Practicals multiblock analyses

Context and dataset introduction

```
library(CCA)
data("nutrimouse")
```

The dataset comes from a nutrigenomic study in the mouse (*Martin et al., 2007, https://doi.org/10.1002/hep.21510*) in which the effects of five regimens with contrasted fatty acid compositions on liver lipids and hepatic gene expression in mice were considered.

Objective: Investigate the impact of five diets with distinct fatty acid profiles on liver lipids and gene expression.

Design: 40 mice, cross-classified by:

Genotype: WT vs PPARα -/-

Diet: corn and colza oils (50/50) for a reference diet (REF), hydrogenated coconut oil for a saturated fatty acid diet (COC), sunflower oil for an Omega6 fatty acid-rich diet (SUN), linseed oil for an Omega3-rich diet (LIN) and corn/colza/enriched fish oils for the FISH diet (43/43/14)

Variables:

Gene expression: 120 selected genes (among about 30,000) as potentially relevant in the context of the nutrition study (macroarray)

Lipid composition: 21 hepatic fatty acids (GC analysis)

Unsupervised multiblock analysis - ComDim

Goals

Explore structure and variability in the complete multiomics dataset consisting of lipid and gene data.

Tasks

1. Data Preparation

Concatenate lipid and gene dataframes and define the number of variables in each block.

2. ComDim Analysis

Run ComDim() from the **MBAnalysis** package.

```
library(MBAnalysis)
ComDim_res <- ComDim(X = ComDim_data,
                     block = n_group,
                     name.block = c("genes", "lipids"),
                     scale = T,
                     scale.block = T)
```

3. Block Contributions

Plot **saliences** to assess how lipids and genes contribute to each dimension.

```
saliences <- ComDim_res$saliences
```

4. Sample Space Visualization

Plot **scores** on Dim.1 vs Dim.2 and Dim.3 vs Dim.4
Identify main sources of variation.

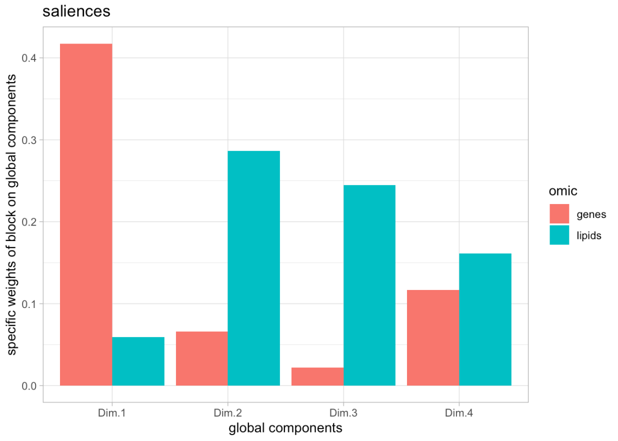
```
scores <- data.frame(metadata, ComDim_res$Scor.g)
ComDim_res$cumexplained[1,"%explX"]
```

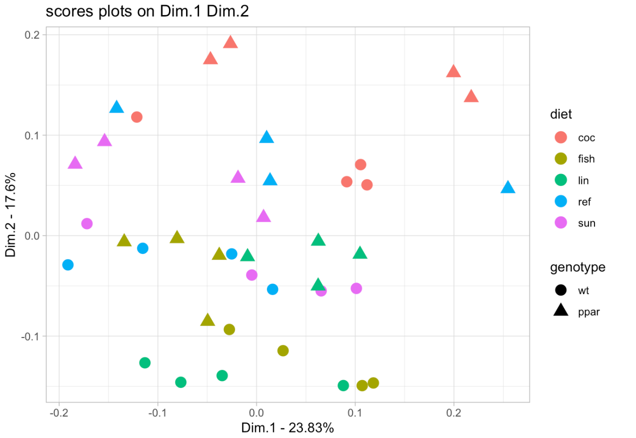
5. Variable Contributions

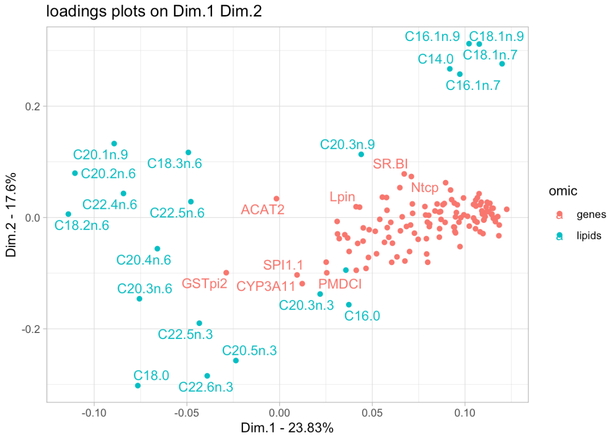
Plot **loadings** on Dim.1 vs Dim.2 and Dim.3 vs Dim.4
Highlight key genes and lipids driving sample differences.

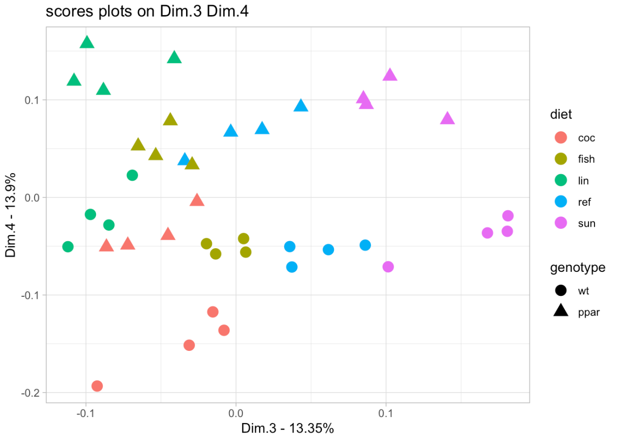
```
loadings <- data.frame(ComDim_res$Load.g)
```

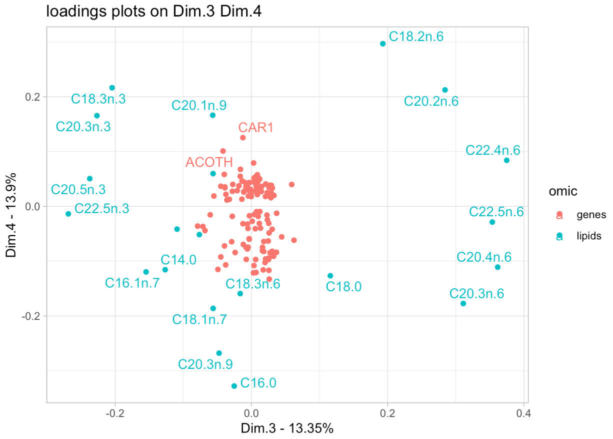
ComDim analysis of complete dataset


saliences

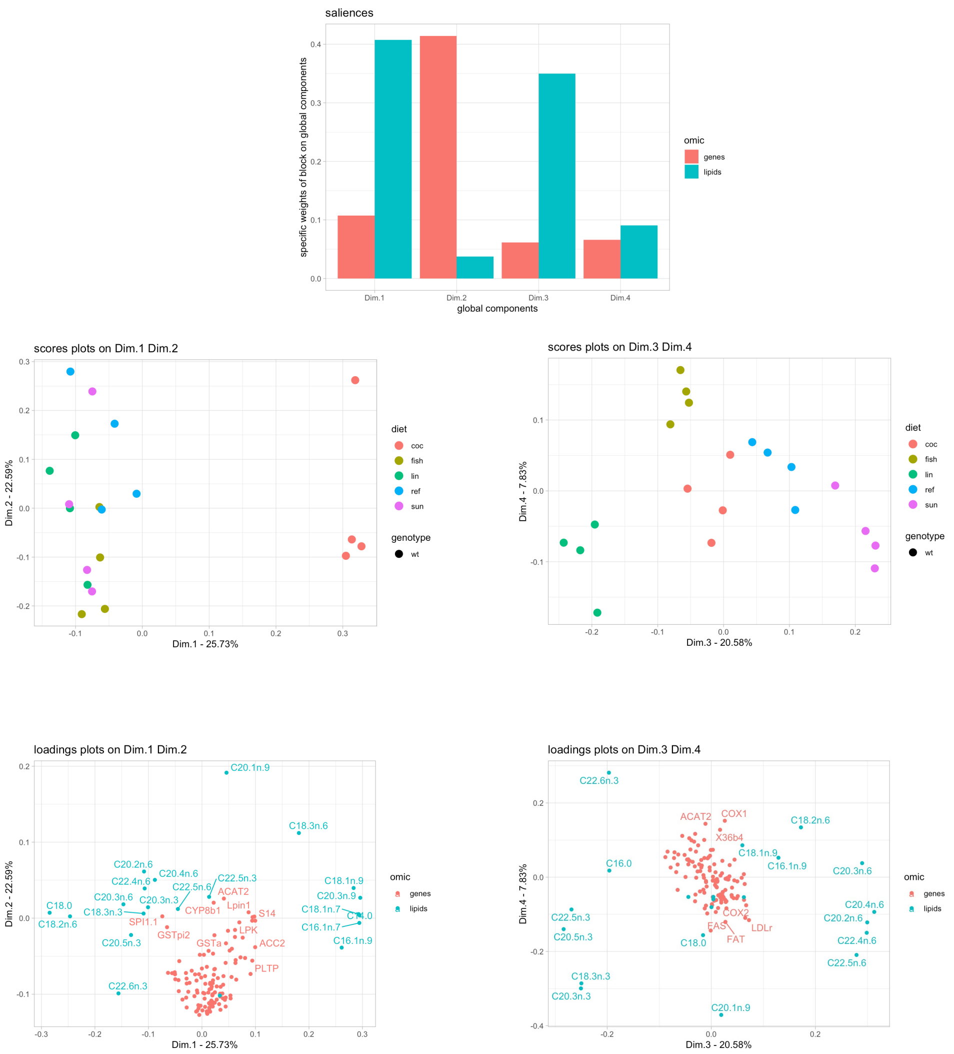

scores plots on Dim.1 Dim.2

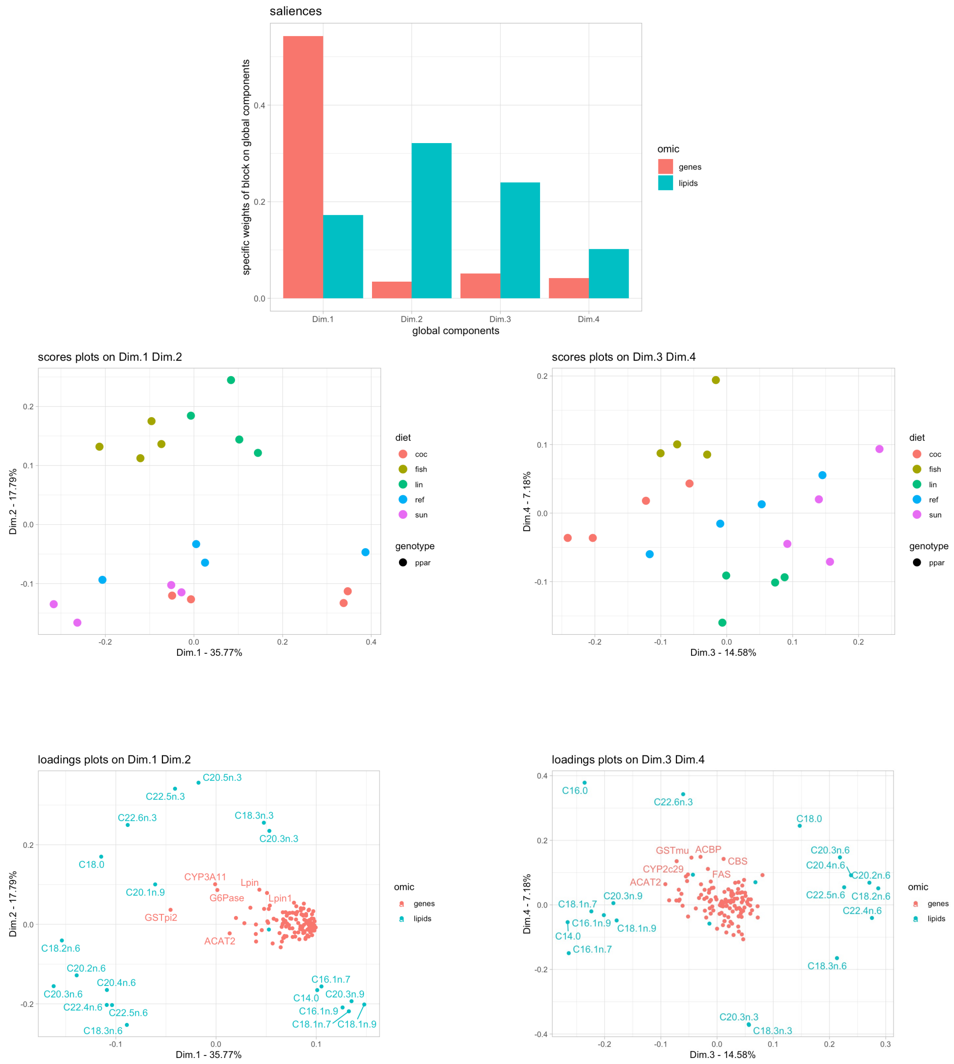

loadings plots on Dim.1 Dim.2


scores plots on Dim.3 Dim.4


loadings plots on Dim.3 Dim.4

ComDim analysis of WT and ppar datasets





blockPLS Discriminant analysis wt vs ppar - mixOmics

Goals

Discriminate between **WT** and **PPAR** samples using integrated lipid and gene data.

Tasks

1. Data Preparation

Format lipid and gene data as a list of dataframes.
Define the outcome variable (genotype) as a factor.

2. Block PLS-DA Analysis

Run block.plsda() from the mixOmics package to model genotype discrimination.

```
block.plsda(X = blockPLS_data,
            Y = genotype,
            design = "full",
            all.outputs = T,
            ncomp = 10)
```

3. Latent Variable Selection

Use perf() to evaluate model performance across components and visualise.
Re-run block.plsda() using the optimal number of latent variables.

```
perf(blockPLS_res, validation = 'Mfold',
folds = 7, nrepeat = 10, auc = TRUE, cpus=2)
```

4. Model Significance

Perform a cross-validation using DIABLO.cv() from the RVAideMemoire package
Perform a permutation test using DIABLO.test() from the RVAideMemoire package.

```
DIABLO.cv(blockPLS_res)
DIABLO.test(blockPLS_res)
```

5. Explained Variance

Plot explained variance per block and globally across latent variables.

```
blockPLS_res$AVE$AVE_X[1:2]
blockPLS_res$AVE[["AVE_outer"]]
```

6. Sample Space Visualization

Use plotIndiv() to visualize sample distributions in latent variable space.

```
plotIndiv(blockPLS_res, block ="weighted.average")
```

7. Variable Contributions

Use plotVar() to identify discriminant genes and lipids contributing to genotype separation.

```
plotVar(blockPLS_res)
```

blockPLS Discriminant analysis wt vs ppar - mixOmics

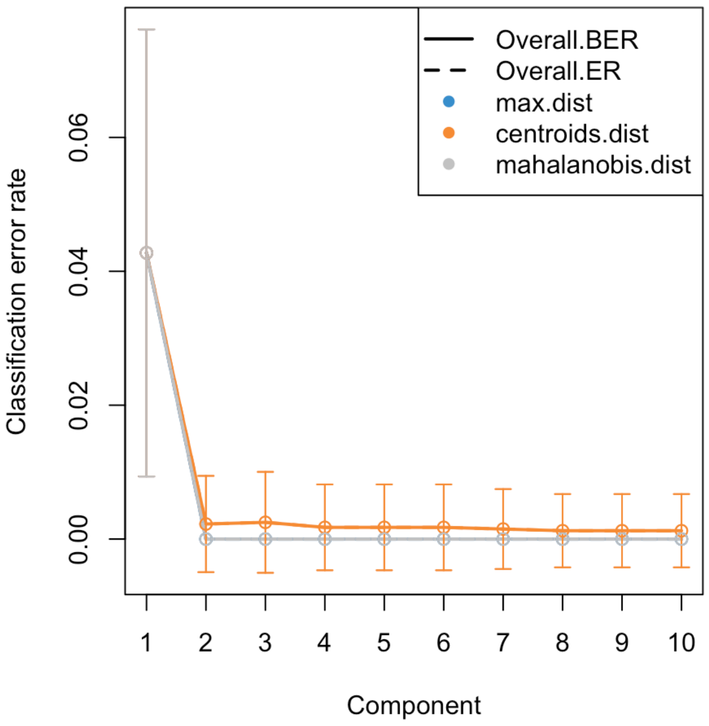
2. Block PLS-DA Analysis

```
blockPLS_res <- block.plsda( X = blockPLS_data,
                             Y = genotype,
                             design = "full",
                             all.outputs = T,
                             ncomp = 10 )
```

Specifies the **connectivity between blocks** in the blockPLS/DIABLO framework: which blocks should share information.
`"full"` means **all blocks are connected**, so the method maximizes covariance between all block pairs.

blockPLS Discriminant analysis wt vs ppar - mixOmics

3. Latent Variable Selection



4. Model Significance

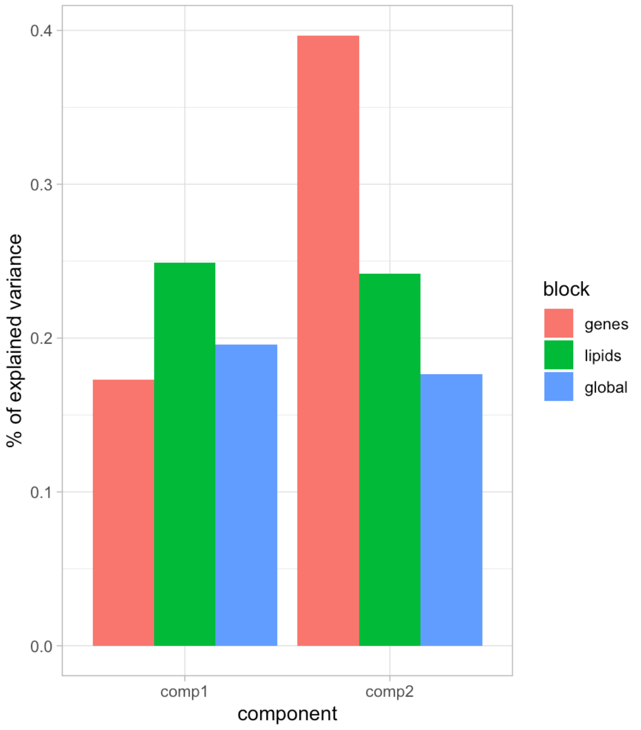
```
Cross validation

Model: DIABLO
7-fold validation
Validation repeated 100 times
2 components

Classification criterion: Mahalanobis distance

Mean (standard error) classification error rate (%): 0.575 (0.015)



Permutation test based on cross-validation

data:  blockPLS_res
DIABLO (2 components)
100 permutations
CER = 0.01375, p-value = 0.009901
```

CER: observed classification error rate

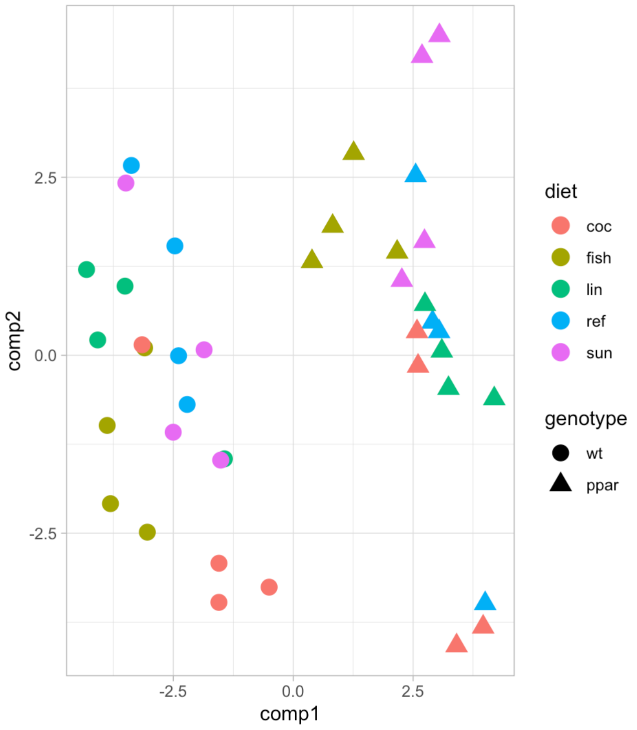
empirical p-value: the proportion of permutations where the permuted model achieved an equal or lower classification error rate than the real model.

blockPLS Discriminant analysis wt vs ppar - mixOmics

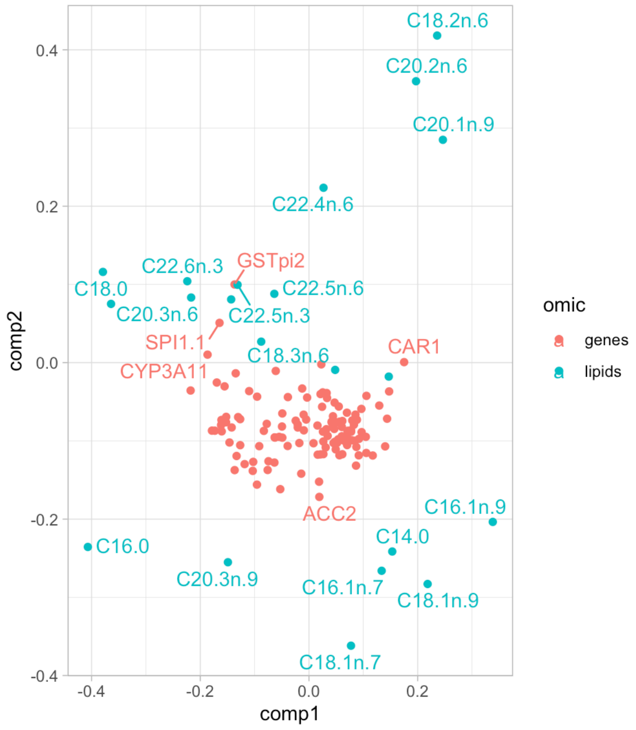
5. Explained Variance



6. Sample Space Visualization



7. Variable Contributions



Summary blockPLS-DA mixOmics

Purpose:

Reduce the dimensionality of multi-block, high-dimensional omics data while simultaneously modeling the relationship with a categorical outcome (e.g., disease status, genotype) preserving covariance between blocks.

How it works:

blockPLS-DA constructs latent variables (LVs), which are linear combinations of variables in each block.
LVs are extracted sequentially: Each LV captures the maximal covariance with Y and across connected blocks. Blocks can be fully connected (design = "full") or selectively connected.

Key properties:

- Supervised: Components maximize discrimination between predefined classes in Y.
- Multiblock: Integrates information across multiple omics datasets, modeling both within-block variance and between-block covariance.
- Latent variables are orthogonal within each block (depending on algorithm settings) and ordered by explained variance in relation to class separation.

Why use blockPLS-DA in omics:

- Identify components that discriminate classes: Extracts latent variables that optimally separate experimental conditions, disease states, or genotypes.
- Visualize class separation in multiblock space: score plots show how well classes are separated in the reduced space (supervised).
- Identify variables contributing most to discrimination: Variable loadings per block highlight biomarkers or key features driving separation between classes.
- Integrate multiple omics layers: Captures correlated features across blocks that jointly explain class variation.

Discriminant analysis wt vs ppar - consensusOPLS

Goals

Discriminate between **WT** and **PPAR** samples using integrated **lipid and gene** data.

Tasks

1. Data Preparation

Organize lipid and gene datasets as a list of dataframes (one per block).
Encode the outcome variable (genotype)

2. Consensus OPLS Analysis

Fit the Consensus OPLS model.

```
COPLS_res <- ConsensusOPLS(data = COPLS_data,
                          Y = genotype,
                          maxPcomp = 1,
                          maxOcomp = 1,
                          modelType = "da",
                          cvType = "nfold",
                          nfold = 5,
                          nperm = 100,
                          verbose = T)
```

3. Model Significance

Assess statistical significance using permutation tests and visualize plotting Q^2, dQ^2, and R^2Y values from permuted models.

```
plotQ2(COPLS_res)
plotDQ2(COPLS_res)
plotR2(COPLS_res)
```

4. Block Contributions

Plot block contributions as a bar plot to see which omics layer drives genotype discrimination.

```
plotContribution(COPLS_res)
```

5. Sample Space Visualization

Plot scores to visualize sample distribution in the space of predictive and orthogonal latent variables.

```
plotScores(COPLS_res)
```

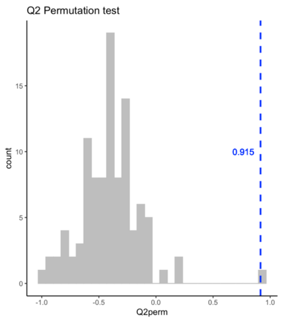
6. Variable Contributions

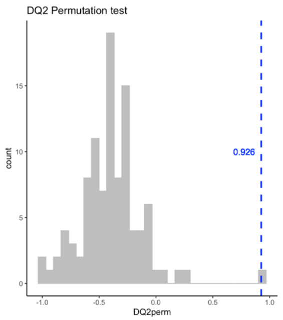
Identify discriminant genes and lipids contributing to genotype separation plotting loadings and VIP.

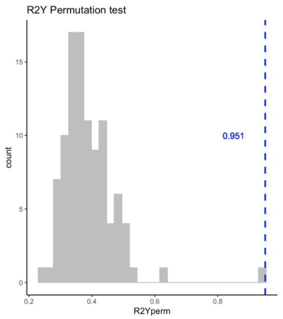
```
plotLoadings(COPLS_res)
```

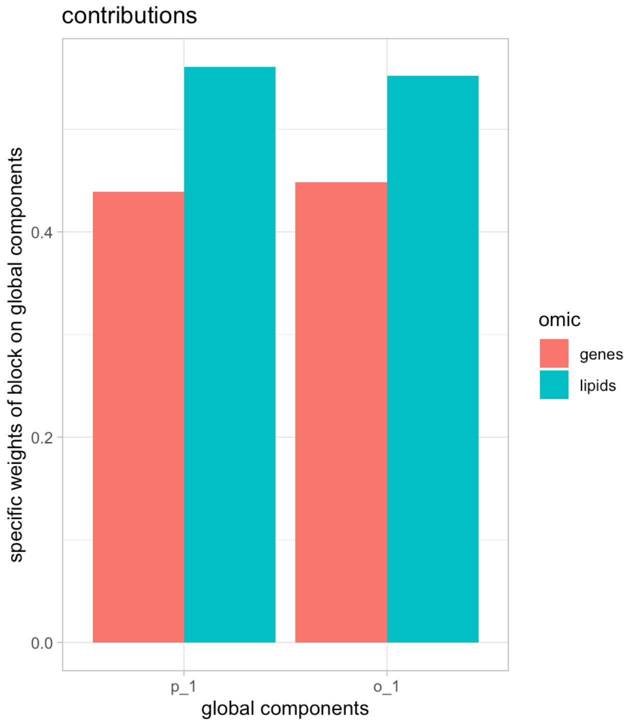
```
COPLS_res$optimal$VIP
```

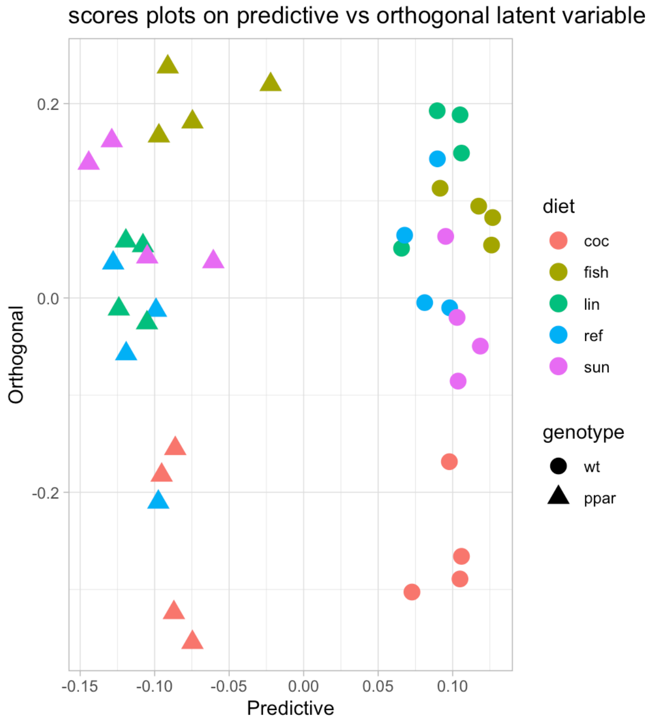
Discriminant analysis wt vs ppar - consensusOPLS

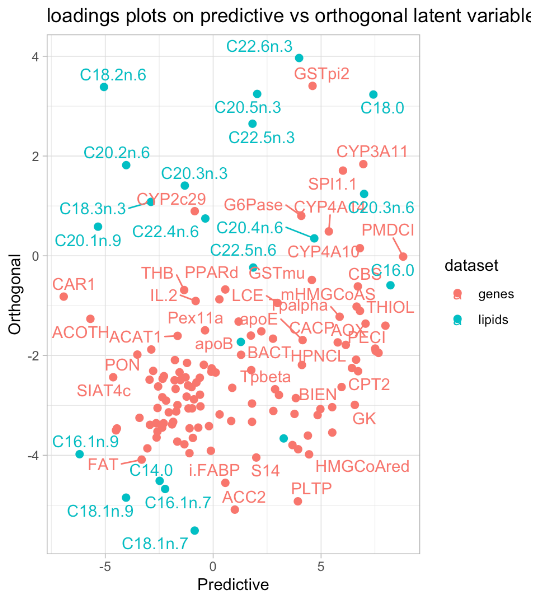

Q2 Permutation test

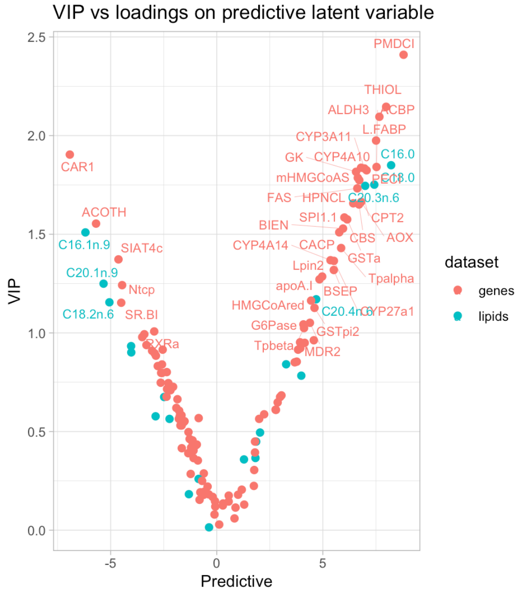

DQ2 Permutation test


R2Y Permutation test


contributions


scores plots on predictive vs orthogonal latent variable


loadings plots on predictive vs orthogonal latent variable


VIP vs loadings on predictive latent variable

Predicting with consensusOPLS

Goals

Train a Consensus OPLS-DA model to discriminate between WT and PPAR samples using a training set and evaluate its performance on an independent test set.

Tasks

1. Data Splitting

Select 30 samples for training and 10 samples for testing.
Organize lipid and gene datasets as lists of matrices for training and testing.
Scale each block separately for training and test sets.
Define the outcome variable (genotype) for the training set.

2. Consensus OPLS Analysis

Fit the Consensus OPLS model on the training set.

```
COPLS_res <- ConsensusOPLS(data = COPLS_data,
                           Y = genotype,
                           maxPcomp = 1,
                           maxOcomp = 1,
                           modelType = "da",
                           cvType = "nfold",
                           nfold = 5,
                           nperm = 100,
                           verbose = T)
```

3. Model Evaluation

Assess statistical significance using permutation tests and visualize plotting Q^2, dQ^2, and R^2Y values from permuted models. Observe block contributions, scores and loadings.

```
plotQ2(COPLS_res)
plotDQ2(COPLS_res)
plotR2(COPLS_res)
```

```
plotContribution(COPLS_res)
```

```
plotScores(COPLS_res)
```

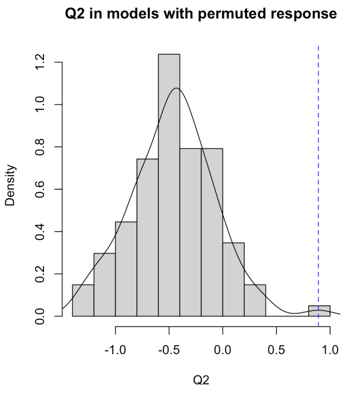
```
plotLoadings(COPLS_res)
```

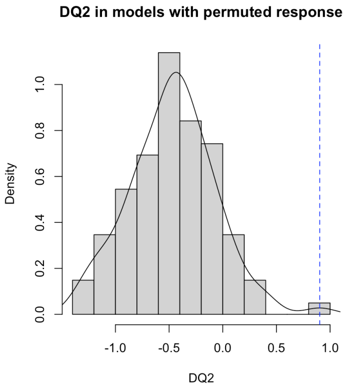
```
COPLS_res$optimal$VIP
```

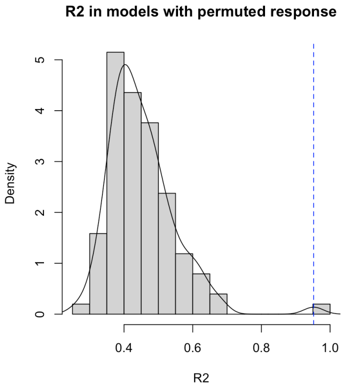
4. External Validation on Test Set

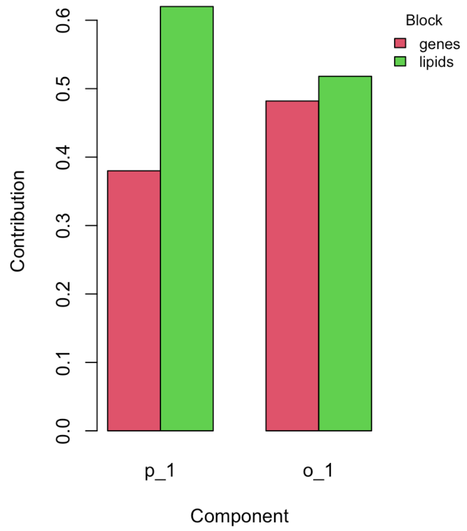
Predict class labels for the independent test samples using predict().
Compare predicted classes and predicted Y values against the true genotypes.
Summarize in a table for accuracy evaluation.

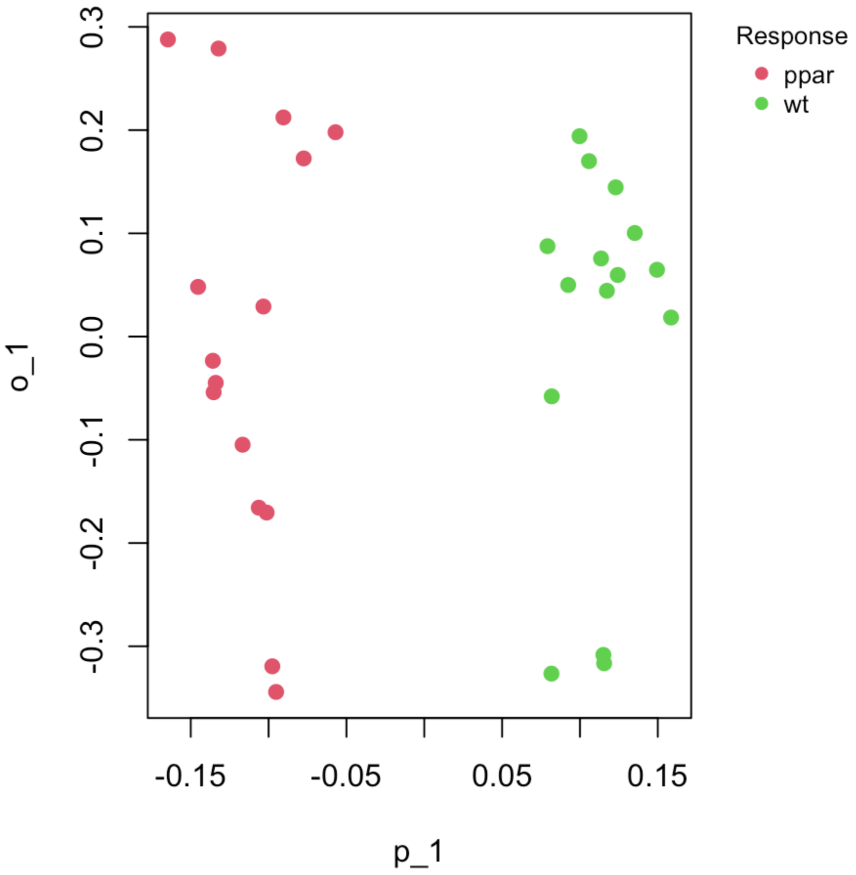
Predicting with consensusOPLS

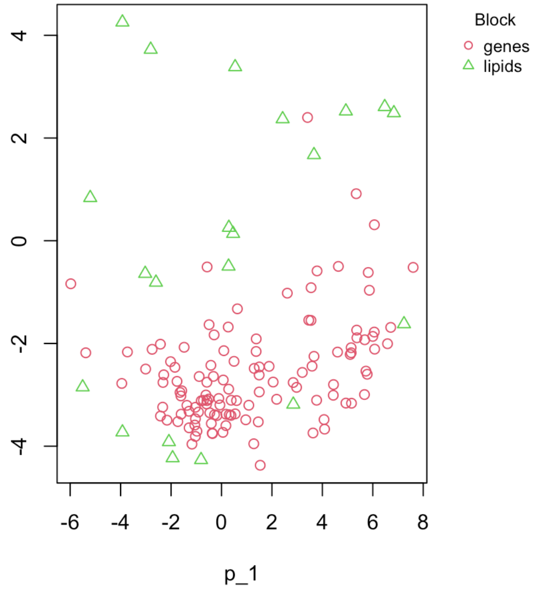

Q2 in models with permuted response

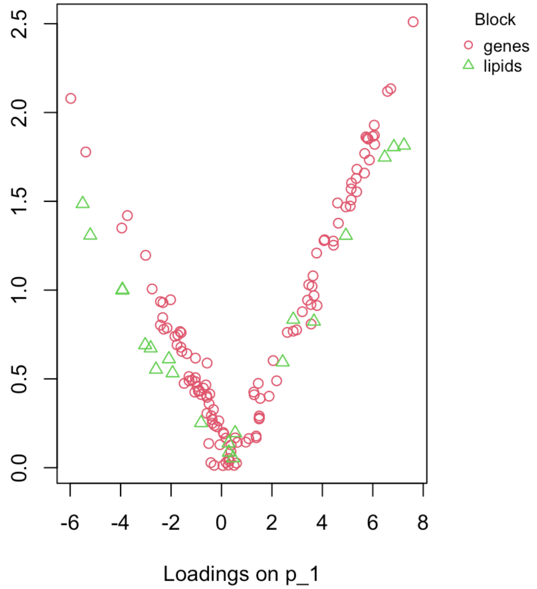

DQ2 in models with permuted response


R2 in models with permuted response









Predicting with consensusOPLS

```
              class        margin  softmax.ppar  softmax.wt   Y.pred.ppar   Y.pred.wt True.genotype
              <chr>         <dbl>         <dbl>       <dbl>         <dbl>       <dbl> <fctr>

8_wt_lin      wt        2.0073085      0.000000  1.00000000    -0.5036542   1.5036542 wt
1_wt_lin      wt        2.2077299      0.000000  1.00000000    -0.6038649   1.6038649 wt
11_wt_fish    wt        2.7191952      0.000000  1.00000000    -0.8595976   1.8595976 wt
16_wt_lin     wt        1.6347977      0.000000  1.00000000    -0.3173988   1.3173988 wt
17_wt_coc     wt        1.6995590      0.000000  1.00000000    -0.3497795   1.3497795 wt
21_ppar_coc   ppar      2.4374451      1.000000  0.00000000     1.7187225  -0.7187225 ppar
34_ppar_sun   ppar      2.1653278      1.000000  0.00000000     1.5826639  -0.5826639 ppar
38_ppar_fish  ppar      0.5705763      0.940192  0.05980796     0.7852882   0.2147118 ppar
30_ppar_lin   ppar      2.5939881      1.000000  0.00000000     1.7969940  -0.7969940 ppar
35_ppar_fish  ppar      2.5012529      1.000000  0.00000000     1.7506264  -0.7506264 ppar
```

The predicted class labels for each sample in the test set.

The difference between the highest predicted score and the second-highest predicted score for a sample.

The probability that the sample belongs to each class

The predicted response values for each sample in each class.

The actual genotype labels for the test samples.

ConsensusOPLS - summary

Purpose:
Reduce the dimensionality of multi-block, high-dimensional omics data while modeling the relationship with a categorical outcome (e.g., disease status, genotype), separating predictive variation from orthogonal variation and preserving covariance between blocks.

How it works:
Consensus OPLS constructs latent variables (LVs): Predictive LVs capture variance in X blocks related to Y.
Orthogonal LVs capture systematic variance in X blocks unrelated to Y.
LVs are extracted sequentially: Maximize covariance with Y (predictive) while modeling correlations across blocks.

Key properties:

- Supervised: LVs explicitly maximize discrimination between predefined classes in Y.
- Multiblock: Integrates multiple omics datasets, modeling both within-block variance and between-block covariance.
- Separation of variance: Predictive vs orthogonal components clarify variance associated with outcome vs unrelated variance.

Why use blockPLS-DA in omics:

- Identify components that discriminate classes: Extracts latent variables that optimally separate experimental conditions, disease states, or genotypes.
- Identify discriminant variables: Loadings and VIP scores highlight genes or lipids driving class separation.
- Integrate multiple omics layers: Captures correlated features across blocks that jointly explain predictive variance.
- Predict new samples and use external validation.