BDA project 2022-2023

G.R. van der Ploeg, J.A. Westerhuis, A.U.S. Heintz-Buschart, A.K. Smilde

09/01/2023 - 20/01/2023

Contents

BDA project 2022-2023

Project description

Welcome to the BDA project 2022-2023 syllabus. In this document you will find all the information you need to do your BDA project.

During the project, we will use a dataset produced by Caldana et al. (2011). During the "do-it-yourself" part of the first 6 practical sessions you should apply the new data analysis method that you have learned on this dataset. Our aim is that you will learn more about the methods and their comparison by applying them to the same data. Over time you will learn what the properties of your data are, what questions can be answered by the different methods, what samples tend to cluster together, and what metabolites or genes are important to distinguish the conditions. Learning the properties of your data is very helpful when you are applying a new method.

Project goals

The following goals are defined for this project:

1. You know the origin of the data and the specific properties of the data.
2. You are able to apply the methods, and you are able to interpret the results.
3. You comprehend the pitfalls of multivariate data and validation strategies to prevent overfit.
4. You are able to critically review data analysis applications of the above mentioned methods.

Keep in mind that the assignments are intended to assess whether you/your group has reached these goals. We hope this helps in answering the questions.

Groups

We encourage you to do the project in pairs. This is because many of the open-ended questions we ask during your project are open for debate. In such cases it is really helpful to have a partner who you can discuss with about what the next step should be. Also you will be able to run RStudio on two computers at the same time, which will help you in splitting up the tasks and completing the assignments faster overall. Please supply your names in the code block below.

```
name1 = "John Williams"
name2 = "Mary Harris Jones"
```

Submission and grading

You will hand in your project markdown file and knitted .pdf files in the first and second week. In the first week you will get an indicative pass/fail grade and some feedback to help you progress. In the second week you will receive a full grade and some feedback. The deadline for the first week submission is Monday January 16th 11:00, for which you need to complete Assignments 1 up to and including Assignment 3. The deadline for the second week submission is Monday January 23rd 11:00, for which you will need to complete Assignment 1 up to and including Assignment 6. Note that you can use the week 1 feedback to improve your

answers from Assignments 1-3 before the second submission! Whichever version of Assignments 1-3 will give you the highest grade will be used for grading in week 2.

The project grade and the exam grade will be combined to produce the final grade of the course. The project grade (P) will have weight 1 and the exam grade (E) will have weight 2. Both grades have to be equal to or higher than 5.0. If you fail the project but not the exam, you will have to do a re-take of the project to finish the course.

$finalGrade = \frac{1*P+2*E}{3}$

Please refer to the course book for more information on the BDA course organisation.

Grading of the project will be done using a rubric for each assignment. An overview of the number of points you can get for every assignment is given below. Every question states how many points you can get for it.

Assignment	Number of points
Assignment 1: Linear Algebra	33
Assignment 2: Gene expression analysis	33
Assignment 3: Principal Component Analysis	34
Assignment 4: Clustering	34
Assignment 5: Classification	33
Assignment 6: ASCA	33
Total	200

It is crucial for our graders to understand what you have done and why. Please elaborate in the supplied text boxes below each question what your reasoning is for making a certain step. To make your plots eligible for grading, you should add a clear x-axis label, y-axis label and title, as well as a clear description of what you have done to make your plot. When writing your code in the code blocks, add comments clarifying what you are doing! You can do this by adding a hash tag ("#") before the comment.

The Caldana data

In this project we will use a dataset produced by Caldana et al. (2011). The paper can be found on Canvas, and we highly recommend you to read it before starting the project. You don't need to understand all parts of the paper, but it's important to focus on the experimental design, the computational methods used, and the biology so you can understand your own results better.

The Caldana et al. (2011) paper describes an experiment where *Arabidopsis thaliana* plants were grown under normal temperature (21°C) and light conditions (150 μE) (see figure 1). After some time, they were moved to a different temperature and/or light condition. The conditions are as follows: the base condition (encoded 21-L), 21 °C and bright light (21-HL), 32 °C and normal light (32-L), 4 °C and low light (4-L), 21 °C and low light (21-LL), 4 °C in the dark (4-D), 21 °C in the dark (21-D) and 32 °C in the dark (32-D). At every timepoint, the rosetta leaves of six new plants were sampled for transcriptomics and metabolomics analysis. This way the adaptation of the plants to the new conditions could be tracked. Samples were taken at 18 timepoints: 20, 40, ..., 360 minutes (and the control timepoint 0).

Data have been obtained on 92 metabolites and 15047 unique genes. In total, samples for 8 conditions x 18 timepoints were measured, plus the baseline condition at timepoint 0. The replicates were summarized per condition-timepoint combination. Both datasets have been transformed to a log-scale as well. In the further project description, we will refer to the condition-timepoint combinations as "samples". Also note that the 21-L samples are from the plants being kept at the baseline condition throughout.

During the project we will work a lot with R markdown coding blocks such as the one below. These are intended for you to use to complete the assignments. During knitting, these code blocks will automatically be run and printed in your output document. If you have any comments to give on what you have done, there are text blocks below each code block where you can elaborate.

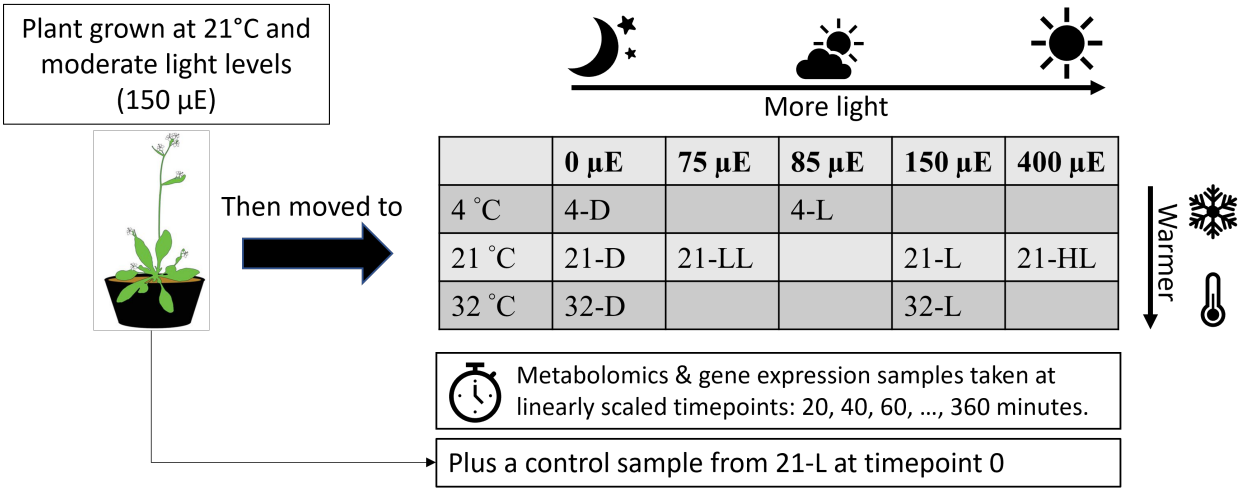

Figure 1: Overview of the experimental design as described in Caldana et al. (2011). Arabidopsis thaliana plants were grown under normal temperature (21 degrees) and light conditions (150 uE). After some time, they were moved to a different temperature and/or light condition. The conditions are as follows: the base condition (encoded 21-L), 21 degrees and bright light (21-HL), 32 degrees and normal light (32-L), 4 degrees and low light (4-L), 21 degrees and low light (21-LL), 4 degrees in the dark (4-D), 21 degrees in the dark (21-D) and 32 degrees in the dark (32-D). At every timepoint, the rosetta leaves of six new plants were samples for transcriptomics and metabolomics analysis. This way the adaptation of the plants to the new conditions could be tracked. Samples were taken at 18 timepoints.

The metabolomics and expression data are supplied in two files in ./Data/ in your unzipped BDA project folder. As your first step in understanding this data, we will load it into R for you and show you what the tables look like. We leave the rest of the exploration to you.

```
df_metabolomics = read.csv("./Data/Caldana_et_al_metabolomics_dataset.csv")
df_expression = read.csv("./Data/Caldana_et_al_expression_dataset.csv")
```

```
print(df_metabolomics[1:4, c(2,3,6,7,8)])
```

```
##   condition time    Alanine   Arabinose    Arabitol
## 1      21-D  100 -0.2299462  0.12785335  0.33647383
## 2      21-D  120 -0.1992554 -0.03012810 -0.05989591
## 3      21-D  140 -0.2964217 -0.04493523 -0.05922931
## 4      21-D  160 -0.2047872  0.07633106 -0.25555183
```

Above, a table is shown of a part of the metabolomics dataset. You can see some values for every metabolite: Alanine, Arabinose and Arabitol. Additionally, you can see the condition of the sample, in this case 21-D, meaning the plant was at 21 degrees in the dark. The timepoint is also shown. Every value that you see is the log2 of the area under the peak of the MS spectrum for that metabolite in that sample, which reflects the amount of that metabolite. If you're interested in how that works, we recommend searching the internet for "GC-MS metabolite quantification". You do not need to know how GC-MS works for our questions and exam.

```
print(df_expression[1:4, 2:7])
```

```
##   condition time AT2G26550 AT2G26570 AT2G26580 AT2G26360
```

```
## 1      21-L    0      7.90      8.14      6.70      2.81
## 2      21-L   20      8.14      8.12      6.00      2.48
## 3      21-L   40      8.08      8.18      6.71      2.51
## 4      21-L   60      8.08      8.19      6.43      2.47
```

Here a table is shown of a part of the gene expression dataset. You can see some values for some Arabidopsis thaliana genes: AT2G26550, AT2G26570, AT2G26580, and AT2G26360. When you look up these genes on TAIR, you can see that they encode a haeme oxygenase-like protein, a coiled-coil protein that moves chloroplasts, a transcription factor and an unnamed protein, respectively. As before, you can see the condition and time combination for the sample as well. Every value that you see is a normalized microarray intensity value, which reflects the transcript abundance of that gene in the sample.

Plant conditions to compare

During the assignments, you will often be asked to compare two plant conditions (see Figure 1). Please discuss which two conditions you want to compare consistently throughout your project, and state them below in the code block. An example choice is given. Use the supplied text box to explain why you want to investigate these two conditions.

```
condition1 = "21-HL"
condition2 = "21-D"
```

Explain your choice here.

End of text block.

Help

As any programmer will tell you, looking stuff up on Google is a major part of any project. Hence looking up specific R programming questions may be helpful in progressing in your project assignments. Coding websites such as StackOverflow are particularly helpful.

Specific questions on how functions work in R can be answered using the R documentation. Say you want to understand how you can run the singular value decomposition function svd(). On the right side of your screen you can find a "help" tab where you can search for the function that you want to use. You can type "svd" in the search field there to reach the documentation of the svd() function. Additionally, you can type "?svd" in your console window at the bottom of your screen to automatically search for the documentation of your supplied function.

This project is a new addition to the BDA course, and will replace the R test from previous years. If you have done BDA in previous years, the rules and regulations regarding grades and exemptions are the same as they used to be for the R test. If you have passed the R-test last year, you do not have to hand in the project assignments again this year. If you have failed the R-test last year, you will need to do the project this year instead.

If things are unclear to you or you have any other question, feel free to ask any of your lecturers or teaching assistants (TAs):
- Johan Westerhuis (Course coordinator, j.a.westerhuis@uva.nl)
- Anna Heintz Buschart (Lecturer, a.u.s.heintzbuschart@uva.nl)
- Roel van der Ploeg (Project coordinator, g.r.ploeg@uva.nl)
- Archontis Goumagias (TA, a.goumagias@student.vu.nl)
- Lucas Jansen (TA, l.s.jansen@student.vu.nl)

- Alex van Kaam (TA, a.t.van.kaam@student.vu.nl)

Assignments

Below are the packages we will use throughout the course. Do not edit this code block!

```
if(!"umap" %in% installed.packages()[,1]){
  install.packages("umap")
}

if(!"MASS" %in% installed.packages()[,1]){
  install.packages("MASS")
}

if(!"gtools" %in% installed.packages()[,1]){
  install.packages("gtools")
}

if(!"stringr" %in% installed.packages()[,1]){
  install.packages("stringr")
}

library(umap)
library(MASS)
library(gtools)
library(stringr)
source("./heatmap.2a.R")
```

Assignment 1: Linear Algebra (Johan Westerhuis, 09-01-2023)

In this assignment, you will start your exploration of the data by checking various summaries per row and per column of your data tables. You will also make a few histograms and inspect them to check the comparability of our samples and variables. Finally you will center and scale your data for use in the other assignments.

A) Import both datasets. You will see that the first column is full of sample names. Change the import code so it automatically uses that column as the row names. (Hint: use read.csv(), use the row.names argument to specify the column. Type "?read.csv" in your console window for more info.) **[1 point]**

Put any text you may want to type here.

End of text block.

B) Create a version of both datasets that only contains numerical data. That means you need to remove the "condition" and "time" columns. **[1 point]**

Put any text you may want to type here.

End of text block.

C) Calculate the total amounts per row and per column of both datasets. Use your numerical-only datasets for this. Plot the row and column sums in a histogram per dataset. What do the plots indicate about the comparability of the samples and variables? (Hint: you can use rowSums() and colSums() for this.) [**10 points**]

Write down your observations here.

End of text block.

D) Calculate the column mean and column standard deviation for each dataset. Make a histogram of your means and a histogram of your standard deviations for each dataset. Explain what the histograms describe in terms of comparability of the variables. (Hint: you can use apply() to calculate the mean and standard deviations per column.) [**10 points**]

Put any text you may want to type here.

End of text block.

E) Extract the amount of Glycine for the metabolomics 21-L samples for all timepoints. Save it in a variable. Plot the amount of Glycine over time in the input data, after you centered your variable, and after you centered & scaled your variable. Check what has happened to the mean and the standard deviation of your variable in each case. Explain what centering and/or scaling does to your data. [**7 points**]

Put any text you may want to type here.

End of text block.

F) Create a centered version of both datasets. You can do this by removing the column mean from every column. (Hint: consider using the sweep() function, you can use the means you have calculated in Assignment 1D for this.) [**2 points**]

G) Create a centered & scaled version of both datasets. You can do this by dividing a centered column by its standard deviation. This is also known as autoscaling. (Hint: consider using the sweep() function, you can use the standard deviations you have calculated in Assignment 1D for this.) [**2 points**]

Note: You have created new versions of the gene expression and metabolomics datasets. In the rest of the assignments we expect you to select the appropriate version of a dataset to use for a given method.

Put any text you may want to type here.

End of text block.

This concludes Assignment 1. Make sure you save your .Rmd file and workspace so you can continue in your next project session.

Assignment 2: Gene expression analysis (Douwe Molenaar, 12-01-2023)

In this assignment we will consider the gene expression data without centering or scaling it. We will perform a differential expression analysis to compare your two chosen plant conditions. While we ask you to do this using t-tests, please be aware that this is not really appropriate given the experimental design of the study. This is because the data points are not randomly drawn from a population, as they come from a time series.

A) Perform a t-test of the expression of AT2G20560 between your two chosen conditions. What does this p-value mean? Is this gene differentially expressed between your two conditions? **[4 points]**

Put any text you may want to type here.

End of text block.

B) Perform a t-test for every gene in the gene expression dataset for your chosen conditions. Collect the p-values in a vector. (Hint: consider using apply() or sapply().) **[3 points]**

Put any text you may want to type here.

End of text block.

C) Make a histogram of all the p-values. Explain what the outcome means for the significance of your t-tests between your two chosen conditions. Is it possible for a gene to not be differentially expressed, and to still have a low p-value? **[5 points]**

Put any text you may want to type here.

End of text block.

D) In your exercises on RNAseq data analysis, you have computed the log(ratio) of gene expression using the total RNA counts of a gene between two conditions. Our gene expression data is already normalized and log transformed, so we can instead calculate the fold change using the averages between the two conditions.
Calculate the fold change (FC) of AT2G20560 between your two chosen conditions. This means that you need to determine the mean of AT2G20560 in each condition separately. You can then subtract the mean of AT2G20560 in condition 2 from the mean of AT2G20560 in condition 1. **[3 points]**

Put any text you may want to type here.

End of text block.

E) Calculate the fold change of all genes between your two chosen conditions this way. **[3 points]**

Put any text you may want to type here.

End of text block.

F) Make a volcano plot using your FC and p-values for every gene. Interpret the result. Where are the important differentially expressed genes in this plot? Why? (Hint: to make a proper volcano plot you still need to do something to your p-values.) **[6 points]**

Put any text you may want to type here.

End of text block.

G) As you may be aware, you have now done over 15.000 t-tests. As such, you will need to control for Type I error. In the exercises you did before the project part of today's computer practical you have calculated the FDR value for any p-value and plotted them. An easy function to obtain the FDR values is p.adjust(). Control the type I error of your p-values using a false discovery rate. **[2 points]**

Put any text you may want to type here.

End of text block.

H) Find the genes that are differentially expressed by setting an "adjusted" p-value and FC threshold. Elaborate on your choice of threshold. What gene has the lowest "adjusted" p-value? What is its biological function? Does this result make sense given your choice of compared conditions? **[7 points]**

Put any text you may want to type here.

End of text block.

This concludes Assignment 2. Make sure you save your .Rmd file and workspace so you can continue in your next project session.

Assignment 3: Principal Component Analysis (Age Smilde, 13-01-2023)

During this Assignment, you will perform Principal Component Analysis on both the gene expression and metabolomics data containing all conditions. We will investigate groupings in the score plots, and have a look at important variables distinguishing your chosen conditions using the loadings.

A) Perform a principal component analysis on the gene expression data: Consider which version of the data from Assignment 1 you want to use. Make a score plot using the first and second principal component. Show the variance explained of each component in your plot labels. Add a legend to identify the colour for each condition. Does your plot show clear groups? What plant conditions does your PCA plot together in a group? Compare your plot with Figure 2 from the paper. (Hint: use the legend() function to clarify which colour belongs to which plant condition.) **[6 points]**

Put any text you may want to type here.

End of text block.

B) Perform a principal component analysis of the metabolomics data: Consider which version of the data you want to use. Make a score plot using the first and second principal component. Show the variance explained by each component in your plot labels. Add a legend to identify the colour for each condition. Does your plot show clear groups? What plant conditions does your PCA plot together in a group? (Hint: use SVD, consider centering and/or scaling your data for PCA specifically!) **[6 points]**

Note: your result is likely different from the PCA shown in Figure 2 of the paper. This is because the authors have chosen to log2 transform and subsequently scale their metabolomics data to median 1 for every column. There is no clear "right" or "wrong" PCA here.

Put any text you may want to type here.

End of text block.

C) Consider your metabolomics score plot. Which of the first two principal components is best at separating your two chosen conditions? Plot the loadings of that component in a bar plot. Which metabolite is considered the most important? What is the biological role of this metabolite? Does this result make biological sense given your choice of compared conditions? **[4 points]**

Put any text you may want to type here.

End of text block.

D) Now consider your gene expression PCA model. Reconstruct your gene expression dataset using the first and second PCs. (Hint: consider the way a dataset is decomposed to reconstruct the data.) **[2 points]**

Put any text you may want to type here.

End of text block.

E) Calculate the residuals of your gene expression dataset by subtracting the reconstructed data from the input data. Inspect the residuals. Report the condition and timepoint that was modeled most poorly. How much worse is it modeled than the average? **[4 points]**

Put any text you may want to type here.

End of text block.

F) Use your reconstructed gene expression data to calculate the variance explained per gene of the entire gene expression PCA model. You can do this by dividing the sum of squares of your reconstructed gene by the sum of squares of the gene in your input data. Which gene was modeled the best? What is its biological role? Does this result make biological sense given your choice of compared conditions? (Hint: consider using a colSums() approach.) **[2 points]**

Put any text you may want to type here.

End of text block.

G) Consider which principal component of the gene expression PCA best separates your chosen conditions. Check its loading. Which genes are considered important? Compare your result with the set of most differentially expressed genes from Assignment 2. What similarities and differences do you find? [**4 points**]

Put any text you may want to type here.

End of text block.

H) Why would the selected genes from differential expression analysis and PCA be different? Explain what the goal is of both methods and thus what it means that a variable is selected in either method. [**6 points**]

Put any text you may want to type here.

End of text block.

That concludes week 1 of the project! Please hand in your .rmd and .pdf files on Canvas for your pass/fail grade and feedback. The week 1 deadline is Monday, January 16th 11:00.

Assignment 4: Clustering (Johan Westerhuis, 16-01-2023)

In this assignment, you will focus on clustering your gene expression and metabolomics data. This will be done using hierarchical clustering and UMAP methods. You will also inspect the validity of the clustering results and compare it with the Caldana paper.

In the code block below, some functions are defined for you to use in subsequent questions. You do not need to understand what happens in this code. Do not edit the code block!

```
conditions = unique(df_expression$condition)
colours = c("grey", "black", "cyan", "blue", "orange", "red", "green", "yellow")

plot_hclust = function(hclust_result){
  dendrogram = dendrapply(as.dendrogram(hclust_result), labelCol)
  plot(dendrogram)
}

labelCol <- function(x) {
  if (is.leaf(x)) {

    ## fetch label
    label = attr(x, "label")

    ## extract condition
    condition = str_split_fixed(label, "_", 2)[,1]

    ## set label color
    index = which(conditions == condition)
```

```
    attr(x, "nodePar") = list(lab.col = colours[index])
  }
  return(x)
}

make_heatmap = function(data, condition1, condition2){
  referenceSample = data[(data$condition == "21-L") & (data$time == 0), 3:94]
  data = data[data$condition %in% c(condition1, condition2),]
  conditionLabels = data$condition
  timepoints = data$time
  data = sweep(data[,3:94], 2, as.numeric(referenceSample), FUN="-")

  selectMetabolites = as.data.frame(cbind(colnames(data), apply(abs(data), 2, mean)))
  selectMetabolites = selectMetabolites[order(selectMetabolites[,2], decreasing=T),]
  selectMetabolites = row.names(selectMetabolites)[1:20]

  reordering = order(factor(conditionLabels, levels=c(c(condition1, condition2))), timepoints)
  data = data[reordering, colnames(data) %in% selectMetabolites]
  conditionLabels = conditionLabels[reordering]

  data = t(data)
  colColours = 1:ncol(data)

  for(i in 1:ncol(data)){
  sampleCondition = conditionLabels[i]
  index = which(conditions == sampleCondition)
  colColours[i] = colours[index]
  }

  heatmap.2a(as.matrix(data[,order(conditionLabels)]), scale="row", hclustfun=function(x){hclust(x,metho
}
```

A) Calculate the euclidean distances between the samples in the gene expression data (consider which version of the data you wish to use). Then, use the distance matrix to create a hierarchical clustering using average linkage. Plot the hierarchical clustering using the plot_hclust() function defined in the code block above. An example of how you can run this custom function is given in the code block below. Compare your result with Figure 3 from the paper. Do you find the same clusters as the paper? Describe what plant conditions the dendrogram clusters together. (Hint: check the dist() and hclust() function documentation.) **[7 points]**

```
#plot_hclust(hclust_result)
```

Put any text you may want to type here.

End of text block.

B) Calculate the euclidean distances between the samples in the metabolomics data (consider which version of the data you wish to use). Use the distance matrix to create a hierarchical clustering using average linkage. Plot the hierarchical clustering using the plot_hclust() function. Compare your result with Figure 3 from the paper. Do you find the same clusters as the paper? Describe what plant conditions the dendrogram clusters together. **[7 points]**

Put any text you may want to type here.

End of text block.

C) Compare the hierarchical clusterings of your metabolomics and gene expression data. Do they cluster in the same way? Explain what similarities and/or differences between the datasets could cause this result. [**6 points**]

Put any text you may want to type here.

End of text block.

D) Execute the code block below. Don't edit it! This creates a heatmap showing only the 20 metabolites whose mean abundance in your chosen conditions differs the most from the control sample (21-L timepoint 0). Compare the result with figure 1 from the paper. Can you see a clear difference in metabolite abundance between your two chosen conditions? Check the biological role of the selected metabolites. Does the selection of metabolites make biological sense given your choice of two conditions to compare? [**6 points**]

```
make_heatmap(df_metabolomics, condition1, condition2)
```

Put any text you may want to type here.

End of text block.

E) Create a UMAP of both datasets. Supply the custom.config as argument to the umap() function when running it. Select an appropriate number of neighbors. Plot the result. Add a legend to identify the colour for each condition. Does UMAP find the same kind of clusters as the hierarchical clustering or the PCA? What can you say about the metabolites that are important for the UMAP clustering? [**8 points**]

```
custom.config = umap.defaults
custom.config$random_state = 123
```

Put any text you may want to type here.

End of text block.

This concludes Assignment 4. Make sure you save your .Rmd file and workspace so you can continue in your next project session.

Assignment 5: Classification (Johan Westerhuis, 19-01-2023)

In this assignment we will use the LDA and PCDA methods to find the metabolites that best discriminate between your two chosen conditions.

A) Make a reduced metabolomics dataset by selecting only the samples of your two chosen conditions. If one of your conditions is "21-L", remove the 21-L sample of timepoint 0 to balance your groups. Create a vector of binary values encoding the condition of each sample (0 = condition1 and 1 = condition2). (Hint: consider recentering and/or rescaling the reduced dataset.) [**5 points**]

Put any text you may want to type here.

End of text block.

B) Perform a linear discriminant analysis on your new metabolomics dataset, using your new vector of binary values as the class of each sample. [**2 points**]

Put any text you may want to type here.

End of text block.

C) You should have gotten a warning from your LDA function stating that the variables in the reduced dataset are collinear. What does this warning mean? Do you think you have created a valid model? If so, analyse it for important metabolites. If not, explain why the model is not valid. [**6 points**]

Put any text you may want to type here.

End of text block.

D) Make a principal component analysis of your reduced metabolomics dataset. Only consider 2 components. Show the amount of variance explained per component in the plot labels. Add a legend to identify the colour for each condition. Does your PCA model separate your two chosen conditions well? (Hint: use svd(). Consider recentering and/or rescaling your data for PCA specifically!) [**7 points**]

Put any text you may want to type here.

End of text block.

E) Make a PCDA model using the PCA scores (from question D) for the discriminant analysis. Inspect the classifications of your samples. Does the model misclassify any of the samples? What is the most important metabolite that differentiates your two chosen conditions? [**9 points**]

Put any text you may want to type here.

End of text block.

F) Compare your metabolomics PCDA result with your metabolomics PCA result (from assignment 3C). Do you find the same important metabolites? If you find any differences, why would the selected metabolites be different? [**5 points**]

Put any text you may want to type here.

End of text block.

This concludes Assignment 5. Make sure you save your .Rmd file and workspace so you can continue in your next project session.

Assignment 6: ASCA (Age Smilde, 20-01-2023)

In this assignment we want you to use the shiny app for ASCA with the Caldana et al. metabolomics data. To answer the questions below, just make a screenshot of your plot window and include it into the markdown as follows (see the .Rmd):

Open up the shiny app for ASCA using the DataTool.R script in the /Shiny App/ folder. From the /Data/ folder, load as your dataset the file "Data_for_ASCA.csv" and as your design file "Design_for_ASCA.csv". The Data_for_ASCA.csv file contains untransformed metabolomics data. We have selected only the 21-D, 21-L, 32-D, 32-L, 4-D and 4-L conditions for you to use in this assignment. There are 4 replicates present for every condition-timepoint combination. The Design_for_ASCA.csv file contains the time, temperature and light condition metadata for every sample. These are encoded as factors 1, 2 and 3, respectively.

A) Consider balancing, transforming, centering and/or scaling the data in the shiny app. Defend your choices. Inspect the PCA plots and change the point colours to light, time or temperature. Does the PCA model separate the different groups well? Which metabolites are important for this separation? Include screenshots of your score plot and biplot. [**7 points**]

Include your screenshots here.

End of screenshot block.

Put any text you may want to type here.

End of text block.

B) Go to the univariate plot window. Plot the metabolites that were important according to your metabolomics PCDA result from Assignment 5. Why do you think these metabolites were selected? Include a screenshot of your univariate plots. [**4 points**]

Include your screenshots here.

End of screenshot block.

Put any text you may want to type here.

End of text block.

C) In the ASCA window of the shiny app, create a model using the factors time, light, and temperature. Do not consider interaction terms yet. Examine the model. Which metabolites are important in separating the different temperature conditions? Which metabolites are important in separating the different light conditions? Have you found these metabolites before using other methods? Include screenshots of your score plots and biplots. [**3 points**]

Include your screenshots here.

End of screenshot block.

Put any text you may want to type here.

End of text block.

D) Use the "Combine terms" text field to create a new ASCA model considering a light factor term and an interaction term of light and temperature. Examine the model in the Combination plots window. Have a look at the levels plot. What is the overall behaviour of the metabolites between the different temperature conditions? Is the overall behaviour different for the light and dark samples? Include a screenshot of your levels plot. [**5 points**]

Include your screenshots here.

End of screenshot block.

Put any text you may want to type here.

End of text block.

E) In the ASCA window of the shiny app, use the "Combine terms" text field to create a new ASCA model considering a time factor term and an interaction term of time and light. Inspect the levels plot of the model. What is the overall behaviour of the metabolites over time? Is that behaviour different for the different temperature conditions? Include a screenshot of your levels plot. [**5 points**]

Include your screenshots here.

End of screenshot block.

Put any text you may want to type here.

End of text block.

F) Compare all of your "most important" metabolites from all of the methods you have applied. What are the similarities? What are the differences? What is causing some methods to produce different results? Which results are the most valid, in your opinion? Which results make the most biological sense? [**9 points**]

Put any text you may want to type here.

End of text block.

That concludes week 2 of the project! Please hand in your .rmd and .pdf files on Canvas for your project grade. The week 2 deadline is Monday, January 20th 11:00.