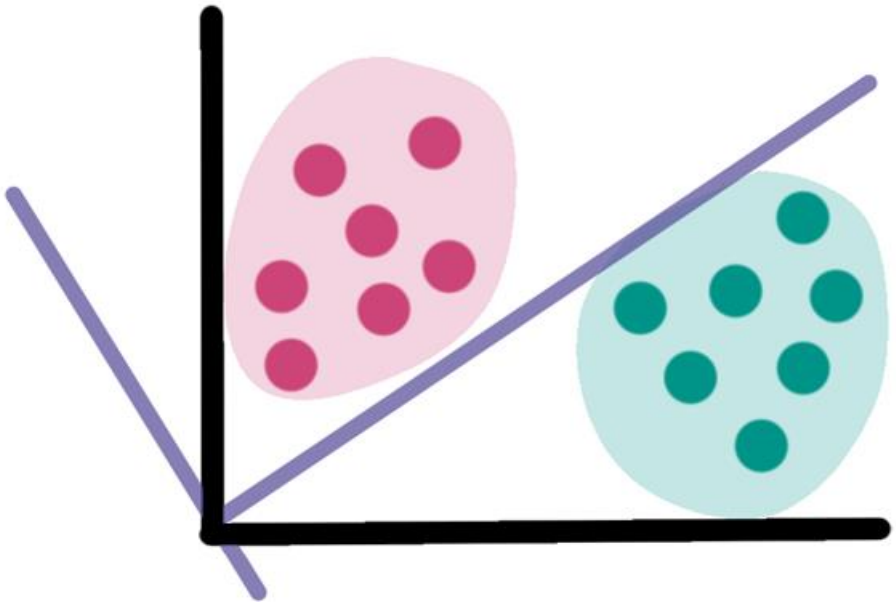
Implementing linear discriminant analysis to identify core in a murine autism model.



By Paul Christian van der Zalm

TU Delft and Erasmus University Rotterdam
Student numbers:
4541979 (TU Delft)
445251 (EUR)
February 2020- May 2021

Supervised by Aleksandra Badura and Lucas Wahl

Table of contents

Introduction

Behaviour data, commonly referred to as "the phenotype", can give us insight into the genotype of an individual [1]. Every individual acts in different ways and there will always be some variation. But if we use a framework, such as existing personality tests for humans, we can distinguish different patterns. In absence of such a framework it is hard to recognise patterns and to define boundaries between different behaviours.

Behaviour analysis on non-human species

When working with non-human species, such as mice, there is no consensus on the methods to define behaviour and existing methods are prone to bias of human social projection; the projection of human mannerisms on non-human behaviour. [1].

When working with basal behaviours and commonly known behaviour types, a framework can help to define and analyse the behavioural characteristics in a generic way. But this gets harder when dealing with genetic disorders that affect the behaviour on multiple aspects or in a subtle way. An example of such a disorder is Autistic spectrum Disorder (ASD).

Autism Spectrum Disorder (ASD)

Autism Spectrum Disorder is a neurological complex condition that affects an array of different developmental aspects [2]. The symptoms of ASD can involve problems in social interaction, verbal and non-verbal communication and restricted or repetitive behaviour. The actual symptoms and also the severity of these symptoms differ per individual, which makes this a very complex disorder to analyse. Genetic research can help with the diagnosis and further understanding of such a versatile disorder. An example of such a genetic research is that studies have shown that an overexpression of the PI3K/AKT/mTOR pathway is linked to neuronal disorders [3]. Although a lot of studies focus on the functional role of mTOR [4], few studies have researched the specific functional role of PI3K, while it has been shown that a mutation in the PI3Kδ gene, over activates the downstream AKT/mTOR pathway [5]. Furthermore, PI3Kδ has been linked to regulation of soma size and dendritic complexity, which indicates a role in neuronal morphology [6].

PI3Kδ

The PI3Kδ gene is involved in the immune response. This gene encodes a heterodimer (protein complex of two different proteins) consisting of a regulatory protein (p85α, p85β, p55γ, p101 or p84) and a catalytic protein, which can be one of the following four: p110α, p110β p110γ or p110δ [10]. Mutations in the PI3Kδ gene are most commonly caused by a deletion in the regulatory subunit p85α or an E1021K substitution in the p110δ catalytic subunit. This last mutation is caused by a substitution of Lysine (K) to Glutamic Acid (E). This mutation is linked to not only immune deficiency, but also impairment of dendritic morphogenesis and increased dendritic spine density, which indicates effects on neurological development [6]. We conducted multiple behavioural experiments on mice with a heterozygous PI3Kδ gain of function mutation ($p110\delta^{E1020K}$ mice) and compared that to a control group, to see what functional role PI3Kδ has in behaviour and whether this murine model shows autism-like features.

Behavioural analysis and data science

To know what aspects of mouse behaviour are characteristic for the PI3Kδ gain of function mutation, one can measure a lot of different variables brute force, like performing many tests for a specific feature, as done in the paper of Ey et al. [31]. The problem of such an approach

is that this creates a lot of data in which a majority is not useful or significant. There is so called too high of a "dimensionality" of the data to properly analyse.
In that way behaviour analysis is following the same trend as many other scientific data fields: the amount of acquisitional data grows faster than the capability of data storage and processing [7]. To do behaviour analysis one needs a way to filter out data that do not contribute to the characterization of normal behaviour and by extension of behavioural deficits in murine models for human disorders. To extrapolate this data, data analysis is needed, but the implementation of data analysis on behaviour data is still in an early stage [8].

Discriminant analysis algorithms

A way of extrapolating the data and finding the most correlating phenotypic features, is by the use of discriminant analysis tools. Many different algorithms exist, but in this thesis I focussed on two different discriminant algorithms: Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA). Both types of analyses belong to a different category.
PCA is an example of an unsupervised classification, where the class separation of the data is not labelled beforehand and the method is based on maximizing the separation of the whole data by the general variance.
LDA is a form of supervised classification, where the data is labelled into multiple pre-determined classes and the method tries to optimize separation following this classification. By combining and comparing the outcome of these two classification methods we can get a better insight into the general structure of the data. Before we dive into the implementation we first will have a look at the mathematical idea of these algorithms.

PCA definition

Principal Component Analysis (PCA) is based on the following separation algorithm: the whole dataset is shifted by relocating the overall mean of the dataset to the origin [12]. After that, separation is found by minimizing the distance from the data points to a rotating line through the origin [12]. The number of vectors that contribute to the separation is equal to the number of variables and every next vector, or Principal Component (PC), is perpendicular to the previous set of PC's *(see Supplementary Figure 1)* . In contrast to the LDA, PCA does not define different classes based on the minimization of this projection, but only separates the dataset itself as much as possible, making it an unbiased separation algorithm. It is important to normalize the data before implementing the algorithm to prevent overrepresentation of certain variables, as different scales can influence the variability [9].

LDA definition

LDA on multiple classes

Linear Discriminant Analysis is based on an eigenvector optimization problem. This problem tries to optimize a set of eigenvectors that separates multiple classes from each other. This separation is based on the minimization of the within variances per class and the variances between the classes **[1].** The intuitive formula for 2 classes would be:

$$Set\ of\ eigenvector \propto \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2}$$

When dealing with multiple classes the within variances can be represented with a within scatter matrix, which also can be seen as the average covariance matrix.
In the formula:

$$S_W = \sum_{i=1}^{k} \sum_{j=1}^{N_i} (x_{ij} - \overline{x_i})(x_{ij} - \overline{x_i})^T \quad [\mathbf{14}]$$

Where k is the amount of classes.

The variances between the classes can be defined by using the between scatter matrix, which is the mean of the classes minus the overall mean of the data, multiplied by its transpose:

$$S_B = \sum_{i=1}^{k} N_i (\overline{x_i} - \bar{x})(\overline{x_i} - \bar{x})^T \quad [\mathbf{14}]$$

This gives the following relation:

$$Set\ of\ eigenvector \propto \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2} = \frac{|S_B|}{|S_W|} = S_W^{-1} S_B \quad [\mathbf{15}]$$

Now we want to find a set of directional vectors, that correspond to this relation. For this we solve the eigenvector problem:

$$S_B \theta = \lambda S_W \theta$$

Where θ resembles the direction of the eigenvectors. This concludes the relation to be:

$$J(\theta) = \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2} = \frac{\theta^T S_B \theta}{\theta^T S_W \theta} \quad [\mathbf{15}]$$

J(θ) is the set of eigenvectors, where the eigenvector with a higher eigenvalue, has a higher separability factor.
Every next eigenvector is perpendicular on the previous set of eigenvectors. Moreover, is it a linear combination of the variables used in the dataset, as every component of every vector correlates to the directional contribution into the space of that specific variable. The number of eigenvectors corresponds to the number of variables one uses. To have a better understanding of the significance of these eigenvectors we take the example of only using 2 classes and 2 variables or features (**Figure 1**).
By projecting the data on these eigenvectors, 2 vectors are created that have specific information about the variances and distances of the classes. These eigenvectors are then sorted by their corresponding eigenvalues, as a higher eigenvalue corresponds to a higher separability power of the eigenvectors for the different classes. By combining the two vectors with the highest eigenvalues in a new 2D plot you get the plot with the best separation value, which in this case with only 2 features is just a simple rotation of the original plot. LDA is mostly useful when

dealing with datasets with multiple variables, as this method can be extended to higher dimensions. In that case, as only the first 2 LDs give the most information, every dataset with high dimensionality can be reduced to a 2D plot.

The above derived formula of $J(\theta)$ can be used for data that contains more than two classes. This is because with more than 2 classes there are multiple differences between the means that can form a between scatter matrix (S_B) containing, i.e. $\mu_1 - \mu_2$ and $\mu_2 - \mu_3$, instead of only a vector, i.e. $\mu_1 - \mu_2$. In our analysis we only dealt with 2 classes. Therefore, we will now discuss this special case of LDA on 2 classes.

Special case: LDA on 2 classes

When dealing with 2 classes, the between scatter matrix can be written as:

$$S_B\theta = \lambda S_W\theta \propto (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T\theta \propto (\mu_1 - \mu_2) \text{ [15]}$$

As the direction of the mean differences only contains the vector from the mean of class 1 to class 2. This makes it that the eigenproblem changes to the following formula:

$$\theta = EIG[S_W^{-1}S_B] = EIG[S_W^{-1}(\mu_1 - \mu_2)]$$

It needs to be mentioned that if the classes are clustered in a spherical shape, the direction of the projection is always in the same direction as the argument: $S_w^{-1}(\mu_1 - \mu_2)$ [15]. This means that instead of computing all the eigenvectors, the direction also can be calculated by using the argument itself: $\theta = S_w^{-1}(\mu_1 - \mu_2)$

The problem with this is that the assumption that behavioural data is always spherically shaped is not a right one to make, as behavioural data can be very varied. Thus, if the data is not spherical, the best projection will not be precisely parallel to the mean difference vector (Figure 1c).

Dimensionality threshold

Until now we assumed that the scatter within matrix is invertible ($\theta = EIG[S_w^{-1}(\mu_1 - \mu_2)]$). This is only the case if the matrix is non-singular, which means that the amount of variables is equal or less than the number of samples minus the number of classes [9][22][23]:

$$d \leq n - c$$

When dealing with more variables than this threshold, some measurements need to be taken to successfully implement LDA. Every used variable creates a new degree of freedom, giving the eigenvector that tries to optimize the separation more directions to find the perfect separation [24]. When using too many variables the LDA will always find a perfect separation, even when random data is used. When the number of variables is greater than the threshold, the separated data will get projected on external dimensions that are higher than the sample size, giving unnatural looking point clusters [24].

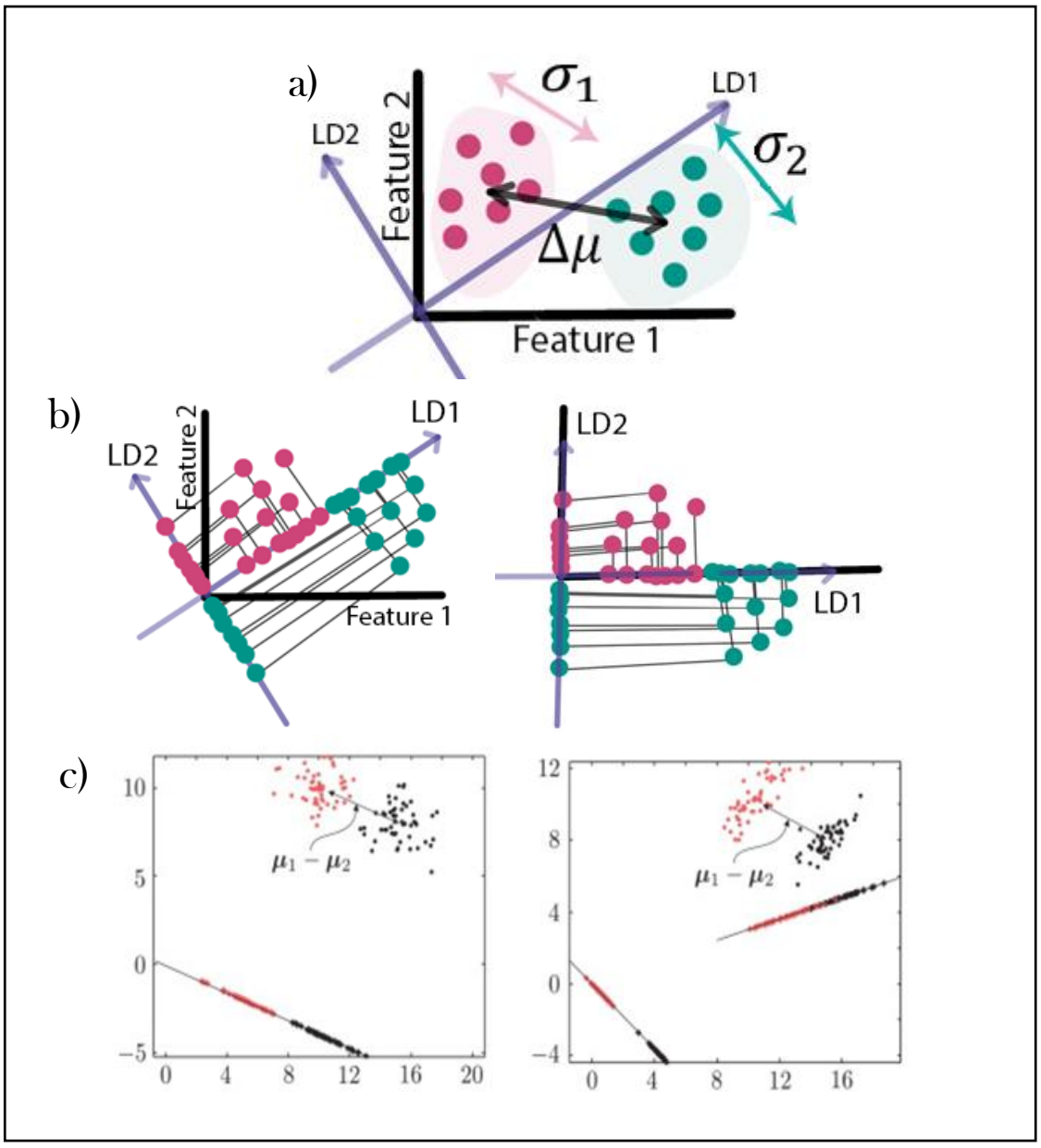


Figure 1 a) *A visual representation of the optimisation problem of the LDA algorithm, where LD1 and LD2 are the separating vectors.* σ_1 *and* σ_2 *are the within variances of each class.* $\Delta\mu$ *is the in between variance or the distance between the mean points of the classes.* **b)** *An visualisation of the LDA projection with two features. Because the data only has 2 features, the LDA projection here is just a rotation.* **c)** *The effect of the shape of the clusters on the angle of the projections; when the clusters aren't spherical, the projection is not parallel to the mean-difference vector. Adaptation from* https://www.sciencedirect.com/topics/engineering/class-scatter-matrix **[15]**

Goal of this project

By comparing the genotype of a group of samples with the outcome of a LDA, we can correlate different behaviours with a genotype and consequently with a disorder.
This way of combining genotype and behaviour data can be applicable in many different ways, but this thesis is a method case study of LDA implementation on data of different experiments on mice with a PI3Kδ gain of function mutation to get a better picture of the behaviour correlated with ASD. The goal is to develop a tool in Matlab that can be used on a wide scale of datasets as addition for further behavioural research.

Methods

Mice used

For the experiments we used 15 wildtype mice and 15 mice that had a heterozygous PI3Kδ gain of function mutation, caused by a E1020K substitution in all cells (p110δ^{E1020K} mice), which is an analogue to the E1021K mutation in humans. All mice were male and between 10 to 13 weeks old at arrival. Before the experiments started the mice were acclimated for two weeks and grouped in cages of 3 to 4 mice per cage. The experimenters were blinded to the genotype of the individual mice to prevent biased interpretations during experiments. 3 days before the experiments started, the mice were handled by the experimenters to get the mice accustomed to human interaction. Before each experiment the mice were weighted. After all the experiments were done the brain tissue of each mouse was collected. The mice were euthanised following animal friendly guidelines.

Behaviour experiments

Most behaviour experiments were done in a behaviour box, except for the Erasmus ladder, Rotarod and Y-maze. This is a 130x80x80 cm wooden box with standardized white and infrared light. All experiments were filmed by a fixed camera (acA 1300-600gm, Basler AG) above the experiment area. The recordings were operated by the open-source software Bonsai (https://bonsai-rx.org). After each recording the videos were uploaded to the OptiMouse software [30] where each mouse was tracked.

Y-maze (YM)

The Y-maze is a white-opaque acrylic box with 3 arms of the same length (20x32x9cm), resembling an Y. This apparatus is placed in a dark chamber with a dark curtain on one side for the experimenter to open and close to get the mouse after each trial. Each arm is filled with water. The goal of the experiment is to analyse the learning pattern of the mice by looking at how fast they can learn and unlearn certain concepts, in other words their cognitive flexibility. After a day of habitation wherein the mice have the opportunity to just swim in the water to accustom to the box, the acquisition experiment takes place. This is done by placing the mouse at the central arm and hiding a platform in one of the two remaining arms. The hiding is done by colouring the water with animal friendly paint, (Basic color, 21 white 30081, Creall®), making the water opaque. The mouse has to find platform to get out of the water, which it has 1 minute to achieve. After the minute has passed, the mouse gets taken out of the box and after drying and a short break, it is placed in the box again. The location of the platform remains the same for each mouse over all the sessions, with the goal that at the end of the experiment, the mouse has learned the location of the platform, by directly swimming to the correct arm. After acquisition day the location of the platform switches to the other arm, forcing the mouse to unlearn the previous location of the platform and to learn the new location. After this reverse day, a second reverse day takes place, switching the platform for a second time, to see again if the mice can unlearn the previous location of the platform. During each trial the mouse is recorded at 30 fps by a fixed camera (Sony PS3 Eye) mounted above the Y-maze and the recording is operated through the open-source software Bonsai (https://bonsai-rx.org). This experiment looks at the lack of cognitive flexibility, which is a known symptom of ASD.

Variable	Description
total distance	The total distance the mouse swims
average speed	The average speed of the mouse
total correct choices	The total amount of times the mouse chooses the correct arm
total platform not achieved	The total amount of time the mouse does not achieve the platform under 60 seconds.
Acquisition (aq)	The total number of times the mouse chose the right arm during the first acquisition
Reverse session 1 (rev1)	The total number of times the mouse chose the right arm in the first reverse session
Reverse session 2 (rev2)	The total number of times the mouse chose the right arm in the second reverse session

Table 1 *The measured variables during the Y-maze experiment.*

Grooming (G)

The mice are placed in a box (30x30x39 cm white-opaque Perspex® arena) for 30 minutes and recorded.(powered by the open-source software BORIS at 30 fps) The recordings are then analysed to look at the grooming behaviour of the mice. Grooming can be classified as obsessive-compulsive behaviour, which is a known phenotype of ASD.

Variable	Description
Grooming time	The total amount the mouse grooms
Rearing bouts	The number bouts the mouse is rearing
Grooming bouts	The number bouts the mouse grooms
% long bouts	The percentage of long bouts relative to the total grooming bouts

Table 2 *The measured variables during the Grooming experiment.*

Marble burying (M)

A mouse is placed in a 26.6 x 42.5 x 18.5 cm cage with 20 marbles under light conditions; 5 rows of 4 marbles. After a certain amount of time the total amount of area under the flooring over all marbles are counted as a measure of the amount of burying. Marble burying can be classified as obsessive-compulsive behaviour, which is a known phenotype of ASD.

Variable	Description
Total area buried	The sum of the buried area of each marble after the session

Table 3 *The measured variables during the Marble burying experiment.*

Open Field (OF)

The Open Field experiment measures the general locomotion and behaviour of a mouse when placed in an empty box under light conditions. A mouse is put in a 50x50x35 cm white-opaque arena in the behavioural box for 30 minutes. It gets filmed from above to, for example, determine how much time the mouse spends in particular areas in the box. The open field looks at anxiety induced behaviour.

Variable	Description
Average speed	The average speed of the mouse
Total distance	The total travelled distance of the mouse
Time in outside	The time the mouse spends in the middle of the area
Total transitions	The amount of times the mouse

Table 4 *The measured variables during the Open Field experiment.*

Social-Interaction (SI)

The setup of the Social-Interaction experiments consists of a 63 x 42.5 x 21 cm transparent acrylic arena with 3 chambers. After habitation while all the areas are empty, two cups are placed in the two side chambers, one empty cup and one cup with another novel mouse of the same age. The ratio of investigating the different cups or being in the central chamber can give an indication on the social ability of a mouse, which is an altered phenotype of ASD.

Variable	Description
Average speed BL	The average speed when the mouse could explore the whole apparatus without a second mouse present (BaseLine)
Average speed test	The average speed during the test
Total distance BL	The total travelled distance of the mouse when the mouse could explore the whole apparatus without a second mouse present (BaseLine)
Total distance test	The total travelled distance of the mouse during the test
Social cup preference	The ratio between the time the mouse touches the cup with the animal inside and the empty cup
Social investigation preference	The time spent with a partner (nose in cup/total time * 100)
Transitions BL	The transitions between the chambers divided by the time in the mouse chamber, when the mouse could explore the whole apparatus without a second mouse present (BaseLine)
Transitions test	The transitions between the chambers divided by the time in the mouse chamber, during the test.
Ratio transitions cups test	The ratio between the transitions of the empty cup to the cup with an animal inside.

Table 5 *The measured variables during the Social Interaction experiment.*

Erasmus Ladder (EL)

The Erasmus Ladder (Noldus, Wageningen, the Netherlands) is a horizontal ladder with 37 high and low steps between 2 boxes at each end. The mouse gets placed in the starting box. After a 9-11s waiting period, a light turns on and an air puff stimulates the mouse to leave the box and pass over the ladder to the second box. The behaviour of the mouse and the moment it leaves the box with respect to the different events (light stimulation etc.) gets recorded, but also the amount of backsteps and jumps are measured. There are in total 5 days of experiments, with 42 trials per day.

Variable	Description
Trial duration	The average duration of a trial
Time on ladder	The average time the mouse spends on the ladder

Leaving before	The amount of times the mouse leaves the box before the cue
Light air ratio	The ratio between leaving on the light cue and the air cue
Short steps	The amount of short steps (only one step) the mouse takes
Jumps	The amount of jumps the mouse takes (multiple steps)
missteps	The amount of times the mouse misses the step

Table 6 *The measured variables during the Erasmus Ladder experiment.*

Elevated plus maze (EPM)

The Elevated Plus Maze consists of a walking area with 4 perpendicular arms with a 29.5 x 28 8.5cm white-opaque acrylic base; each arm supported by a 30x3cm cylindrical pole. 2 of them are open to a wider space and 2 of them are closed off by a 20cm high black-opaque acrylic wall. At the beginning of the experiment, the mouse is placed in the centre of the cross-section of the 4 arms and during 10 minutes the location of the mouse is analysed; Particularly the ratio of being in the closed off areas and the open areas could give an indication of anxiety-like behaviour.

Variable	Description
Average speed	The average speed
Total distance	The total travelled distance
Time in closed arms	The time that the mouse spends in the closed arms
Total transitions	The amount of times the mouse transits between the open arms and closed arms

Table 7 *The measured variables during the Elevated Plus Maze experiment.*

Rotarod (R)

The Rotarod is a rotating rot that procedurally gets faster. The first four days the speed of the rotating rot accelerates to 40 rpm. The acceleration increases to 80 rpm on the fifth day which is the last day. The time the mouse stays on the rotating rod gets recorded and gives an indication on the motor abilities of the mouse. The duration of the session is defined by the mouse falling of the rod, grabbing the moving rod performing a 360º turn without actually walking on the rod or if the maximum time of 300 seconds is reached.

Variable	Description
Rotarod	The amount of time the mouse spends on the rotarod (the slope of the learning curve is taken for the analysis)

Table 8 *The measured variables during the Rotarod experiment.*

Pre-processing of the data

Requirements for the data format and data distribution

The coded LDA accepts a dataset containing 2 different classes. The rows of the dataset are the individual samples, ordered in genotype class, and the columns are the values for the different variables of these samples. The class labels are located at the first column and the feature names at the first row, but because this first step of formatting is relatively easy to do manually per dataset, the format of the imported file does not really matter; As long as for the input for the

code the data matrix is formatted as mentioned above, there is a vector with the class labels and a vector with the names of the different features.
The data is preferred to not be a mixture of continuous and discrete valued features [16], but as long as the distribution of these features resemble or approximate a normal distribution it can be used with caution. This is due to the fact that for analyses used in the code, the data first needs to be normalized, to scale the whole data, and for now this is done by the z-score. The normalisation step is needed as the data that is used for LDA must have similar variance as different scales can influence the outcome of the LDA; variables with a much higher variance contribute more than variables with a lower variance [9].
The distributions of the variables, of the independent classes as well as of the whole feature, are plotted when the code is run, by using the normal probability plot function of Matlab [17], to check visually if the data approximates a normal distribution. If this is not the case at all and the feature diverges too much, this variable will be excluded from the analysis.

Pre-processing: outlier removal

The first step of pre-processing implies investigation of the data and looking for any extreme value cases. This was an important step, because faulty measurements can give rise to non-representable values that influence the analyses of the data [18]. It was crucial to make a distinction on the different definitions of an outlier, especially as we worked with different classes: an individual can be an outlier of the overall data or an outlier of its associated class. We chose to take the definition of an outlier of its class, because clustering characterizes a class and extreme outliers can illegitimately influence its variance.
There is an argument to be made that outliers can be just a characterization of the behaviour of a certain class and by blindly removing outliers, this feature can be lost. Therefore, it was necessary to have a good base definition for an outlier. We used the base definition of Matlab. This is the following: if a value is more than 3 times scaled MAD (median absolute deviations) away from the median, it is an outlier [19].
The MAD is defined as:

$$MAD = c * median(|A - median(A)|) \qquad where\ c = -\left(\sqrt{2} * erfcinv\left(\frac{3}{2}\right)\right)^{-1}$$ [19]

Erfcinv() is the Inverse complementary error function

Pre-processing: normalization: 2 ways

We made a distinction between the 2 ways of normalizing, both with a different goal, to prevent confusion: normalization over the variable as a whole and normalization over the variable per class. To see if the variables are indeed normally distributed, we z-scored the variables as a whole and compare it to a standard normal distribution.
We also checked if the classes themselves were approximately normally distributed as the Matlab definition of an outlier is based on a normal distributed dataset.
The normalization we used to implement in the LDA and PCA was over the whole data and not per class. This is because if both classes would be normalized separately, there would be no difference in the mean and variance between the classes, negating all useful information.

Visualizing outliers and distributions

In addition to the normal probability plots from the Matlab function, we made a custom function that plots a cumulative distribution function per variable in a non-squashed space. This was done by ordering the z-scored data from minimum to maximum. Every value was labelled with the corresponding index of its class, to further compare samples in different variables. The defined outliers that would get removed were coloured in pink. The black dashed line represented the standard normal distribution, so the z-scored data would approximate this line in the ideal case.

These custom distribution plot per variable were created for both classes to view the outliers and see if the classes were approximately normally distributed. This was also done for both classes after the removal of their outliers to see the effect of the filtering.

Pre-processing: outlier interpolation and z-score

After visualisation, interpretation and eventual removal of variables, we used the z-scores distributions of the classes to find the indices of the outliers per class; creating a sparce matrix for every class, of in general zeros, with on the index of an outlier an '1'. These index matrices were used to remove the outliers from the original data and setting those to "Not a Number" (NaN). Subsequently, this data, now containing "NaN" values, were z-scored over the whole variable by using the mean and variance; both only using the "non-NaN" values. The result was a z-scored matrix containing both classes, with "NaN" values for the outliers of the two classes. Next we implemented an interpolation method with the mean of the class per variable to substitute the "NaN" values of the matrix.

Pipeline pre-processing

It is important to note that when talking about z-scoring the data, we made a distinction in using it to visualise the distributions of the data or altering the data itself; The data only gets normalized at the end of the process over the whole variable, so the 2 classes combined. Figure 2 gives a schematic overview of the whole pre-process pipeline before inputting the classes into the LDA.

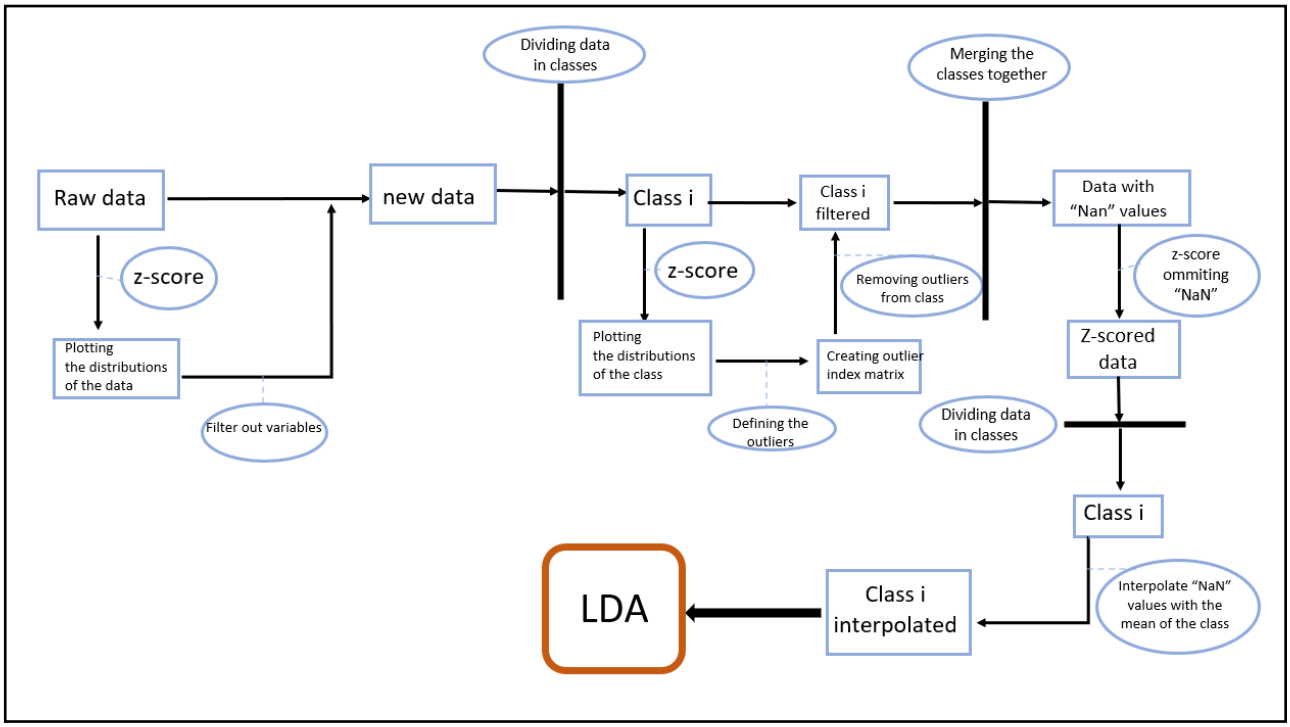


Figure 2 *The pipeline to pre-process the data for the LDA, where 'i' is the amount of classes the data contains. This pipeline stays the same ones the code is adapted to handle multiple classes as this process is repeated for every class. The rectangles are processed data, ovals are processing steps.*

LDA methods

For the LDA analysis we used a custom-written Matlab code that performed a two class LDA. As mentioned in the introduction, the substitution of the eigenvector with its argument is only possible if the data is spherical. This is not always the case with behavioural data.
For this reason, we decided to use the eigenvector method: $\theta = EIG[S_w^{-1}(\mu_1 - \mu_2)]$
Another benefit of this method is that the eigenvalues give more insight into the separability contributions of the individual eigenvectors. We used the Moore-Penrose pseudo inverse instead of the normal inverse to solve the high dimensionality problem, as this approximates the inverse of uninvertible matrices. Moreover, to validate the outcome of the Moore-Penrose pseudo inverse method, we compared it to the outcome of the PCA and a scoring method, based on the parameter used for the LDA: $score = \frac{|\mu_1 - \mu_2|}{{\sigma_1}^2 + {\sigma_2}^2}$.

Validation of the LDA outcome by shuffling

As stated above, the variable contributions of every LD are the directional components for the projection vectors that separate the classes the best.
The order of the LDs, should therefore be unique for our genotype classification, most importantly the ordering of LD1 as this is the most separating one.
To validate this uniqueness, we shuffled the labels of our datasets to separate our data in two equally sized random classes that differ from the real genotype classification. Subsequently we did LDA on this random classification and compared the outcoming LD1 order with our original LD1 order.
We repeated this process multiple times, creating a probability distribution per variable of their rank placement on an optimal separation vector.
This distribution quantified the shape of the variable, which indicated in what sense the variable was separable into 2 classes, relative to the other variables. We shuffled the data 200.000 times as this reduced the error margin below 0.05.

Results

Multi-point variables

Data that comes from multiple measurements within a single test, such as experiments that measure the progress over multiple days, needed to be to be reduced to a single value if we wanted to use it for the LDA analysis. We chose to use the interpolated linear slope value for the multi-valued variables, because all the experiments had the goal of quantifying the performance or learning curve. From now on class 1 contains the PI3Kδ gain of function mice and class 2 the wildtype.

Visualising the data

Firstly, we needed to visualize the data itself to know which variables are useable and which could have a potential negative influence on the following analysis. Because the variables will be z-scored and pre-processed, the data should approximate a normal distribution or at least be distributed in a not too extreme manner. Given the distribution of "YM: rev2" **(Figure 4b)** with almost all the points being 0 and the only points that are deviating being from class 2, this variable would have given potential problems in the separation process; The values that are deviating are small, but after normalizing the data, it would have influenced the separation significantly, while all the data points are virtually the same. Therefore, we excluded this variable from the analysis.

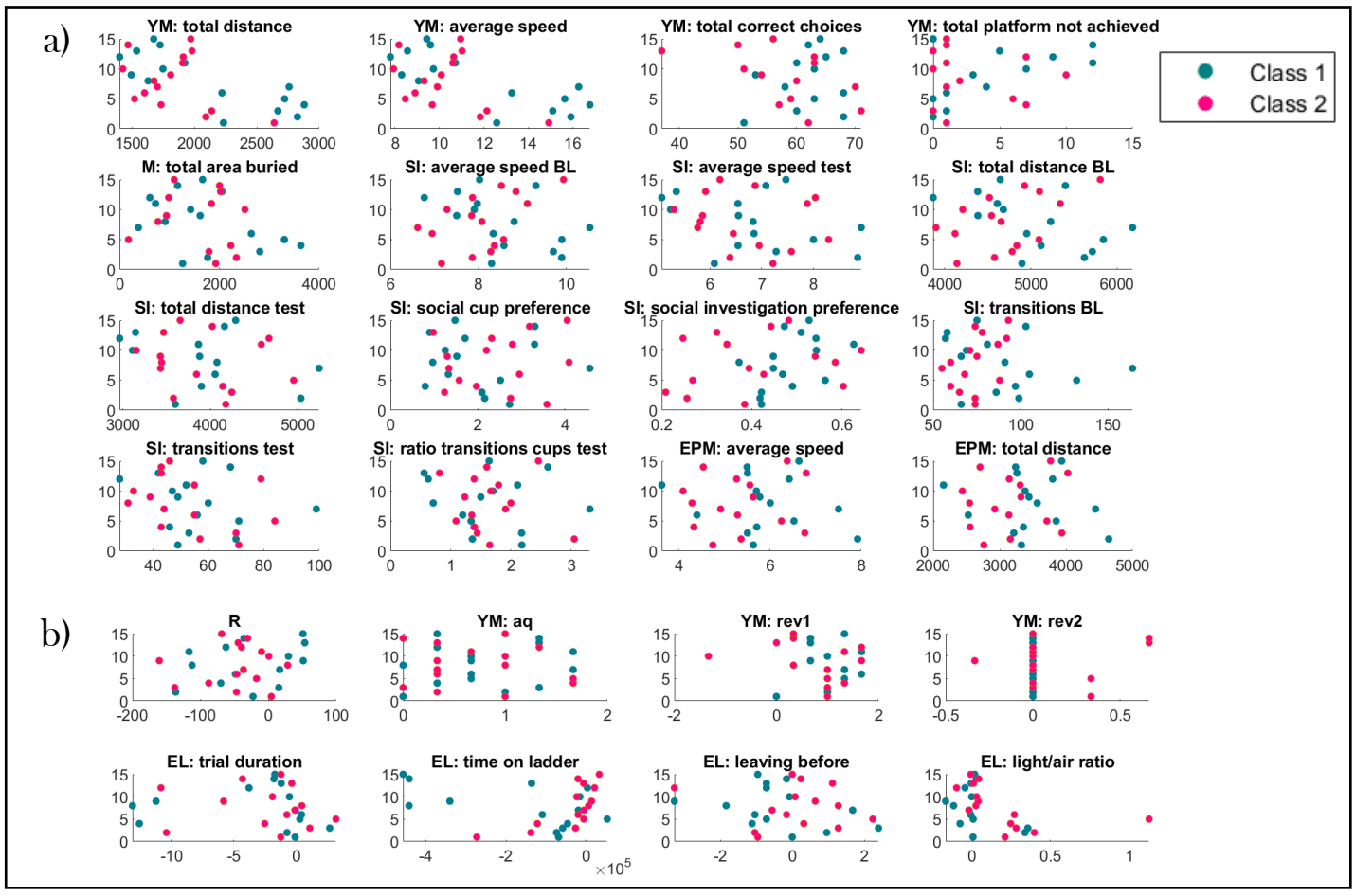


Figure 4 a) *The scatterplot of a subset of the data from the single-point measurements. The experiment abbreviations stand for: M: marble burying, SI: social interaction, EPM: elevated plus maze, G: Grooming, OF: Open Field.* **b)** *The scatterplot of a subset of the data from the multi-point measurements. The distribution of 'YM: rev2' shows too much discrepancies and will be removed from the analysis.*

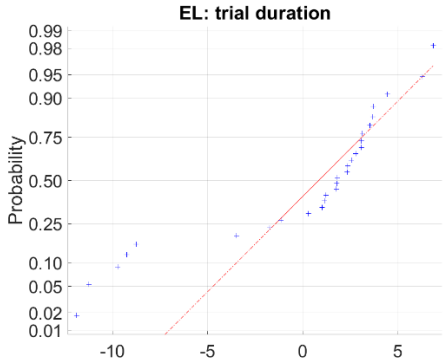
Variable and class distributions

Like already discussed in the methods, we needed to examine the distributions of the variables to see if we could normalize the data, as we used the z-score for the LDA. As example we can see in Figure 5a, the variable "EL: trial duration" seems to have deviating points from the standard distribution. These points will be most likely be filtered out of the analysis and as the rest of the data seem to still approximates the standard normal distribution line, we chose to keep this variable into the analysis. In Figure 5b the variable "G: Grooming time" approximates the red line of the normal distribution well with a small amount of deviating points.

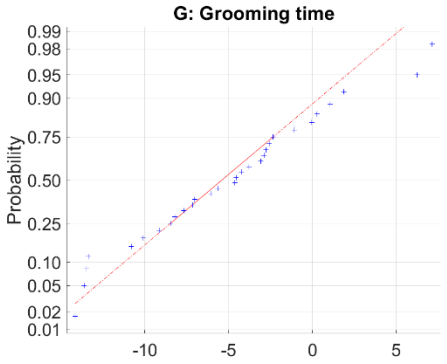
The next step was to see if the all the variables approximate a normal distribution for both classes, given that outlier filtering was based on that the classes approximate a normal distribution. To make the class and variable distributions examples consistent, the same two variables are displayed in Figure 5c-f as the examples of the normal probability plots in Figure 5a-b. The difference being that now the distributions are over the individual classes. For this we used a custom distribution plot. All other distributions can be found in the supplementary ***(Supplementary Figure 3a-f).*** In Figure 5c-d, the class distributions before and after the outlier removal are displayed. It is clear that the majority of the distribution in Figure 5c approximates the standard distribution line and that the outliers that will be removed, marked in pink, are indeed far away. In Figure 5e it looks like the distribution is more stretched, but as can be seen in Figure 5f, does the distribution look much better after outlier removal. In the distribution plots after the outlier removal (**Figure 5d** and **5f**) we can see the new outliers that will be removed in pink if we would choose to do outlier removal again. Which we concluded to not be necessary.

In Figure 5g are the outliers plotted that we removed from each class for all variables. The variables "EPM: total distance" and "EL: time on ladder" of class 1 and "YM: total platforms not achieved" and "Grooming time" of class 2 all have a total amount of outliers of 4, which is 37.5% of both class sizes. This is a level 4 outlier percentage following the definitions in the paper of *A. J. Tallón-Ballesteros and J. C. Riquelme (2014)* **[26].** This paper states 4 levels of outlier percentages depending on the kind of data. These are defined as follows: level 1 (0%-10%), level 2(10%-20%), level 3 (20%-30%) and level 4 (30%-40%). As behavioural data is prone to outliers, it was expected to have a higher level of outlier percentage.

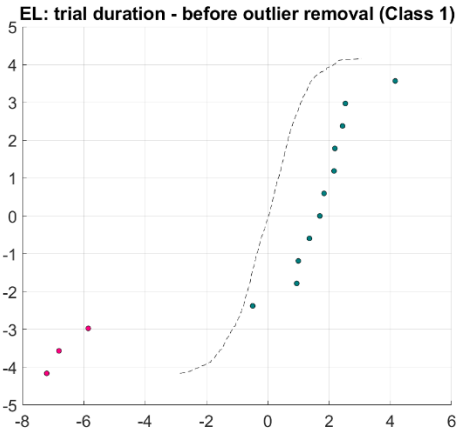
a)


EL: trial duration
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-10
-5
0
5

b)


G: Grooming time
Probability
0.99
0.98
0.95
0.90
0.75
0.50
0.25
0.10
0.05
0.02
0.01
-10
-5
0
5

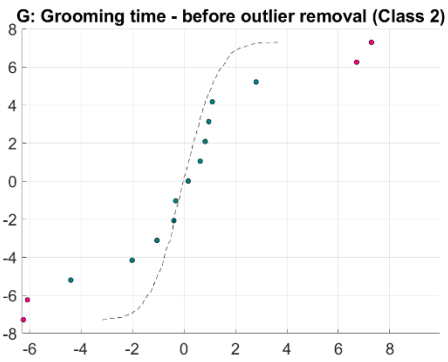
c)


EL: trial duration - before outlier removal (Class 1)

d)


EL: trial duration - after outlier removal (Class 1)

e)


G: Grooming time - before outlier removal (Class 2)

f)


G: Grooming time - after outlier removal (Class 2)

g)

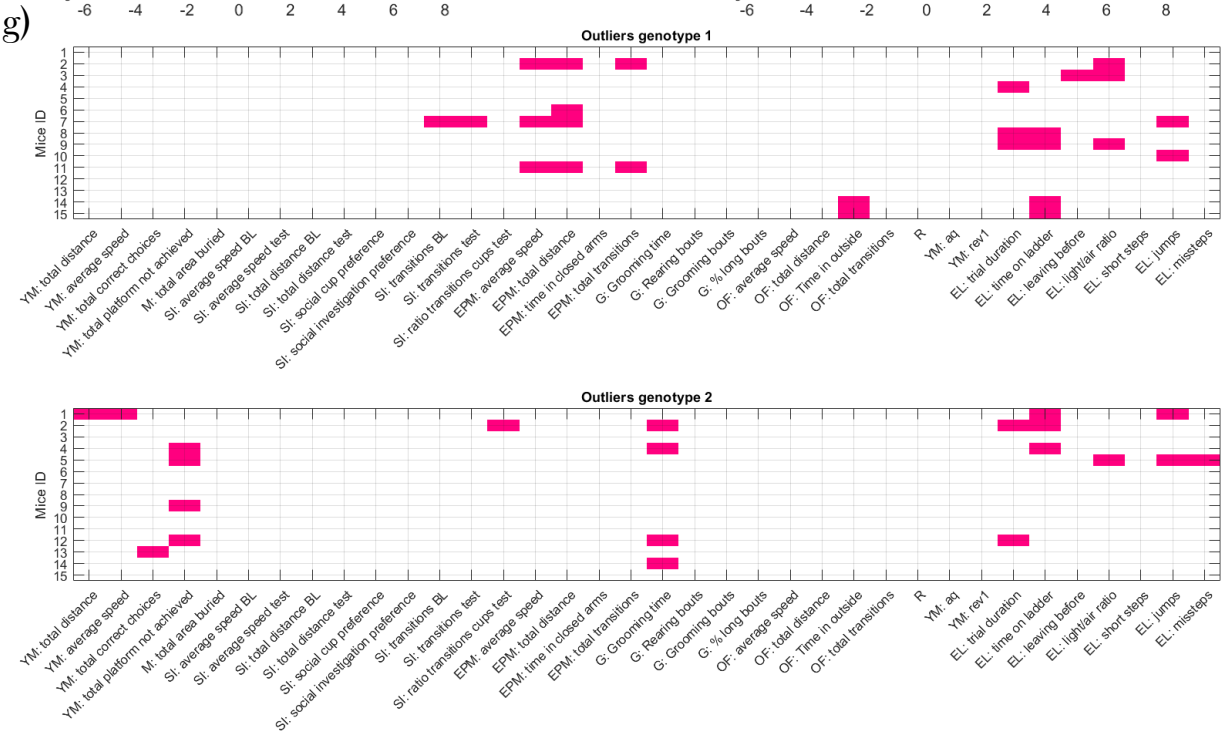

Outliers genotype 1
Outliers genotype 2
Mice ID
YM: total distance
YM: average speed
YM: total correct choices
YM: total platform not achieved
M: total area buried
SI: average speed BL
SI: average speed test
SI: total distance BL
SI: total distance test
SI: social cup preference
SI: social investigation preference
SI: transitions BL
SI: transitions test
SI: ratio transitions cups test
EPM: average speed
EPM: total distance
EPM: time in closed arms
EPM: total transitions
G: Grooming time
G: Rearing bouts
G: Grooming bouts
G: % long bouts
OF: average speed
OF: total distance
OF: Time in outside
OF: total transitions
R
YM: aq
YM: rev1
EL: trial duration
EL: time on ladder
EL: leaving before
EL: light/air ratio
EL: short steps
EL: jumps
EL: missteps

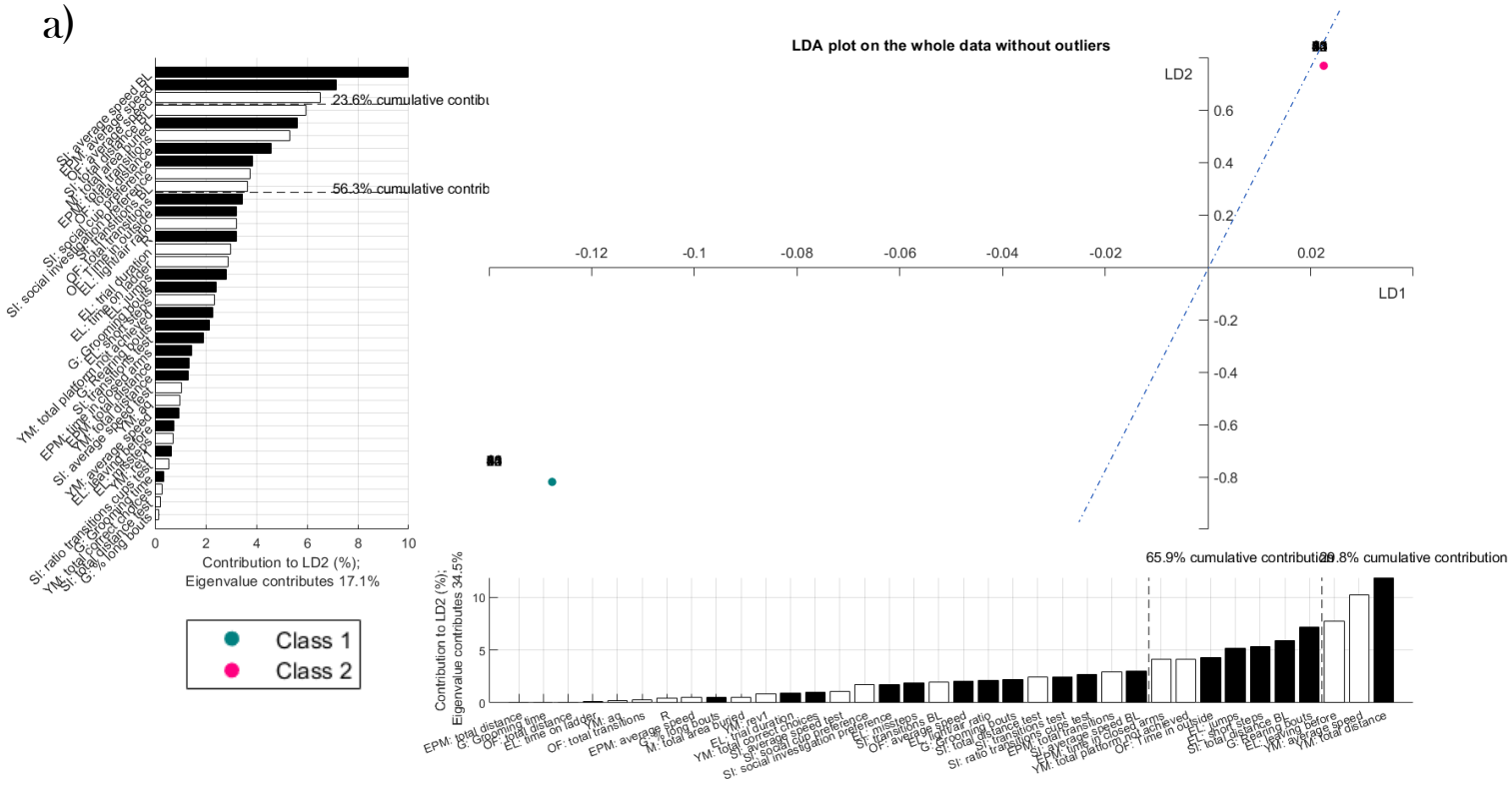
Figure 5 *The norm plot of "EL: trial duration" over the whole variable.* **b)** *The norm plot of "G: grooming time" over the whole variable.* **c)** *The distribution plot for "EL: trial duration" of class 1 before outlier removal. The pink datapoints will be removed.* ***d)*** *The distribution plot for "EL: trial duration" of class 1 after outlier filtering. The pink point are the new outliers, which are not that extreme so will not be filtered out again.* **e)** *The distribution plot for "G: Grooming time" of class 2 before outlier removal. The pink datapoints will be removed.* ***f)*** *The distribution plot for "G: grooming time" of class 2 after outlier filtering.* **g)** *The outlier plot for both classes (genotypes). The variables "EPM: total distance" and "EL: time on ladder" in class 1 and "YM: total platforms not achieved" and "Grooming time" in class 2 have all a total amount of outliers of 4, which is 37.5% for of total datapoints both class classes.*

First LDA implementation

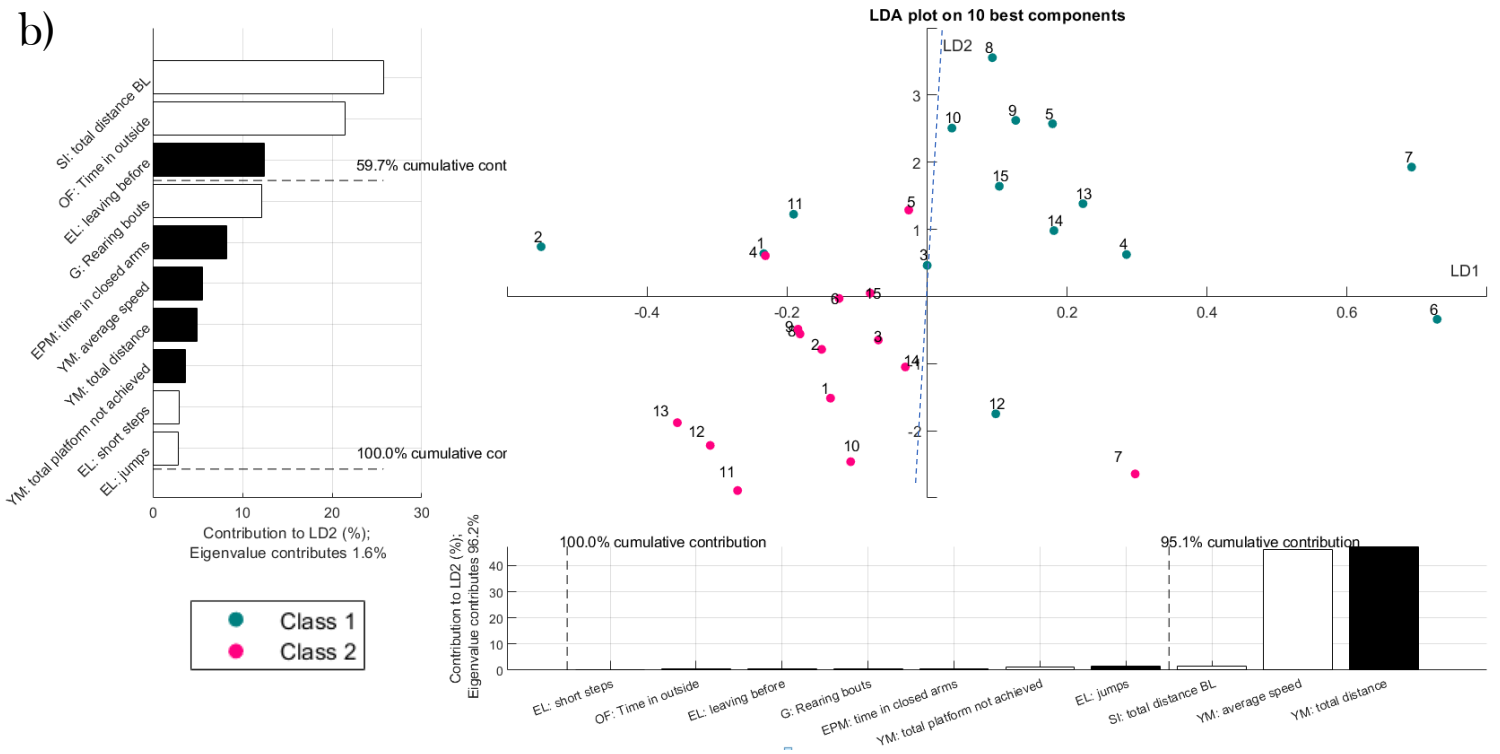
After the pre-processing was done, the new filtered classes were implemented in the coded LDA function. As mentioned in the introduction, if the number of variables is relatively higher than the sample size, the data gets over separated. With a sample size of 30 and 2 classes, the number of variables should not pass the threshold of $d \leq n - c$=28, if we wanted to use the normal inverse for the within scatter matrix. After pre-processing we had 36 variables, which was 8 variables too many causing the output of the LDA to have a typical form of two clustered dots (**Figure 6a**). The eigenvalue of LD2 contributed 19.1% of the total eigenvector set, while the eigenvalue of LD1 contributed 34.5%, which means that LD1 contributed twice as much as LD2 to the separation of the two classes. This was indeed the case as the imaginary line that would separates the two classes the best, would be the arctan of the ratio of the two eigenvalues. We calculated the angle for this particular outcome: arctan(Eigenvalue_1/Eigenvalue_2)= arctan(34.5/17.1)$\approx 63°$. This is represented by the blue line in Figure 6a. The separation itself did not give a lot of information, but the sorted bar graph of the first LD did; By using a subset of the components with the highest contributions, the dimensionality of the LDA could be reduced significantly. We chose to use two different subsets: the first 10 components that contribute 65.9% of the total separation and the first 3 components, which contribute a total of 29.8%.

Using the 10 most contributing variables already gave a more natural separation (**Figure 6b**). The eigenvalue of LD2 in this LDA was very low, but it still gave more context about the shaped behaviour of both classes. Using only the 3 best contributing variables of the LDA gave a worse but more natural looking separation, with some overlapping samples (**Figure 6c**). When comparing the clustering of class 1 in the 10-dimensional and the 3-dimensional LDA, while labelling the points by their class index, we saw that the variance was mostly caused by sample 6 and 7. We used this to revisit the distributions of the variables of interest for class 1 to see if there were any missed outliers. Sample 6 and 7 of class 1 were both not showing up as potential outliers for the top 3 variables of LD1 in Figure 5, so the removal of these two samples was not necessary. Moreover in Figure 6c it looks like that the first two variables of LD1 contribute virtually everything to the separability and LD2 contribute almost nothing. We saw that "YM: total distance" and "YM: average speed" have a really strong linear correlation. (**Figure 6d**) A strong correlation is favourable as long as those variables aren't dependent, because strong correlating independent variable gives us an indication of what variables contribute to the separation. Dependency on the other hand can overrepresent a certain feature and thus bias the outcome. Speed and distance are dependent variables, so out of this result we concluded that these dependent variables should be filtered out to give room to other variables to contribute more to the outcome.

a)


LDA plot on the whole data without outliers
LD2
LD1
23.6% cumulative contrib
56.3% cumulative contrib
Contribution to LD2 (%);
Eigenvalue contributes 17.1%
Class 1
Class 2
65.9% cumulative contribution
29.8% cumulative contribution
Contribution to LD2 (%);
Eigenvalue contributes 34.5%

b)


LDA plot on 10 best components
LD2
LD1
SI: total distance BL
OF: Time in outside
EL: leaving before
G: Rearing bouts
EPM: time in closed arms
YM: average speed
YM: total distance
YM: total platform not achieved
EL: short steps
EL: jumps
59.7% cumulative cont
100.0% cumulative cor
Contribution to LD2 (%);
Eigenvalue contributes 1.6%
Class 1
Class 2
100.0% cumulative contribution
95.1% cumulative contribution
Contribution to LD2 (%);
Eigenvalue contributes 96.2%

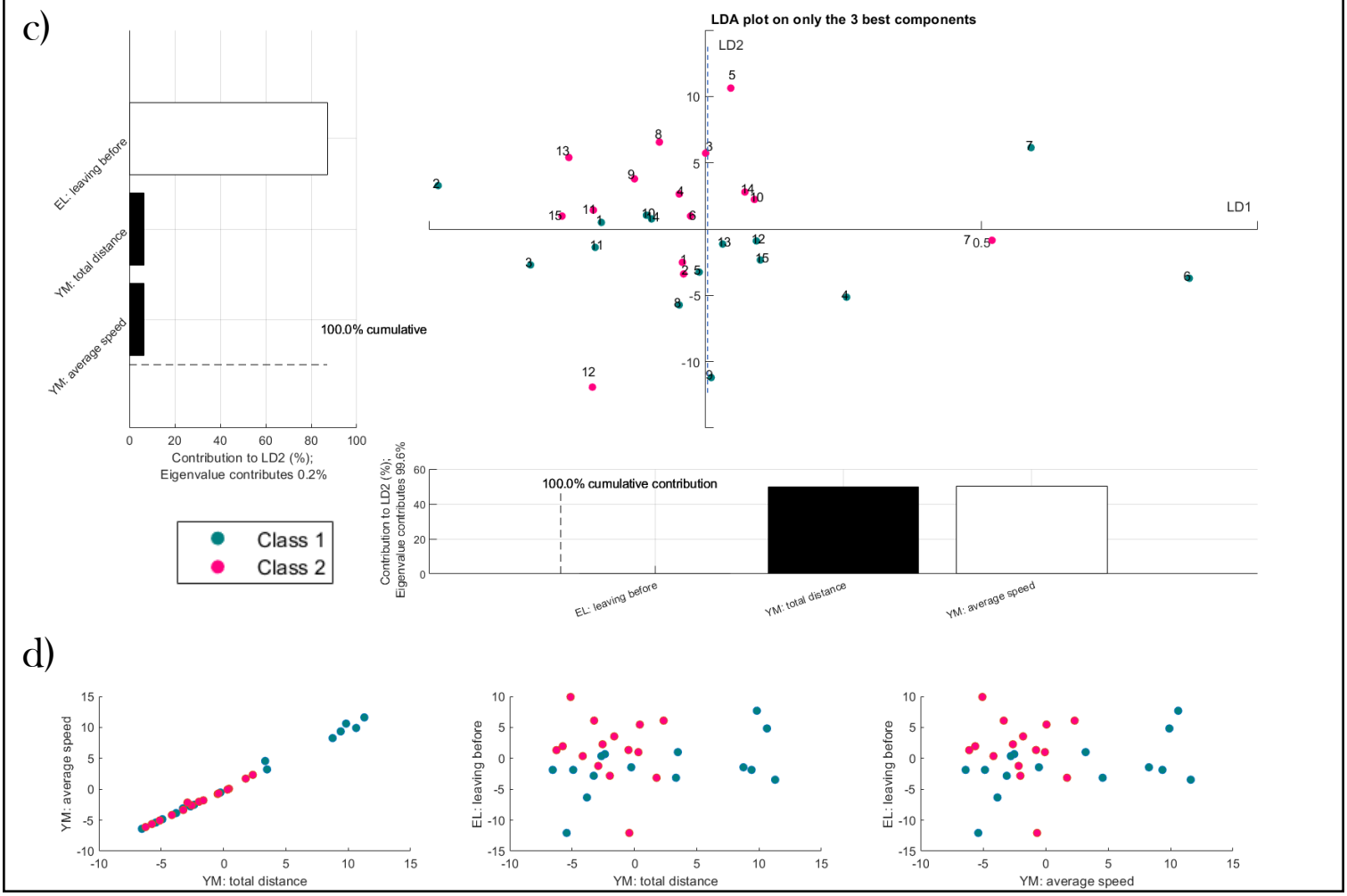


Figure 6 *The LDA plot for the whole dataset after pre-processing. As expected, the data is clustered into 2 extreme dense point clusters because of the high dimensionality.* **b)** *The LDA plot for the subset of the first 10 best separating variables. The blue dotted line indicates the separation line following the arctan of the ratio of the two eigenvalues.* **c)** *The LDA plot for the subset of the first 3 best separating variables. YM: average speed and YM: total distance contribute almost everything in this space.* ***d)*** *The 3 best variables plotted against each other. YM: average speed and YM: total distance give a linear dependency, which is as expected. This causes the other variables to perform worse.*

Revisiting and filtering variables

After the first attempt to implement LDA on the data we optimized the analysis by revisiting the variables. Because the ultimate goal of LDA was to reduce the dimensionality to look for the variables that give the most information, we tried to filter out dependent variables. When too many dependent variables are present in the dataset, certain characteristics will be overrepresented, giving a false outcome **[9].** Reducing the number of variables also gives less degrees of freedom; which is favourable as we had to many variables to begin with. We then plotted the absolute correlation matrix to look for dependent variables. (**Figure 7a**) What we saw was that the 'total distance' variables with their corresponding 'average speed' variables had a high correlation value (corr. >= 0.8642). The correlation between 'EPM: average speed' and 'EPM: total transitions' was also high with a correlation value of corr. = 0.8592. Both types of dependencies were expected and after knowing their correlation value we chose to remove the speed related variable, as the correlation value between 'EPM: total distance' and 'EPM: total

transitions' is a bit lower (corr. = 0.7875). After excluding the speed related variables from the analysis, we ended up with the variables as shown in Figure 7b. After this step we still had 31 variables, 3 variables above the threshold criterium to do a successful LDA.

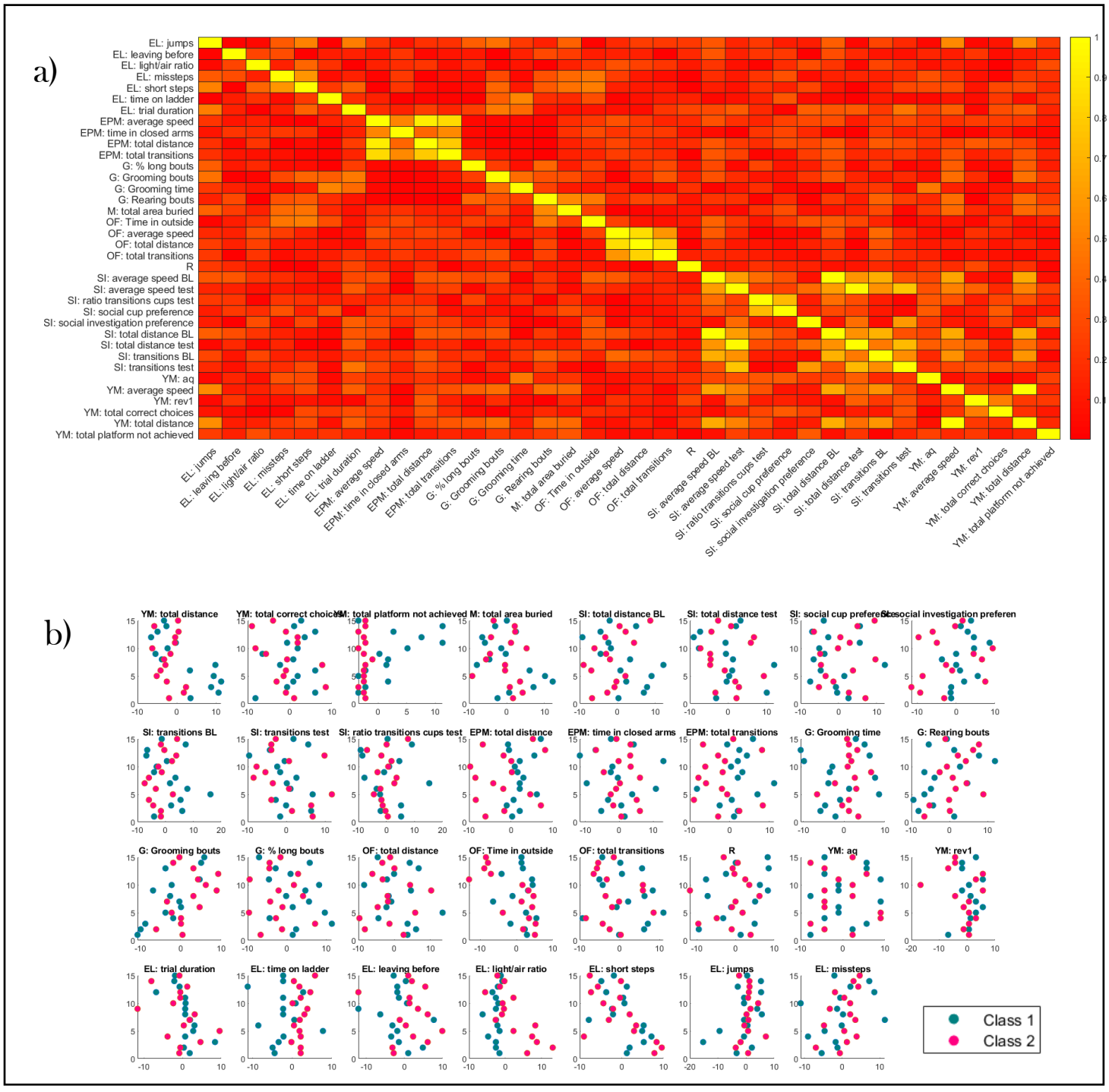


Figure 7 a) *The heatmap of the absolute correlation matrix of all the variables. Mostly the variables that relate to distance and their associated speed variable correlate the most. From this we concluded to filter out the speed variables from further analyses.* **b)** *The scatterplots of the data that will be used for the updated LDA.*

Solving the high input dimensionality problem

The limiting factor of the number of variables we can use, stated a real problem for the use of the LDA for behavioural data, because the initial idea was to use LDA for data with too much variables to begin with. One way to deal with this problem was to reduce the number of variables we had as an input for the LDA. But by removing variables manually we would neglect the objectivity of using an algorithm like LDA; manual filtering will always be prone to subjectivity. There are multiple ways to objectively filter out the least contributing variables, but we focused on three and compared them to look for the best suiting one.

Scoring method

To have a general idea which variables were separated the best, we calculated for every variable the parameters that were used to compute the LDA: the mean difference between the 2 classes, the variance for every class and a scoring that is parallel to the LDA optimization argument:

$$score = \frac{|\mu_1 - \mu_2|^2}{\sigma_1^2 + \sigma_2^2}$$

We then ordered the variables on their highest score, so that we had an indication on what variables were prone to end up higher on the LDA, like in Figure 8a. This method could be used as a filtering method on its own, but also as a comparison method, as the scoring corresponds to a LDA on a lower dimension.

PCA

Another way of ordering the variables for potential filtering was to look at the outcome of the PCA, making use of the inbuild PCA function of Matlab. We plotted the contributions of all the variables on the first PC to compare the variable ordering to the scoring method. (**Figure 8b**) The ordering of this PCA did not correlate to the scoring method, which was remarkable. But in theory the ordering of the scoring method would be more in line with the outcome of the LDA, given that it works with same parameters. Moreover, both the scoring method and the LDA focusses on the separation of the 2 classes, while PCA looks at the separability of the data in general.

Pseudo inverse

A third option of handling the high number of variables did not include filtering out variables, but instead made use of the whole range of variables. This method used the Moore-Penrose pseudo inverse instead of the normal inverse for the within scatter matrix **[22].** The pseudo inverse approximates the inverse when the within matrix is singular and this method has been proven to handle larger amounts of variables **[22].** This allowed us to use more variables than the LDA criterium ($d \leq n-c$). In Figure 8c we see the outcome of implementing the pseudo inverse; The classes clustered in a natural way, despite the high number of variables. This seemed like the ultimate solution, but we had to take in consideration the question if we favoured to approximate data, which in theory is already usable, over filtering data; taking the risk of losing significant data due to bias. Therefore, we compared this outcome to the other methods of dealing with high input dimensionality.

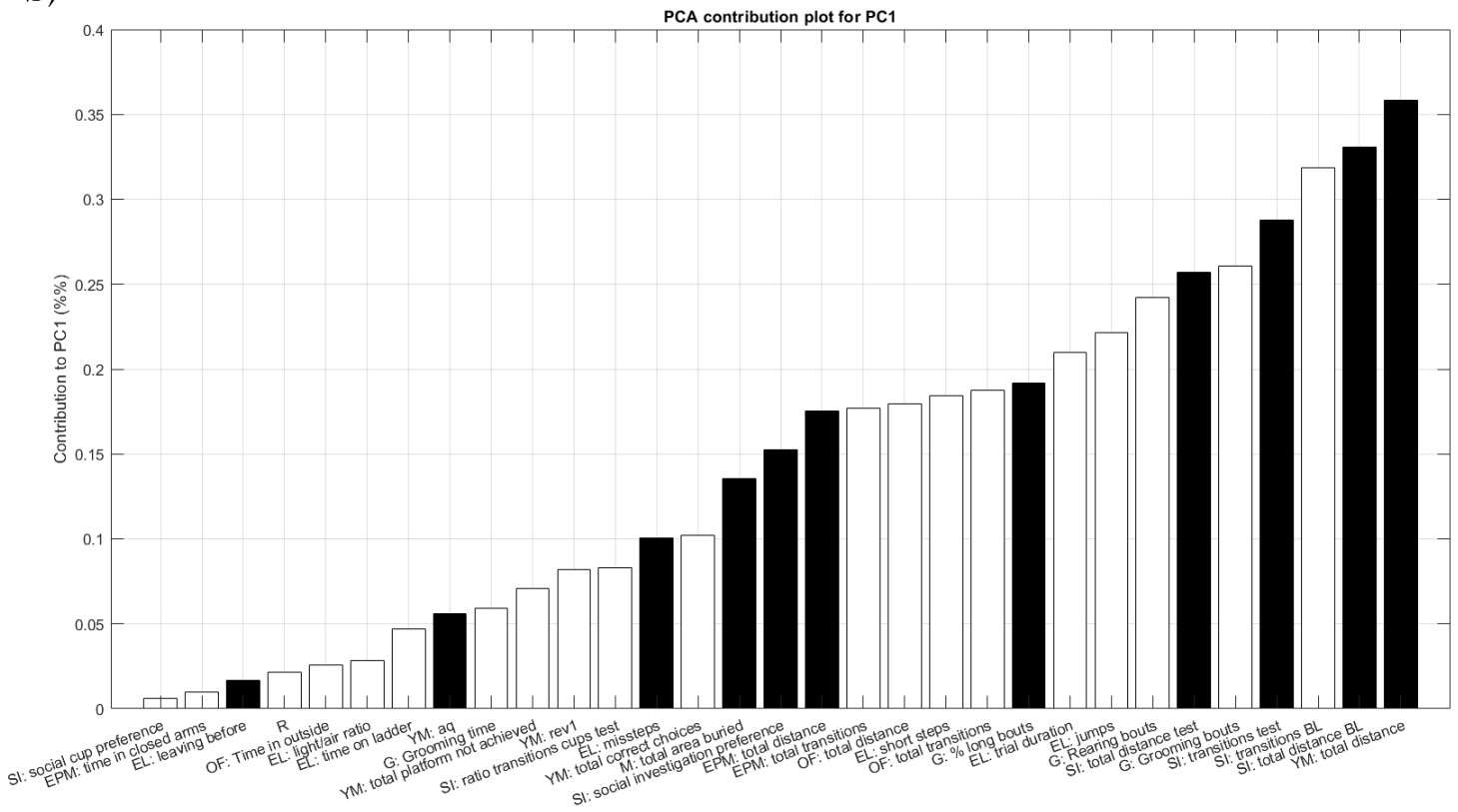
Comparing the pseudo inverse with the scores and the PCA

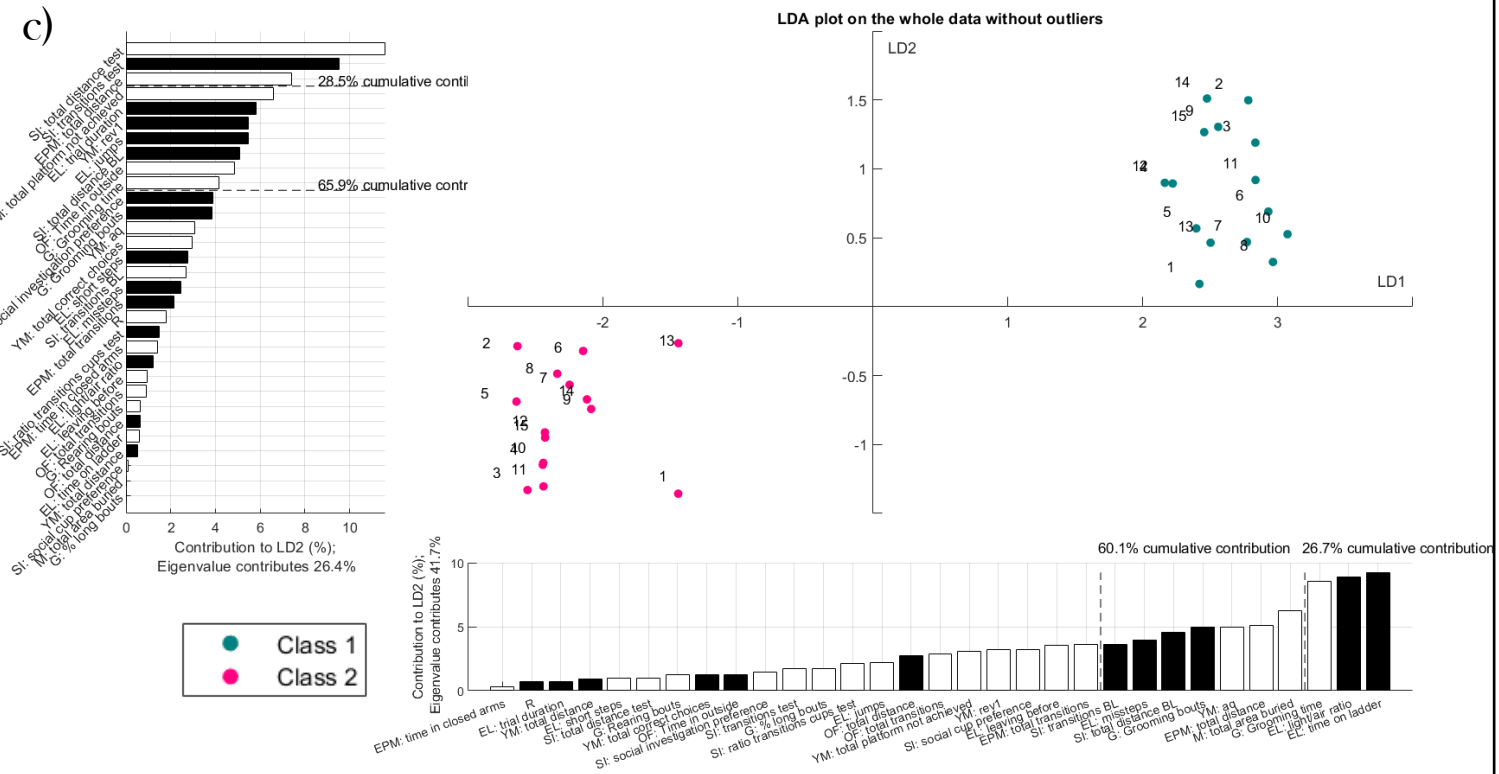
After comparing the pseudo inverse LDA with the PCA outcome and the scoring method, we saw that the scoring method corresponded the most, as predicted; Of the 10 best variables from each method, the scoring method had 5 variables in common with the pseudo inverse LDA, while only 3 with the PCA. Not only that, but in the PCA ordering there were many more variables that ended up really low, while scoring high in the LDA. We plotted the contribution ordering of the pseudo inverse LDA, the scoring of the variables with the LDA parameters and the ordering of PC1 to compare the ordering from the different methods. (**Figure 8d**)

a)

	mean difference	std class 1	std class 2	score: (m1-m2)/(s1+s2)
SI: transitions test	0.2419	4.308	6.278	0.02285
SI: total distance test	0.4561	5.81	4.913	0.04253
YM: aq	0.6667	5.606	5.521	0.05992
SI: ratio transitions cups test	0.6351	6.738	3.519	0.06192
EL: short steps	0.7308	4.208	6.002	0.07158
M: total area buried	0.8807	6.086	4.071	0.08671
EL: missteps	1.071	6.115	3.493	0.1114
EPM: time in closed arms	1.37	6.808	3.845	0.1286
EL: trial duration	1.262	3.244	4.997	0.1532
R	1.826	6.058	4.863	0.1672
EL: jumps	1.436	5.647	2.579	0.1746
SI: social cup preference	1.982	5.596	5.359	0.1809
YM: total correct choices	1.756	4.165	4.913	0.1934
OF: total distance	2.446	5.547	5.308	0.2253
G: Grooming time	2.04	5.992	2.94	0.2284
G: Rearing bouts	2.703	6.201	4.447	0.2538
OF: Time in outside	2.59	3.364	6.273	0.2687
YM: rev1	2.487	3.409	5.651	0.2745
OF: total transitions	3.064	5.764	4.887	0.2877
SI: total distance BL	3.074	5.894	4.726	0.2895
EL: leaving before	2.793	4.493	5.119	0.2905
SI: transitions BL	3.181	6.302	3.613	0.3208
G: Grooming bouts	3.484	5.999	4.434	0.334
SI: social investigation preference	3.601	3.249	6.681	0.3627
EPM: total transitions	3.868	4.395	5.219	0.4024
YM: total distance	3.801	6.595	2.652	0.4111
G: % long bouts	4.3	5.459	4.736	0.4217
EPM: total distance	3.666	2.483	5.845	0.4402
EL: time on ladder	4.44	4.456	1.663	0.7256
EL: light/air ratio	4.84	1.382	5.249	0.7299
YM: total platform not achieved	4.248	5.076	0.62	0.7458

b)





d)

	LD1
EPM: time in closed arms	0.3126
R	0.6809
EL: trial duration	0.7023
YM: total distance	0.9426
EL: short steps	0.9523
SI: total distance test	0.984
G: Rearing bouts	1.231
YM: total correct choices	1.236
OF: Time in outside	1.246
SI: social investigation preference	1.471
SI: transitions test	1.702
G: % long bouts	1.727
SI: ratio transitions cups test	2.16
EL: jumps	2.19
OF: total distance	2.708
OF: total transitions	2.894
YM: total platform not achieved	3.091
YM: rev1	3.222
SI: social cup preference	3.24
EL: leaving before	3.553
EPM: total transitions	3.606
SI: transitions BL	3.609
EL: missteps	3.982
SI: total distance BL	4.551
G: Grooming bouts	4.988
YM: aq	4.997
EPM: total distance	5.089
M: total area buried	6.269
G: Grooming time	8.52
EL: light/air ratio	8.915
EL: time on ladder	9.229

	score: (m1-m2)/(s1+s2)
SI: transitions test	0.02285
SI: total distance test	0.04253
YM: aq	0.05992
SI: ratio transitions cups test	0.06192
EL: short steps	0.07158
M: total area buried	0.08671
EL: missteps	0.1114
EPM: time in closed arms	0.1286
EL: trial duration	0.1532
R	0.1672
EL: jumps	0.1746
SI: social cup preference	0.1809
YM: total correct choices	0.1934
OF: total distance	0.2253
G: Grooming time	0.2284
G: Rearing bouts	0.2538
OF: Time in outside	0.2687
YM: rev1	0.2745
OF: total transitions	0.2877
SI: total distance BL	0.2895
EL: leaving before	0.2905
SI: transitions BL	0.3208
G: Grooming bouts	0.334
SI: social investigation preference	0.3627
EPM: total transitions	0.4024
YM: total distance	0.4111
G: % long bouts	0.4217
EPM: total distance	0.4402
EL: time on ladder	0.7256
EL: light/air ratio	0.7299
YM: total platform not achieved	0.7458

	PC1
SI: social cup preference	0.6362
EPM: time in closed arms	0.9939
EL: leaving before	1.677
R	2.169
OF: Time in outside	2.554
EL: light/air ratio	2.864
EL: time on ladder	4.686
YM: aq	5.598
G: Grooming time	5.934
YM: total platform not achieved	7.09
YM: rev1	8.183
SI: ratio transitions cups test	8.313
EL: missteps	10.04
YM: total correct choices	10.23
M: total area buried	13.57
SI: social investigation preference	15.24
EPM: total distance	17.55
EPM: total transitions	17.7
OF: total distance	17.95
EL: short steps	18.42
OF: total transitions	18.73
G: % long bouts	19.17
EL: trial duration	20.98
EL: jumps	22.16
G: Rearing bouts	24.2
SI: total distance test	25.69
G: Grooming bouts	26.05
SI: transitions test	28.8
SI: transitions BL	31.85
SI: total distance BL	33.09
YM: total distance	35.85

Figure 8 **a)** *The mean differences standard deviations of both classes and the score per variable ordered by this score:* $\frac{|\mu_1-\mu_2|^2}{\sigma_1^2+\sigma_2^2}$*, which is analogue to the optimization argument of the LDA.* **b)** *The outcome of the first PC from the PCA.* **c)** *The LDA after using the pseudoinverse. The clustering seems more natural despite the high input dimensionality.* **d)** *The ordering of LD1 using the pseudoinverse, the scoring method and PC1. The score method and the ordering of LD1 correlate the most.*

Variable	Rank in pseudo inverse LD1	Rank in scoring	Rank in PC1
EL: time on ladder	1	3	25
EL: light/air ratio	2	2	26
EPM: total distance	5	4	15
G: Grooming bouts	7	9	5
SI: transitions BL	10	10	3
SI: total distance BL	8	12	2

Table 9: *the indices of some overlapping variables in the different methods.*

As we can see in Figure 8d and table 9, there are 5 variables that are overlapping in the top 10 of both the scoring method and the pseudo inverse based LDA. Even the individual placements of these variables are close to each other. The placement of the variable "YM: aq" on the other hand is very different in both orders; In the pseudo inverse LDA it is on the 6th place and in the scoring method on the 29th place. This could be due to the fact that the distribution of this variable is discrete and LDA had difficulties dealing with a mixture of continuous and discrete variables **[16].** Concluding, as the pseudo inverse method did not seem to distort the expected ordering too much from the scoring method, we decided to go for the pseudo inverse. Also taking in consideration the applicability of this method for datasets in the future that will have even more variables.

Second LDA implementation

After applying the pseudo-inverse method to our LDA algorithm, we ran the LDA again with the new dataset to see if the outcoming variables or the separation had changed. We saw a remarkable change in the 10 variable LDA comparison to the previous 10 variable LDA. The distributions of the variable contributions of both LD's looked more natural than the previous distributions; the previous 10 variable LDA had 2 two strong correlating variables that contributed almost everything. It looked like LD2 did not contribute much to the separation of the classes, but when drawing a line following the arctan of the ratio of the 2 eigenvalues, we saw that the separation was right. (**Figure 9a**)

The classes in the LDA on 3 variables were a bit more clustered with still a high variance in class 1, with some points of class 1 overlapping the domain of class 2 and vice versa. (**Figure 9b**) We then paired the 3 best variables by plotting them to each other in 3 2D scatter plots, that way we could see the relation between these variables. It was noticeable that the relations of these variables indeed separated the data in 2 classes in a significant way. (**Figure 9c**)

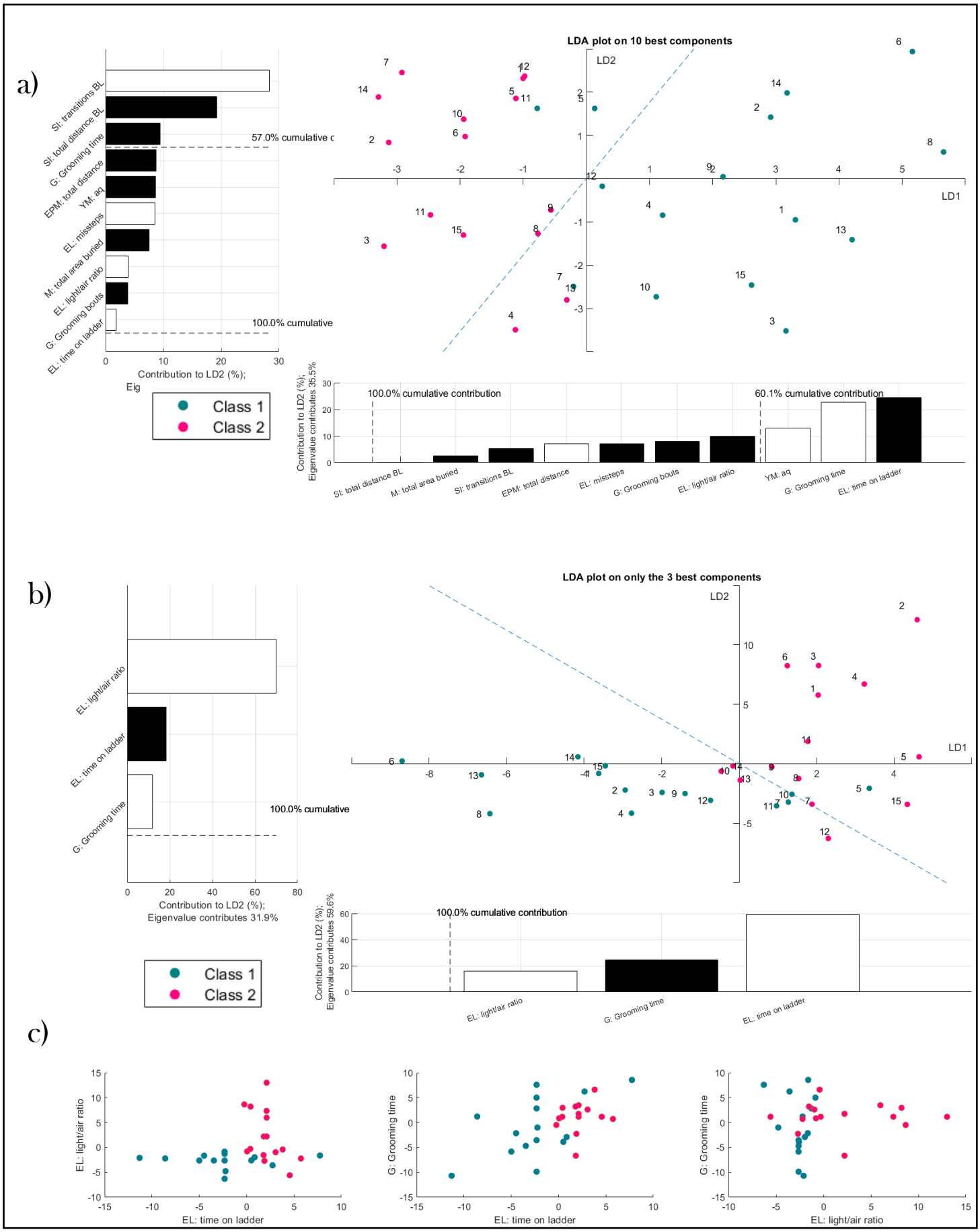

a)
SI: transitions BL
SI: total distance BL
G: Grooming time
EPM: total distance
YM: aq
EL: missteps
M: total area buried
EL: light/air ratio
G: Grooming bouts
EL: time on ladder
57.0% cumulative c
100.0% cumulative
Contribution to LD2 (%);
Eig
Class 1
Class 2
LDA plot on 10 best components
LD2
LD1
100.0% cumulative contribution
60.1% cumulative contribution
Contribution to LD2 (%);
Eigenvalue contributes 35.5%
SI: total distance BL
M: total area buried
SI: transitions BL
EPM: total distance
EL: missteps
G: Grooming bouts
EL: light/air ratio
YM: aq
G: Grooming time
EL: time on ladder
b)
EL: light/air ratio
EL: time on ladder
G: Grooming time
100.0% cumulative
Contribution to LD2 (%);
Eigenvalue contributes 31.9%
Class 1
Class 2
LDA plot on only the 3 best components
LD2
LD1
100.0% cumulative contribution
Contribution to LD2 (%);
Eigenvalue contributes 59.6%
EL: light/air ratio
G: Grooming time
EL: time on ladder
c)
EL: light/air ratio
EL: time on ladder
G: Grooming time
EL: time on ladder
G: Grooming time
EL: light/air ratio

Figure 9 a) *The LDA plot of the first 10 variables of the updated data, the blue dotted line has the slope is calculated by the ratio of the eigenvalues of LD1 and LD2.* **b)** *The LDA plot of the first 3 variables of the updated data.* ***c)*** *The first 3 variables are plotted versus each other to visualise the relation between the variables; there is no linear correlation, but the classes do get separated well.*

LDA validation: Shuffling of the data

We validated the LDA by performing a shuffling test (**Figure 10**). The distribution of the first 5 variables of LD1 with respect to the original LD1 after shuffling 200.000 times revealed that the order of the variables of LD1 was indeed significant.

The bar graphs in Figure 10 can be interpreted as the quantification of the clustering degree of the whole variable. Ideally would like to have high peaks for variables that score low on our LD1. This can be interpreted by the following statement: Although variable X is highly clustered, the variable still appears low on the ranking of the best separating vector of our analysis. This means that the ranking of this variable is unique for our classification and thus our genotype.

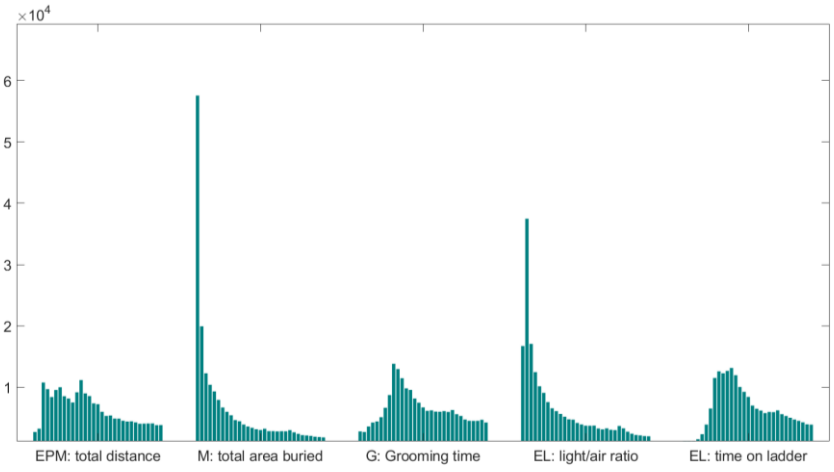
The first variable 'EL: time on ladder' had a low peak on the first ranking, so this made it significant that it did appear in the LD1 of the LDA; it meant that the variable was less geometrically clustered but still important for our classification, which makes it significant for our genotype. "EL: light/air ratio" did have a high peak on its respective place, which made its rank less significant. This is most likely caused by the fact that the data can be split in two geometrical clusters, which meant that this data had a high probability to be split into two random clusters (**Figure 10b**). This means that the clustering coincides partially with the labelling of our classes.

Furthermore, we saw a high peak on the first ranking of "M: total area buried" (**Figure 10a**). This peak could be explained by the distribution of this variable (**Figure 10c**). We saw that the distribution of the data was moderately clustered, separated by a vector from the origin. This clustering did not correlate to the classification of our genotype, as the data from both classes were scattered over the 2 spatial classes of this variable. Because of the shape of its distribution, "M: total area buried" has a high chance to appear high on the ranking for a LD. But in our classification this was not the case, which made the outcome of our LD1 more unique.

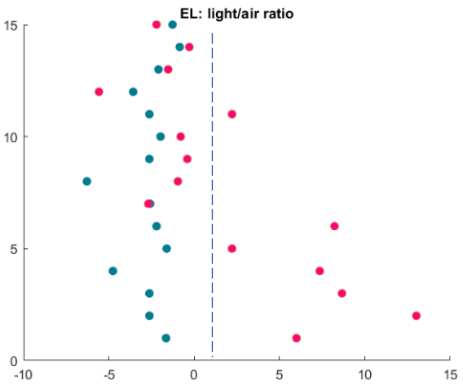
We then plotted for the amount of times the variables appeared on their own respective ranking during the shuffling (**Figure 10d**). Except for the "EL: light/air ratio" variable, each peak seemed to be low ($\leq 7.69\%$), which could be interpreted as a quantification for the significance of the appearance of these variables individually. We also looked at the placements for the second rank (**Figure 10e**). We saw that the high peak of "EL: light/air ratio" was not the highest; the variable "SI: transition test" appeared 24.925% of the 200.000 shuffles on the second place which was 6.3% more than "EL: light/air ratio". This meant that there was still some significance in "EL: light/air ratio" being on the 2nd place on the original LD1.

The next thing we did was looking for the dependent appearances between the first 3 variables, or in other word: how many times does variable 1 and 2 appear together while shuffling the data 200.000 and then the same question for the first 3 variables. We found that only 0.01% of the time the first 2 variables appeared together and the first 3 variables appeared 0% of the time together in their order. This meant that the specific order of these variables was significant.

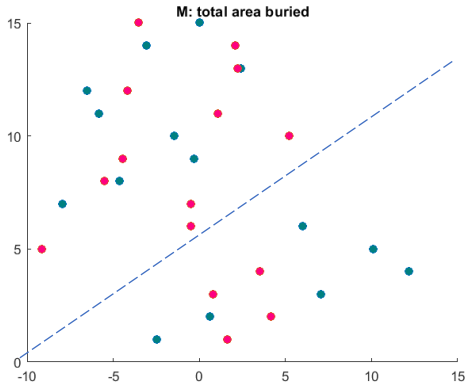
a)


×10^4
EPM: total distance
M: total area buried
G: Grooming time
EL: light/air ratio
EL: time on ladder

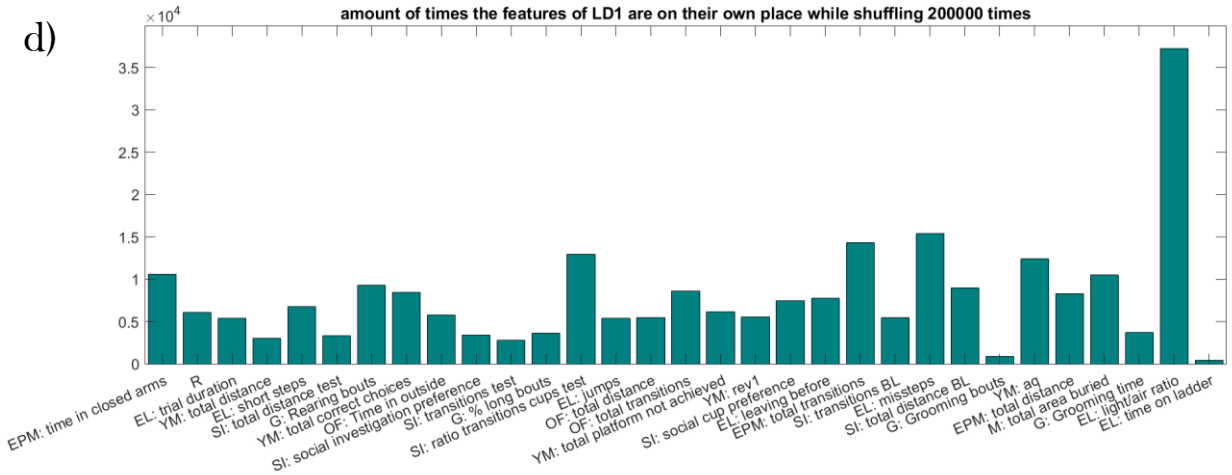
b)


EL: light/air ratio

c)


M: total area buried

d)


amount of times the features of LD1 are on their own place while shuffling 200000 times
×10^4
EPM: time in closed arms
R
EL: trial duration
YM: total distance
EL: short steps
SI: total distance test
G: Rearing bouts
YM: total correct choices
OF: Time in outside
SI: social investigation preference
SI: transitions test
G: % long bouts
SI: ratio transitions cups test
EL: jumps
OF: total distance
OF: total transitions
YM: total platform not achieved
YM: rev1
SI: social cup preference
EL: leaving before
EPM: total transitions
SI: transitions BL
EL: missteps
SI: total distance BL
G: Grooming bouts
YM: aq
EPM: total distance
M: total area buried
G: Grooming time
EL: light/air ratio
EL: time on ladder

e)

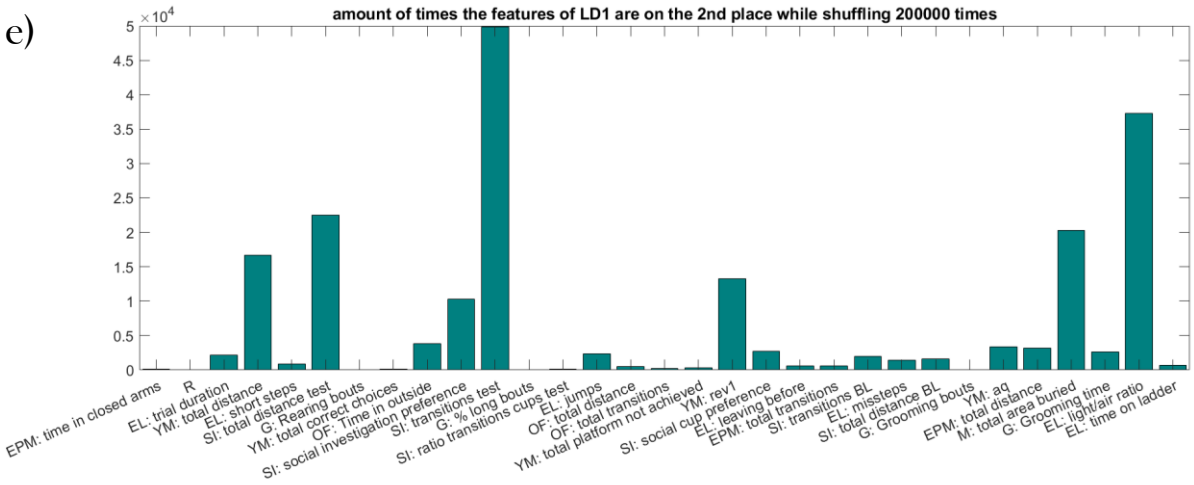

amount of times the features of LD1 are on the 2nd place while shuffling 200000 times
×10^4
EPM: time in closed arms
R
EL: trial duration
YM: total distance
EL: short steps
SI: total distance test
G: Rearing bouts
YM: total correct choices
OF: Time in outside
SI: social investigation preference
SI: transitions test
G: % long bouts
SI: ratio transitions cups test
EL: jumps
OF: total distance
OF: total transitions
YM: total platform not achieved
YM: rev1
SI: social cup preference
EL: leaving before
EPM: total transitions
SI: transitions BL
EL: missteps
SI: total distance BL
G: Grooming bouts
YM: aq
EPM: total distance
M: total area buried
G: Grooming time
EL: light/air ratio
EL: time on ladder

Figure 10 **a)** *The distribution bar graphs for the first 5 variables of LD1. There seem to be a high peak on the 2nd place bar of "EL: light/air ratio" ,its place in LD1, which could indicate a dominant feature. There is also a high peak on the first place of "M: total area buried" but in LD1 it appears on the 4th place, which makes the particular ordering of LD1 more significant.* **b)** *The scatterplot of "M: total area buried". If we draw an imaginary line through the origin we can see that this data is geometrically clustered, but not in favour of the classification. This explains why this variable has a high chance of appearing high on a separation vector, but not in our classification case.* **c)** *The scatterplot of "EL: light/air ratio". If we draw an imaginary line, like the blue line, through the origin we can see that this data is geometrically clustered, but in favour of the classification. This explains why this variable has a high chance of appearing high on a separation vector and also does on our LD1.* **d)** *The percentage of appearance on their own respective place for every variable after shuffling 200.000 times. "EL: light/air ratio" appears to be very high as previously mentioned.* **e)** *The percentage of appearance on the second place for every variable after shuffling 200.000 times. "SI: transition test" has a higher chance of appearing on the second place than "EL: light/air ratio", which tells us that the placement of "EL: light/air ratio" in our LD1 is somewhat significant.*

Heatmaps

Next, we used the quantified absolute contribution values of all the variables per LD, grouped per experiment. We visualized the correlation is of every variable with every LD and investigated the clustering patterns among experiments using the heatmap visualization (**Figure 11a**). The Erasmus ladder experiment (EL) seemed to have a high correlation with this classification as 2 variables: "EL: light/air ratio" and "EL: short steps" have high values in LD1. Except from that, there does not seem to be real experiment related clustering.

The next thing we did was combining all the contributions of the first 10 LD's and taking the weighted sum of the absolute value of the variable, weighted by the eigenvalue of the LD's. This way we quantified the contextual contributions of each variable in a single number for the first 10 LD's (**Figure 11b**). With this method, "G: grooming time" was the most contributing variable, but the experiment itself has 2 high scoring variables and 2 low contributing ones. The clustering in the Erasmus ladder experiment seemed to be clearer than the clustering in only the LD1, as 5 out of 7 variables were above or equal to the median of the weighted sum (median=6.002). Furthermore the "SI: total transitions" "SI: total distance" seemed to score high. Rotarod and the open field experiment seemed to score the lowest overall.

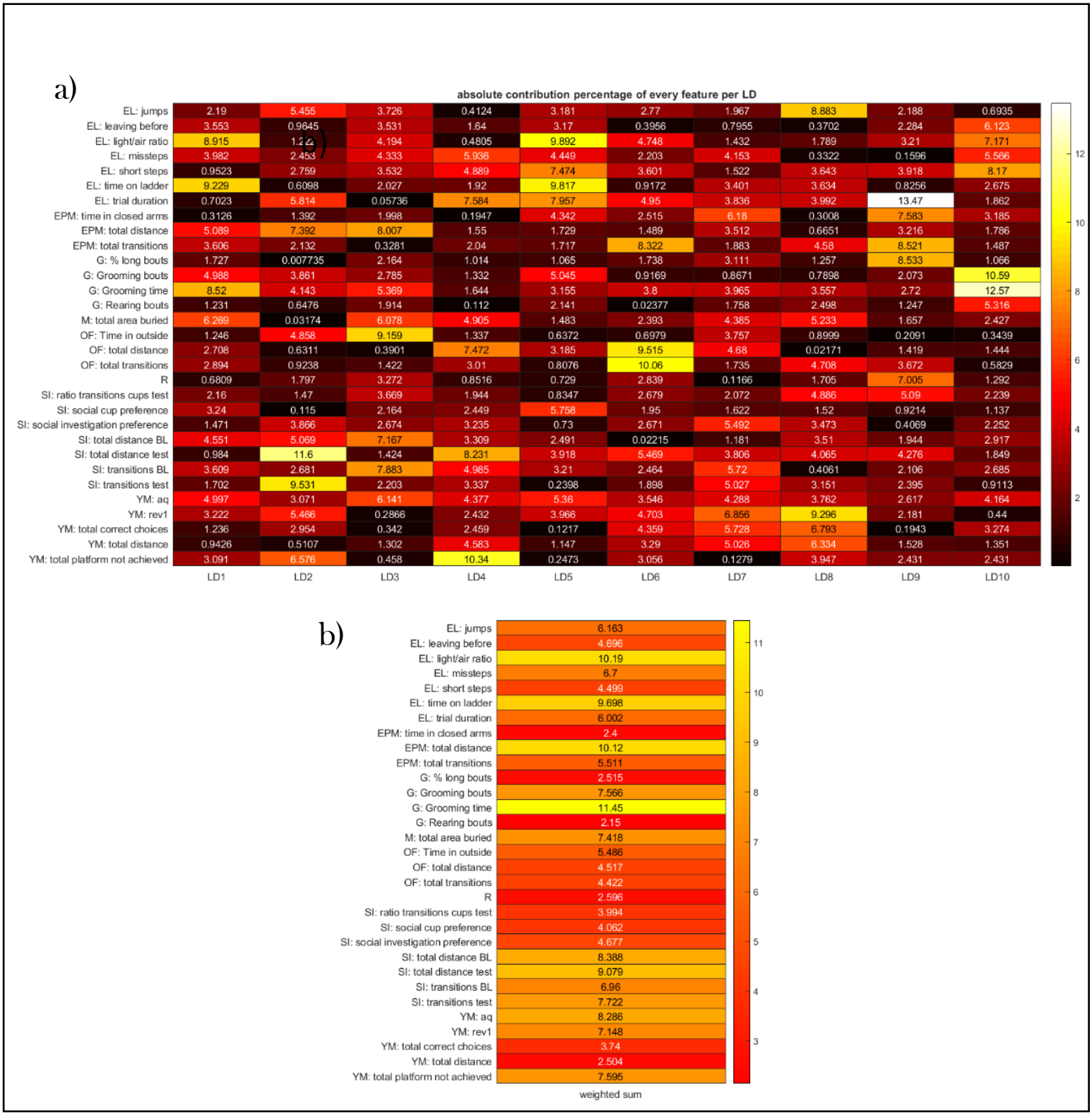


Figure 11 ***a)*** *The heatmap of the all the variables absolute contribution for the first 10 LDs ordered by their ranking in LD1, ordered by experiment.* **b)** *The weighted sum of the contributions of the variables of the first 10 LD's, with the eigenvalue of the LD's as weight. The experiment clustering seems to be more present for the Erasmus Ladder (EL) experiment as it has 5 experiments above the median.*

LDA on PCA

Lastly, we implemented the LDA on the PCA outcome of the data (Figure 12). This gave a linear combination of all the variables, independent of the amount of PC's used, giving an extra contextual insight of the relation between the variables related to the separation. The best contributing variable in the LDA on the first 10 PCs was 'EL: light/air ratio', which was on the second place of LD1 from the original LDA. The first variable from the LDA, 'EL: time on ladder' was placed 7th, which is low but still in the top 10. Out of all the results we tried to make a set of variables that contribute the most to the characterising behaviour of Pi3Kδ gain of function mice. Figure 12b shows the difference in ordering between the LDA and the LDA on the PCA data.

Although we saw that there was some overlap between the different methods, there was still significant variability and some extreme cases where high scoring variables in one method scored lower in a different method. This instability was due to the high dimensionality; in a high dimensionality space a small difference can redirect a separability vector to different directions [6]. It was still a good sign that there were some variables that were present in all methods. To quantify the overlap between the LDA and the LDA on PCA methods we looked which variables had a similar placement in a particular range. We did that for the 10 best variables of LD1, as in that way we could see which variables score the best in both methods and are in that way the most valid. In table 10 we see what variables we get for different boundaries:

Boundary range (from -x to x)	variables
1	'EL: light/air ratio' 'YM: aq'
6	'EL: time on ladder' 'EL: light/air ratio' 'YM: aq'
7	'EL: time on ladder' 'EL: light/air ratio' 'G: Grooming time' 'YM: aq'
8	'EL: time on ladder' 'EL: light/air ratio' 'G: Grooming time' 'EPM: total distance' 'YM: aq' 'SI: transitions BL'

Table 10: *The correlating variables for different ranges.*

Using a boundary range of 7 gave us the top 3 of the LDA ordering. This gave us more confidence that the 3 best variables of the LDA were indeed the most valid separable variables from the dataset and so in line with the expectation.

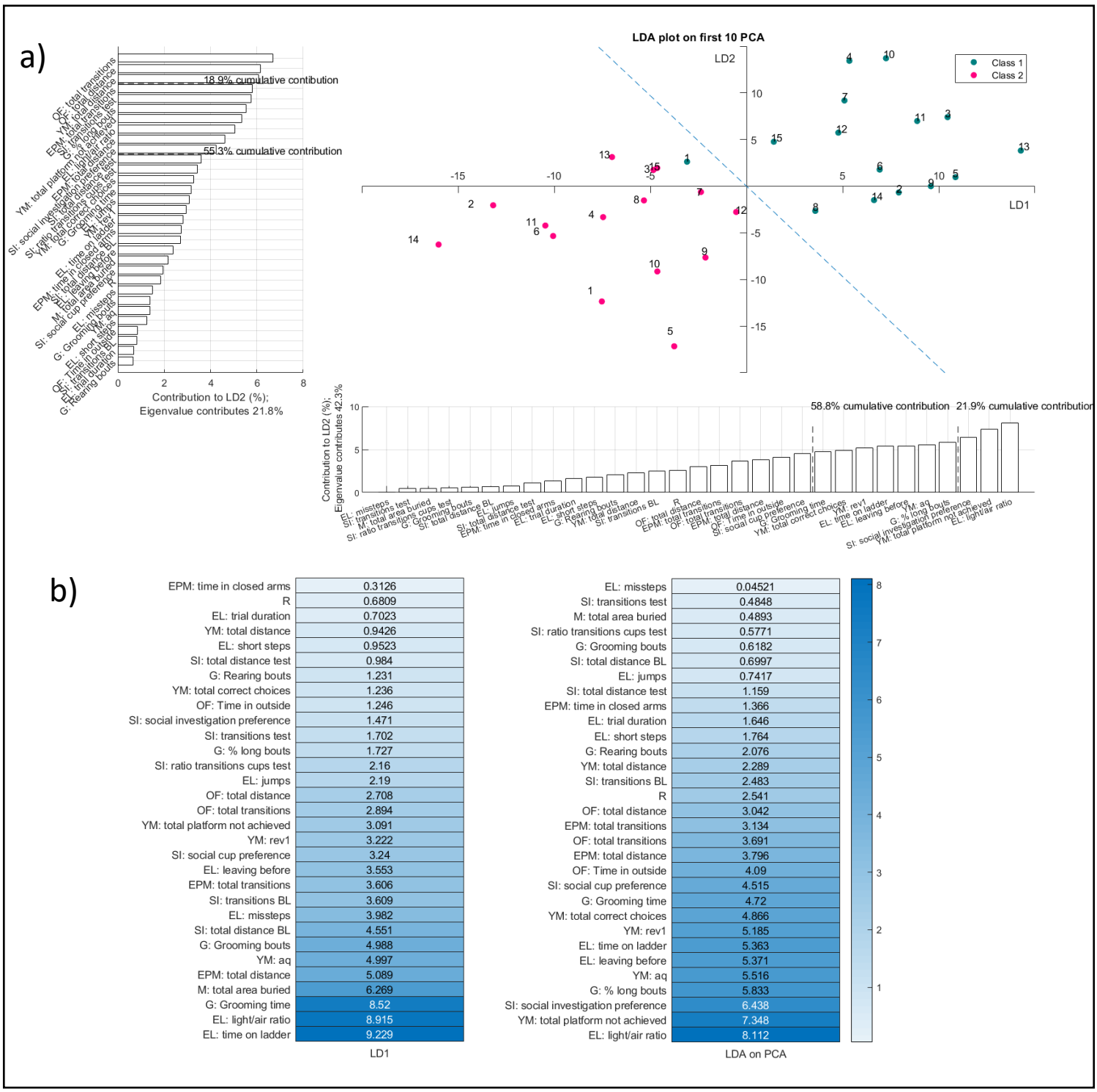


Figure 12 a) *The LDA on the first 10 PCs of the data. "EL: light air ratio" appears on the first place, which was on the second place in the LD. This is as expected as this variable was also spatial clustered, which is what PCA focusses on.* **b)** *The ordering of LD1 of the LDA on the original data and LD1 of the LDA on the PCA.*

General Conclusion

Having performed all the optimization we arrived at the following conclusions regarding the classification on this specific PI3K dataset: The two most defining variables are locomotion and learning related ('EL: time on ladder' and 'EL: light to air ratio' respectively), with a combined contribution of 18.14% on the first LD. The third variable: 'G: Grooming time' is associated with repetitive behaviour. This means that this analysis captures an altered locomotive performance, learning and repetitive behaviour between wildtypes and p110δ^{E1020K} mice.

Discussion

Mean interpolation

Both LDA and PCA are not totally capable of handling "NaN" values [9][13], so we needed to interpolate those values. Outlier interpolation for LDA and PCA is still an active field of research [20], so a perfect solution is yet to be discovered. We chose to go for interpolation with the mean of the class per variable, because of the following reasons.

First, PCA is done by shifting the overall mean to the origin, so by interpolating the outliers with the mean, the relative shifting of the data would not change.
Secondly, LDA works by minimizing the variance within the classes and maximizing the variance between the classes. By interpolating the outliers with the mean of the class, these values do not contribute to the variance within the classes, because of the definition of the variance:

$$\sigma^2 = \frac{\sum(x_i - \bar{x})^2}{N - 1}$$

The distances of the interpolated outliers to the mean contribute nothing to the sum in the numerator of the variance, but the outliers do contribute to the total amount of datapoints -N-, making the overall variance smaller. This leads to that classes which have variables with a large number of outliers will appear more clustered than they really are. Because there is no standard interpolation method for LDA, we chose this one, as reducing the within variances only gives us a small positive bias.

Extreme outliers greatly influence the variances both within, as in between the classes if not taken out, so by removing them you create a positive bias towards perfect separation. Therefore, it strengthens the rule that we must only remove outliers if they are extreme, as outliers can characterize a class and removing an excess of outliers can delete the important distinction between classes. As a result, we have multiple classes with interpolated outliers, that are z-scored together to normalize all the variables, but will function as distinct classes for the LDA.

Validation of the LDA outcome by shuffling

As stated above, the variable contributions of every LD are the directional components for the projection vectors that separate the classes the best.
The order of the LDs, most importantly LD, as this is the most separating one, should therefore be unique for our genotype classification.
To validate this uniqueness we shuffled the labels of our datasets to separate our data in two equally sized random classes that differ from the real genotype classification. Subsequently we did LDA on this random classification and compared the outcoming LD1 order with our original LD1 order.
By repeating this process multiple times, we got per variable the probability distribution on what place they will end up in the ranking of a separation vector.
This distribution quantifies the shape of the variable, which indicates in what sense the variable is separable into 2 classes, relative to the other variables.
Variables that are heavily geometrically clustered in general, will have a high chance to end up as a high separating variable, while uniformly distributed variables will have a lower chance (Figure 13a). Also the presence of outliers can influence the probability of separating the data into two classes as can be seen in Figure 13b.

To get a full representation of the probability distributions of the variables, we shuffled and compared the amount of times that 2 random classes can be defined from the whole dataset, without worrying about the order of the classes. This is the definition of the binomial coefficient, which can be descripted as follow:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

where n is the total amount of samples and k is the class size.
As we are dividing the data into 2 equally sized classes, the binomial coefficient is as follow:

$$\binom{n}{\frac{n}{2}} = \frac{n!}{(\frac{n}{2})!\,(\frac{n}{2})!}$$

The problem with defining the amount of shuffles by the binomial coefficient is that this number increases logarithmically with the amount of total samples.
In our analysis we used a dataset with 30 samples, which means that if we would shuffle the amount of times following the binomial coefficient it would be:

$$\frac{30!}{(15)!\,(15)!} = 1.55 * 10^8$$

This amount of shuffling would cost too much computational power and time for an easy to use analysis, which is the end goal of the implementation of this LDA method.
Therefor we needed another way of defining the amount of shuffles required to get a good indication of the distribution of the variables.

For this we used the concept of a Monte Negro Simulation **[25].** This method dictates that the distribution of a dataset can be approximated by an uniform random data.
If we extrapolate this concept to our case, we use "d" amount of uniform variables with each "N" datapoints, where d is the amount of variables used for the real LDA and N the total amount of samples.
As stated before, a uniform variable will have a low probability to appear as the most dominant one if the other variables have a more clustered shape, but as every variable now has virtually the same uniform distribution, we would expect that every random variable distribution will converge to a constant probability function, where the probability to appear on every place is equal to the mean: the amount of shuffling divided by the amount of variables (**Figure 13d**).
In practice there would be always a certain bias towards a variable that by chance is relatively more clustered than the rest of the variables, due to the random characteristics of that particular dataset, but by repeating this process multiple times with each time new random uniform distributions, this bias will cancel out. These distributions will in theory converge to the mean after the amount of shuffling satisfies the binomial coefficient, as stated before, but by computing the error after each shuffle we can reduce the amount of shuffling, by compromising the outcome by a certain error percentage. This error can be calculated by taking the maximum absolute distance from all the variables to the mean, which is the amount of shuffling divided by the number of variables used.

$$|\epsilon|\ \% = \frac{\max\left(|X - \bar{\mu}|\right)}{\bar{\mu}} * 100\%, \qquad where\ \bar{\mu} = \frac{N_{shuffles}}{d_{variables}}$$

We then plot this error to see when this error passes a certain threshold. The shape of the error curve will be noisy, as with every shuffle the mean chances linearly, so every variable needs "to keep up" to this mean, but the general direction will be converging to zero. To get a better understanding we plot this error percentage in a semi-logarithmic plot, with the error displayed linearly and the amount of shuffling logarithmically. The error threshold has been set to 5%, which we stated as a good error margin. In Figure 13e we can see the error margin plot for a random data set with the same dimensions as we have used for the final LDA. The point where the error margin is $|\epsilon| \leq 0.05$, is after shuffling 200.000 times. So we choose this amount.

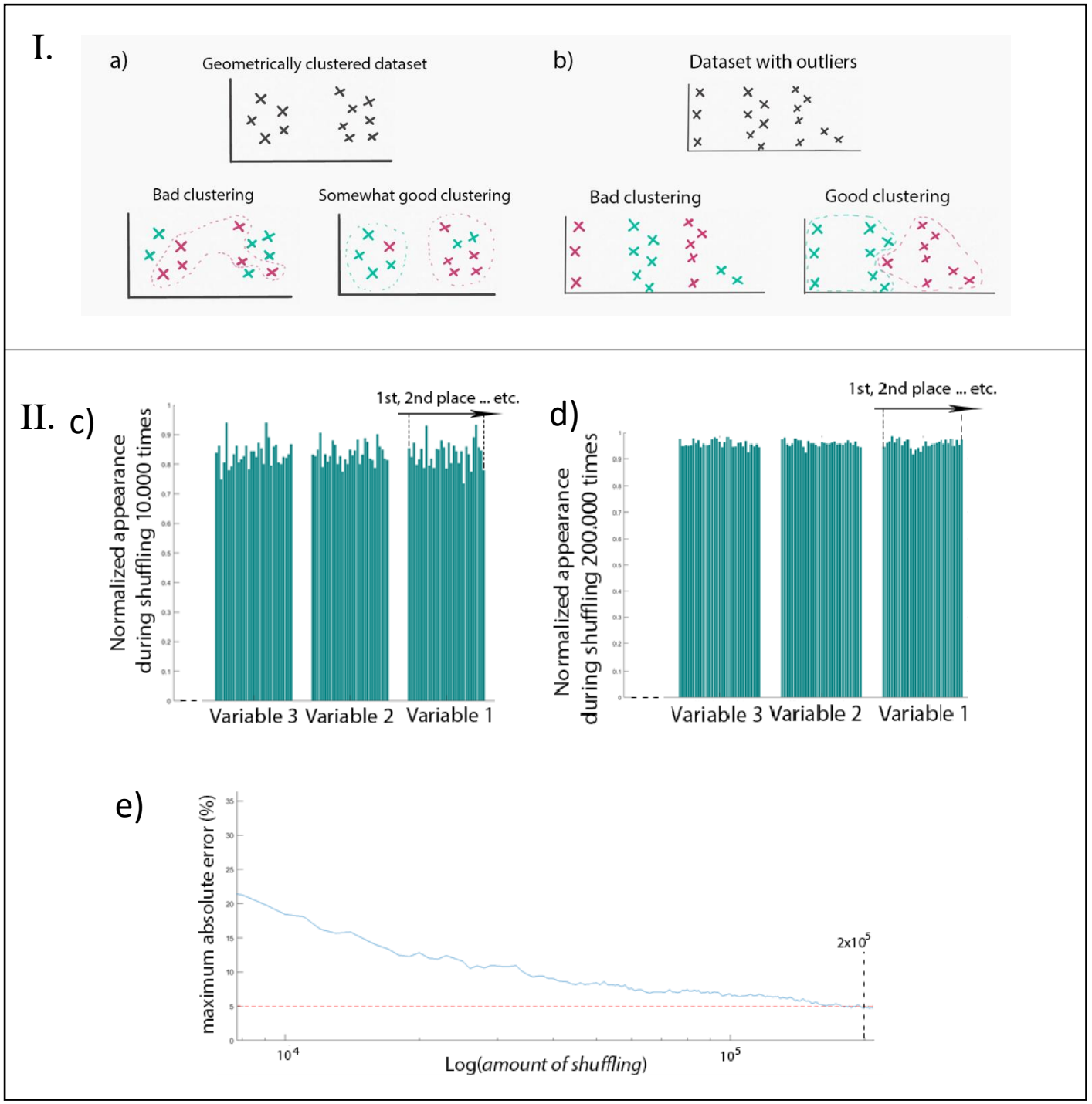


Figure 13 I. *Visualisation of the effect of the shape of the data on the classification probability.* ***a)*** *Geometrically clustered data has a higher chance of being separated into classes, but this classification does not have to follow the geometrical clustering.* ***b)*** *Data with outliers can bias the separation of the data as some datapoints are so far away that it is impossible to create a continuous connected cluster with a linear approach.* **II.** *The bar graphs of every variable are ordered from left to right from highest rank to lowest rank.* The shuffling was done with a dataset of in total 31 uniform random variables, with each 30 datapoints. **a)** *The distribution of the first 5 uniform variables after shuffling 10.000 times. The maximum absolute divergence to the mean is 19.66%.* **b)** *The distribution of the first 5 uniform variables after shuffling 200.000 times. The maximum absolute divergence to the mean is* 4.73%. **c)** *The error percentage plotted against the log of the amount of shuffling. The noise comes from the chance in mean that correlates to the error. The noise declines relatively as the number of shuffling increases. The error margin passes the 5% line after around 200.000 shuffles.*

Future functional grouping

In future research we can, with the results we have, further look into the correlation between the variables by grouping them by functionality such as: motor based, social based, intellect based, anxiety based etc. If a LD has set of dominant features related to a specific phenotypical functionality, we can use that LD to characterize a specific identity as was done in Forkosh et al. [1]

Loading plots

To make a step in this direction we can create a loading plot from the 10 best variables of the LDA to see if these variables create any clusters. This is done by looking at the relation of LD1 and LD2 with respect to every variable. In Figure 14, we see the loading plot using only the 10 most contributing variables. It shows that the variables can be clustered into 2 or 3 groups, indicated with the red, blue and optional green circle. It is remarkable that the blue circle consists of variable mostly from the same experiment: "Erasmus Ladder". This is the first step to grouping the variables into different functional clusters. *(see **Supplementary Figure 4** for the loading plot with all the variables)*

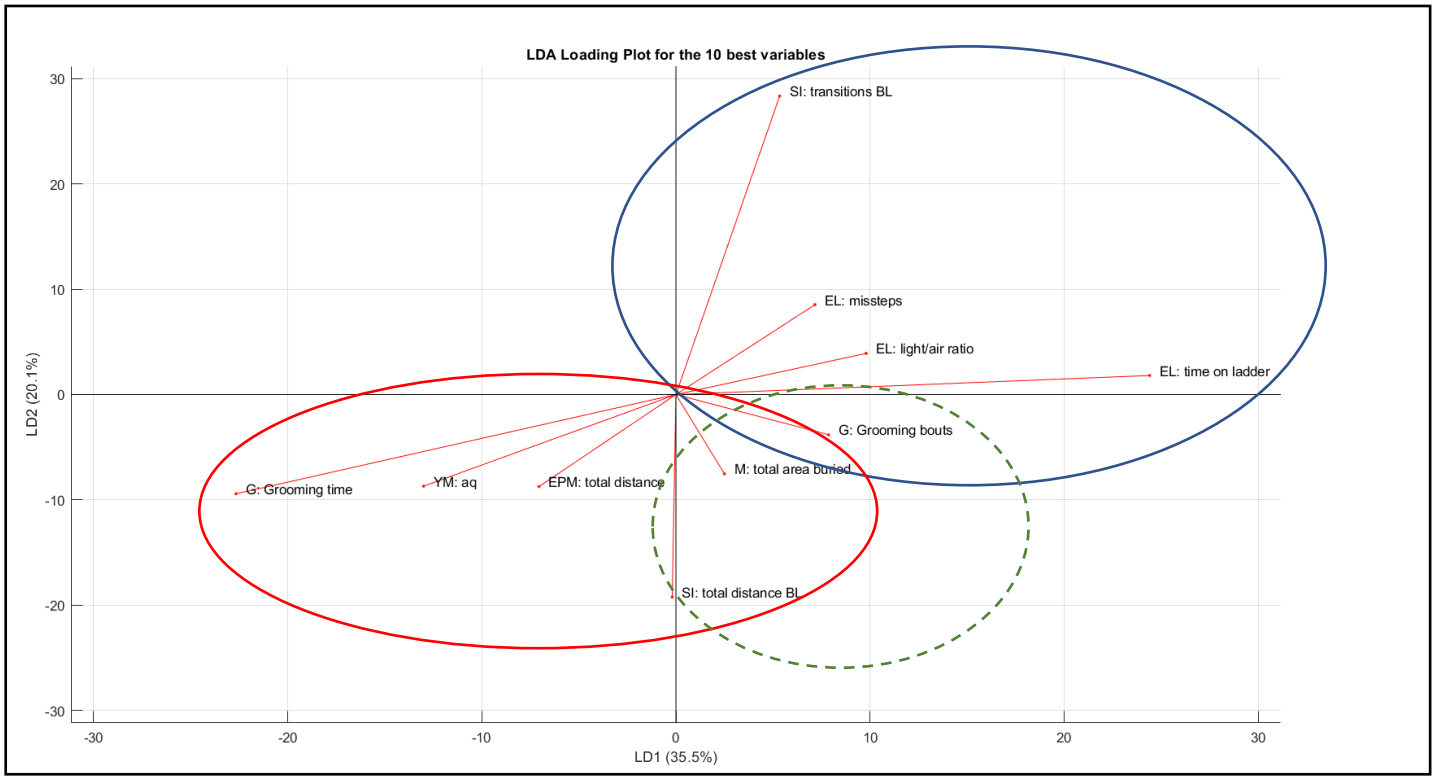


Figure 14 *Loading plots of the 10 best variables for LD1 and LD2, the percentage on the y-axis stands for the eigenvalue contribution of that LD with respect to all other LD's.*

LDA code development goal

With this analysis tool, focussed on LDA in Matlab, we tried to give an insight into the significance order of variables when dealing with a modest big dataset containing 2 classes. While developing this tool, the reusability was kept in mind, the applicability for a range of different datasets and the user-friendliness of the code. The goal was to develop a tool that can be used by researchers that are familiar with handling pre-written code, but not hard-coding themselves. By encapsulating certain frequently used functions, the amount of coding the user needs to do to get a certain result, gets reduced. Also by generalizing the function to be input dependent of the particular dataset itself, the code gets more flexible; the only requirement now for a dataset to get an automatic output is to only use 2 classes and to set the class size, besides being in the right format. The amount of lines of code to perform individual parts of the main analyses, like the LDA calculation, mostly do not exceed 5 lines. There is always a balance between encapsulation and readability for the user and we tried to find that balance by extensive commenting and the existence of a function description document. Moreover we tried to use the right amount of input and output variables per function to give the user enough information to what those mean, but encapsulating it enough to prevent unnecessary writing.

Future enhancements: supporting multiple classes

One of the most important enhancements that can be made on the present tool is the extension to handle multiple classes. This is an important next step as a lot of biological research has a third class or even more, that function as a control, represent a heterozygous group or a second type of mutation. For this analysis it was not yet needed, but for further analysis the implementation of supporting multiple classes will be a priority.

The drawbacks of behaviour data and the role of LDA in defining research

Ideally we would like to make a defining connection with the Pi3Kδ gain of function mutation and the specific behaviour characteristics from our data. We did tried to look at the unique combination of strong correlating variables measured from the conducted experiments, like a fingerprint for this specific mutation, by using heatmaps, comparison tables and loading plots. But this fingerprint can only give us an indication of the right direction we have to go for further analysis as this outcome is not only heavily influenced by the variable combination we took, but also the measurements themselves; Because, like we mentioned in the introduction, to define a behaviour and to quantify it, we will always bring a certain amount of bias **[27].** The experiments we did, involved putting the mice in a controlled environment and measuring a specific behaviour by stimulating the mice under standard, but unnatural conditions. Examples of this are fixing the head of the mice, putting the mice in a white box for 30 minutes or letting the mice swim in a maze with only 3 possible directions. These experiments have to be standardized to enhance the reproducibility of the measurements, the recording of the experiments and consistency over the overall data, but this will simultaneously bias towards unnatural behaviour as these experiments are set to measure a specific task. The question is if these measurements actually measure a natural behaviour or if this data is biased by the setup of the experiment itself. Moreover is it important to note that when an experiment will show natural behaviour, how to quantify it. There are already some standardized methods, like measuring the average speed by taking the central point of mass or total travelled distance, but this will not give information for more complex behaviour. Putting a number on a specific behaviour almost always create a level of bias **[27].** Concluding, it is important to interpret the

outcome of behaviour analysis in its context. Especially for tools like LDA, it is important to combine its outcome with a combination of interdisciplinary research, like neural analysis, biomolecular research and other data analysis methods. The outcome of a LDA does not replace the outcome of another analysis, but will only help and validate the result of a combination of methods.

Future enhancements: Comparison with other analysis techniques

We already did combine different analysis methods, such as the PCA and LDA. From this combination of analyses we saw that there was quite a large difference between the two outcomes, but this is due to the difference between the two algorithms: PCA tells us something about the separation of the data, but not about the role of our genotype, while LDA takes the genotype labelling into account. It is still important to combine these two methods to look at the overall context of the data. By extending this principle and comparing the present methods with other classification methods such as Support Vector Machine (SVM) and k-fold cross-validation, we can have more in depth information about the validity of their outcomes. These two data analysis methods are based on testing the validity of an already classified dataset.

SVM is done by dividing the data in training data set and a testing one. It first creates a hyperplane to separate the defined classes from the LDA, focusing on the boundary point; the points that are the most difficult to classify **[28].** It then tries to classify the new data based on this hyperplane and scores the classifier, in our case the LDA, on its separability power for new data. Besides scoring the LDA we could also try to score the separability of our raw genotype labelling. The advantage of this method is that it can handle quadratic hyperplanes for data that is not linearly separated. The drawback of SVM is that it focusses on the points that are the most difficult to classify, while LDA will account for the whole distribution. The outcome of the SVM is also less interpretable. Moreover the non-linearity problem could be solved by using a Quadratic Discriminant Analysis (QDA) instead of LDA, if one chooses to use a similar approach. Lastly SVM has problems with working for more than 2 classes, while LDA can be expanded to multiple classes.

K-fold cross-validation works by randomising and dividing the data into k groups. One group will function as a training set. The rest of the k-1 groups act like the training data, by doing the LDA on this particular subset of the data **[29].** Subsequently, this fitted model will be evaluated and scored by testing it on the test data. By repeating this process multiple times with a different number groups the LDA separability power can be scored.

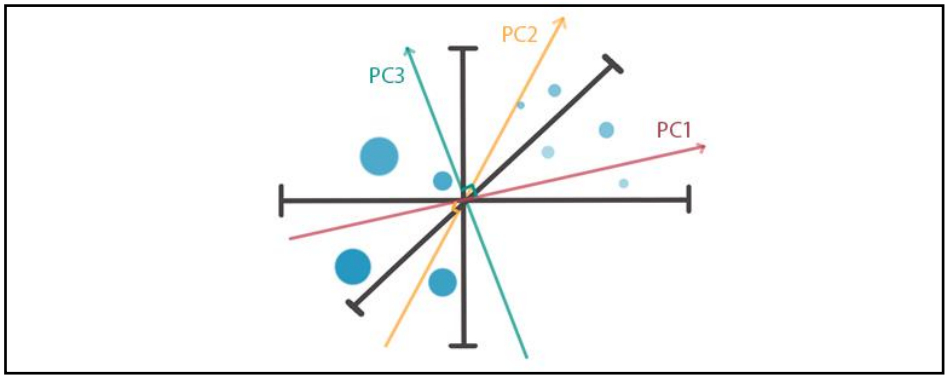
These two methods are just examples that can be implemented to enhance the validity and reproducibility of the current LDA method. This, with the expansion to handle multiple classes and the user friendly coding approach, all have the common goal of making this implemented method usable for future research.

References

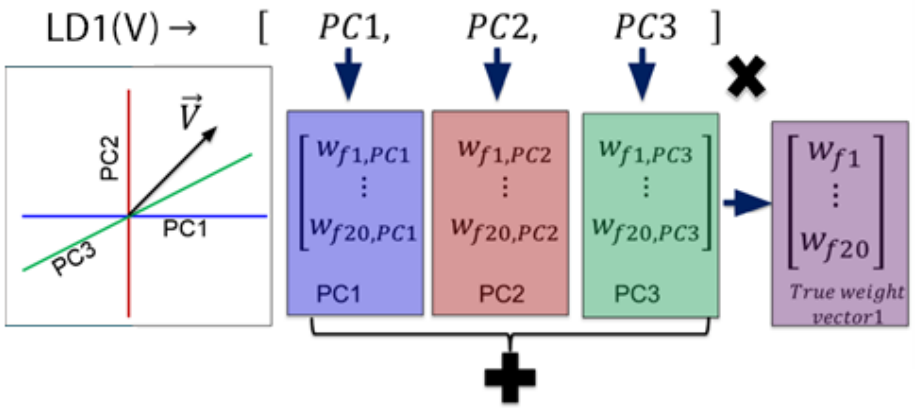
1. Forkosh, O., Karamihalev, S., Roeh, S., Alon, U., Anpilov, S., Touma, C., . . . Chen, A. (2019). Identity domains capture individual differences from across the behavioral repertoire. *Nature Neuroscience, 22*(12), 2023-2028. doi:10.1038/s41593-019-0516-y
2. (n.d.). Retrieved November 29, 2020, from https://www.psychiatry.org/patients-families/autism/what-is-autism-spectrum-disorder
3. Costa-Mattioli, M., & Monteggia, L. M. (2013). MTOR complexes in neurodevelopmental and neuropsychiatric disorders. *Nature Neuroscience, 16*(11), 1537-1543. doi:10.1038/nn.3546
4. Onore, C., Yang, H., Water, J. V., & Ashwood, P. (2017). Dynamic Akt/mTOR Signaling in Children with Autism Spectrum Disorder. *Frontiers in Pediatrics, 5*. doi:10.3389/fped.2017.00043
5. Hoegenauer, K., Soldermann, N., Stauffer, F., Furet, P., Graveleau, N., Smith, A. B., . . . Zécri, F. (2016). Discovery and Pharmacological Characterization of Novel Quinazoline-Based PI3K Delta-Selective Inhibitors. *ACS Medicinal Chemistry Letters, 7*(8), 762-767. doi:10.1021/acsmedchemlett.6b00119
6. Hood, V. L., Paterson, C., & Law, A. J. (2020). PI3Kinase-p110δ Overexpression Impairs Dendritic Morphogenesis and Increases Dendritic Spine Density. *Frontiers in Molecular Neuroscience, 13*. doi:10.3389/fnmol.2020.00029
7. ISD-2014, Structured Data vs. Unstructured Data: The Balance of Power Continues to Shift
8. Todd, J. G., Kain, J. S., & Bivort, B. L. (2017). Systematic exploration of unsupervised methods for mapping behavior. *Physical Biology, 14*(1), 015002. doi:10.1088/1478-3975/14/1/015002
9. Hart, P. E. (1973). *Pattern classification and scene analysis: Richard O. Duda, ... Peter E. Hart, ..* New York: J. Wiley & Sons.
10. Jean, S., & Kiger, A. A. (2014). Classes of phosphoinositide 3-kinases at a glance. *Journal of Cell Science, 127*(5), 923-928. doi:10.1242/jcs.093773
11. Fruman, D. (2017). Faculty Opinions recommendation of Conformational disruption of PI3Kδ regulation by immunodeficiency mutations in PIK3CD and PIK3R1. *Faculty Opinions – Post-Publication Peer Review of the Biomedical Literature*. doi:10.3410/f.727289141.793531351
12. Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*(4), 433-459. doi:10.1002/wics.101
13. PCA. (n.d.). Retrieved September 04, 2020, from https://nl.mathworks.com/help/stats/pca.html
14. Class Linear Discriminant Analysis. (n.d.). Retrieved November 29, 2020, from https://multivariatestatsjl.readthedocs.io/en/latest/mclda.html
15. Class Scatter Matrix. (n.d.). Retrieved November 29, 2020, from https://www.sciencedirect.com/topics/engineering/class-scatter-matrix
16. Knoke, J. D. (1982). Discriminant Analysis with Discrete and Continuous Variables. *Biometrics, 38*(1), 191. doi:10.2307/2530302
17. Normplot. (n.d.). Retrieved November 29, 2020, from https://nl.mathworks.com/help/stats/normplot.html
18. Gomez, M., Benzo, Z. D., Gomez, C., Marcano, E., Torres, H., & Ramirez, M. (1990). Comparison of methods for outlier detection and their effects on the classification results for a particular data base. *Analytica Chimica Acta, 239*, 229-243. doi:10.1016/s0003-2670(00)83857-x

19. Isoutlier. (n.d.). Retrieved September 04, 2020, from https://nl.mathworks.com/help/matlab/ref/isoutlier.html
20. Cai, T. T., & Zhang, L. (2019). High dimensional linear discriminant analysis: Optimality, adaptive algorithm and missing data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 81*(4), 675-705. doi:10.1111/rssb.12326
21. https://nl.mathworks.com/help/stats/classification-example.html
22. Ng, M. K., Liao, L., & Zhang, L. (2010). On sparse linear discriminant analysis algorithm for high-dimensional data classification. *Numerical Linear Algebra with Applications, 18*(2), 223-235. doi:10.1002/nla.736
23. Ekenel, H. K., & Stiefelhagen, R. (2007). Two-class Linear Discriminant Analysis for Face Recognition. *2007 IEEE 15th Signal Processing and Communications Applications*. doi:10.1109/siu.2007.4298761
24. *Springer texts in statistics*. (n.d.). New York: Springer
25. S. Raychaudhuri, "Introduction to Monte Carlo simulation," 2008 Winter Simulation Conference, Miami, FL, USA, 2008, pp. 91-100, doi: 10.1109/WSC.2008.4736059.
26. Tallon-Ballesteros, A. J., & Riquelme, J. C. (2014). Deleting or keeping outliers for classifier training? *2014 Sixth World Congress on Nature and Biologically Inspired Computing (NaBIC 2014)*. doi:10.1109/nabic.2014.6921892
27. Berman, G. J. (2018). Measuring behavior across scales. BMC Biology, 16(1). doi:10.1186/s12915-018-0494-7
28. Pupale, R. (2019, February 11). Support Vector Machines(SVM) - An Overview. Retrieved December 02, 2020, from https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989
29. Brownlee, J. (2020, August 02). A Gentle Introduction to k-fold Cross-Validation. Retrieved December 02, 2020, from https://machinelearningmastery.com/k-fold-cross-validation/
30. Ben-Shaul, Y. (2017). OptiMouse: A comprehensive open source program for reliable detection and analysis of mouse body and nose positions. *BMC Biology, 15*(1). doi:10.1186/s12915-017-0377-3
31. Ey, E., Leblond, C. S., & Bourgeron, T. (2011). Behavioral profiles of mouse models for autism spectrum disorders. *Autism Research, 4*(1), 5-16. doi:10.1002/aur.175

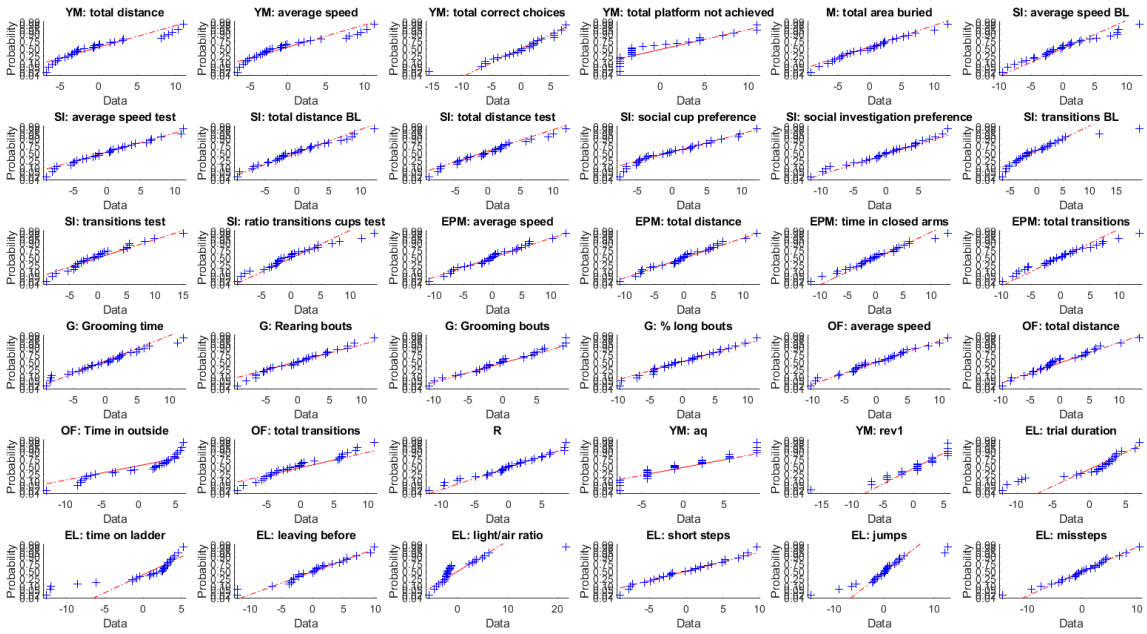
SUPPLEMANTARY



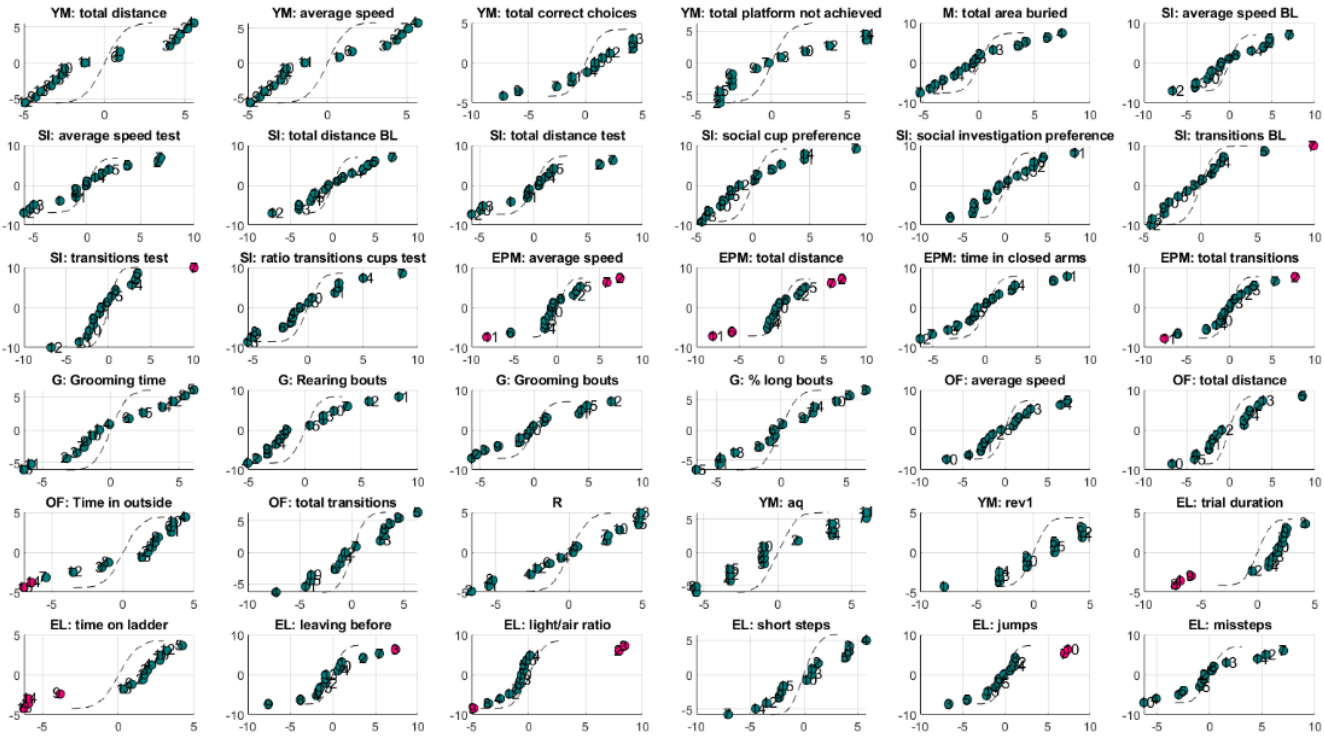
Supplementary Figure 1 *An visualisation of the orientations of the eigenvectors in a 3 dimensional space. Each PC is perpendicular to the previous set of eigenvectors.*



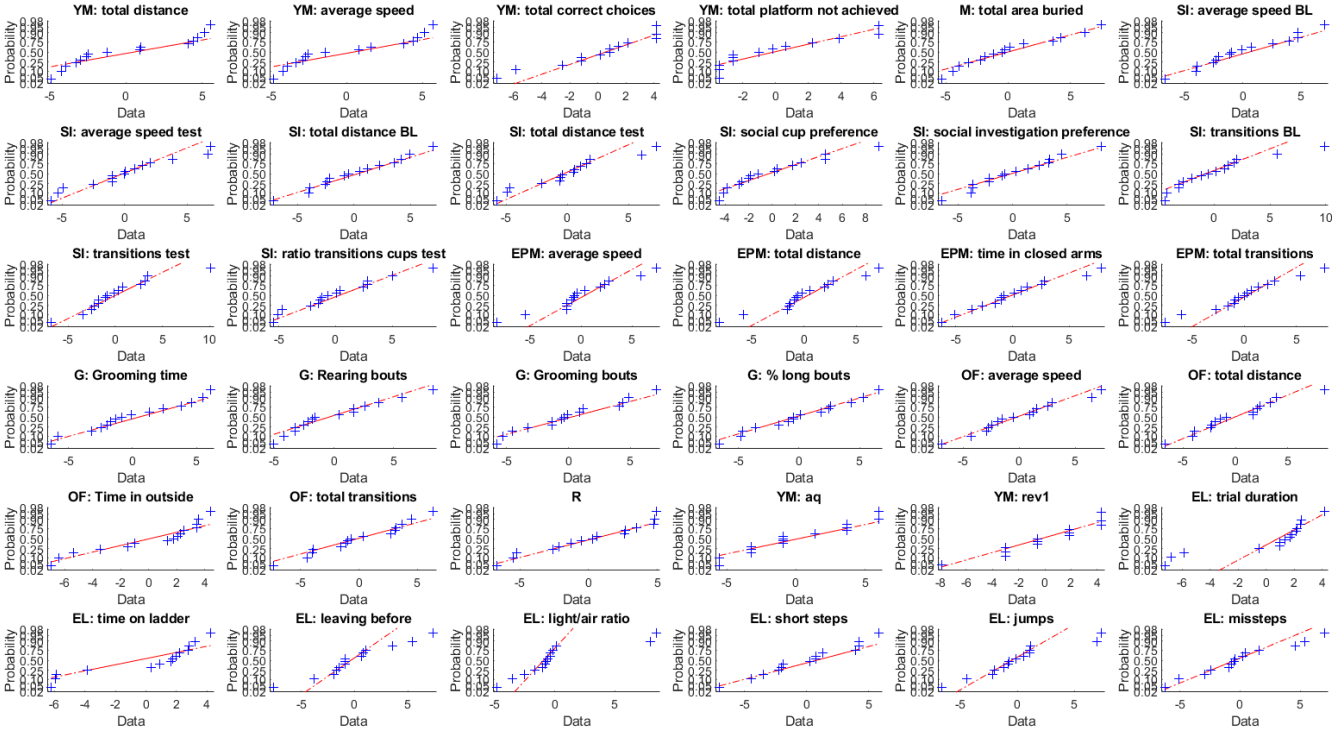
Supplementary Figure *2* *visual representation of the weight conversion that is needed when doing LDA on PCA data. The contribution values, or directional coefficients, of LD1 need to be multiplied with the directional coefficients of their corresponding PC. All this need to be added up to create the true weight LD vector.*



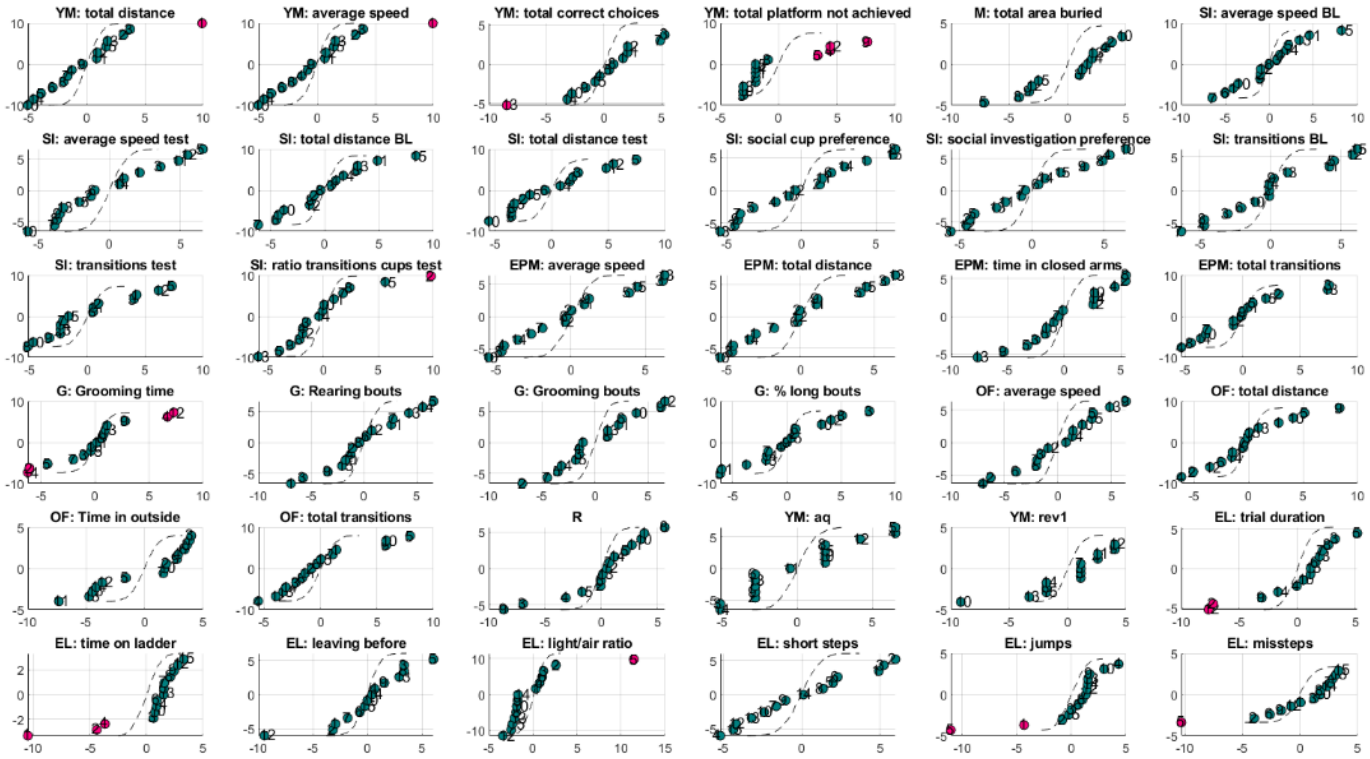
Supplementary Figure 3a *Normplots as a whole*



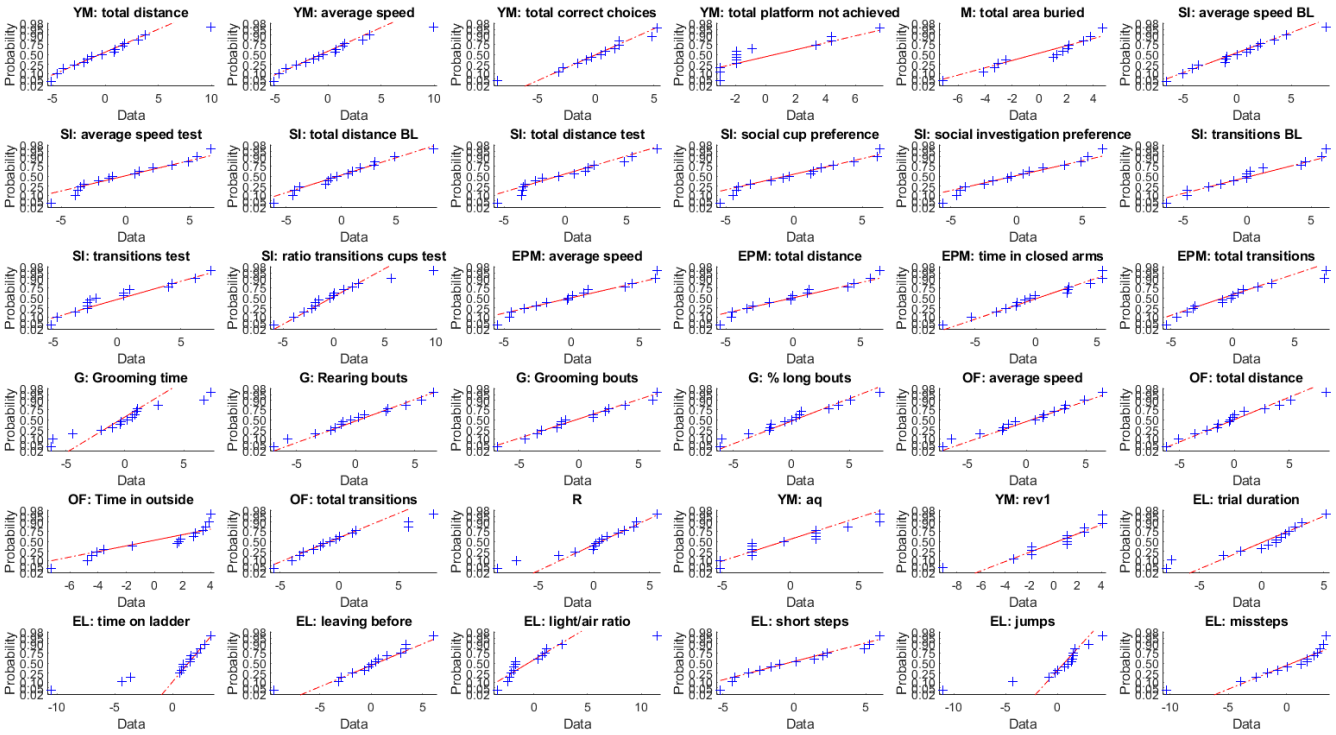
Supplementary Figure 3b *Distributions of class 1 before outlier removal*



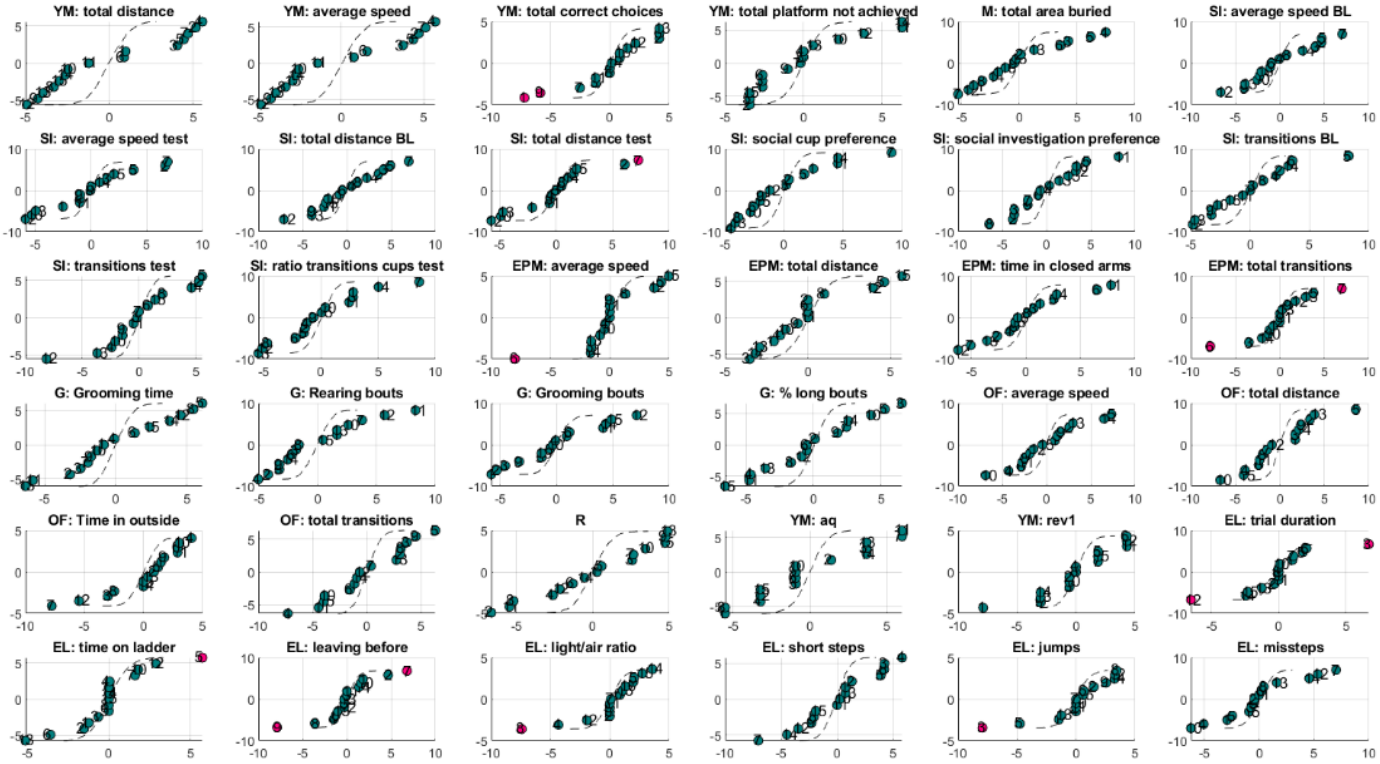
Supplementary Figure 3c *Norm plots of class 1 before outlier removal*



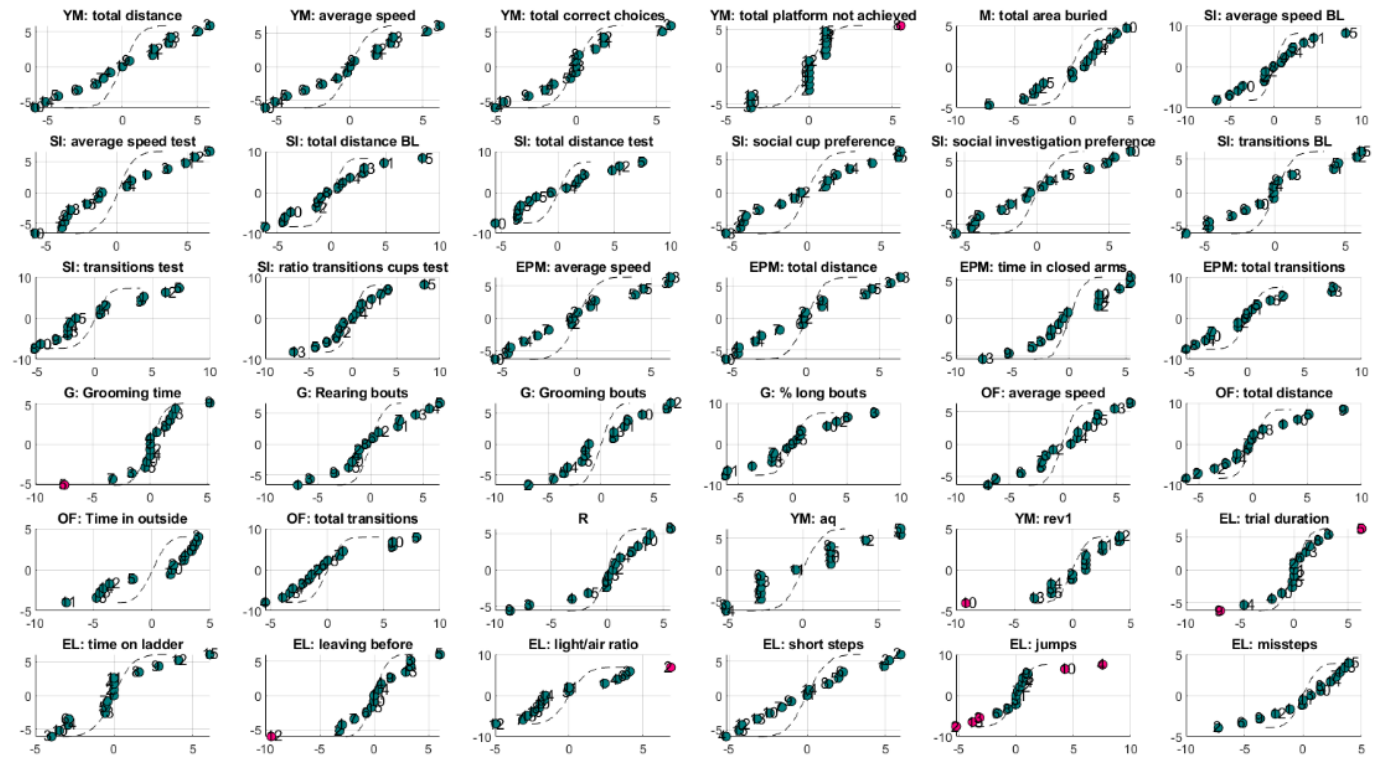
Supplementary Figure 3d *Distributions of class 2 before outlier removal*



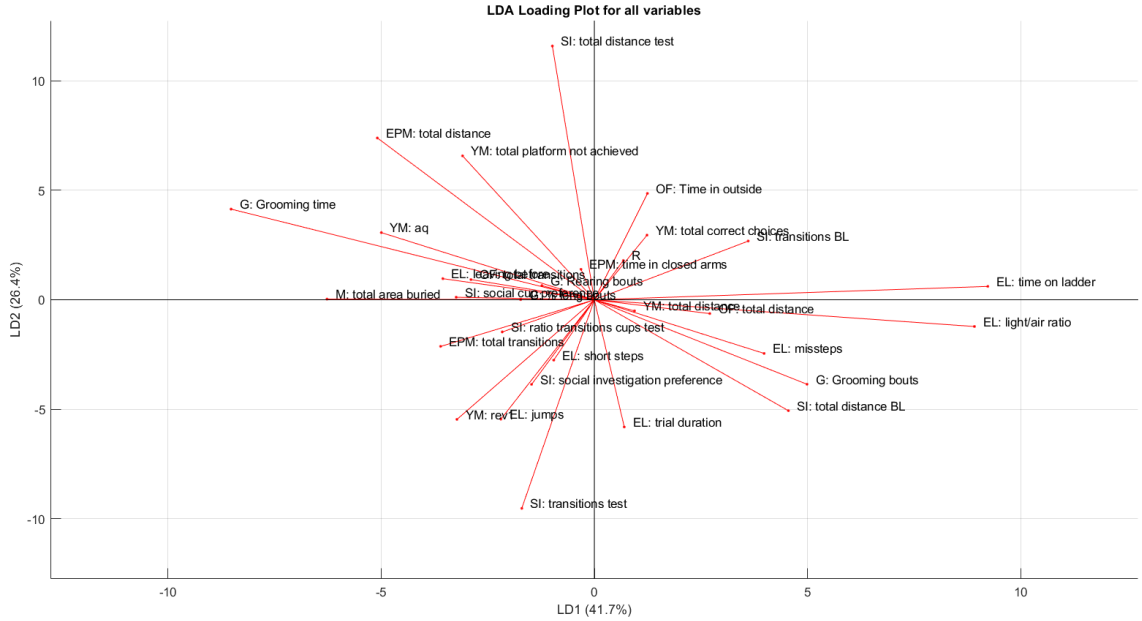
Supplementary Figure 3e *Norm plots of class 2 before outlier removal*



Supplementary Figure 3f *Distributions of class 1 after outlier removal*



Supplementary Figure 3g *Distributions of class 2 after outlier removal*



Supplementary 4 *Loading plot with all the variables for LD1 and LD2.*