X\perp}$ to have orthonormal columns. JIVE and DISCO-SCA additionally restrict the columns of the matrices $[WP_{Y\perp}]$ and $[CP_{X\perp}]$ to be orthonormal. Furthermore, both methods assume that $U = T$, while O2PLS only assumes a linear relation between U and T.

The O2PLS algorithm for estimating the O2PLS components is provided in [14]. Briefly, singular vectors of the covariance matrix $X^{\mathrm{T}}Y$ are calculated. From these vectors, loadings and scores containing both joint and specific variation are estimated. Then, specific variation is estimated using SVD and subtracted from the data. Finally, using the corrected data, the joint parts are re-estimated.

Interpretation. Within each part, the components have a similar interpretation as PCA. In particular, the loading value w_{jk} indicates the importance of the *variable* x_j for component k. If w_{jk} and $w_{j'k}$ have the same sign, the corresponding variables x_j and $x_{j'}$ are positively correlated within component k. The same interpretation holds for the other parts. The scores can be used to define similarity between *subjects* within each component: for example, if $t_{ik} \approx t_{i'k}$, then subjects i and i' are similar in component k. Between the joint parts, in the k'th joint component, the loading values w_{jk} and $c_{j'k}$ indicate correlation between x_j and $y_{j'}$. High positive or negative loading values indicate high positive or negative correlation in this component between these variables, respectively. As a consequence, the joint loading values w_k and c_k can be sorted to prioritize variables in X and Y based on high covariation.

3.3.2 Implementation

The functions in OmicsPLS can be organized as follows

- Cross-validating: Functions to determine the number of O2PLS components.
- Fitting: Functions to fit the O2PLS model.
- Summarizing & visualizing: Functions to summarize and visualize the results.

Cross-validating. Cross-validation is a well-known technique to choose tuning parameters of a model, while limiting the risk to overfit. All samples are partitioned in k blocks (denoted as folds), and the model is fitted on $k-1$ folds. The left out fold is used to evaluate the model fit. For O2PLS, an approach to determine the number of components is to maximize the prediction error over a three-dimensional grid of possible integers and select the triple (n, n_X, n_Y) that minimizes this error. As O2PLS is symmetric in X and Y, the sum of the two prediction errors $||Y - \hat{Y}||^2 + ||X - \hat{X}||^2$ is taken as error measure. Here, $||A||^2 := \sum_{ij} a_{ij}^2$. This approach is implemented in the `crossval_o2m` function:

```
crossval_o2m(X, Y, a, ax, ay, nr_folds)
```

Here, `a`, `ax` and `ay` are vectors of integers to consider for the number of components n, n_X and n_Y. The vector `a` must have positive elements, while both `ax` and `ay` may contain zeros. The number of folds is specified by `nr_folds` and should be between two and N. The `crossval_o2m` function returns a three-dimensional array with the prediction errors.

Cross-validation over a three-dimensional grid can be computationally intensive, especially with many grid points. For this reason, we have proposed an alternative cross-validation procedure [2]. The rationale behind this approach lies in the interpretation of the specific parts: specific variation in the data will affect the joint scores, thereby reducing the covariance between T and U. Correcting for specific variation will increase this covariance. On the other hand, overcorrecting will again reduce the covariance between the joint scores. Candidates for n_X and n_Y, given n, are those integers for which the covariance of the joint scores are maximized. This approach is called by:

```
crossval_o2m_adjR2(X, Y, a, ax, ay, nr_folds)
```

It performs the cross-validation over a one-dimensional grid `a`, while maximizing the covariance between the joint scores T and U over a two-dimensional grid given by `ax` and `ay`. The last maximization does not involve cross-validation. Consequently, the looping over `nr_folds` folds is omitted in two dimensions. This can drastically reduce computation time, while often yielding similar minimizers to those obtained with the full cross-validation approach. The output is a matrix containing the prediction errors and the number of components (n, n_X, n_Y).

Note that these two cross-validation strategies can be combined: The alternative cross-validation is used to find candidate minimizers of the prediction error. Based on these minimizers, a three-dimensional grid is constructed on which the full cross-validation is performed. Both cross-validation implementations support parallel computation.

Fitting. In its simplest form, the function call for fitting the O2PLS model is

```
o2m(X, Y, n, nx, ny)
```

The input parameters are the two data matrices X and Y, and the number of components in the joint, X-specific and Y-specific part. The output is a list containing scores and loadings in the notation of [14], as well as proportions of explained variance and residual matrices; these proportions are defined below. The user can choose a 'stripped' output, by adding `stripped=TRUE` as an argument, to discard the residual matrices (and reduce memory usage).

By default, a Singular Value Decomposition (SVD) of the covariance matrix between X and Y is used to calculate joint and specific components. If both X and Y are high-dimensional, the covariance matrix $X^{\mathrm{T}}Y$ will use a high amount of memory. Therefore, an alternative algorithm is implemented in the OmicsPLS package, named NIPALS [21]. The NIPALS algorithm is an iterative algorithm that avoids construction and storage of the covariance matrix. Moreover, the NIPALS-based joint components are numerically equal to the SVD-based PLS components (up to sign) if the number of NIPALS iterations is large enough. In the case that p or q is not too large, the NIPALS approach can be slower than the SVD approach. Therefore, a check on data dimensionality is performed to determine the proper approach. The threshold is by default at $p = q = 3000$ and can be adjusted.

Summarizing & visualizing A summary of the modeled variation is given by `summary(object)`. Here, `object` contains the O2PLS fit as produced by the `o2m` call. The output includes proportions of:

- variation in X and Y explained by the joint, specific and residual parts, e.g., $||TW^{\mathrm{T}}||/||X||$.
- variation in U and T that is predictable by T resp. U, e.g., $||TB_T||^2/||U||^2$.

Note that the proportion of predictable variation in Y by X is then $||TB_T||^2/||U||^2 \times ||UC^{\mathrm{T}}||^2/||Y||^2 = ||TB_T||^2/||Y||^2$.

The OmicsPLS package provides a flexible framework to plot loadings in each component. As this framework is built on the ggplot2 package, several plotting layers can be added to enhance visualization and aid interpretation of the results. The command for constructing a plot is

```
plot(x, loading_name).
```

Here `x` is the O2PLS fit and the only required object. The parameter `loading_name` represents which of the four parts (X-joint, Y-joint, X-specific or Y-specific) should be plotted. The plot command calls `geom_text` from the ggplot2 package. Its documentation contains information about editing, for example, text color, transparency and size. These attributes can be changed within the OmicsPLS `plot` function.

Workflow A workflow for OmicsPLS analysis is provided in Figure 3.1. The steps in the workflow are based on the genetic and glycomic data analysis showed the next section.

3.4 Results

3.4.1 Analysis of genetic and glycomic data

We consider $p = 333858$ genotyped Single Nucleotide Polymorphisms (SNPs) and $q = 20$ quantified IgG1 glycan (glycopeptide) abundances, measured with nano-LC-ESI-MS, for $N = 885$ participants in the CROATIA_Korcula cohort [6]. Both data sets contain highly correlated measurements and are heterogeneous (as they differ in scale, distribution and measurement error).

Our aim is to investigate how IgG1 glycans relate to genetic variation by determining the statistical overlap between IgG1 glycan data and genetic data, as in Eq. (3.3.1). To this end, we use the OmicsPLS package to obtain estimates of the amount of joint variation and estimate the contribution of the genetic and IgG1 glycan measurements to this joint variation.

The SNPs were summarized by taking, for each gene (in the UCSC hg18 database), all SNPs within 50 kilobases from that gene and applying Principal Components Analysis. For each gene, the set of corresponding SNPs were replaced by as many principal components as needed to explain at least 80% of this set of SNPs. This provided a new data set with one or several variables, which we denote as Genetic PCs, per gene. This 'Genetic PCs' data set contains 37819 variables and is referred to as X. The glycan measurements were log-transformed, batch-corrected [4] and quantile-normalized [1]. The resulting data matrix is referred to as Y.

Scree plots of XX^{T}, $Y^{\mathrm{T}}Y$ and $X^{\mathrm{T}}Y$ are shown in Figure 3.2. By identifying an elbow in these scree plots, the number of joint and specific components are determined. Based on the plots, 5 joint and 5 genetic-specific components were retained. Note that no glycan-specific parts were detected. The O2PLS fit took around 5 seconds.

Regarding the five IgG1 glycan joint components, they account for 96% of the total IgG1 glycan variation. The amount of IgG1 variation that can be predicted with the Genetic PCs is 70%. The loading values of each IgG1 glycan variable are depicted in Figure 3.3. The first joint component is proportional to the 'average' IgG1 glycan, as all glycans get approximately the same loading value. The second joint component distinguishes fucosylated (negative loading values) and non-fucosylated (mostly positive loading values) IgG1 glycans. This component is referred to as the 'fucosylation' component. The third joint component involves especially non-galactosylated (negative loading values) and di-galactosylated (positive loading values) IgG1 glycans, while mono-galactosylated glycans have estimated loading values around zero. This component is referred to as the 'galactosylation' component. In the fourth joint component, G1NS and G2NS glycans have high loading values. The fifth joint component distinguishes, apart from G1NS and G2NS, glycans for which bisecting GlcNAc is present (negative loading values) or absent (positive loading values).

Regarding the five joint components in the Genetic PCs data set, they account for 0.8% of the total variation. For the specific parts, this percentage is 1.9%. The top five genes in each Genetic PCs joint component are shown in Table 3.1. In the first Genetic PCs joint component, the gene with the highest loading value is *DNAJC10*. The corresponding protein is involved in recognizing and degrading misfolded glycoproteins. This first joint component corresponds to the 'average' glycan pattern in the first glycan joint component. The top gene in the second joint component, which corresponds to the 'fucosylation' component, is *FUT8*. It encodes a fucosyltransferase enzyme that catalyzes the transfer of fucose to a glycopeptide. In the third joint component, which corresponds to the 'galactosylation' component, the gene *AKAP9* has second highest loading value. It encodes an A-kinase anchor protein, which is involved in maintaining the integrity of the Golgi apparatus. Note that in the Golgi apparatus, glycosylation (in particular galactosylation) takes place. In the fourth and fifth component, no directly relevant genes were found. More research is needed to further elucidate these relationships.

For comparison purposes, r.jive was also applied to the data. However, the algorithm did not converge after 500 iterations (and 3000 seconds). We will investigate possible reasons in the simulation study.

3.4.2 Simulation study

A simulation study is conducted to compare r.jive and OmicsPLS in terms of accuracy and speed. To gain insight into the robustness of r.jive, possible reasons for the lack of convergence of r.jive are investigated. The simulated data follow a model that satisfies the assumptions of both O2PLS and JIVE:

$$\begin{aligned} X &= TW^{\mathrm{T}} + T_{Y\perp}P_{Y\perp} + E, \\ Y &= UC^{\mathrm{T}} + U_{X\perp}P_{X\perp} + F, \end{aligned} \tag{3.1}$$

where $U = T$. Note that in the O2PLS formulation, $B_T = I_r$ and $H = 0$. In the first scenario, we take $N = 500$, $p = q = 100$, $n = 2$, $n_X = 3$ and $n_Y = 1$. In the second scenario, we consider $p = q = 10^4$. Elements of W, C, $P_{Y\perp}$ and $P_{X\perp}$ are drawn independently from a standard normal distribution. The JIVE constraints are applied by orthogonalizing each column in both joint and specific parts with respect to each other. Elements of T, $T_{Y\perp}$ and $U_{X\perp}$ are drawn independently from a standard normal distribution. Noise, represented by E and F, is added to X and Y to account for about 10% of the total variation. For both r.jive and OmicsPLS, loading matrices are extracted. To evaluate estimation accuracy, the absolute value of the inner product between corresponding columns are calculated. Here, higher values represent lower estimation errors. For each scenario, we generated 1000 replicates.

To investigate the lack of convergence of r.jive in the data analysis, two additional scenarios are considered. In the first additional scenario, elements in U have a standard deviation of 10, i.e., $U = 10T$. In the second additional scenario, elements in the specific parts will be normally distributed with a standard deviation of 10. The dimensions and sample size are taken as above. Note that both scenarios represent an 'imbalance' in the amount of variation per part. Here, r.jive is considered converged if it needs less than 500 steps. In these additional scenarios, we generated 100 replicates.

In Table 3.2, median inner product values, together with Median Absolute Deviations (MAD) are shown for $p = q = 100$. It can be seen that for balanced scenario settings, OmicsPLS performs as good as r.jive in terms of median inner product. The results for $p = q = 10^4$ were very similar to these results (not shown).

In Table 3.3, elapsed time and convergence ratios are shown. OmicsPLS runs about 3500 times faster in the first scenario ($p = q = 100$) and 7 times faster in the second ($p = q = 10^4$) scenario. In both additional scenarios in which there is an imbalance in the amount of variation between the joint and specific parts, r.jive failed to converge in the majority of runs. In case $U = 10T$, r.jive did not converge in more than 90% of the runs. In case the specific parts contain more variation, r.jive failed to converge in 74 and 63 out of 100 runs, for $p = q = 100$ and $p = q = 10^4$, respectively.

3.5 Discussion

In this article, we introduced the OmicsPLS package for integration of two (omics) data sets. We evaluated its performance with a simulation study and demonstrated it using genetic and IgG1 glycomic data. Regarding the data analysis, the proportion of joint variation in the Genetic PCs data set was 0.8%. This proportion is expected to be small since it is not likely that a large fraction of genetic variation (in particular

SNPs) is related to IgG1 glycosylation. In the joint components, several genes were found that might play a role in the genetic regulation of IgG1 glycosylation. Some of these genes are known to be directly involved (e.g., *FUT8*), while others (*DNAJC10* and *AKAP9*) are localized to cell compartments where the majority of glycosylation takes place (the endoplasmic reticulum and Golgi Apparatus). However, much is still unknown about the genetic regulation of (IgG) glycosylation.

Additionally, we considered JIVE for this type of data, but without success: the algorithm did not converge. A potential cause for this lack of convergence is the different data-specific characteristics of the two data sets. In particular, the dimensionality and amount of variation differ. Therefore, the JIVE assumption $U = T$ might not be reasonable. This is confirmed by our simulation: the r.jive algorithm is not robust against an 'imbalance' in the amount of variation between the two joint parts, or between the joint and specific parts. In particular, when $U = 10T$, r.jive did not converge in more than 90% of the replicates. This suggests that r.jive might be inappropriate for analyzing heterogeneous data sets (in which data-specific characteristics differ across data sets). Note that in DISCO-SCA the same assumption ($U = T$) is made, therefore we expect a suboptimal performance of this method as well when analyzing heterogeneous data.

As part of a future update of the OmicsPLS software package, we intend to deal with missing data. To impute missing values and simultaneously estimate O2PLS components, the OmicsPLS algorithm can be extended [21]. The imputation step can also be performed prior to analysis. For multiple omics data, Ensemble Regression Imputation [8] and Multiple Factor Analysis imputation [19] have been proposed. Note that, as with all imputation methods, uncertainty due to missing data should be assessed and presented to the user. A probabilistic framework for O2PLS would facilitate imputation and simultaneously addresses additional uncertainty due to missing data.

An important extension of OmicsPLS involves obtaining standard errors for the estimates. To this end, bootstrap approaches, similar to those found in PLS literature, can be applied [20]. A drawback of using resampling methods is the computational burden, especially with high-dimensional data sets. To avoid such procedures, a probabilistic framework for O2PLS can be used to directly calculate asymptotic standard errors.

Interpretability of the OmicsPLS output can be increased by extending the algorithm to produce sparse estimates. This extension can be implemented by considering Sparse PLS [7] or by considering a probabilistic framework for O2PLS and obtaining penalized maximum likelihood estimates.

We are currently investigating the possibilities of Probabilistic O2PLS for data integration, which facilitates multiple imputation and statistical inference, such as calculation of asymptotic standard errors. By penalizing the likelihood, sparse estimates can be obtained.

As OmicsPLS is open-source, it is straightforward to extend the current implementation to handle more complex situations. For example, in the GitHub repository, several 'branches' can be initialized in which new functionalities can be developed.

3.6 Conclusion

We propose OmicsPLS, an open-source and freely available R package for robust integration of heterogeneous data with O2PLS. It includes functions to determine the number of components, fit, and inspect results. For high-dimensional data, a memory-efficient implementation is used.

3.7 Tables

Table 3.1: **Top 5 genes and loading values of the Genetic-Glycan joint principal components.** The results are displayed per component. Only the first three components are shown.

Component 1:‘average’ glycan		**Component 2**:‘fucosylation’	
Gene Symbol	**Loading value**	**Gene Symbol**	**Loading value**
DNAJC10	-0.0929	FUT8	-0.0844
ARID3B	-0.0880	LGALS8	-0.0781
ZNF502	0.0756	LDB3	0.0766
TBC1D13	0.0611	ARID3B	-0.0701
ZC2HC1C	0.0601	LCE2D	-0.0677

Component 3:‘galactosylation’	
Gene Symbol	**Loading value**
MTO1	0.0875
AKAP9	-0.0627
MRPL33	-0.0622
MYLPF	0.0562
POLR2F	0.0554

Table 3.2: **Simulation results for OmicsPLS and r.jive: inner products.** These results are for $p = q = 100$. 1000 replicates were generated. Median (MAD) values of (the absolute value of) inner products between true and estimated loading vectors for O2PLS and JIVE. Higher values indicate better agreement with true loadings. The results are very similar for high-dimensional data ($p = q = 10^4$).

	OmicsPLS	**r.jive**
X joint	0.88 (0.09)	0.88 (0.09)
X specific	0.79 (0.08)	0.78 (0.09)
Y joint	0.85 (0.08)	0.85 (0.08)
Y specific	0.93 (0.013)	0.92 (0.014)

Table 3.3: **Performance comparison of OmicsPLS and r.jive w.r.t. median (MAD) total elapsed time in seconds across 1000 replicates, and convergence across 100 runs.** For the convergence, the heterogeneity scenario $U = 10T$ was used.

	CPU time (sec)		**Convergence (%)**	
Dimensions	**OmicsPLS**	**r.jive**	**OmicsPLS**	**r.jive**
Low ($p = q = 100$)	0.04 (0.007)	14 (2.8)	100	9
High ($p = q = 10^4$)	18 (4.1)	132 (16)	100	8

3.8 Figures

Figure 3.1: **Workflow of the OmicsPLS package.** Firstly, each data set is preprocessed. Secondly, O2PLS is used to decompose each data set in joint, specific and residual parts. Finally, the output is visualized and interpreted.

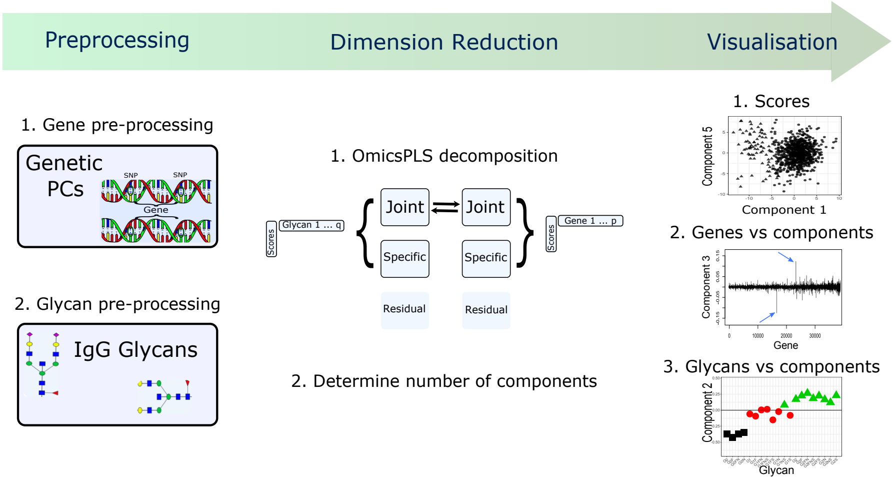


Figure 3.2: **Eigenvalues of the covariance matrices of the genetic and glycan data.** The relative contribution of each eigenvalue towards the sum of all eigenvalues is shown for the Genetic PCs and IgG1 glycan data, and their covariance, respectively.

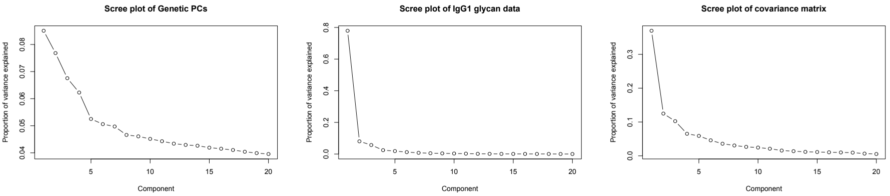
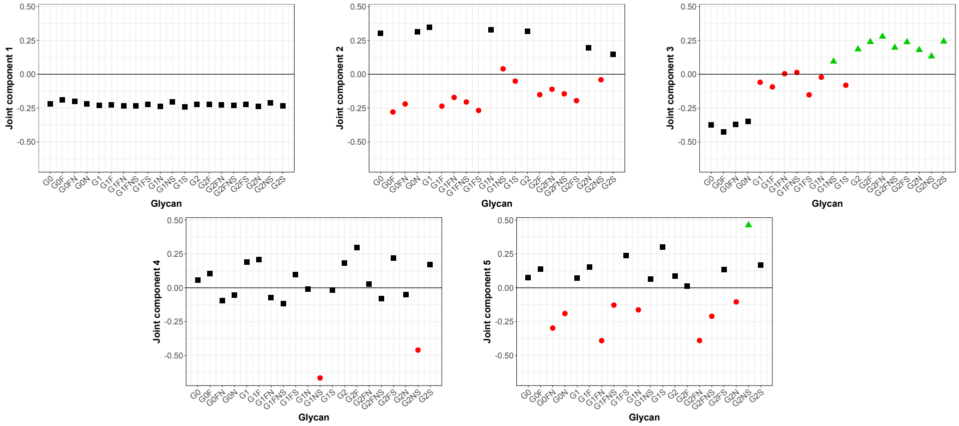


Figure 3.3: **Genetic-Glycan joint principal components obtained with the OmicsPLS R-package.** Loading values of each IgG1 glycan variable are depicted per component. The colors and shapes represent the biological grouping of the glycans. In the last row and column, a graphical representation of the structure of a particular glycan is shown.



Bibliography

[1] D. Amaratunga and J. Cabrera. Analysis of Data From Viral DNA Microchips. *J. Am. Stat. Assoc.*, 96(456):1161–1170, dec 2001.

[2] S. el Bouhaddani, J. Houwing-Duistermaat, P. Salo, M. Perola, G. Jongbloed, and H.-W. Uh. Evaluation of O2PLS in Omics data integration. *BMC Bioinformatics*, 17(S2):S11, dec 2016.

[3] H. Hotelling. Relations Between Two Sets of Variates. *Biometrika*, 28(3/4):321, dec 1936.

[4] W. E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, jan 2007.

[5] KULEUVEN. DISCO-SCA software, 2017.

[6] G. Lauc, J. E. Huffman, M. Pučić, L. Zgaga, B. Adamczyk, A. Mužinić, M. Novokmet, O. Polašek, O. Gornik, J. Krištić, T. Keser, V. Vitart, B. Scheijen, H.-W. Uh, M. Molokhia, A. L. Patrick, P. McKeigue, I. Kolčić, I. K. Lukić, O. Swann, F. N. van Leeuwen, L. R. Ruhaak, J. J. Houwing-Duistermaat, P. E. Slagboom, M. Beekman, A. J. M. de Craen, A. M. Deelder, Q. Zeng, W. Wang, N. D. Hastie, U. Gyllensten, J. F. Wilson, M. Wuhrer, A. F. Wright, P. M. Rudd, C. Hayward, Y. Aulchenko, H. Campbell, and I. Rudan. Loci Associated with N-Glycosylation of Human Immunoglobulin G Show Pleiotropy with Autoimmune Diseases and Haematological Cancers. *PLoS Genet.*, 9(1):e1003225, jan 2013.

[7] K.-A. Lê Cao, D. Rossouw, C. Robert-Granié, and P. Besse. A sparse PLS for variable selection when integrating omics data. *Stat. Appl. Genet. Mol. Biol.*, 7(1), 2008.

[8] D. Lin, J. Zhang, J. Li, C. Xu, H.-w. Deng, and Y.-p. Wang. An integrative imputation method based on multi-omics datasets. *BMC Bioinformatics*, 17(1):247, dec 2016.

[9] E. F. Lock, K. A. Hoadley, J. S. Marron, and A. B. Nobel. Joint and individual variation explained (JIVE) for integrated analysis of multiple data types. *Ann. Appl. Stat.*, 7(1):523–542, 2013.

[10] C. Meng, O. A. Zeleznik, G. G. Thallinger, B. Kuster, A. M. Gholami, and A. C. Culhane. Dimension reduction techniques for the integrative analysis of multi-omics data. *Brief. Bioinform.*, 17(October 2015):bbv108, 2016.

[11] M. J. O'Connell and E. F. Lock. R.JIVE for exploration of multi-source molecular data. *Bioinformatics*, 32(June):btw324, 2016.

[12] M. D. Ritchie, E. R. Holzinger, R. Li, S. A. Pendergrass, and D. Kim. Methods of integrating data to uncover genotype–phenotype interactions. *Nat. Rev. Genet.*, 16(2):85–97, 2015.

[13] F. Rohart, B. Gautier, A. Singh, and K.-A. Lê Cao. mixOmics: An R package for ‘omics feature selection and multiple data integration. *PLOS Comput. Biol.*, 13(11):e1005752, 2017.

[14] J. Trygg and S. Wold. O2-PLS, a two-block (X–Y) latent variable regression (LVR) method with an integral OSC filter. *J. Chemom.*, 17(1):53–64, 2003.

[15] UMetrics. SIMCA O2PLS software, 2017.

[16] F. M. van der Kloet, P. Sebastián-León, A. Conesa, A. K. Smilde, and J. A. Westerhuis. Separating common from distinctive variation. *BMC Bioinformatics*, 17(S5):S195, dec 2016.

[17] K. Van Deun, A. K. Smilde, M. J. van der Werf, H. a. L. Kiers, and I. Van Mechelen. A structured overview of simultaneous component based data integration. *BMC Bioinformatics*, 10:246, 2009.

[18] K. Van Deun, I. Van Mechelen, L. Thorrez, M. Schouteden, B. De Moor, M. J. van der Werf, L. De Lathauwer, A. K. Smilde, and H. a. L. Kiers. DISCO-SCA and Properly Applied GSVD as Swinging Methods to Find Common and Distinctive Processes. *PLoS One*, 7(5):e37840, 2012.

[19] V. Voillet, P. Besse, L. Liaubet, M. San Cristobal, and I. González. Handling missing rows in multi-omics data integration: multiple imputation in multiple factor analysis framework. *BMC Bioinformatics*, 17(1):402, 2016.

[20] R. Wehrens and W. E. van der Linden. Bootstrapping Principal Component Regression Models. *J. Chemom.*, 11(April 1996):157–171, 1997.

[21] H. Wold. Nonlinear iterative partial least squares (NIPALS) modelling: some current developments. In *Multivar. Anal. III (Proc. Third Internat. Symp. Wright State Univ., Dayton, Ohio, 1972)*, pages 383–407. Academic Press, New York, 1973.