Progeno
identifies genetic potential

Progeno Analytics Tutorial

Steven Maenhout

17-04-2020

Progeno BVBA | Technologiepark 82, B-9052 Zwijnaarde | Belgium
Tel: +32 (0)499 45 60 05 | E-mail: info@progeno.net | Site: www.progeno.net
TAV: BE 0508.909.015 | IBAN: BE28 0016 8499 8720 | BIC/SWIFT: GEBABEBB

Contents

1 Introduction

The Progeno Analytics module provides a comprehensive set of tools for breeding data manipulation, visualization and analysis. The Analytics module relies heavily on R which is an open-source software environment for statistical computing and graphics (https://www.r-project.org/). The Progeno Analytics module sends all analysis and visualization tasks to an R session that runs in the background and presents the results by means of a graphical user interface. Users do not require knowledge of the R scripting language to use the Progeno Analytics module. However, users that are familiar with R can switch back and forth between the graphical user interface and the R scripting environment at their convenience.

This document provides a tutorial that covers most analysis and visualization tools of the Progeno Analytics module. The examples extract data from a dedicated Tutorial database which is accessible for all EUCLEG project partners. This Tutorial database can be accessed by logging in to the EUCLEG Progeno server at https://eucleg.progeno.tk. While the Progeno web application works with all major browsers, the best performance is achieved by using the Google Chrome browser (https://www.google.com/intl/nl/chrome/). To access the EUCLEG Progeno server you need to provide your username and password that also give access to the EUCLEG collaborative workspace. Select the Tutorial database as shown in Figure 1 and press the login button.

Figure 1: Login screen of the Progeno web application

Once logged in, the Analytics module can be accessed by clicking on the Analytics button in the module button array on the left side of the screen. This will replace the module button array by the application button array of the Progeno Analytics module. You can always return to the module selection buttons by clicking on the “SELECT MODULE” bar that is on the left side of the button array.

2 Data management

The Data management app provides a toolbox to save, restore and manipulate datasets. These datasets are used as input to the other Analytics apps that effectively analyse or visualize the data. Figure 2 shows the Data management screen when the app is activated for the first time. The ACTIVE DATA panel is the only active part of the screen and lists the datasets that are currently active. This list is obviously empty if this is the first time you use the Progeno Analytics module. Below the ACTIVE DATA header there are three buttons namely "Load", "Upload Excel" and "Upload RDS".

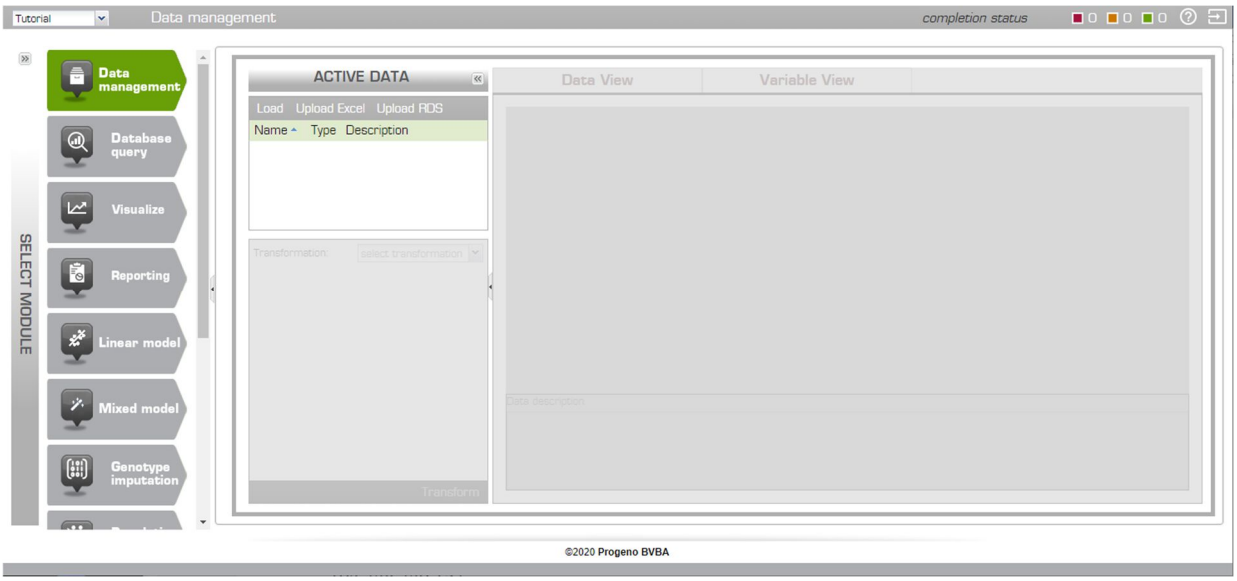


Figure 2: View of the Data management app when activated for the first time.

Clicking on the "Load" button opens the "Database explorer" screen depicted in Figure 3. The left part of this window contains two entries namely "Personal" and "Shared". These represent two distinct data spaces namely the personal storage space where the user can save private datasets and the shared storage space that is accessible by all users. Clicking on the "Shared" node opens this shared storage space which has a subdirectory named "Tutorial". Clicking on this Tutorial nodes shows the available tutorial datasets in the right part of the screen. You can load one of these datasets, for example the carrot data, by double-clicking on the name of the dataset. This closes the Database explorer screen and loads the carrot dataset in the active data space. The "Data View" and "Variable View" tab panels on the right part of the screen provide respectively a raw and summarized view of the selected dataset.

A dataset that is loaded in the ACTIVE DATA panel is basically a copy of the dataset that is stored in the database. Making changes to such an active dataset does not affect the original dataset that is stored in the database. You can, however, store your (possibly altered) dataset by right-clicking on the name of the dataset in the ACTIVE DATA panel and choosing the "Save" or "Save as" menu options. This same menu also provides the options to Copy, Delete, Rename and Export the active dataset. Clicking on the Export menu entry generates a download link that allows to download the dataset to your local hard drive as an Excel file.

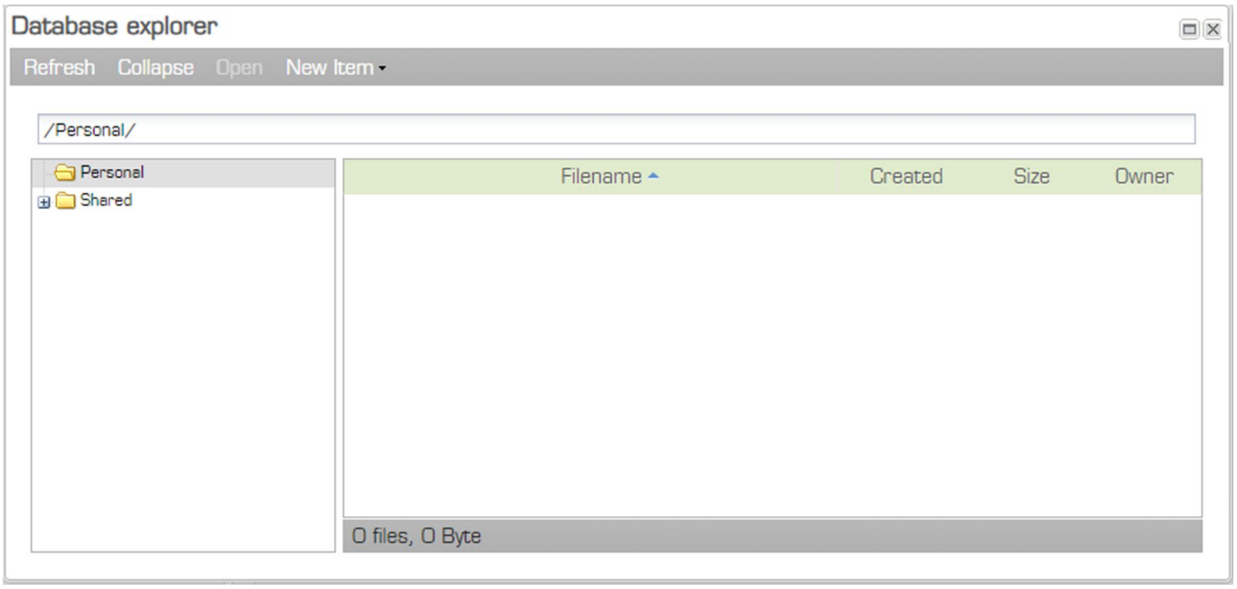


Figure 3: The Database explorer screen of the Data management app.

Active datasets are preserved over Progeno sessions which means that you can logout or close your browser window and the datasets will still be active the next time you login to Progeno. However, it is generally advised to save important datasets in your Personal storage space as server updates and/or reboots might clear all active datasets.

2.1 Variable View

The Variable View panel provides a summary of the dataset that has been selected in the ACTIVE DATA panel. Each column an active dataset has one of the following types:

- numeric: integer and decimal numbers
- character: textual data
- factor: limited set of textual values like A, B and C
- logical: (TRUE or FALSE)

The Variable View separates the numeric variables (quantitative) from the non-numeric variables (categoric) and provides a summary of the observed values for each column.

2.2 Data View

The Data View provides a spreadsheet-like interface to the dataset that is selected in the ACTIVE DATA panel. You can make changes to the dataset by clicking in a particular cell and changing its value. This kind of change will add a small red triangle at the top left of the dataset name in the ACTIVE DATA panel. This red triangle means that the dataset is now different from the data that is stored on disk and that you need to save the data if you wish to overwrite the original dataset.

Below the header of this spreadsheet there is an empty row that allows to filter the rows of the active dataset. For numeric variables one can enter equality, greater than and less than

constraints. For textual variables a partial text can be provided which will filter out the rows that have that have a matching value. If, for example, one types character “x” in the filter field of the Entry column of the carrot dataset only rows that contain the character x in the Entry column are shown.

Applying a filter to a dataset merely changes the visual representation of the data while the dataset itself remains unchanged. You can, however, create a new dataset that only contains the filtered rows. For this we need to select the “Copy filtered data” option in the Transformation drop-down box below the ACTIVE DATA panel as show in Figure 4. In this example I have used the carrot dataset which was loaded from the Shared storage space. I have filtered the rows of this dataset as to only show results that contain the character ‘x’ in the Entry column. In the “Data name” text field below the Transformation drop-down box I have typed the name ‘filteredCarrot’. Clicking the “Transform” button below this text field generates a new dataset that only contains the filtered rows. This new dataset contains but 10 of the 60 rows that are in the original carrot dataset.

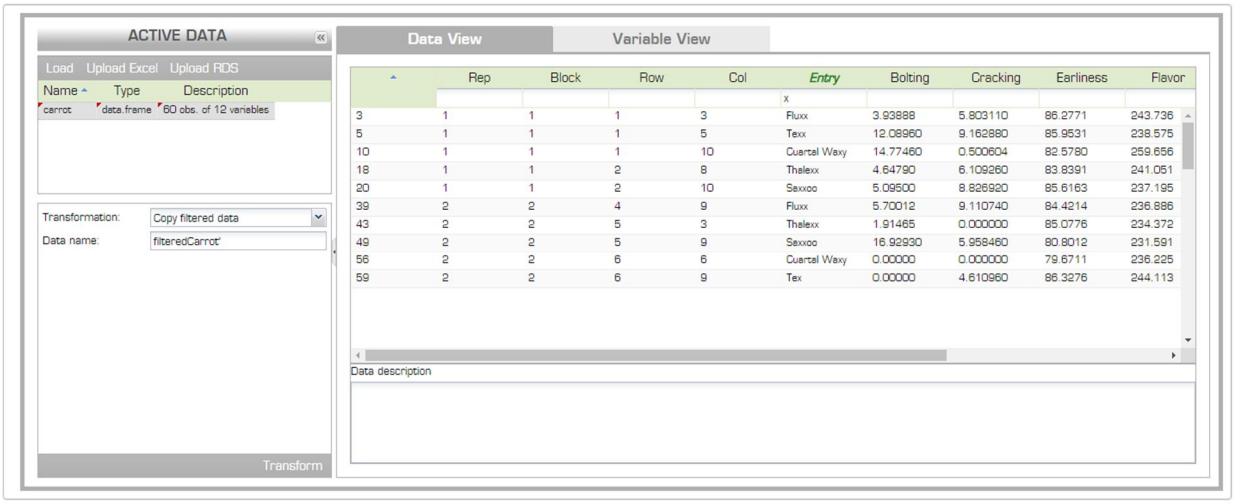


Figure 4: Creating a new dataset by filtering.

2.3 Transformations

- The Transformation drop-down menu provides various other data manipulation functions:
- Add rows from other datasets: combines the rows of two datasets
- Add variable: adds a new column to the dataset
- Aggregate: calculate summaries for numeric variables
- Change variable type: transform numeric variables to character variables and vice-versa
- Copy filtered data: copies filtered rows into a new dataset
- Delete variables: remove column(s) from the dataset
- Merge with other dataset: combine the columns of two datasets
- Rename variable: rename a column

To demonstrate the “Merge with other dataset” transformation we will merge the carrot dataset with another dataset from the Shared storage named “otherCarrotData”. You open this other dataset by clicking the Load button and navigating to the Tutorial node under the Shared node and double-clicking the “otherCarrotData” entry.

You should now have both the “carrot” and the “otherCarrotData” datasets in the ACTIVE DATA panel. The “otherCarrotData” contains additional observations of the trait “ForgottenTrait”. We want to add this ForgottenTrait column to the carrot dataset but we have to make sure that the observations are matched to the correct entry and replication. To achieve this we perform the following steps:

- click on the carrot dataset in the ACTIVA DATA panel
- select the “Marge with other data set” from the Transformation drop-down menu
- select both “Entry” and “rep” (in that exact order) in the Merge column(s) drop-down menu
- keep all rows remains checked
- select “otherCarrotData” from the Other data drop-down menu
- select both “Entry” and “rep” (in that exact order) in the Other column(s) drop-down menu
- keep other rows remains checked
- type “mergedCarrot” in the Data name text field

Figure 5 shows how the Data management screen looks like when all of the above options have been set. Clicking the Transform button produces the new mergedCarrot dataset that combines the columns of the carrot and otherCarrotData datasets. We can save this mergedCarrot dataset to our Personal storage space by right-clicking on the mergedCarrot entry in the ACTIVE DATA panel and selecting Save from the menu. This opens the Database explorer window in which we click on the Personal node and press the Save button on the bottom right of the screen to store the dataset under the default name “mergedCarrot”.

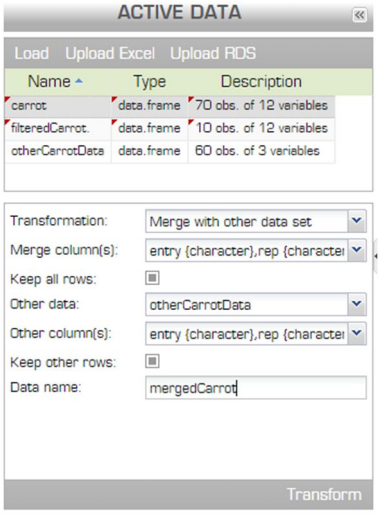


Figure 5: Example of the Merge with other data set transformation.

3 Database query

The buttons in the header of the ACTIVE DATA panel allow to retrieve datasets from the Personal or Shared storage space, upload data as an Excel file and upload RDS files (i.e. compressed native R objects). The Database query app allows to query the data from the actual Progeno database. The app allows to extract trial observations, passport information, pedigree records, molecular markers scores and (genomic) breeding values.

The screen of the Database query app is split in a top and a bottom part. The top part contains two tabs named “Trial” and “Individual” and the bottom part contains the QUERY RESULTS grid. The idea is to filter the data (i.e. build the database query) in the top part of the screen and load the resulting data records in the bottom part of the screen.

The two tabs in the top part allow to filter out the data you wish to retrieve from the database:

1) Trial: filter on trial records
2) Individual: filter on accession properties

3.1 Trial filters

The Trial tab serves to filter on trial records. The panels below the CHOOSE A FILTER header contain all the known properties of the trials including the trial name, organization, country, … You define a filter by double-clicking on an entry in one of these panels. For example, if I wish to retrieve the observations for all trials that were conducted in the Czech Republic, I double-click the CZE entry in the COUNTRY panel. You can also drag-and-drop the CZE entry to the ACTIVE FILTERS panel on the left of the screen. In both case the ACTIVE FILTERS panel now contains the entry “Country: CZE”. The content of all filter panels are now adjusted so they only show values that are relevant to trials conducted in the Czech Republic. Apparently there are two trials named “ART trial 2018” and “ART trial 2019”. The EUCLEGID panel contains all accessions that have been tested in these two trials and the MONTH and DAY panels contain the observation dates. We further reduce the observation data by adding the following filters:

- year: 2018
- month: 10
- day: 29

This combination of filters implies that we wish to retrieve the observations made in the “ART trial 2018” on October 29 of 2018. If we press the “Fetch Trial data” button in the header right to QUERY RESULTS we are asked to select the traits we wish to retrieve. Clicking on the “Select trait(s)” field opens the drop-down menu in which we can select one or more traits that have been observed. We click on “Select all” and then on the “Continue” button at the bottom right of this popup window. 140 observation rows have now been loaded in the QUERY RESULTS panel at the bottom of the screen. We can enlarge this section by moving the mouse pointer just above the QUERY RESULTS (140) header and dragging the panel upwards. Figure 6 shows how this screen looks like after expansion of the QUERY RESULTS section.

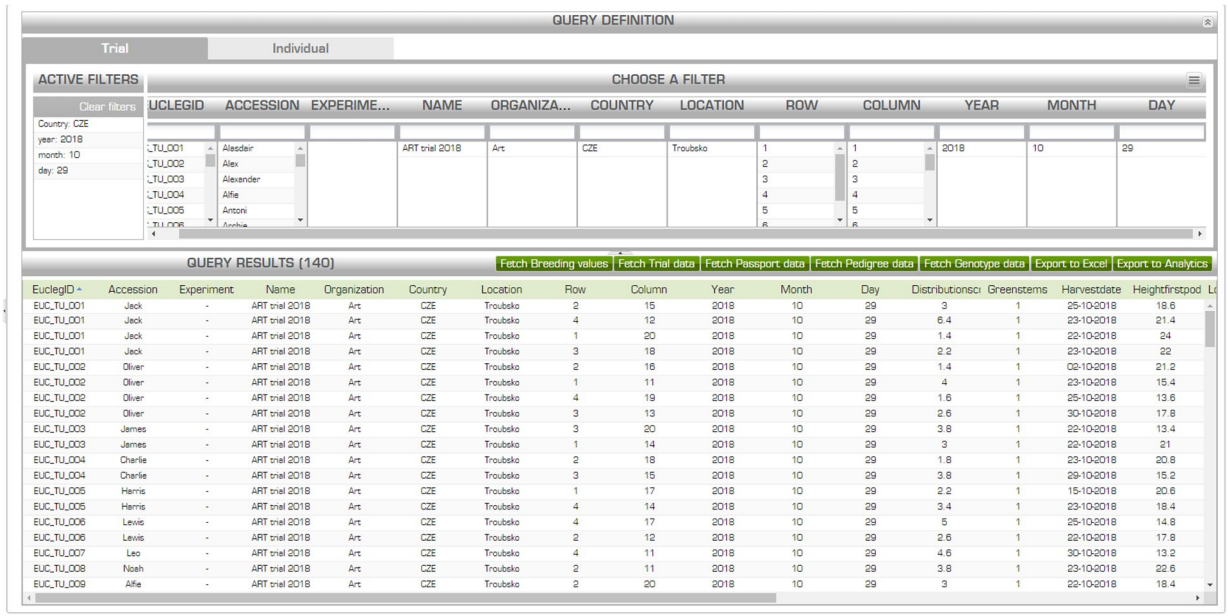


Figure 6: Example of filtering on trial records in the Database query app.

We can now export these query results to the Progeno Analytics module by pressing the "Export to Analytics" button. We are asked to provide a name for this dataset. The default name is "ResultsAndValuesExport" but that is not very informative so I replace this name with the value "CZE.2018.10.29" and press the ok button. This opens the Data management app in which the CZE.2018.10.29 dataset is loaded in the ACTIVE DATA panel.

Notice how this dataset contains a column for each of the filter panels in the Database query app. Unknowingly we selected the default representation which means that each filter becomes a separate column and each trait becomes a separate column.

Imagine we would like to focus on the trait Rstage which is a visual score from 0 to 8 that represents the reproductive stage at the time of observation. We want each observation date of this trait to become a separate column. One could also say that month and day of observation should be moved from the rows to the columns. To achieve this we perform the following actions:

- click on the Database query app button
- click on the Clear filters button just below the ACTIVE FILTERS window
- double-click value CZE in the COUNTRY filter panel
- double-click 2018 in the YEAR filer panel
- click the "Fetch Trial data" button
- select the trait "Rstage" in the Trait selection window
- click on the Row/column tab header of this popup window
- scroll down the list and select "columns" in the drop-down menus for Month and Day
- press the Continue button

The QUERY RESULTS panel now contains the 16 observation dates of the trait Rstage as different columns.

Notice how this dataset still contains quite a few uninformative columns like Experiment, Name, Organization, … where each cell has the same value. We can remove these columns by clicking "Fetch Trial data" again, select the trait Rstage in the Trait selection tab and select the following configuration for the Row/column tab:

- Trait: columns
- EuclegID: rows
- Accession: rows
- Experiment: drop
- Name: drop
- Country: drop
- Location: drop
- Block1: drop
- Row: rows
- Column: rows
- Year: drop
- Month: columns
- Day: columns

Pressing Continue produces a dataset of 140 rows in which all of these uninformative columns have been removed.

You might not be particularly interested in the row and column coordinates of the plots and therefore choose to drop these as well. This will, however, result in a dataset that contains only 100 rows instead of the 140 expected rows. This can be explained by the fact that there are exactly 100 accessions in this trial. 8 of these accessions have 4 plots, 16 accessions have 2 plots and the remaining 76 accessions only have 1 plot resulting in a total of 140 plots. If we include the row and column coordinates of the plots in the query, the system returns the observations for each individual plot which results in 140 rows. If we drop the row and column coordinates, the system returns the average observation for each accession, resulting in 100 rows.

The above example demonstrates that the properties (i.e. trait, accession, row, column…) that we put in the rows and the columns are used to uniquely define an observation. If we use a combination of trait, accession, row and column, the cells of the query result contain the observed trait values of each plot. If we only use the properties trait and accession, the cells of the query result contain the average trait observation per accession. In the next example we will take advantage of this feature to calculate trait averages over trials:

- click on the Clear filters button just below the ACTIVE FILTERS window
- hold down the Ctrl key and select the "ART trial 2018" and "ART trial 2019" in the NAME filter panel, drag-and-drop the two selected trials in the ACTIVE FILTERS panel
- click the Fetch Trial data button
- in the Trait selection tab of the popup window, click on the "Select trait(s) dropdown menu and click the Select all button at the top of the list

- in the Row/column tab, set Trait to rows, Name to columns and set all other properties to drop.
- click the Continue button

The query result containing 19 rows is shown in Figure 7. Each cell contains the average observation of a trait in one of the two selected trials. For some non-numeric traits like Harvest date and Vstage it is not possible to calculate the average of the observations. In this case the system simply returns one of the possibly many observed values.

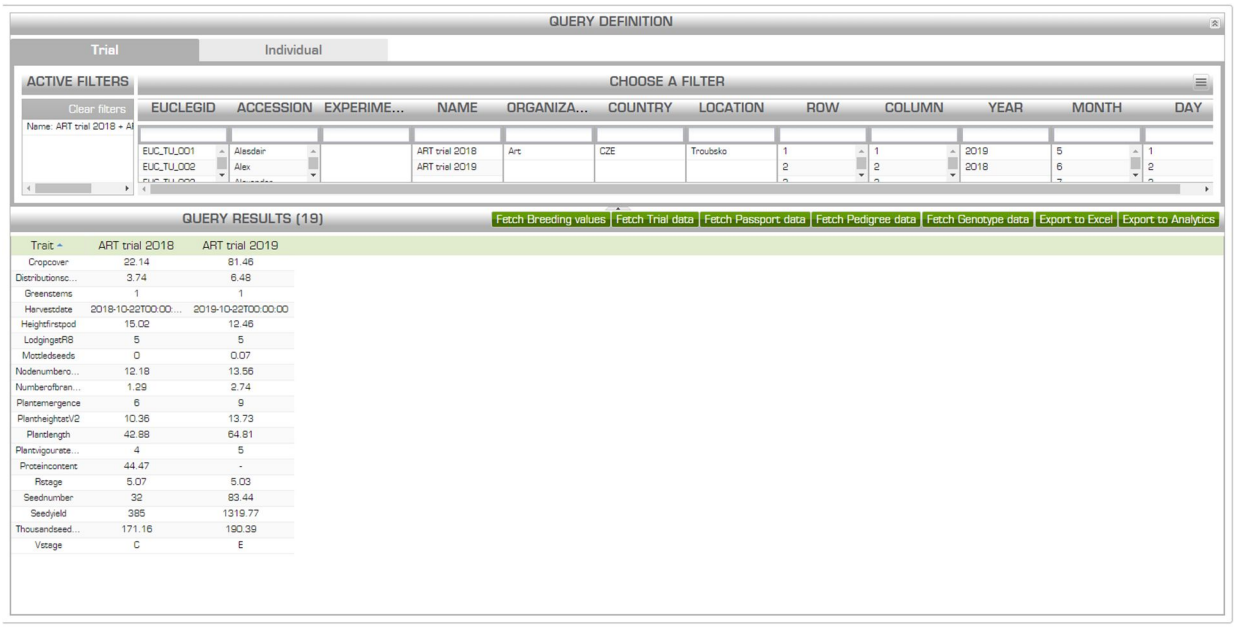


Figure 7: Example of a trial summary.

3.2 Individual filters

The trial filters provide a EUCLEGID and an ACCESSION filter which allow to retrieve observations for specific accessions. The Individual tab, however, makes it more convenient to select accessions for which you wish to retrieve observations. Both filter tabs work together meaning that if you filter on the trial Name “ART trial 2018” in the Trial tab and also define a filter on accession EUC_TU_001 in the Individual tab, the query result will contain observations on accession EUC_TU_001 in trial “ART trial 2018”.

The Individual tab consists of 3 panels (from left to right):

- the SELECTION panel containing the filtered individuals
- the CANDIDATES panel containing all accessions that are in the database
- the ACTIVE FILTERS panel containing passport filter definitions

The idea is to populate the SELECTION panel with the accessions for which you wish to retrieve the observations. There are three ways to do this:

1. drag-and-drop individuals from the CANDIDATES panel to the SELECTION panel

2. upload an Excel file containing the EuclegID's or accession names
3. loading a predefined selection of candidates (see Selection app in the Breeding activities and Decision support modules).

You can type part of the EuclegID or accession name in the text field just below the CANDIDATES header to reduce the list of candidates. If, for example, I type the characters "Dor" in this text field the list of candidates reduces to accession Theodore and Dorian. I can select both these entries by clicking them while holding down the Ctrl key. Then I drag-and-drop these accessions in the SELECTION panel. If you click the Fetch Trial data button, select all traits and press continue you receive all 69 observations on these accessions. If you receive the "No trial observations available" warning it is very likely that there are other filters still active in the Trial tab. If you remove these filters by clicking the "Clear filters" button below the ACTIVE FILTERS header on the Trial tab and click the Fetch Trial data button again, you should receive all 69 observation rows.

You can clear the SELECTION panel by clicking the "Clear Selection" button just below the SELECTION header in the Individual tab. Also remove the "Dor" characters in the text field below the CANDIDATES header so that all 642 accession are listed in the CANDIDATES panel.

We can now select individuals based on their passport properties. For example, we will retrieve all observations for accessions that originate from China. To achieve this we click on the "Add filter" button just below the ACTIVE FILTERS header in the Individual tab. In the Add filter popup window we choose the following settings:

- Filter: ProvenanceCountry
- Operator: Equals
- Value: CHN

Press the Save button. The ACTIVE FILTERS window now contains an entry "ProvenanceCountry = CHN" and the list of CANDIDATES has been reduced to 21 entries. We could select these one by one whilst holding down the Ctrl key but we can also select them all by right-clicking on any of the 21 accession and choosing the "Select all" option. We can now drag-and-drop all 21 accessions to the SELECTION panel. If we click on Fetch Trial data we can retrieve all observations in all trials for these Chinese accessions.

It should be clear from the list of buttons that the Database query app can be used to retrieve more than just trial observations. Pressing the Fetch Passport data button allows us to retrieve passport entries for each of the 21 Chinese accessions. In the Trait selection tab you can select the desired traits and click the Continue button. The QUERY RESULTS panel is now filled with 21 rows containing the selected passport properties of the selected accessions.

The Fetch Genotype data allows to extract molecular marker scores for the selected accessions. Pressing this button produces a Marker selection window in which each available marker is listed. You can filter markers by their chromosome, physical position, reference and alternative allele and by their percentage of completeness over the selected list of accessions. If you hover with your mouse over the filter field below the Complete column header a green equality sign appears. If we type in 100 the list of available markers is reduced from 212.689 markers to the

199.792 markers for which an allelic score is available for the 21 selected Chinese accessions. The number of candidate and filtered markers is printed at the bottom of the Marker selection window. If I also enter a 1 in the Chromosome filter I reduce the list further to the 9.973 complete markers on chromosome 1. Clicking on the checkbox at the top left corner of the Marker selection popup window selects these 9.973 markers. If you now click the "Done" button at the bottom right of the popup, the markers scores are loaded in the QUERY RESULTS panel.

The marker scores are expressed as the frequency of the reference allele. This means that for a diploid species the possible values are 0, 0.5 or 1. If you use the scrollbar at the bottom of the QUERY RESULTS panel to scroll to the right you will notice that there are only 100 columns instead of the expected 9.973 columns. This is related to a technical limitation of the browser that limits the number of columns that can be shown in a table. If you press the Export to Analytics button and give your dataset a name like "Chrom1.CHN" the matrix containing 21 rows and 9.973 columns is exported to the Progeno Analytics module. This matrix can now be used for GWAS, genomic prediction or other marker-based analysis methods.

4 Phenotypic data analysis

4.1 Single trial analysis

Most EUCLEG field trials make use of partially replicated (p-rep) designs which prevent the use of traditional analysis methods, such as the Analysis of Variance (ANOVA), that require balanced trial designs. While there are modifications of these linear model-based approaches that allow for the analysis of unbalanced datasets, the linear mixed model framework is the preferred workhorse in an unbalanced setting. The Progeno software provides a linear mixed model engine that can be used to fit all sorts of linear mixed models. The Progeno Analytics module provides a graphical interface to fit a subset of linear mixed model definitions. The Progeno R library allows to fit more advanced models but requires some knowledge of the R scripting language.

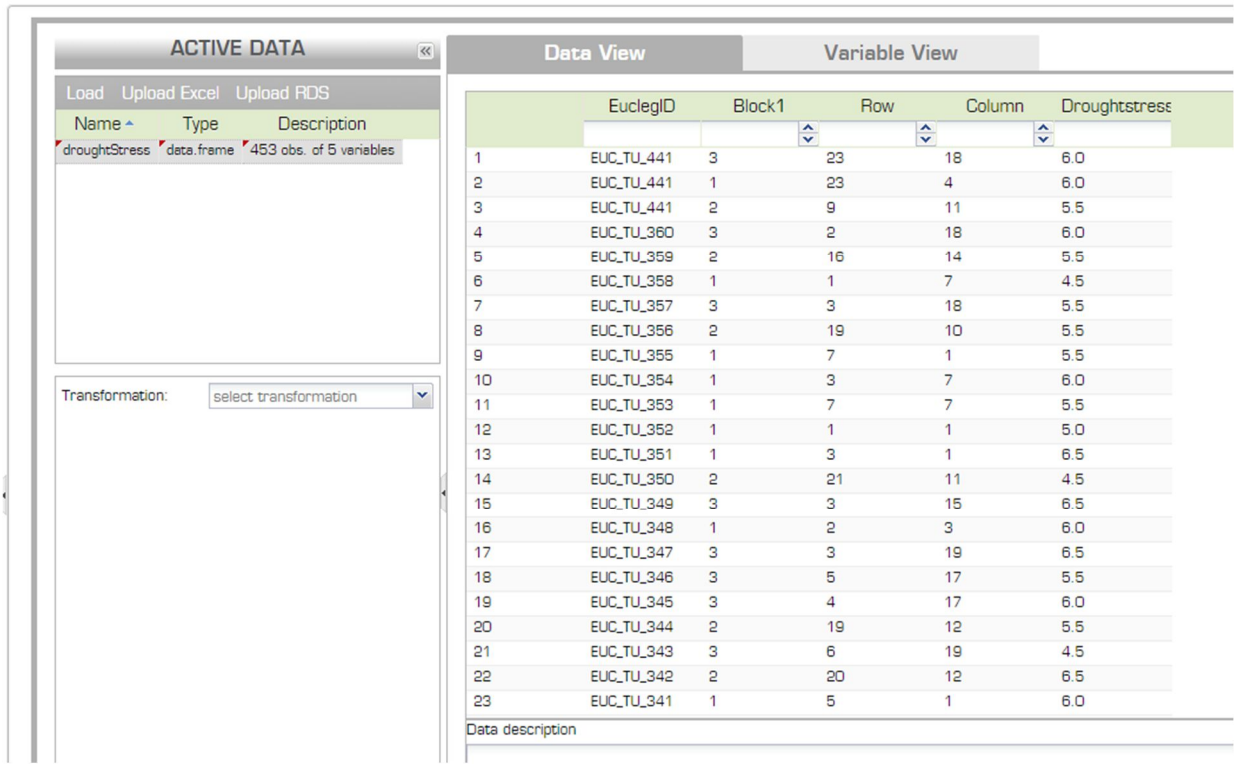
The example dataset that we use to demonstrate the phenotypic analysis tools is a drought stress trial that is named "ILVO drought trial 2018". Use the Database query app to construct a dataset that contains the observations of the trait " Droughtstressnum" that were collected on 27-06-2018. This dataset should have the columns EuclegID, Block1, Row, Column and observed values in the column Droughtstressnum. This dataset should contain 453 observations and is partially shown in Figure 8.

Looking at the Variable View tab of the Data management app we can see that this trial is laid out in a grid of 30 rows and 21 columns. The variable block1 represents a blocking structure within this grid. To visualize this blocking structure we can use the Visualize app. Activate this app and click on the droughtStress dataset in the ACTIVE DATA window and choose the following settings:

- Plot type: Scatter
- X-variable: column
- Y-variable: row

- Color by: block1

Leave the remaining fields blank and press the Visualize button at the bottom. In the resulting scatter plot, depicted in Figure 9, we can see that the block1 variable represents a blocking structure that groups columns of the design (block 1 = columns 1 to 7, block 2 = columns 8 to 14, block 3 = columns 15 to 21). It is assumed that these blocks agree with the physical dimensions of the rainout shelters. We can also see that there is a systematic pattern of missing observations in columns 2, 6, 9, 13, 16 and 20.



	EuclegID	Block1	Row	Column	Droughtstress
1	EUC_TU_441	3	23	18	6.0
2	EUC_TU_441	1	23	4	6.0
3	EUC_TU_441	2	9	11	5.5
4	EUC_TU_360	3	2	18	6.0
5	EUC_TU_359	2	16	14	5.5
6	EUC_TU_358	1	1	7	4.5
7	EUC_TU_357	3	3	18	5.5
8	EUC_TU_356	2	19	10	5.5
9	EUC_TU_355	1	7	1	5.5
10	EUC_TU_354	1	3	7	6.0
11	EUC_TU_353	1	7	7	5.5
12	EUC_TU_352	1	1	1	5.0
13	EUC_TU_351	1	3	1	6.5
14	EUC_TU_350	2	21	11	4.5
15	EUC_TU_349	3	3	15	6.5
16	EUC_TU_348	1	2	3	6.0
17	EUC_TU_347	3	3	19	6.5
18	EUC_TU_346	3	5	17	5.5
19	EUC_TU_345	3	4	17	6.0
20	EUC_TU_344	2	19	12	5.5
21	EUC_TU_343	3	6	19	4.5
22	EUC_TU_342	2	20	12	6.5
23	EUC_TU_341	1	5	1	6.0

Figure 8: The example dataset used for phenotypic analysis.

To correctly analyze these observations we need to know if there are accessions that are represented in all three blocks. If each block would contain a unique set of accessions, the blocks are said to be disconnected and the analysis should be performed on each of the blocks separately. We create a two-way contingency table using the following settings in the Data management app:

- Transformation: Two-way contingency
- Variable1: euclegID
- Variable2: block1
- Matrix name: droughtContigency

Clicking the transform button produces a matrix named “droughtContigency” in the ACTIVE DATA panel. Click on this matrix and examine its content in the Data View panel. Each cell indicates how many times an accession (rows) is represented in a block (columns). We can

immediately see that there are several accessions that are represented multiple times in multiple blocks. The blocks are therefore connected and we can analyze them together.

To get a feel for the data we create a heatmap in the Visualization app:

- Plot type: Heatmap
- X-variable: column
- Y-variable: row
- Color by: Droughtstressnum
- Label variable: euclegID

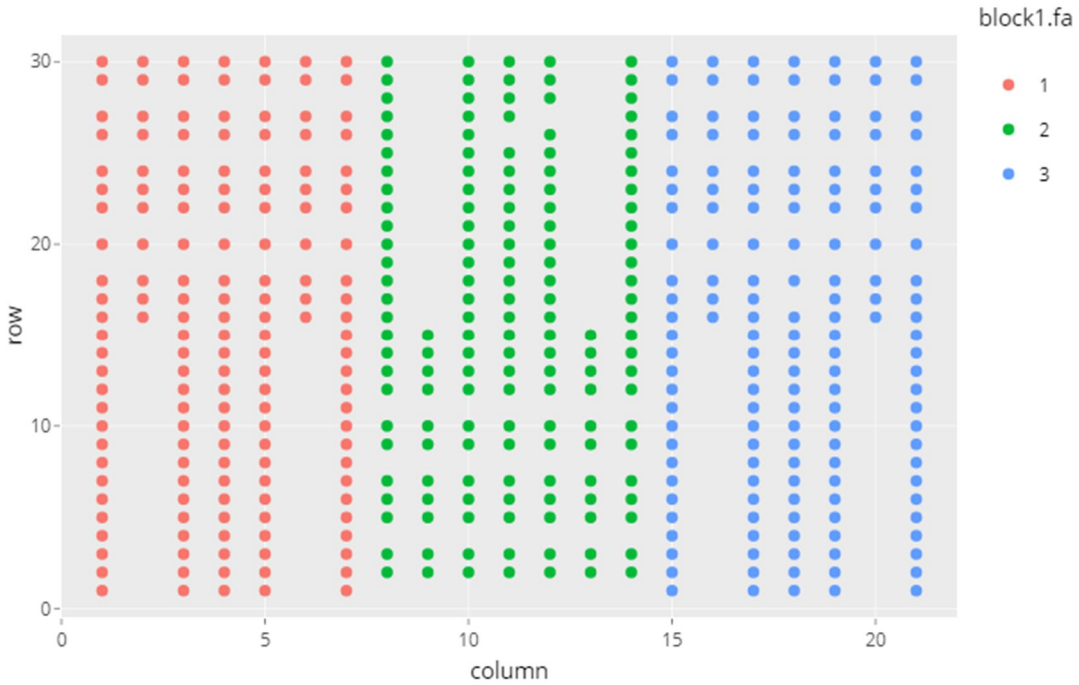


Figure 9: Scatter plot showing the layout and blocking structure of the trial.

The resulting plot is show in Figure 10. Assuming that a higher observation signifies with more drought stress, we can clearly see that block 2 demonstrates less drought stress compared to blocks 1 and 3. We will assume that this deviation is an artefact of unbalanced precipitation and is not related to the genetic capacity to endure drought stress of the accessions in block 2. As the block effect is assumed to be non-genetic, we will correct for it during the analysis. In reality, however, this assumption needs to be validated with climate records prior to analysis.

The difference between blocks is also demonstrated in Figure 11 that shows a density plot of the drought stress observations for each block. This plot was made by applying the following settings in the Visualize app:

- Plot type: Density
- X-variable: Droughtstressnum
- Smooth: 1
- Color by: block1

- Opacity: 0.5

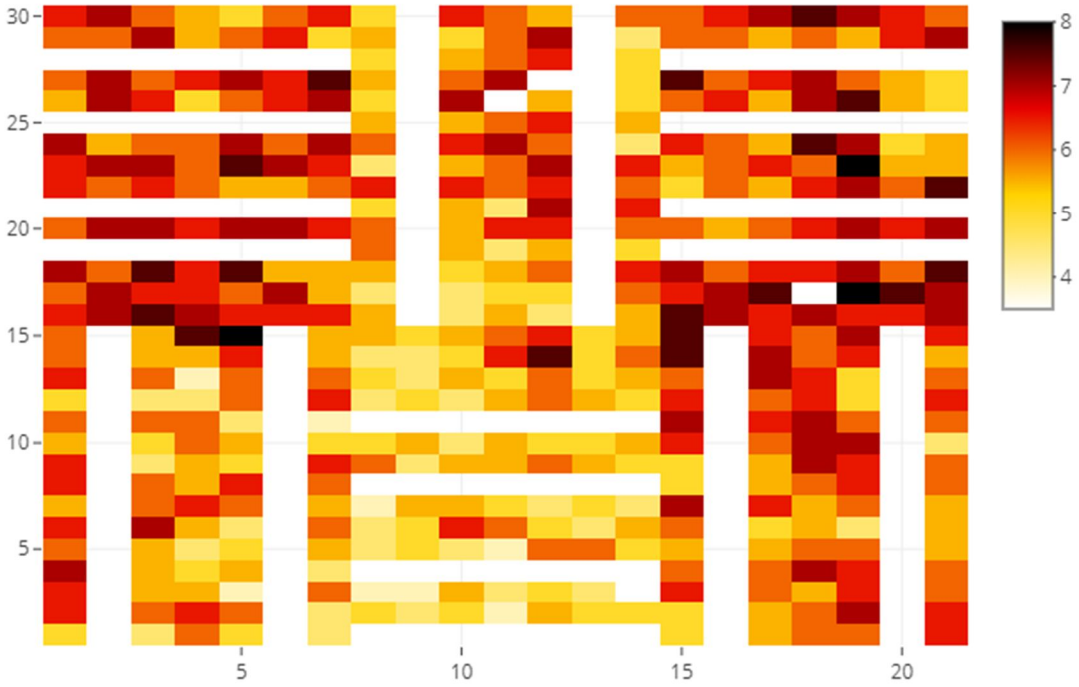


Figure 10: Heatmap of drought stress observations.

The actual analysis is performed by clicking on the Mixed model app button and activating the droughtStress dataset in the ACTIVE DATA panel. We then apply the following settings:

- Analysis type: Single trial analysis
- Response: Droughtstressnum
- Subject: euclegID
- Pedigree data: blank
- Fixed rep: block1
- Random incomplete block: blank
- Random row: row
- Random column: column
- Spatial row: row
- Spatial column: column

Pressing the Analyze button produces the “Single trial analysis” report. The model fit statistics are as follows:

- Likelihood convergence: TRUE
- LogLikelihood: -186.5918
- AIC: 385.1836
- Residual variance: 0.2803186

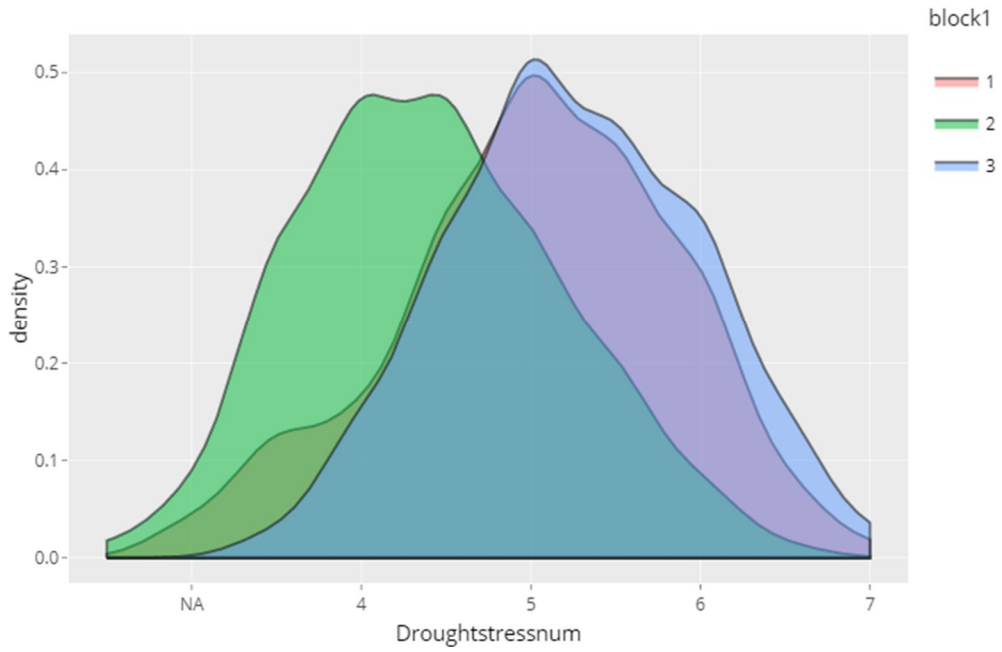


Figure 11: Density plot of drought stress observations by block.

The variance parameters have been estimated by the Restricted Maximum Likelihood or REML procedure. It is indicated that the likelihood has converged which means that the estimation procedure has finished normally and that resulting variance components effectively maximize the likelihood function. A non-converged likelihood is a strong indicator that the model does not fit very well and / or that the data does not contain sufficient information to estimate the required variance parameters. If the estimation procedure terminates before convergence, the estimated variance components are unreliable.

The LogLikelihood value of -186.5918 itself is not very informative but allows to compare the fit of various models that have different variance components. The model that produces the higher LogLikelihood is the model that shows the best fit for the data at hand. The AIC or Akaike Information Criterion judges model fit by trading of the LogLikelihood with the number of variance components as AIC = -2(LogLik – n) where n is the number of estimated variance components. Lower AIC values indicate a better model fit and parsimony.

Table 1 shows the estimated variance for each of the random model terms. The residual variance S2 = 0.2803 is the variance in drought stress values that has not been explained by the model. The gamma column expresses the variance component estimates in the component column as a relative proportion of the residual variance. For example, the estimated variance of the genetic effects (euclegID term) is 0.3281 which results in a gamma of 1.17 indicating that the euclegID variance is 17% greater than the residual variance. This helps to judge the importance of the model terms. The gamma for the column term is 0.0478 indicating that the column variance is less than 5% of the residual variance which implies that there is very little variation between the columns. The gamma for the rows is fixed at 8e-06 which is the default value for parameters for which no variance could be estimated. This basically means that there

is no (measurable) effect of the rows. The residual ANISO components of 0.141 and 0.1594 are not variances but Pearson correlations. For these types of parameters the gamma value should be ignored. The model fits an AR(1) autoregressive component in the direction of the rows and the direction of the columns. This assumes that there is a correlation between the residuals of neighboring plots that decays as plots are further apart. The estimated correlations in rows (0.14) and columns (0.16) are quite low so they might have very little impact on the model fit.

origin	term	vartype	gamma	component
residual	S2	NA	1	0.2803
random	euclegID	ID	1.17	0.3281
random	row	ID	8e-06	2.243e-06
random	column	ID	0.0478	0.0134
residual	NA	ANISO	0.5031	0.141
residual	NA	ANISO	0.5686	0.1594

Table 1: Overview of estimated variance components.

We therefore rerun the analysis with the following settings:

- Analysis type: Single trial analysis
- Response: Droughtstressnum
- Subject: euclegID
- Pedigree data: blank
- Fixed rep: block1
- Random incomplete block: blank
- Random row: row
- Random column: column
- Spatial row: blank
- Spatial column: blank

The AIC of this model that does not fit AR(1) residual structures in rows and columns is 166 which is a vast improvement over the previous model that had an AIC of 385. Furthermore, the variance of the row component now has gamma of 0.47 while it was not estimable before.

Instead of randomly trying to add or remove model terms we can also ask to run a set of predefined models and select the one with the best AIC value. We accomplish this with the following settings:

- Analysis type: Single trial model search
- Response: Droughtstressnum
- Subject: euclegID
- Pedigree data: blank
- Fixed rep: block1
- Random incomplete block: blank
- Random row: row
- Random column: column

Also check the "Extract BLUPs" and "Extract residuals" checkboxes and press Analyze. This analysis takes some time as the system is now trying several mixed model definitions. The winning model is identical to the one we fitted manually.

The Fixed effects section contains the estimates for the mean and the blocks. As block1 is a factor that has only three levels it is not very sensible to try to estimate a variance component for these levels. If there were numerous block levels we could also fit block1 as a random effect by leaving the "Fixed rep" field blank and filling block1 in the "Incomplete block" field. The effect of block 1 has been set to NA (not available) which means that block 1 is the reference level and the effects of the other levels are expressed relatively to this block 1. For instance, the effect of block 3 is 0.1315 which means that on average the drought stress observations in block3 are 0.1315 units higher compared to the drought stress observations in the reference block 1. The effect of block 2 is -0.6445 so 0.6445 units lower than the effect of block 1 which agrees with our visual assessment in the heatmap in Figure 10.

The diagnostic plots at the bottom of the report allow to visually assess the validity of the model assumptions. The "Actual vs Fitted values" plot compares the original observations with the fitted observations (i.e. observations that are reconstructed from the estimated effects of block, euclegID, row and column). We clearly see that the drought stress trait has been measured on a discrete scale having 10 levels (from 3.5 to 8). The fitted values, however, are more continuous which explains why a single value on the actual scale agrees with multiple values on the fitted scale.

The "Residuals vs Fitted values" plot shows a similar systematic pattern that is also caused by the use of a discrete measuring scale. The "Residual vs Row order" plot shows no signs of a systematic relationship between the position of an observation in the dataset and its residual. Any systematic pattern in this scatter plot would be a strong indicator that there is something seriously wrong with this model fit and/or dataset. The "Histogram of residuals" shows a Gaussian distribution which is expected when fitting a linear mixed model that inherently assumes and therefore imposes a normal distribution on the residuals.

Checking the "Extract residuals" checkbox has produced a new dataset in the ACTIVE DATA panel named "droughtStress_Residuals". This is a copy of the original droughtStress dataset where a column named "Resid" has been added that contains the residuals of the fitted model.

These residuals have been corrected for block, accession, row and column effects and should therefore demonstrate a random pattern when laid out over the trial grid. We verify this by making another heatmap by selecting this new dataset in the Visualize app and supply the following settings:

- Plot type: Heatmap
- X-variable: column
- Y-variable: row
- Color by: resid
- Label variable: euclegID

Checking the "Extract BLUPs" checkbox generated a second dataset named "droughtStress.euclegID". This dataset contains the Best Linear Unbiased Predictors or BLUPs of the genetic effects. You can use this dataset to create the scatter plot in Figure 12 where the standard estimate of the BLUPs are plotted against the BLUPs themselves. The three accessions with the lowest standard errors each have 9 plots evenly distributed over the 3 blocks. The 9 accession that have a slightly larger standard error have only 6 plots. The bulk of the accessions has a single plot in the trial which results in BLUPs with elevated standard errors.

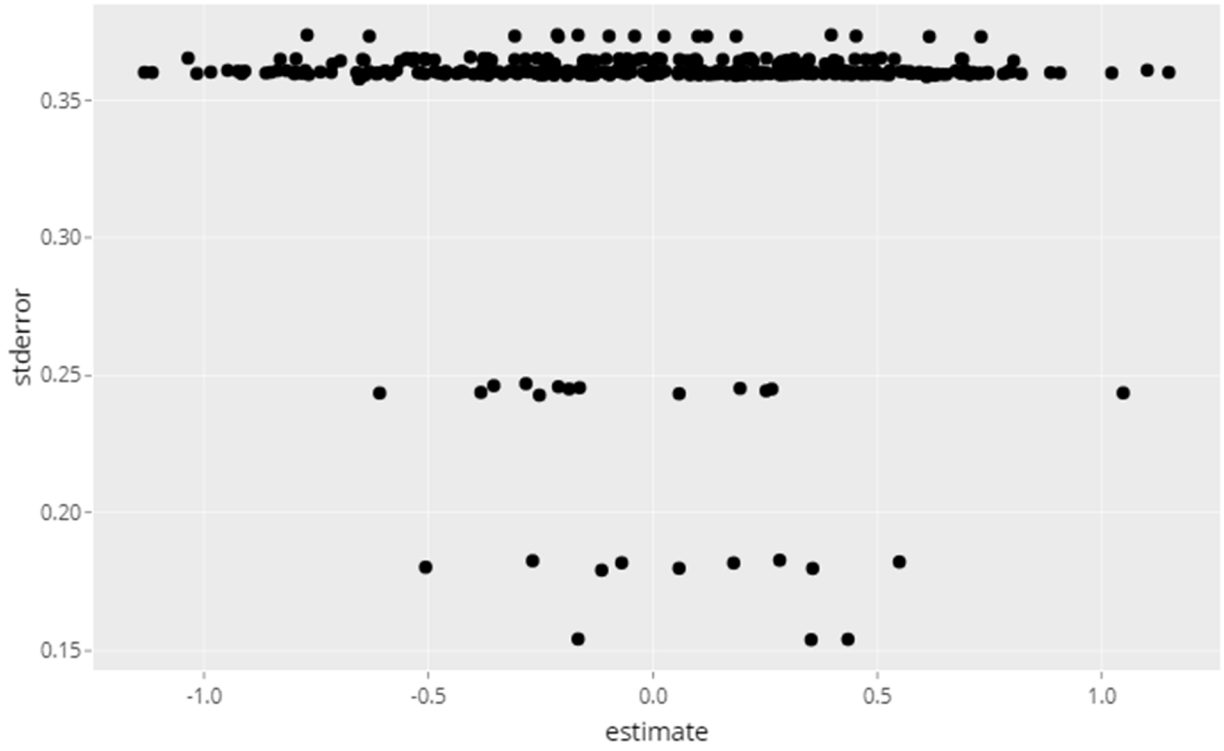


Figure 12: Scatter plot showing the BLUPs of the accessions and their associated standard error.

4.2 Multi-trial analysis

To demonstrate the analysis of multi-environment data we create a dataset containing observations of the trait "Proteincontent" that was measured on 519 accessions in 9 different trials. This dataset named "multiTrial" should have 1081 rows and the following 6 columns:

- EuclegID
- Name (of the trial)
- Block1
- Row
- Column
- Proteincontent

To get a feel for the data we create the boxplot in Figure 13 by applying the following settings in the Visualize app:

- Plot type: Boxplot
- X-variable: Name
- Y-variable: Proteincontent

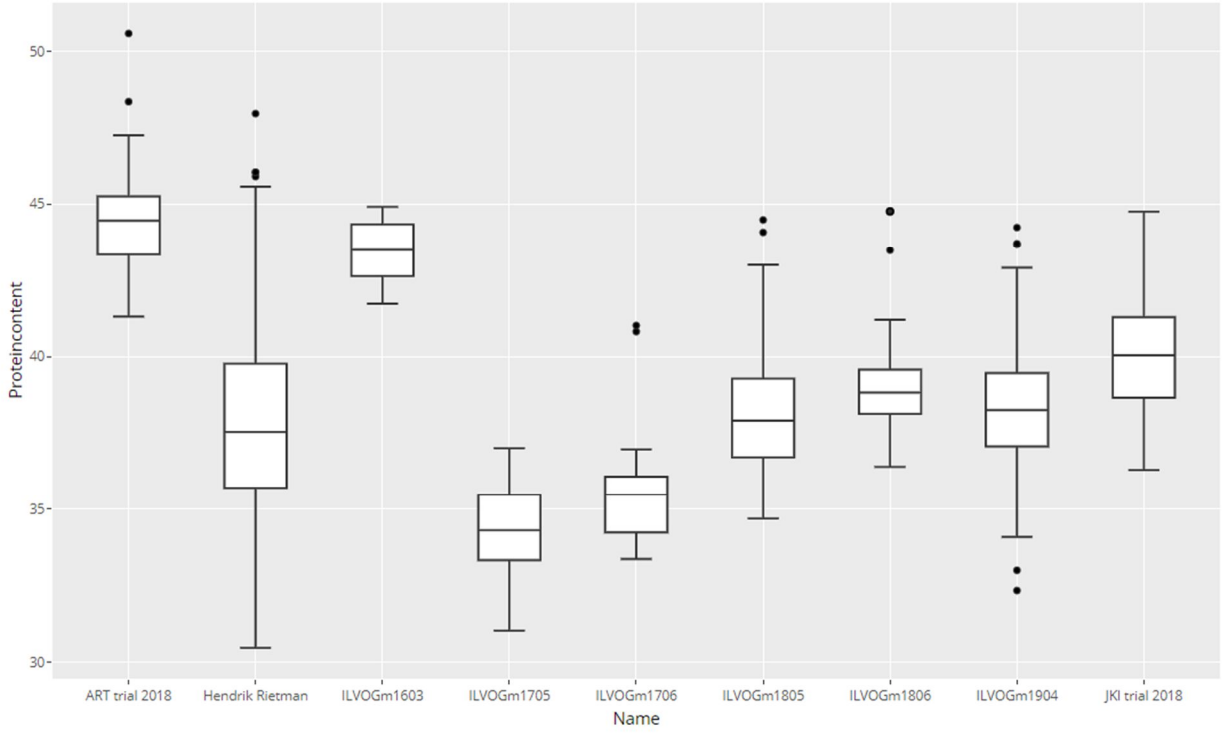


Figure 13: Boxplot of the multi-trial Proteincontent observations.

We clearly see that the both the median and the variance differ extensively between trials.

We perform the actual analysis by clicking the Mixed model app button and selecting the multiTrial dataset in the ACTIVE DATA window. We apply the following settings:

- Analysis type: Multi trial analysis

- Response: Proteincontent
- Subject: euclegID
- Trial: Name
- Fixed rep: block1
- Random row: row
- Random column: column

Check the “Fit trial-specific residual variances” checkbox and press the Analyze button.

The first section of the analysis report is named Trial connectivity. To analyze these trials together we need to make sure that the trials are connected. Two trials are connected if they have one or more accessions in common. We could create a 2-way contingency table between variables euclegID and Name but this would only allow to easily identify accessions that are common to all trials. However, different pairs of trials can be connected through different accessions and this type of connectivity is more difficult to verify in a contingency table. This is why the multi trial analysis routine always performs a connectivity analysis. If one ore more trials would be disconnected for the other trials, the analysis would be performed on the largest connected subset of trials. In this case, however, all trials are connected so they are analyzed together.

The multi-trial model we fitted contains the following fixed effects:

- Mu: the mean
- Name: the effect of the trial
- Name.block1: the effect of the block within each trial

The following random effects have been fitted:

- euclegID: the genetic effect of the accession
- Name.column: the effect of the column within each trial
- Name.row: the effect of the row within each trial

A separate residual variance has been fitted to each trial. This has the implicit effect that good trials (i.e. trials with low residual variance) will contribute more to the genetic estimates compared to bad trials.

While this model fits well, it makes some simplifying assumptions. For example, it is assumed that there is an effect of the rows and the columns in each of the 9 trials. Furthermore, the variance of these row and column effects is assumed to be identical in all these trials.

The multi-trial model search is an extension of the single trial model search. The idea is to find the best fitting model for each of the 9 single trials and then to combine these single trial models in a multi-trial model definition.

We perform a multi-trial model search by applying the following settings:

- Analysis type: Multi trial model search

- Response: Proteincontent
- Subject: euclegID
- Trial: Name
- Fixed rep: block1
- Row: row
- Column: column

Check the “Extract BLUPs” checkbox.

The resulting multi-trial model is a lot more complicated compared to the previous one with simplifying assumptions. We can see that separate column and row variances are estimated for some, but not all, trials.

Looking at the trial-specific variances there is something peculiar going on with trial ILVOGm1705. The residual variance of this trial has been fixed at the minimum value namely 0.000008. This indicates that the residual variance in this trial is so small that it cannot be estimated. So either this trial is so good that there is 0 residual variance or there is a hidden issue that prevents the estimation of a residual variance.

Try to find the reason for this anomaly as an exercise. Filter out the observations of the ILVOGm1705 trial in a separate dataset and perform a single trial analysis. Can you find the origin of the problem?

Create a new multiTrial dataset for the Proteincontent trait that does not include the ILVOGm1705 trial. Perform the multi trial model search and extract the BLUPs. Store this dataset containing 495 BLUPs in your Personal storage area as “proteinBLUPs”. Later we will use this dataset to demonstrate other analysis methods.

5 Genotypic analysis

The tutorial database contains marker scores for 212.689 SNPs and 477 accessions. This score matrix is 99,78% complete which implies that it has 225.273 missing values. These missing values are problematic for most analysis routines for population structure analysis, GWAS and genomic prediction. It is therefore advised to use genotype imputation to fill in the blanks with sensible values.

Analysing genotypic datasets can be computationally demanding due to the data volume that has to be processed. To reduce the waiting time this tutorial will make use of a reduced genotypic dataset that only contains markers that are physically located on chromosome 11. You can create this dataset by means of the Database query app. Make sure that there are no active filters in the Trial and Individual tabs and press the “Fetch Genotype data” button. In the Marker selection window type 11 in the filter field below the header of the Chromosome column. As can be seen at the bottom of the Marker selection window, the filter on chromosome 11 results in a subset of 8.733 markers. We check the checkbox at the top left corner of the window to select all these markers and press the Done button at the bottom right. Export the QUERY RESULT to the Progeno Analytics module by pressing the “Export to Analytics” button. Give your dataset the name “Chrom11”. This opens the score matrix in the

Data management app. In the Variable View tab we can see that this matrix contains 477 rows (accessions) and 8.733 columns (markers).

5.1 Genotype imputation

Imputation is the statistical inference of unobserved genotypes. Figure 14 shows an example of the most common imputation scenario where the pattern of missing data is random. There are numerous imputation algorithms, each having their own requirements and specificities in terms of input data, missingness pattern and computation time.

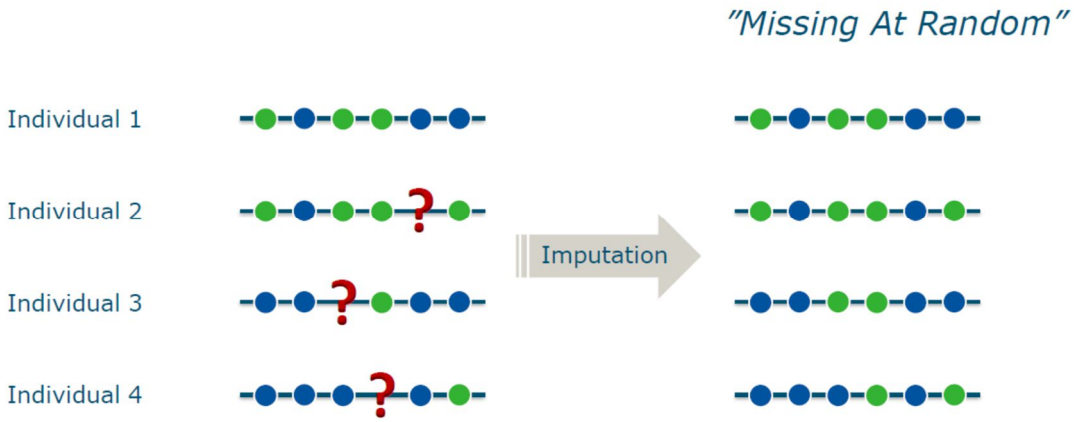


Figure 14: Graphical presentation of genotype imputation where the pattern of missingness is random.

The Progeno Analytics modules provides three different imputation methods:

1) Simple means: the most basic imputation method. For each marker the average of the reference allele frequency is calculated from the non-missing observations. Each missing value is then replaced by the average reference allele frequency of the marker. This method is very fast but not very accurate.
2) FastPHASE: is an implementation of the imputation model described in Scheet & Stephens (2006). Allows for more advanced imputation but is computationally quite demanding which implies a long processing time.
3) Beagle: imputation method described in B L Browning, Y Zhou, and S R Browning (2018). Generally considered as one of the best performing imputation algorithms. Very demanding in terms of computation so you should expect very long processing times.

To reduce the processing time both FastPHASE and Beagle process multiple chromosomes in parallel. This does not help for our tutorial dataset as the selected markers all belong to the same chromosome. We therefore use Simple means to impute the missing values as follows:

- open the Genotype imputation app
- click on the Chrom11 matrix in the ACTIVE DATA panel
- analysis type: Impute missing genotypes
- method: Simple means (use FastPhase or Beagle if you can let this run overnight)

- click on the Analyze button

From the Genotype Imputation analysis report we can see that 8.090 missing scores have been replaced by their marker average. The ACTIVE DATA window now contains a new score matrix named “Chrom11.imputed”. In the Variable View of the Data management app you can verify that this imputed matrix does not contain missing values.

Besides imputation the Genotype imputation app allows to identify and remove redundant markers. A redundant marker is a marker that is in (near) perfect correlation with another marker. In this case all the information that is contained in the scores of this marker is also contained in the other marker so it is safe to remove one of the two. The algorithm calculates the Pearson correlation between each pair of markers. In case the absolute value of this correlation exceeds a certain threshold (the default is R = 0.99), one of the two correlated markers is removed. To be able to calculate the Pearson correlation between two markers, both markers should not be monomorphic. This explains why this method also imposes a minimum threshold on the marker variance.

We remove redundant columns by applying the following actions:

- click on Chrom11.imputed in the ACTIVE DATA panel
- Analysis type: Remove redundant markers
- Minimum variance: 0.01
- Maximum correlation: 0.99
- click on the Analyze button

The Marker redundancy analysis result shows that 2.806 markers have been removed as these were monomorphic. 3.933 markers have been removed as they were redundant. This results in a new score matrix named ‘Chrom11.imputed.informative” that contains 1.994 complete, polymorphic and non-redundant markers. Save this matrix in your Personal storage space.

5.2 Population analysis

The Population analysis app tries to detect population structure in a set of genotyped accessions. Principal Component Analysis (PCA) is used to reduce the dimensionality of the marker data and represent the accessions in one or several 2-dimensional spaces. If a population structure is known beforehand, the accessions can be colored according to this existing stratification. One can also detect the presence of population structure by means of the Hierarchical Clustering on Principle Components (HCPC) approach. This algorithm performs an agglomerative clustering on the scores from the principal component analysis. The desired number of clusters can be provided or estimated from the data. We apply the following actions to analyze population structure:

- activate the Population analysis app
- click on the Chrom11.imputed.informative matrix in the ACTIVE DATA panel
- Analysis type: Perform population analysis
- PCA components: 3
- Scale marker scores: checked

- Export PCA scores: checked
- Population clusters: Auto
- Export population clusters: checked
- click on the Analyze button

The Population Structure Analysis report provides the results of the PCA and HCPC analyses. We can see that the first three principal components together explain approximately 22% of the variance in the chromosome 11 dataset. These three components have been used to construct a cluster dendrogram. The HCPC algorithm estimates the presence of four population clusters which are depicted in the subsequent PCA plots. If you hover over the points in these plots a popup window appears giving you the EuclegID of the accession at that position.

The Population analysis created a new dataset in the ACTIVE DATA panel named "Chrom11.imputed.informative.scores.clusters". We can examine this dataset in the Data management app. It contains the scores for each of the three PCA scores and the putative cluster (i.e. population) number for each accession. We can use this dataset to make a 3D visualization of the PCA space:

- activate the Visualize app
- click on the "Chrom11.imputed.informative.scores.clusters" dataset in the ACTIVE DATA window
- Plot type: 3D scatter
- X-variable: Dim.1
- Y-variable: Dim.2
- Z-variable: Dim.3
- Color by: cluster
- Opacity: 0.6
- Size by: blank
- Label variable: name
- click on the Visualize button

Figure 15 shows the resulting 3D scatter plot. We can rotate this plot by holding down the left mouse button somewhere in the plot and moving the mouse from left to right or top to bottom. Hovering the mouse pointer over a point in the 3D space shows a popup window that contains the EuclegID of the accession.

Save the "Chrom11.imputed.informative.scores.clusters" dataset in your Personal storage space.

5.3 GWAS

In a genome-wide association study (GWAS) one generally examines a genome-wide set of marker scores for associations with a phenotypic trait of interest. Each marker is analyzed separately which implies that the computational workload scales linearly with the number of markers that are examined.

Most GWA studies rely on linear mixed model analyses, more specifically the QK model (Yu et al, 2006), for the calculation of metrics such as p-values or association scores that express the strength of the association between marker and trait. The QK model incorporates a correction for population stratification (Q matrix) and kinship (K matrix).

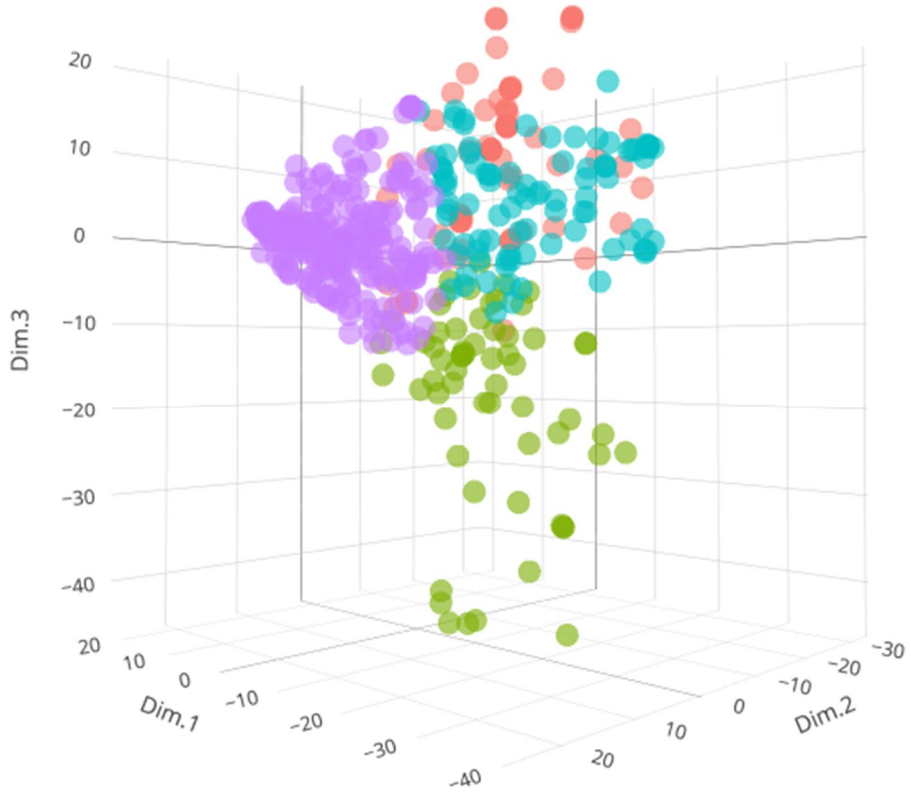


Figure 15: 3D visualization of the genetic space made from the first three principal components.

We perform a GWAS study using the following data sources:

1. Chrom11.imputed.informative: matrix containing reference allele frequencies for 477 accessions and 1.994 polymorphic, non-redundant markers located on chromosome 11
2. proteinContentBLUPs: dataset containing BLUPs for 495 accessions obtained from a multi-environment analysis of protein content observations in 8 EUCLEG trials.
3. Chrom11.imputed.informative.scores.clusters: dataset containing PCA scores and population cluster identifiers for the 477 accession in the Chrom11.imputed.informative matrix

If you followed the tutorial, these three datasets should be stored in your Personal storage space. If this is not the case, you can find these datasets in the Tutorial folder on the Shared storage space.

To use the population structure in a GWAS the identifiers of the population clusters should be added to the phenotypic dataset. We will therefore merge the proteinContentBLUPs dataset with the Chrom11.imputed.informative.scores.clusters dataset as follows:

- click on the Data management app button
- click in the proteinContentBLUPs dataset in the ACTIVE DATA panel
- Transformation: Merge with other data set
- Merge column(s): level
- Keep all rows: uncheck
- Other data: Chrom11.imputed.informative.scores.clusters
- Other column(s): name
- Keep other rows: uncheck
- Data name: mergedProteintContent
- click the Transform button

By unchecking the “Keep all rows” and “Keep other rows” checkboxes we request to only keep rows in the proteinContentBLUPs dataset for which the level (i.e. accession) also appears in the name column of the Chrom11.imputed.informative.scores.clusters dataset and vice versa. The resulting “mergedProteinContent” dataset contains 394 rows. These are the 394 accession for which both phenotypic and genotypic observations are available.

We proceed to the GWAS by applying the following actions:

- click in the GWAS app button
- click the “Chrom11.imputed.informative” score matrix in the ACTIVE DATA panel
- Analysis type: Perform GWAS analysis
- Phenotypic data: mergedProteinContent
- Response: estimate
- Subject: level
- Population: cluster
- MAF cutoff: 0.01
- FDR level: 0.05
- Reuse variance estimates: uncheck
- Extract marker scores: uncheck
- click the Analyze button.

The MAF cutoff is a threshold that filters markers on their Minor Allele Frequency. Markers for which the MAF is below this threshold will not be analyzed. The idea is to reduce computational effort by skipping markers that are nearly monomorphic and will therefore never reach the requested significance threshold.

The FDR level is the False Discovery Rate at which we wish to examine the associations. The default of 0.05 implies that the probability of finding a false positive over all tested markers is equal or less than 5%. This type of error control is less stringent compared to familywise error rate controlling procedures such as the Bonferroni correction.

The Reuse variance estimates checkbox allows to speed up the calculations at the cost of reducing the power of the analyses. If this box is unchecked, a linear mixed model is fitted for each marker and the required variance components are also estimated separately for each marker. If the Reuse variance estimates box is checked, the required variance components are

estimated once (using a mixed model without marker effect) and reused for each marker. This considerably reduces the computational workload.

The resulting GWAS report shows the distribution of minor allele frequencies and the Polymorphism Information Content (PIC) for the examined markers. The Marker scores section lists the GWAS scores for each marker sorted in descending order. This score is the minus log transformation of the p-value (-log(p)) that expresses the probability of an association between marker and trait.

Figure 16 shows the resulting Manhattan plot in which the scores are potted against their position on the genome. While there are several positions that have elevated scores, none of these reach the significance threshold. As the FDR procedure requires at least one marker to exceed the significance threshold, the FDR corrected threshold cannot be calculated. Figure 17 shows a Manhattan plot of a different dataset in which one marker is found to be significantly associated with the trait of interest. In this case the FDR threshold can be calculated and is plotted as a horizontal red line.

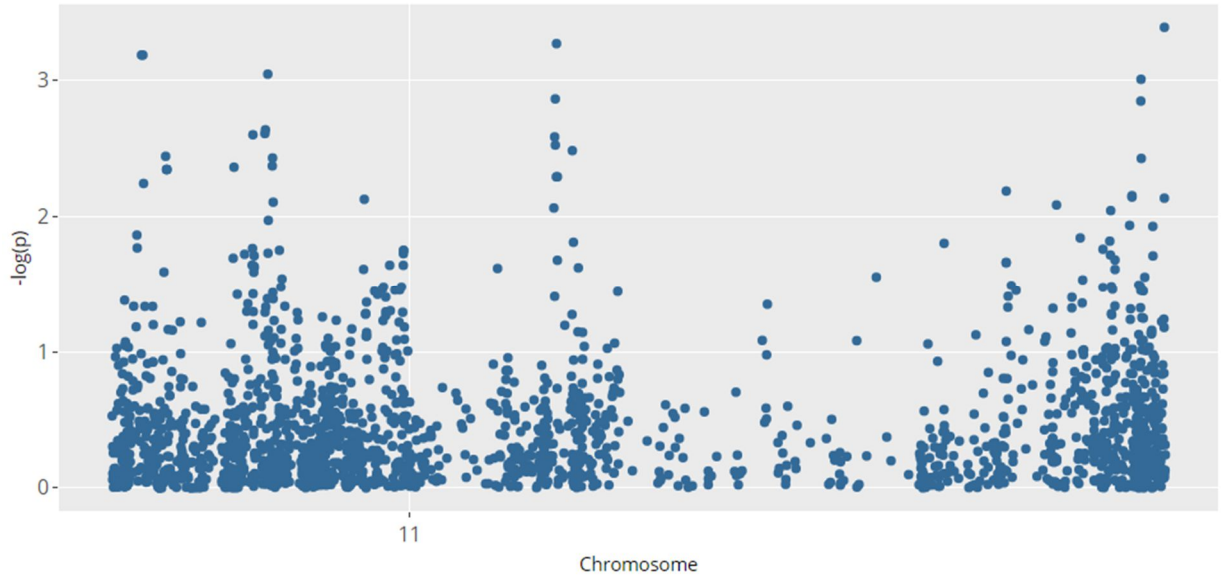


Figure 16: Manhattan plot of the tutorial GWAS.

Figure 18**Error! Reference source not found.** shows the Q-Q plot of the tutorial GWAS. In this plot the observed scores (-log(p) values) are plotted against the expected -log(p) values when one assumes that there are no associated markers. The bulk of the scores should be located on the line. The markers that are located above the red line show elevated scores compared to their expectation and therefore indicate a possible association. If most of the points would be located above the line, it is very likely that there is a confounding factor that introduces a bias in the resulting p-values. In most cases this bias is caused by population stratification that is not (sufficiently) corrected for in the analysis. As an exercise you can compare the scores, Manhattan and Q-Q plots for a GWAS with and without correction for population stratification. In this case the difference between the two approaches is negligible indicating that the population stratification is not that pronounced for this selection of markers.

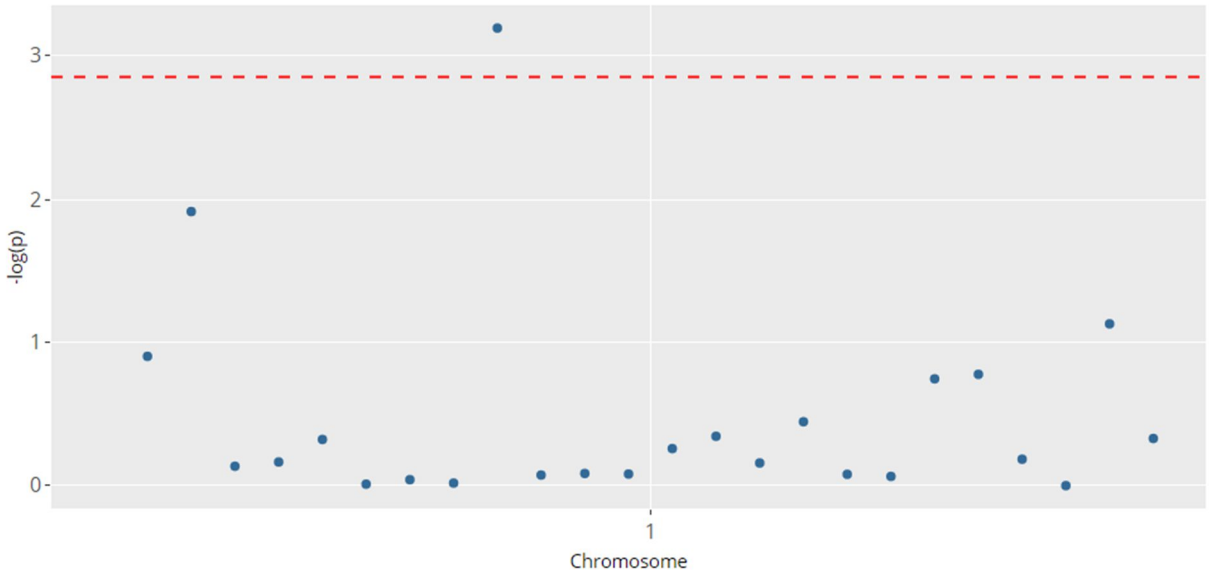


Figure 17: Example of a Manhattan plot with a marker that exceeds the significance threshold.

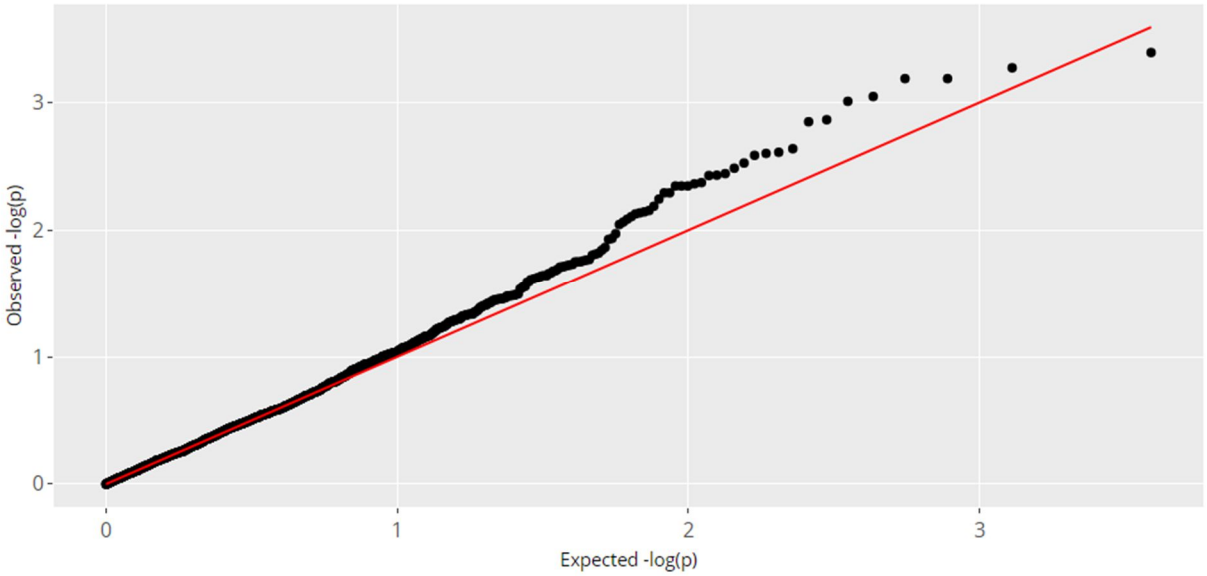


Figure 18: Q-Q plot of the tutorial GWAS.

5.4 Genomic prediction

Genomic prediction is the field in which the genetic value of accessions is estimated from a (generally genome-wide) set of molecular marker scores instead of phenotypic observations. Similar to GWAS, however, genomic prediction requires the availability of both phenotypic and genotypic data to train a prediction model. In a GWAS one tries to find variants that are associated with the trait of interest and these associated markers can then be used to track down the causal variants or to make predictions on the phenotypic performance of certain genotyped accessions. However, the associated markers that are found in most GWA studies generally explain only a small part of the genetic variance which implies that the derived prediction models demonstrate moderate to low prediction accuracies. Genomic prediction uses all available marker scores to make predictions of genetic potential which generally results in superior prediction accuracies compared to GWAS. Most genomic prediction methods do not allow for a reliable identification of the causal variants of the predicted trait.

Using a genomic prediction model to make selections is called genomic selection for which the concept is shown in Figure 19. The training population consists of accessions that are both genotyped and phenotyped for the trait of interest. This training population is used to train a genomic prediction model using a genomic prediction model. The Genomic prediction app fits three types of genomic prediction models:

1) RRBLUP or Ridge Regression BLUP: estimates an effect for each marker. Is computationally fast but the method does not allow for the calculation of reliabilities of predictions
2) GBLUP or Genomic BLUP, mathematically equivalent to RRBLUP but fits an effect for each accession. Computationally slower than RRBLUP but provides reliabilities of the predictions
3) RandomForest: a technique from the domain of machine learning. Computationally slow and several parameters require tweaking to achieve good prediction accuracy.

Once the genomic prediction model has been trained, it can be applied to the selection population which contains accessions that have been genotyped but have no phenotypes. The genomic prediction model predicts genomic estimated breeding values (GEBVs) which replace the missing phenotypes. The GEBVs of the selection candidates allow to rank the accessions according to their genetic value for the trait of interest.

To train a genomic prediction model for protein content we will use a more complete set of molecular marker scores which can be found in the Tutorial folder of the Shared storage space as “allMarkers.imputed.informative”. This matrix contains the reference allele frequencies of 53.937 markers which have been found to be polymorphic and non-redundant. We train a genomic prediction model by applying the following actions:

- click on the Genomic prediction app button
- click on the allMarkers.imputed.informative matrix in the ACTIVE DATA panel
- Analysis type: Train genomic prediction model
- Phenotypic data: proteinContentBLUPs
- Response: estimate
- Subject: level

- Method: RRBLUP
- MAF cutoff: 0.01
- Extract subject predictions: unchecked
- Accuracy iterations: 3
- click on the Analyze button

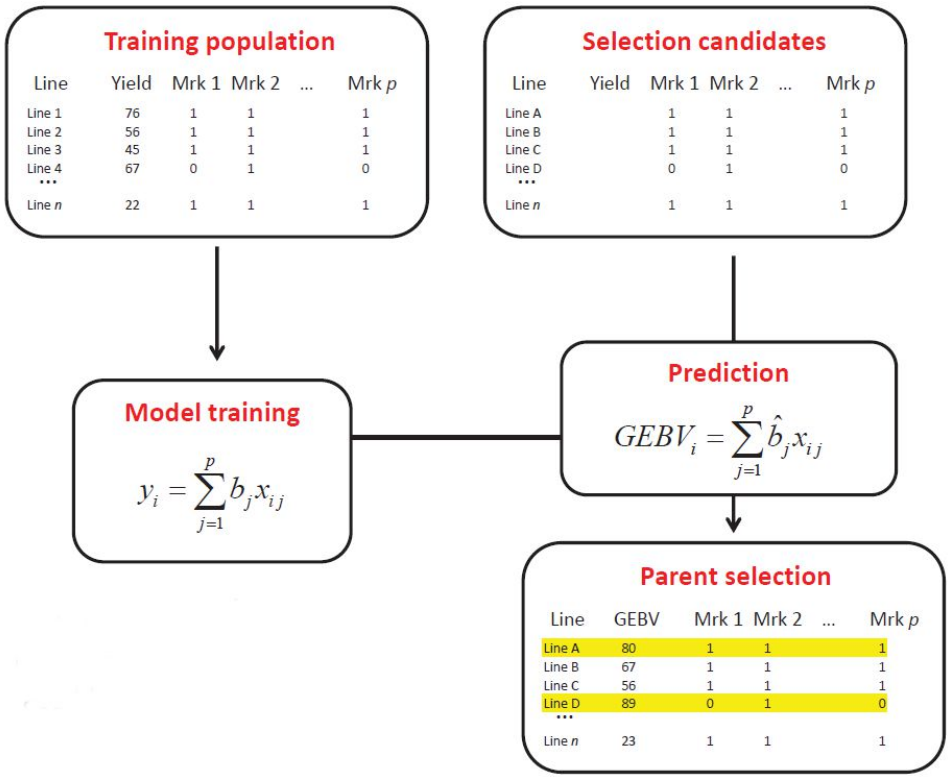


Figure 19: The concept of genomic selection.

Setting the Accuracy iterations count to a positive number estimates the prediction accuracy of the genomic prediction model. This procedure is a 5-fold cross-validation in which the set of genotyped and phenotyped accessions is randomly split into 5 groups. For each of these 5 groups a genomic prediction model is trained using the accessions in the 4 other groups as training dataset. The resulting prediction model is then used to predict the phenotypic performance of the accessions in the group that was not part of the training set. The Pearson correlation between the actual and predicted phenotypes is used as a measure of accuracy. This cross-validation procedure is repeated three times resulting in three Pearson correlations.

The resulting Genomic prediction model training report contains a bar chart of these cross-validation prediction accuracies which shows 0.53, 0.54 and 0.55 (numbers may differ due to sampling). This means that the average explained variation of this prediction model is around 29% which is not brilliant but perhaps usable to perform negative selection on a set of selection candidates.

The ACTIVE DATA panel now contains the trained genomic prediction model as "estimate.RRBLUP.predictionModel". We will use this prediction model to make genomic predictions on the protein content of a set of 100 selection candidates. The genotypic scores for these selection candidates can be found in the "selectionMatrix" file that is located on the shared storage space in the Tutorial folder. We make predictions of the protein content for these 100 selection candidates by applying the following actions:

- click on the Genomic prediction app button
- click on the selectionMatrix entry in the ACTIVE DATA panel
- Analysis type: Make genomic predictions
- Prediction model: estimate.RRBLUP.predictionModel
- Fixed effects: blank
- click the Analyze button

This produces a new datset name "estimate.RRBLUP.predictionModel.predictions" that contains the genomic predictions for the 100 selection candidates.

Now we train a second prediction model that uses GBLUP instead of RRBLUP as prediction method. We follow the exact same steps as before but choose GBLUP instead of RRBLUP in the Method field. The resulting GBLUP prediction model has a similar accuracy compared to the RRBLUP model (i.e. Pearson correlations around 0.54). This is not surprising as GBLUP and RRBLUP are related techniques. We use the GBLUP prediction model to make predictions of the protein content of the 100 selection candidates. The resulting dataset is called "estimate.GBLUP.predictionModel.predictions" and contains, besides the prediction column, a reliability column. This reliability is a number between 0 and 1 that expresses how much we can trust the prediction. Figure 20 shows a scatter plot in which the genomic predictions are plotted against their reliabilities.

When we compare the GBLUP predictions with the RRBLUP predictions it is evident that these predictions are similar but not identical. For example the GBLUP prediction for candidate1 is 0.2115 while the RRBLUP prediction is 0.176. By merging the two prediction datasets we can show that the difference between the two is just a matter of scale. Before merging we rename the "prediction" variable of the estimate.RRBLUP.predictionModel.predictions dataset to "RRBLUP" and the "prediction" variable of the estimate.RRBLUP.predictionModel.predictions dataset to "GBLUP". We then merge the two datasets to produce a "mergedPredictions" dataset containing 100 rows and an RRBLUP, GBLUP and reliability column. In the Visualize app we select this mergedPredictions dataset and apply the following options:

- Plot type: Scatter
- X-variable: RRBLUP
- Y-variable: GBLUP
- Regression: Linear

The resulting scatter plots shows that all points are located on a straight line, demonstrating their linear dependency. Another way of visualizing this linear dependency is achieved as follows:

- Plot type: Correlation
- Variables: RRBLUP, GBLUP, reliability

The resulting correlation matrix shows that the Pearson correlation between RRBLUP and GBLUP predictions is exactly 1 while there is not correlation with the reliability.

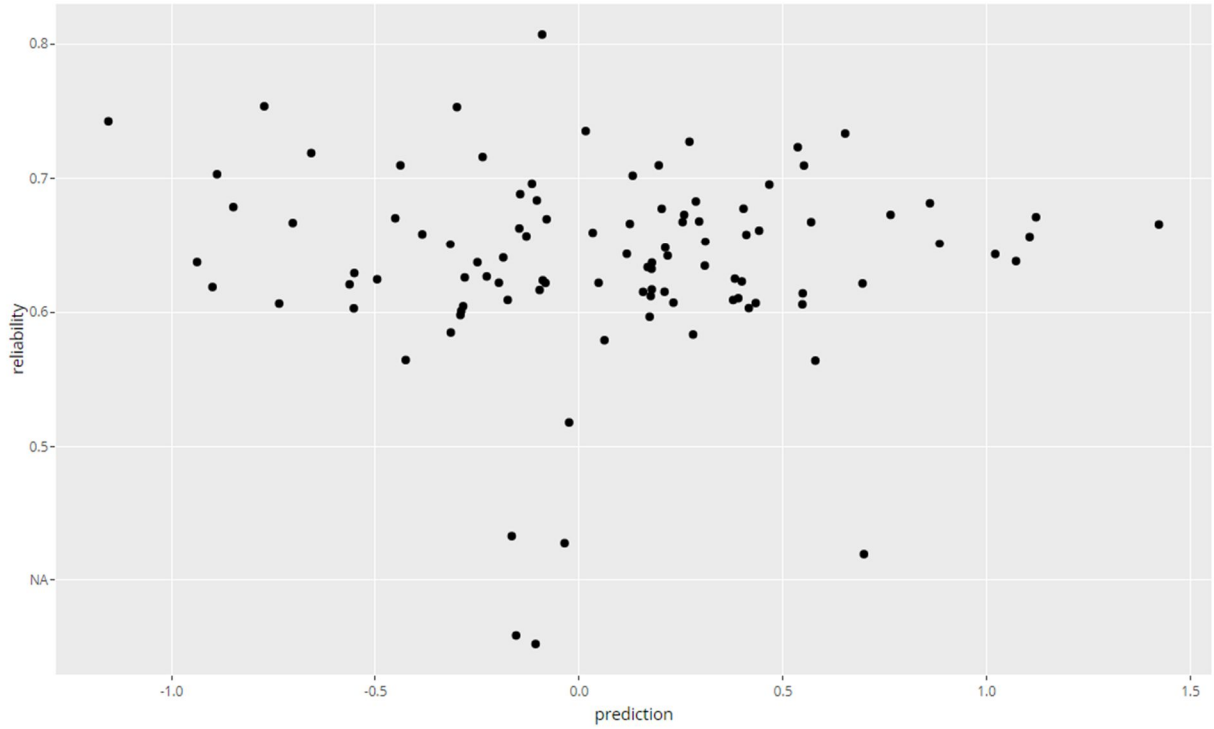


Figure 20: Scatter plot where genomic predictions of protein content are plotted against their reliability.

Imagine that the trait protein content has a major QTL that is closely linked to marker AX-93673197. This marker is currently not present in the "allMarkers.imputed.informative" matrix that we have been using. Simply adding this particular marker as an additional column would be a suboptimal solution as the RRBLUP and GBLUP models assume that the effect of each marker is very small and we know in advance that the effect of marker AX-93673197 is considerable. Q-GBLUP is a variant of GBLUP where one or more QTL (or markers linked to those QTL) are fitted as fixed effects. This has the effect of precorrecting the phenotypic observations for the effect of the major QTL before fitting the effects of the accessions (i.e. the quantitative part of trait).

First we need to create a separate matrix containing only the scores for marker AX-93673197 and all 437 genotyped accessions. Use the Database query app, click the Fetch Genotype data button without defining any filters and use the Name filter in the Marker selection popup window to select the AX-93673197 marker. Export the resulting matrix to the Analytics module as "majorQTL".

Proceed to the Genomic prediction app, select the "allMarkers.imputed.informative" score matrix and apply the following settings:

- Analysis type: Train genomic prediction model
- Phenotypic data: proteinContentBLUPs
- Response: estimate
- Subject: level
- Method: GBLUP
- MAF cutoff: 0.01
- Accuracy iterations: 3
- Fixed effects: majorQTL

Click the Analyze button. In the resulting Genomic prediction model training report we can see that there is no improvement of the prediction accuracy so our marker AX-93673197 is not really a QTL. In the Fixed marker effects section we can see that the effect of this marker is estimated at 0.1415 which is the difference in protein content between an accession that is homozygous for the reference allele and an accession that is homozygous for the alternative allele.

If we want to make predictions with this Q-GBLUP model we need to have the reference allele frequency of marker AX-93673197 of each selection candidate. This matrix can be found in the shared storage space under the Tutorial folder as “selectionMajorQTL”. We make Q-GBLUPs for the selection candidates by selecting the “selectionMatrix” in the ACTIVE DATA panel of the Genomic prediction app and applying the following settings:

- Analysis type: Make genomic predictions
- Prediction model: estimate.GBLP.predictionModel.1
- Fixed effects: selectionMajorQTL

Clicking the Analyze button produces the estimate.GBLUP.predictionModel.predictions dataset containing the Q-GBLUPs of the candiates.