Genomic Mating:
Identifying the most promising crosses

Martine Jean

Bioinformatics workshop

4th SoyaGen Annual Meeting, Université Laval, Québec

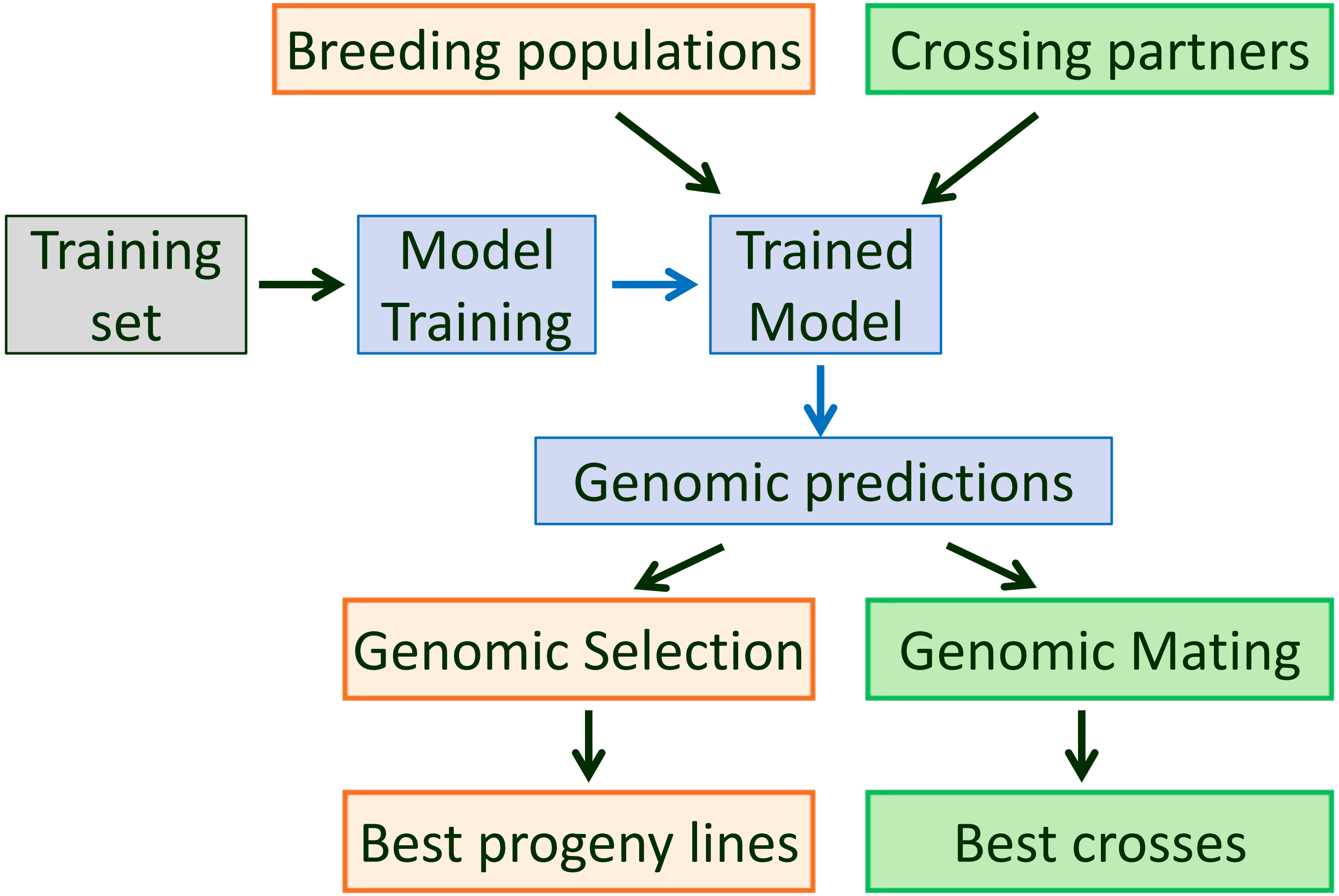
December 6th, 2019

Overview of workshop

- Section 1. Introduction
- Section 2. Getting started with R and RStudio
- Section 3. Data handling with SelectionTools and PopVar
- Section 4. Selecting crosses using conventional approaches with SelectionTools
- Section 5. Genomic Mating: Selecting crosses using genome-wide predictions generated with SelectionTools and PopVar
    - Model training and selection
    - Predicting progeny phenotypes
    - Selecting crosses with SelectionTools
    - Selecting crosses with PopVar

Section 1. Introduction

Genomic predictions can help breeders select breeding lines as well as crosses to perform



Breeding populations

Crossing partners

Training set

Model Training

Trained Model

Genomic predictions

Genomic Selection

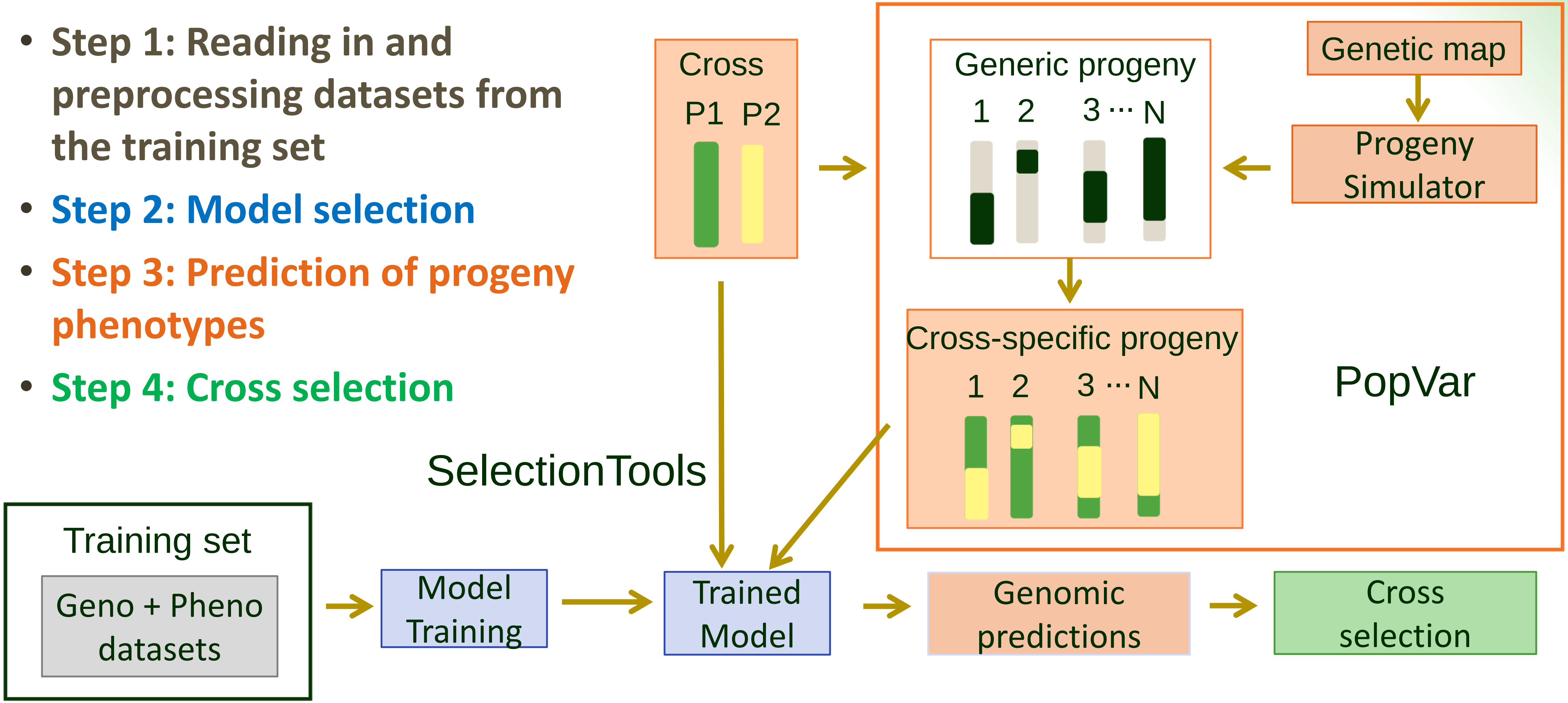
Genomic Mating

Best progeny lines

Best crosses

There are 4 main steps in genomic mating

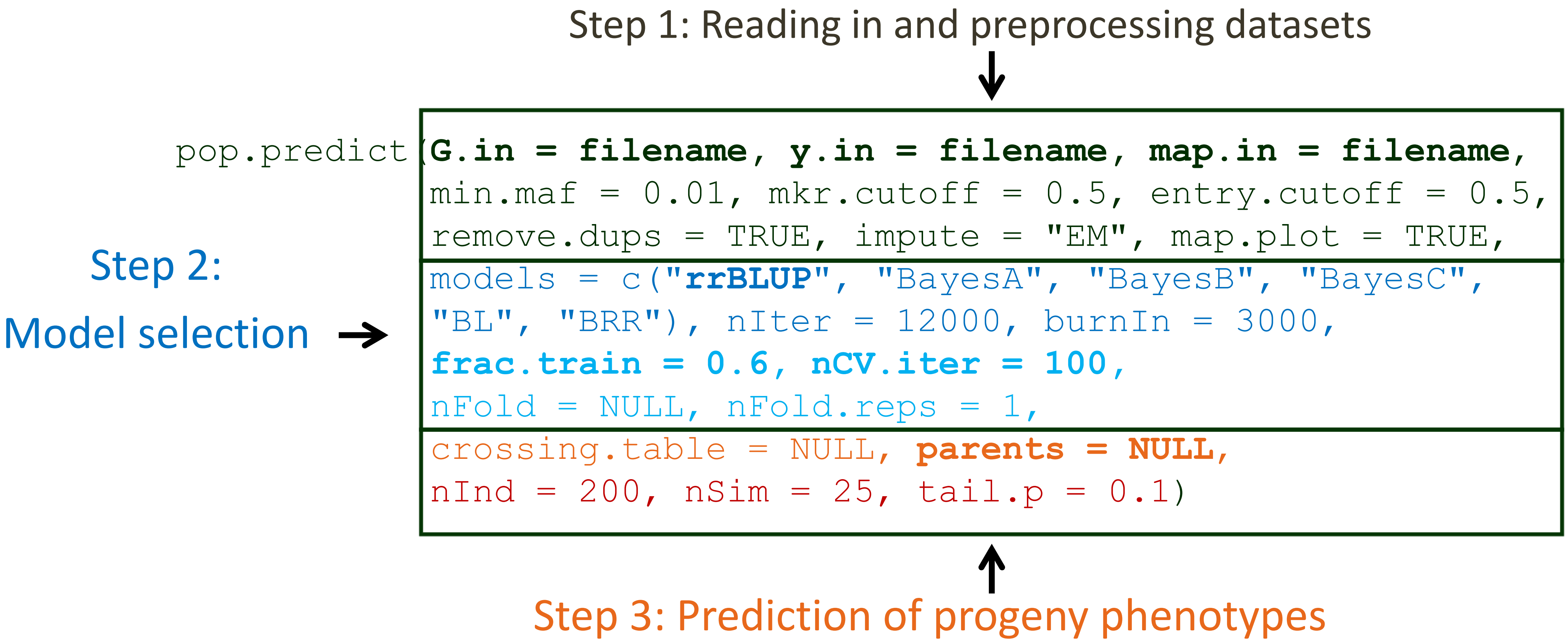
- **Step 1: Reading in and preprocessing datasets from the training set**
- **Step 2: Model selection**
- **Step 3: Prediction of progeny phenotypes**
- **Step 4: Cross selection**



Two R packages are available for genomic mating

	SelectionTools (Osthushenrich et al. 2018 Front. Plant Sci. 9:1899)	**PopVar** (Mohammadi et al. 2015 Crop Sci. 55:2068-20177)
R package content	Collection of bioinformatic tools	Bioinformatic pipeline
Tools available		
- Conventional selection	Yes	No
- Genome-wide predictions		
- Genomic selection	Yes	No
- Genomic mating	Yes	Yes
Statistical approach	Analytical (Models)	Experimental (Simulations)
Calculation speed	Fast	Slow

The heart of the PopVar genomic mating pipeline : the “pop.predict” function



Step 1: Reading in and preprocessing datasets

Step 2:

Model selection

```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
            min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
            remove.dups = TRUE, impute = "EM", map.plot = TRUE,
            models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
            "BL", "BRR"), nIter = 12000, burnIn = 3000,
            frac.train = 0.6, nCV.iter = 100,
            nFold = NULL, nFold.reps = 1,
            crossing.table = NULL, parents = NULL,
            nInd = 200, nSim = 25, tail.p = 0.1)
```

Step 3: Prediction of progeny phenotypes

SelectionTools offers individual functions to perform genomic mating

Step 1: Reading in and preprocessing datasets

st.read.marker.data( )

st.read.performance.data( )

st.read.map( )

st.marker.data.statistics( )

st.copy.marker.data( )

st.restrict.marker.data( )

Step 2: Model selection

gs.esteff.rr( )

gs.esteff.external( )

gs.predict.genotypes( )

gs.cross.validation( )

gs.plot.validation( )

Step 3: Prediction of progeny phenotypes

gs.cross.info( )

gs.cross.eval.gd( )

gs.cross.eval.mi( )

gs.cross.eval.ma( )

gs.cross.eval.mu( )

gs.cross.eval.va( )

gs.cross.eval.es()

Tools for conventional selection

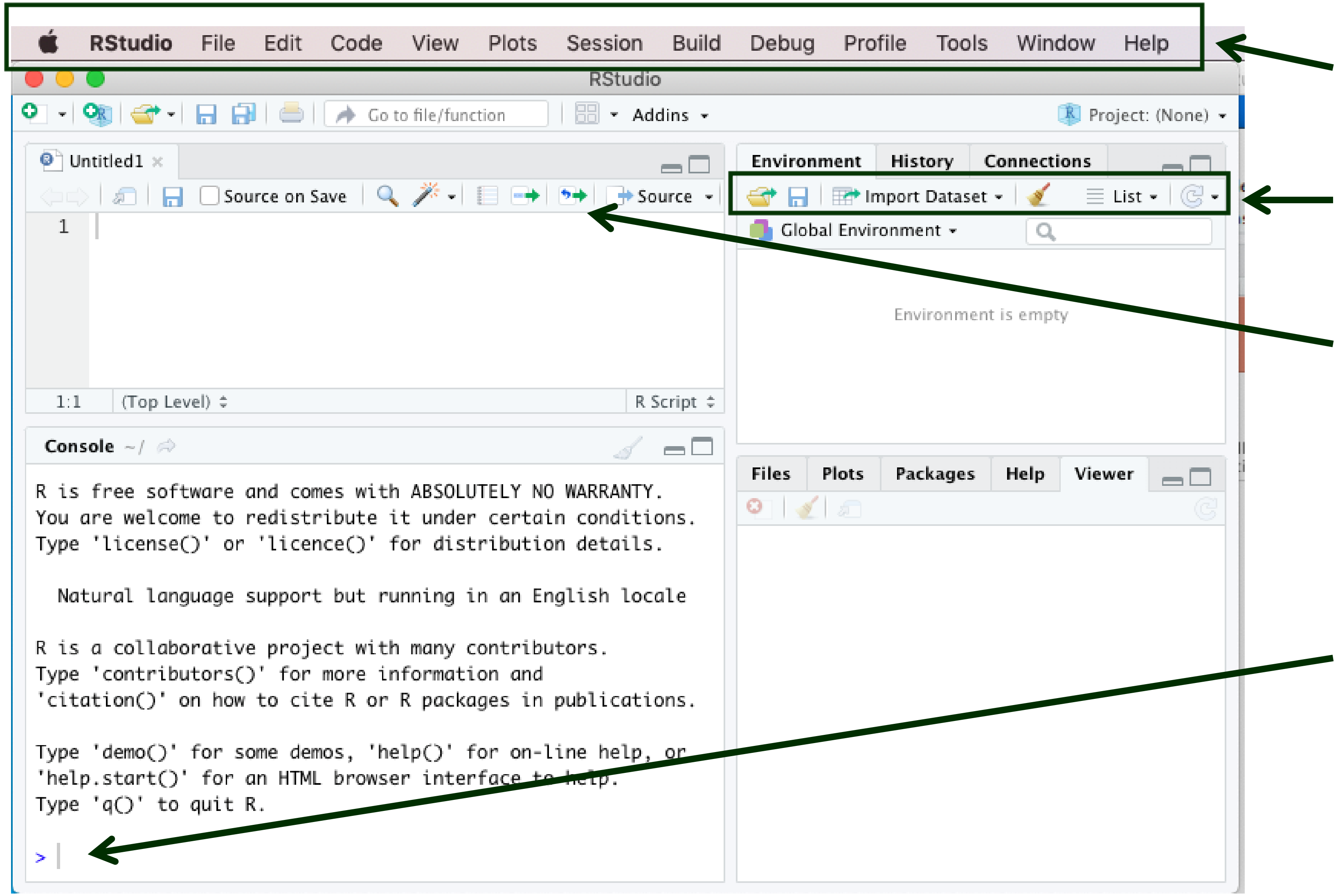
st.select.phen( )

st.genetic.distances( )

st.plot.ggt( )

Section 2. Getting started with R and RStudio

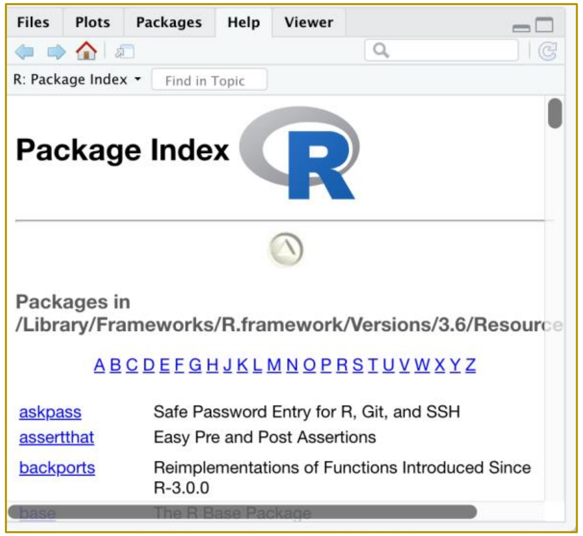
There are many ways to do most basic tasks in RStudio



```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

- **Selecting options in pull-down menus**
- **Clicking on buttons in panels**
- **Running R commands from a script in the Script Editor panel**
  - This makes your analyses more reproducible.
- **Writing R commands in the Console panel**

There are many ways to get help

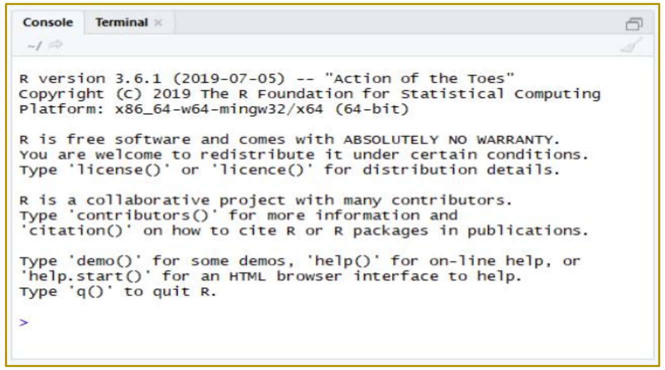


Package Index

Packages in
/Library/Frameworks/R.framework/Versions/3.6/Resourc

A B C D E F G H J K L M N O P R S T U V W X Y Z

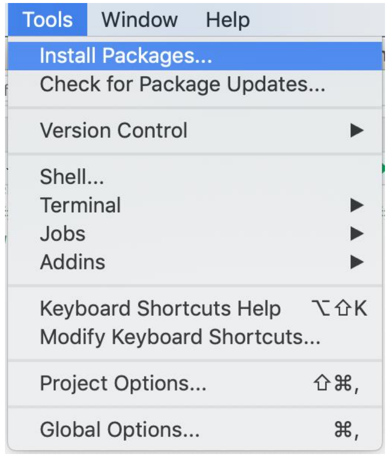
askpass	Safe Password Entry for R, Git, and SSH
assertthat	Easy Pre and Post Assertions
backports	Reimplementations of Functions Introduced Since R-3.0.0
base	The R Base Package

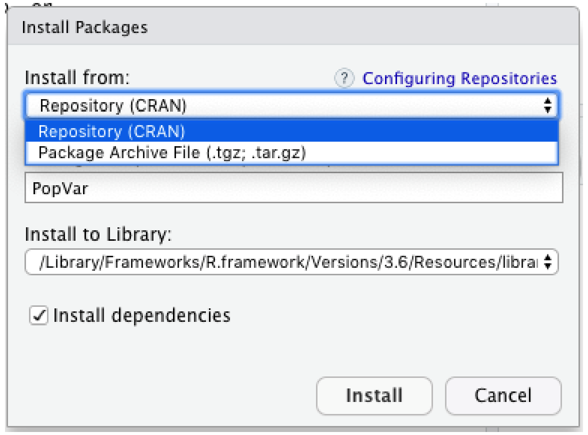


```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

- **Use the reference manuals**
- **Use the Help panel**
  - Click on a function name to learn more about it and its options
- **Start writing a command in the Console panel**
  - RStudio will show the variables and functions starting with those letters
  - Word completion is your friend
    - It helps to avoid spelling mistakes
- **Hovering above a command in the Console panel**
  - RStudio will show you the available options

Installing the SelectionTools and PopVar packages



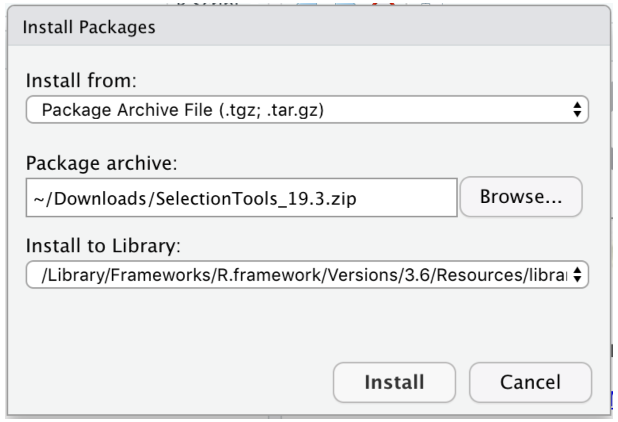


Install from: Repository (CRAN)

PopVar

Install to Library: /Library/Frameworks/R.framework/Versions/3.6/Resources/librai

☑ Install dependencies



Install from: Package Archive File (.tgz; .tar.gz)

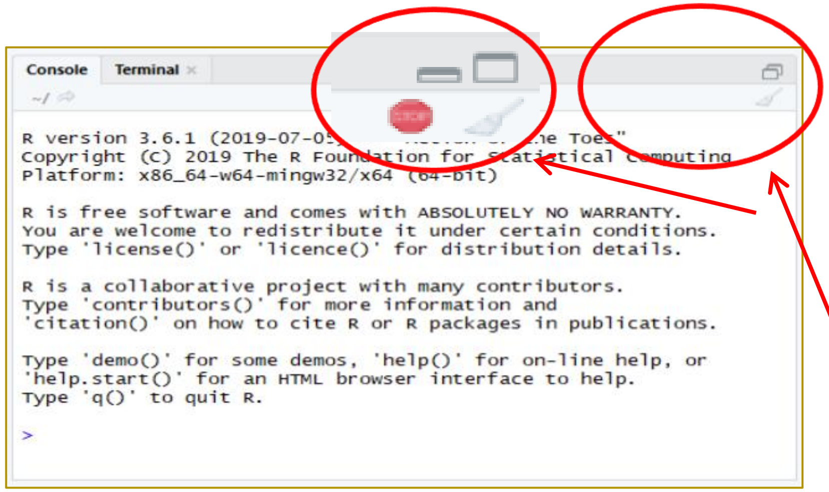
Package archive: ~/Downloads/SelectionTools_19.3.zip

Install to Library: /Library/Frameworks/R.framework/Versions/3.6/Resources/librai

- **The PopVar package can be installed directly from the CRAN repository**
- **The SelectionTools package must first be downloaded as a Package Archive File before being installed :**

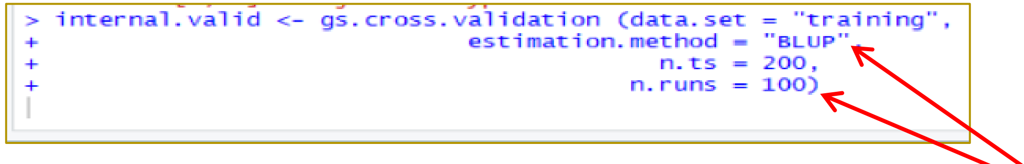
population-genetics.uni-giessen.de/~software/

The R console is where you execute R commands



```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

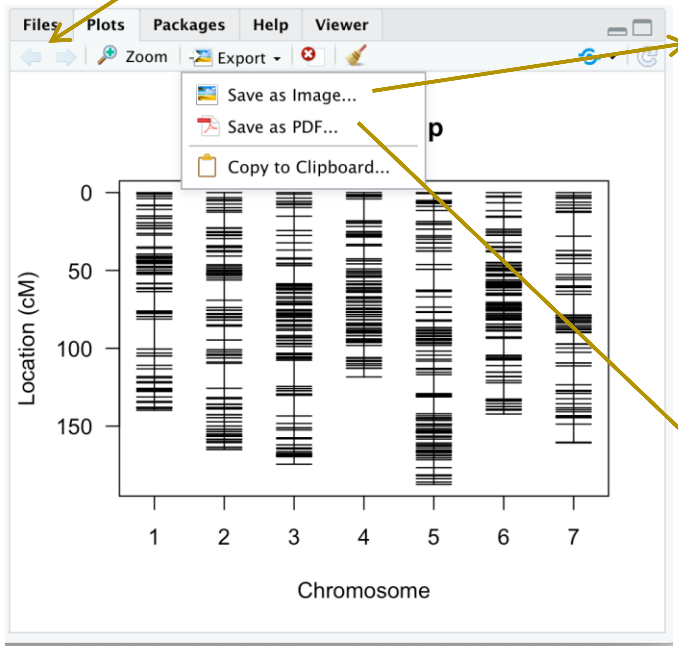
- **There are many ways to input and run R commands :**
  - By highlighting them in a script and clicking the “Run” button from the “Script Editor” panel
    - This makes your analyses more reproducible.
  - Using the arrow keys to scroll through the previous commands
  - Copying and pasting from a text editor
    - Warning: Mac, Windows and Linux word editors use different codes at the end of a line. Some are not recognized correctly by R.
  - Writing them directly written in the console
- **Frequent problems**
  - Want to know if a command is still running?
    - Check if there is a “stop” button at the top of the panel
  - Stuck” in a command (a + symbol is showing on the left) ?
    - Check for unpaired symbols such as ‘, “, (or [



```
> internal.valid <- gs.cross.validation (data.set = "training",
+                                        estimation.method = "BLUP",
+                                                       n.ts = 200,
+                                                     n.runs = 100)
|
```

Analysis results and plots are not saved automatically

Use arrows to move between plots



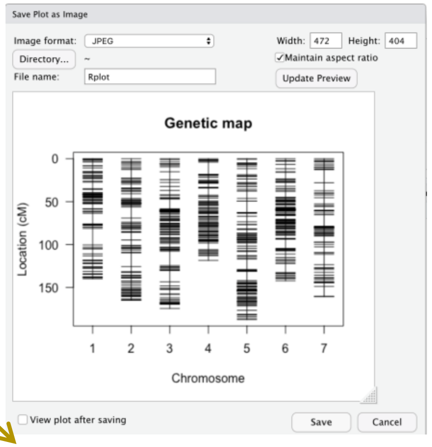

Save Plot as Image

Image format: JPEG

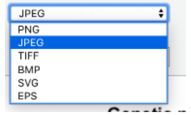
Directory... ~

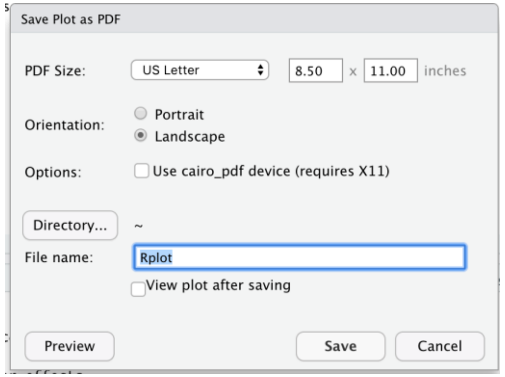
File name: Rplot

Width: 472 Height: 404

☑ Maintain aspect ratio

□ View plot after saving




Save Plot as PDF

PDF Size: US Letter 8.50 x 11.00 inches

Orientation: ○ Portrait ● Landscape

Options: □ Use cairo_pdf device (requires X11)

Directory... ~

File name: Rplot

□ View plot after saving

This plot was generated with PopVar “map.plot = TRUE” option.

- **Many functions output results in a table or list format**
  - Results can be viewed in the Viewer or the Console panels.
  - However, when there are many columns, it is usually easier to export tables and use Excel to get an overall view of them.

- **Statistics and plots can be easily generated from those results with R commands.**
  - The Plot panel can be used to visualize and save/export plots.
    - They can be exported as an image (6 available formats) or in PDF format.

Section 3. Data handling with SelectionTools and PopVar

Reading in options

Preprocessing options

- **Reading in options**
  - Genotypes
  - Phenotypes
  - Map
- **Preprocessing options**
  - Filtering, subsetting and duplicating
  - Imputing

Reading in datasets

SelectionTools

```
st.read.marker.data(filename,
                    format = "m",
                    data.set = "default")
```

```
st.read.performance.data(in.filename,
                         data.set = "default")
```

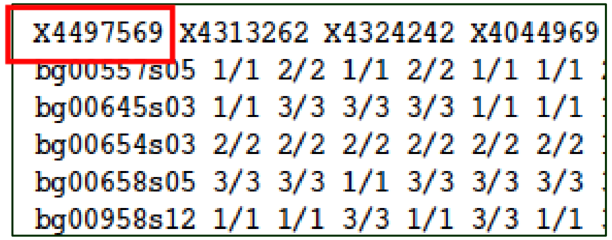
```
st.read.map(filename,
            format = “mcp”
            skip = 1,
            data.set = "default")
```

PopVar

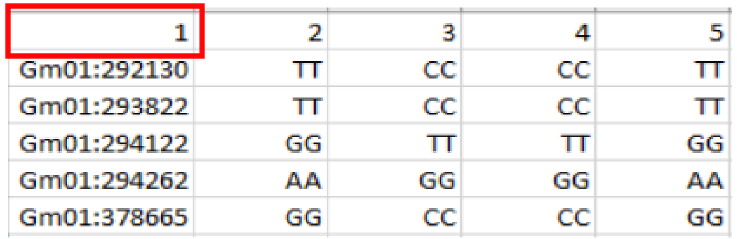
```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
            min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
            remove.dups = TRUE, impute = "EM", map.plot = TRUE,
            models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
            "BL", "BRR"), nIter = 12000, burnIn = 3000,
            frac.train = 0.6, nCV.iter = 100,
            nFold = NULL, nFold.reps = 1,
            parents = NULL, crossing.table = NULL,
            nInd = 200, nSim = 25, tail.p = 0.1)
```

Reading in genotypic data with SelectionTools

```
st.read.marker.data(filename,
                format = ”m",
              data.set = "default")
```



```
 X4497569 X4313262 X4324242 X4044969
bg00557s05 1/1 2/2 1/1 2/2 1/1 1/1
bg00645s03 1/1 3/3 3/3 3/3 1/1 1/1
bg00654s03 2/2 2/2 2/2 2/2 2/2 2/2
bg00658s05 3/3 3/3 1/1 3/3 3/3 3/3
bg00958s12 1/1 1/1 3/3 1/1 3/3 1/1
```

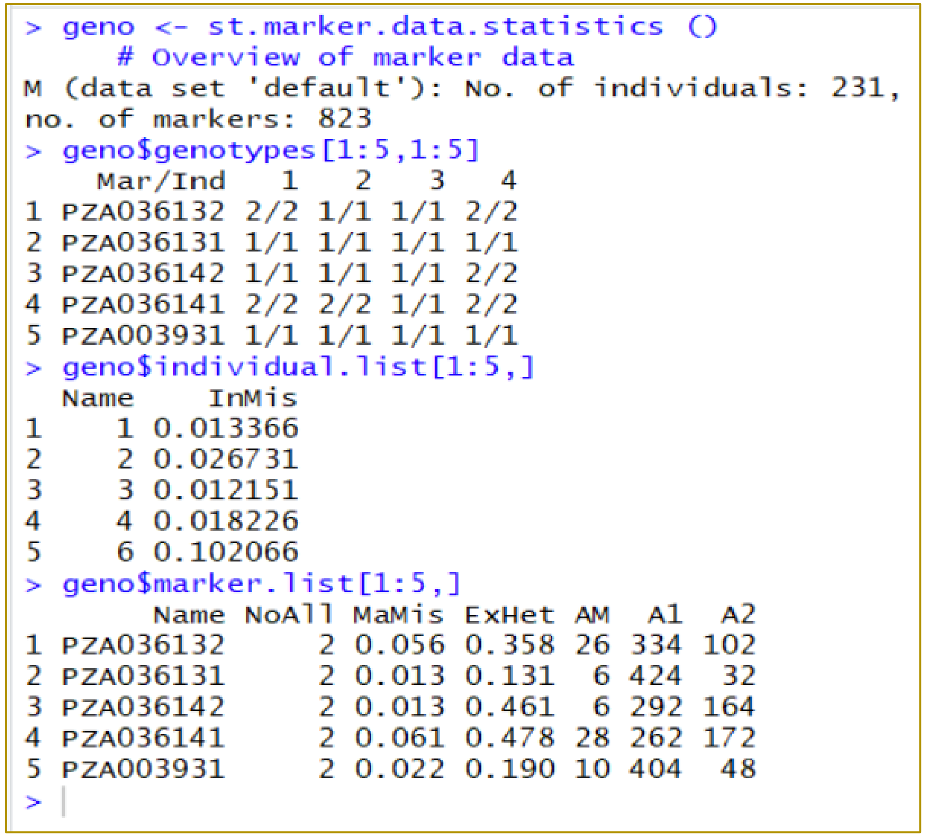


1	2	3	4	5
Gm01:292130	TT	CC	CC	TT
Gm01:293822	TT	CC	CC	TT
Gm01:294122	GG	TT	TT	GG
Gm01:294262	AA	GG	GG	AA
Gm01:378665	GG	CC	CC	GG

- **The default file format is the matrix format ("m"):**
  - First row = individual names
    - No ID for the marker column
  - First column = marker names
  - Genotype separators = tabs, blanks and ";"
- **Three other file formats are accepted :**
    - "t" = transposed matrix, "l" = list, "n" = NTSys
- **The genotypes must be in diploid format:**
  - Allele separators = nothing or a slash
  - Allele codes = numbers or alphanumeric codes
  - Missing value codes = - or -1
- **Warnings**
  - Alphanumeric codes are recoded internally as numbers.
  - There are no data-imputation options available.

Visualizing genotypic datasets used by SelectionTools requires running a special function

st.marker.data.statistics(filename="marker.stats",
                                                            data.set="default")



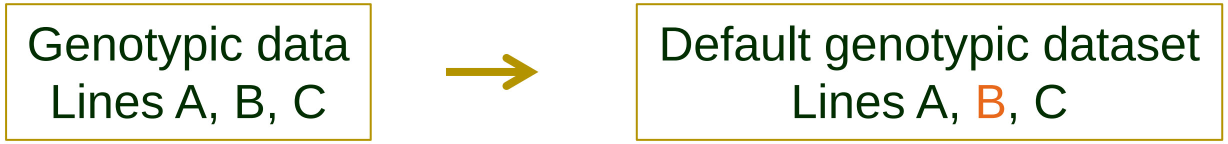
```
> geno <- st.marker.data.statistics ()
     # Overview of marker data
M (data set 'default'): No. of individuals: 231,
no. of markers: 823
> geno$genotypes[1:5,1:5]
    Mar/Ind    1   2   3   4
1 PZA036132 2/2 1/1 1/1 2/2
2 PZA036131 1/1 1/1 1/1 1/1
3 PZA036142 1/1 1/1 1/1 2/2
4 PZA036141 2/2 2/2 1/1 2/2
5 PZA003931 1/1 1/1 1/1 1/1
> geno$individual.list[1:5,]
  Name     InMis
1     1 0.013366
2     2 0.026731
3     3 0.012151
4     4 0.018226
5     6 0.102066
> geno$marker.list[1:5,]
       Name NoAll MaMis ExHet AM  A1  A2
1 PZA036132     2 0.056 0.358 26 334 102
2 PZA036131     2 0.013 0.131  6 424  32
3 PZA036142     2 0.013 0.461  6 292 164
4 PZA036141     2 0.061 0.478 28 262 172
5 PZA003931     2 0.022 0.190 10 404  48
> |
```

- **Marker data are stored internally in “default” variables.**
  - These “default” variables are not displayed in the “Environment” panel.
- **To get an overview of the marker dataset, run the st.marker.data.statistics function.**
  - It creates 3 variables that can be used by R commands
    - **$genotypes** = genotypic dataset
    - **$indivdiual.list** = individual information
      - frequency of missing data for each individual (InMis)
    - **$marker.list** = marker information
      - number of alleles observed at the marker (NoAll),
      - frequency of missing values for each marker (MaMis)
      - expected heterozygosity (ExHet)
      - count of the observed alleles (A1, A2…)

Many functions alter the hidden ”default” variables storing the information about the genotypic dataset

Importing datasets

Step 1. Import genotypic data with the “st.read.marker.data” function

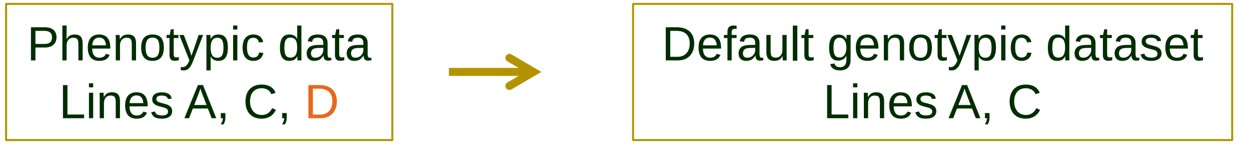


Genotypic data
Lines A, B, C

→

Default genotypic dataset
Lines A, B, C

Step 2. Import phenotypic data with the “st.read.performance” function



Phenotypic data
Lines A, C, D

→

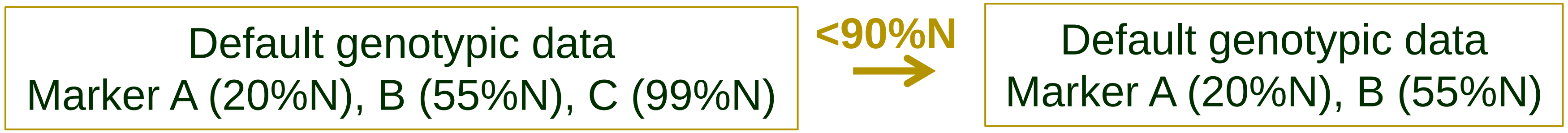
Default genotypic dataset
Lines A, C

- **Warning: In SelectionTools, the genotypic dataset is adjusted according to the content of the phenotypic dataset.**
  - If the genotypic dataset contains individuals that don’t have a phenotype, they are discarded.
- **In PopVar, individuals from the genotypic dataset are kept even if they don’t have a phenotype.**

It is often hard to keep track of changes made to these hidden “default” variables

Testing different settings of a function
(ex. the “MaMis.MAX” option of the “st.restrict.marker.data” function)

Test 1. Testing a relaxed setting for

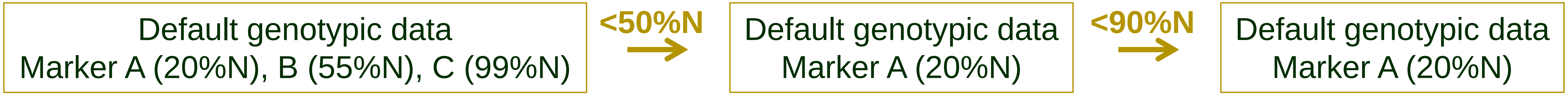


Default genotypic data
Marker A (20%N), B (55%N), C (99%N)

<90%N →

Default genotypic data
Marker A (20%N), B (55%N)

Test 2. Testing a more stringent setting before a more relaxed one



Default genotypic data
Marker A (20%N), B (55%N), C (99%N)

<50%N →

Default genotypic data
Marker A (20%N)

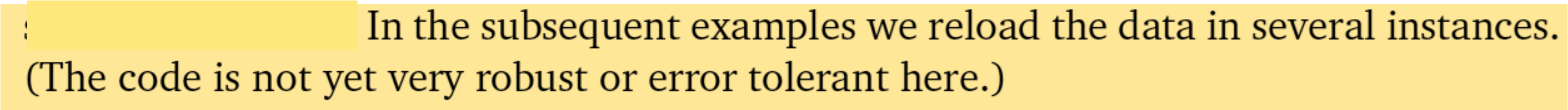
<90%N →

Default genotypic data
Marker A (20%N)

Best practices when working with SelectionTools

- **In the manual, it is highly recommended to reload the datasets between each analysis.**

ST-manual, p55



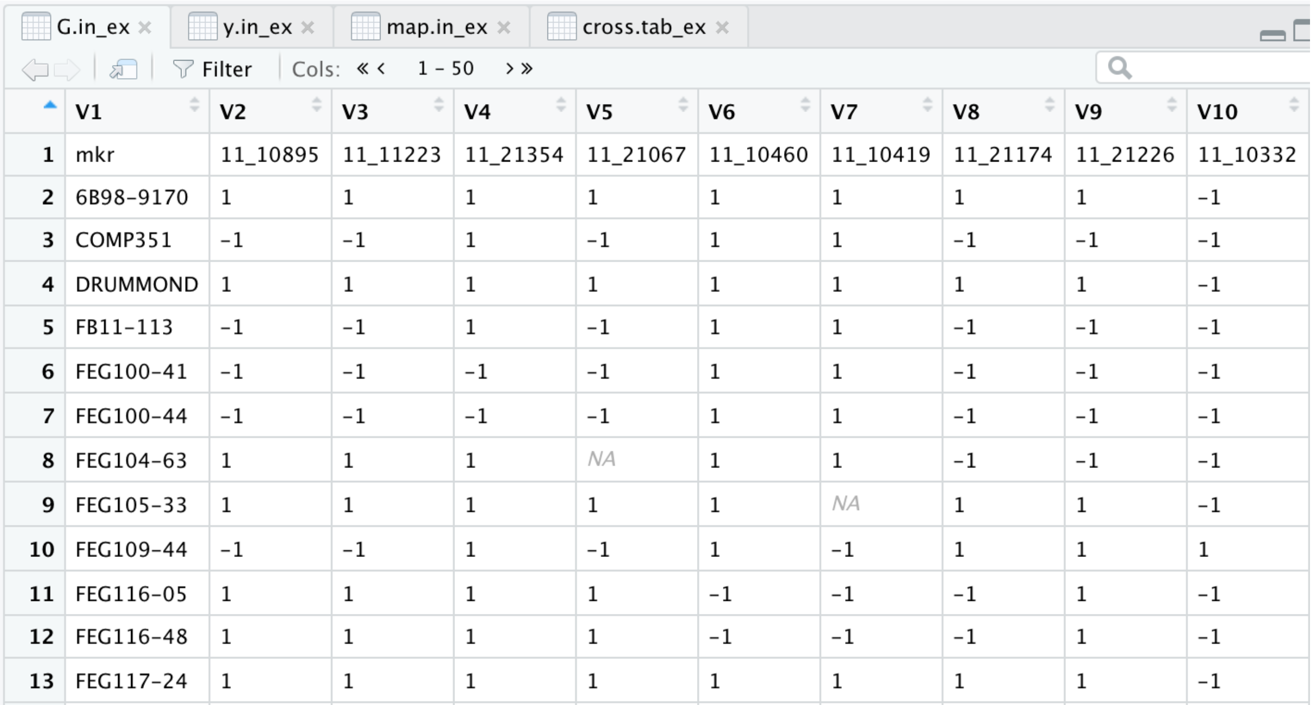
In the subsequent examples we reload the data in several instances. (The code is not yet very robust or error tolerant here.)

- **Since it is often hard to keep track of changes to the "default" variables:**
  - I highly recommend to use already filtered and imputed genotypic datasets when working with SelectionTools.
  - It is also a good idea to use named datasets when working with the data.set option instead of using the ”default” dataset.

data.set = "default” => data.set = ”newfilename"

Reading in genotypic datasets with PopVar

PopVar “G.in” option

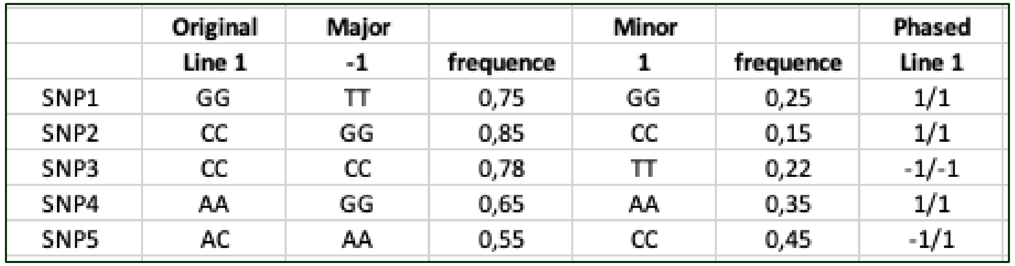


	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
1	mkr	11_10895	11_11223	11_21354	11_21067	11_10460	11_10419	11_21174	11_21226	11_10332
2	6B98-9170	1	1	1	1	1	1	1	1	-1
3	COMP351	-1	-1	1	-1	1	1	-1	-1	-1
4	DRUMMOND	1	1	1	1	1	1	1	1	-1
5	FB11-113	-1	-1	1	-1	1	1	-1	-1	-1
6	FEG100-41	-1	-1	-1	-1	1	1	-1	-1	-1
7	FEG100-44	-1	-1	-1	-1	1	1	-1	-1	-1
8	FEG104-63	1	1	1	NA	1	1	-1	-1	-1
9	FEG105-33	1	1	1	1	1	NA	1	1	-1
10	FEG109-44	-1	-1	1	-1	1	-1	1	1	1
11	FEG116-05	1	1	1	1	-1	-1	-1	1	-1
12	FEG116-48	1	1	1	1	-1	-1	-1	1	-1
13	FEG117-24	1	1	1	1	1	1	1	1	-1

- **File format:**
  - First row = marker names
  - First column = entry (individual) names
- **Genotype format:**
  - PopVar requires **phased** genotypic data.
  - Allele codes:
    - 1: homozygous for minor allele
    - 0: heterozygous
    - -1: homozygous for major allele
    - NA: missing value
      - Warning: NA will automatically be imputed by PopVar using the rrBLUP package.

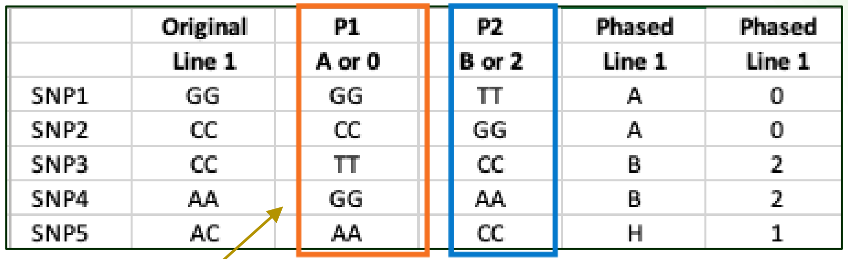
What is a phased genotypic dataset?

Phasing to allelic frequency



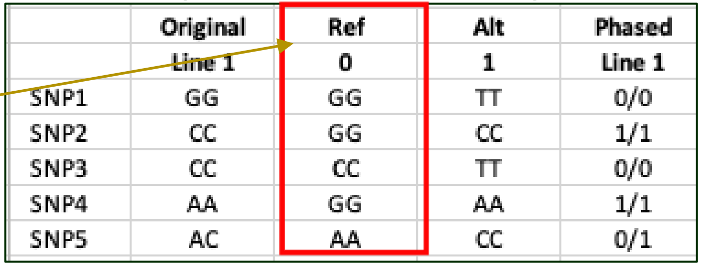
	Original	Major		Minor		Phased
	Line 1	-1	frequence	1	frequence	Line 1
SNP1	GG	TT	0,75	GG	0,25	1/1
SNP2	CC	GG	0,85	CC	0,15	1/1
SNP3	CC	CC	0,78	TT	0,22	-1/-1
SNP4	AA	GG	0,65	AA	0,35	1/1
SNP5	AC	AA	0,55	CC	0,45	-1/1

Phasing to parental origin



	Original	P1	P2	Phased	Phased
	Line 1	A or 0	B or 2	Line 1	Line 1
SNP1	GG	GG	TT	A	0
SNP2	CC	CC	GG	A	0
SNP3	CC	TT	CC	B	2
SNP4	AA	GG	AA	B	2
SNP5	AC	AA	CC	H	1

Phasing to a reference genome



	Original	Ref	Alt	Phased
	Line 1	0	1	Line 1
SNP1	GG	GG	TT	0/0
SNP2	CC	GG	CC	1/1
SNP3	CC	CC	TT	0/0
SNP4	AA	GG	AA	1/1
SNP5	AC	AA	CC	0/1

Warning:
reference genome =
genotypes from one line

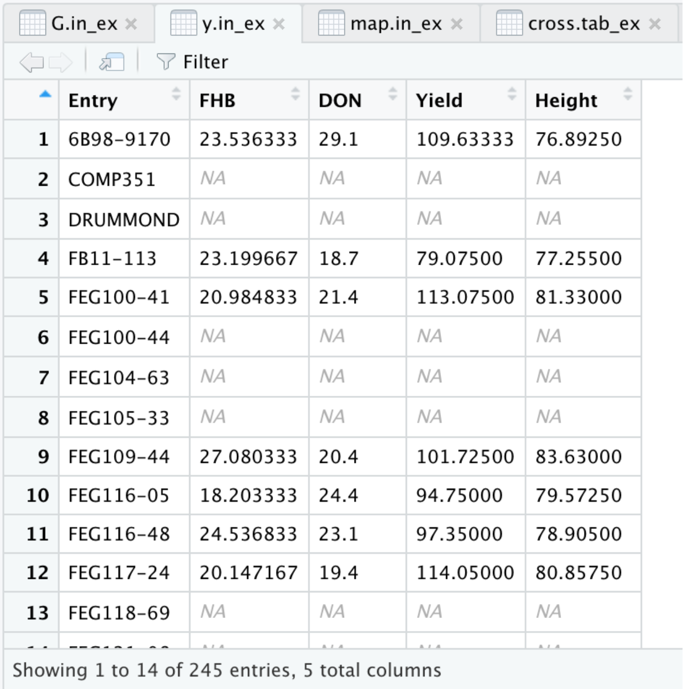
- **In a phased dataset, genotypes are recoded according to a reference.**
- **Different softwares and analyses may require different references.**
  - SNP-calling softwares will score alleles according to a reference genome.
  - Mapping softwares will require alleles to be score according to parental origin.
  - Most genomic prediction softwares will require alleles to be recoded according to their allele frequency in the training set.

Reading in phenotypic datasets

SelectionTools

```
st.read.performance.data(in.filename,
                         data.set = "default")
```

PopVar "y.in" option



	Entry	FHB	DON	Yield	Height
1	6B98-9170	23.536333	29.1	109.63333	76.89250
2	COMP351	NA	NA	NA	NA
3	DRUMMOND	NA	NA	NA	NA
4	FB11-113	23.199667	18.7	79.07500	77.25500
5	FEG100-41	20.984833	21.4	113.07500	81.33000
6	FEG100-44	NA	NA	NA	NA
7	FEG104-63	NA	NA	NA	NA
8	FEG105-33	NA	NA	NA	NA
9	FEG109-44	27.080333	20.4	101.72500	83.63000
10	FEG116-05	18.203333	24.4	94.75000	79.57250
11	FEG116-48	24.536833	23.1	97.35000	78.90500
12	FEG117-24	20.147167	19.4	114.05000	80.85750
13	FEG118-69	NA	NA	NA	NA

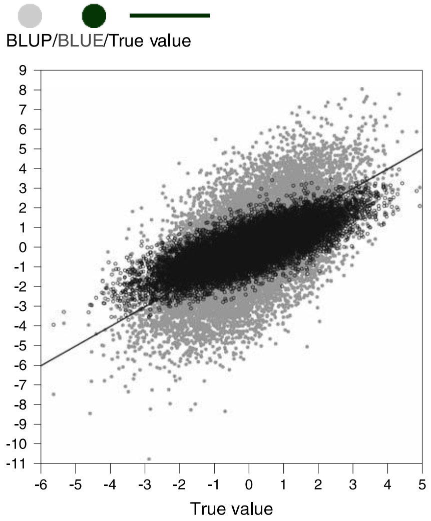
Showing 1 to 14 of 245 entries, 5 total columns

- **SelectionTools and PopVar accept the same format.**
  - First row = column names
    - Name should reflect the trait
  - First column = entry names
  - Additional column (s) = phenotypic data
- **However, they have different input data requirements.**
  - SelectionTools :
    - Only individuals with a phenotype are allowed.
    - Only one trait is allowed.
  - PopVar :
    - All individuals from the genotypic dataset must be included in the phenotypic dataset, **even those without a phenotype.**
    - Multiple traits are accepted.

Most prediction models only allow a single value per trait

$$\text{Trait} = \underbrace{\text{Grand mean} + \text{Line}}_{\text{EBV}} + \text{Environment} + e$$

Year
Location
Block



Piepho et al. 2008
Euphytica 161:209-228

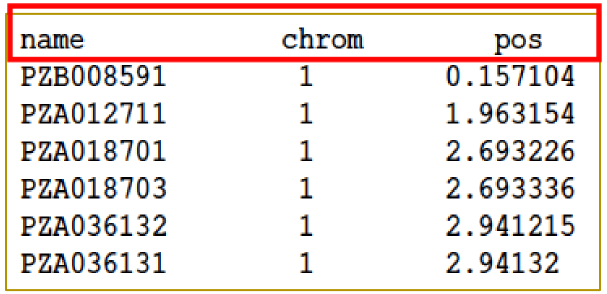
- Because BLUP involves a shrinkage toward the mean, extreme values may be slightly under- or over-estimated.
- If needed, BLUPs can be “deregressed” to account for this effect.

- **The estimated breeding value (EBV) is often used as input for prediction models instead of the raw phenotypes.**
  - How to generate EBV will not be demonstrated in the current workshop.
- **Using EBV as input means that environmental effects cannot be taken into account by the prediction model.**
  - If appropriate datasets are available, prediction models taking into account environmental effects and genotype X environment effects can be used.
  - However, in many cases, predictions made with these more sophisticated models have a similar accuracy to those derived using EBV.
- **EBV can be calculated using BLUE (Lines = fixed effects) or BLUP (Lines = random effects).**
  - Best linear unbiased **estimations** (BLUE) can be used for multiple environment trials with very little missing data.
  - Best linear unbiased **predictions** (BLUP) can be used with highly unbalanced datasets like official provincial trials (= tables with high number of missing cells).
  - Both approaches usually give similar prediction accuracy.

Reading in genetic maps with SelectionTools

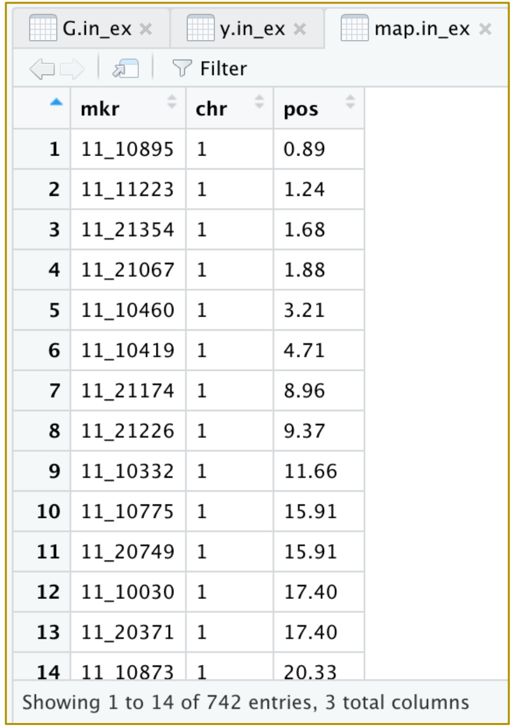
SelectionTools

```
st.read.map(filename,
        format = “mcp”
          skip = 1,
     data.set = "default")
```



```
name              chrom        pos
PZB008591           1     0.157104
PZA012711           1     1.963154
PZA018701           1     2.693226
PZA018703           1     2.693336
PZA036132           1     2.941215
PZA036131           1     2.94132
```

PopVar “map.in” option



	mkr	chr	pos
1	11_10895	1	0.89
2	11_11223	1	1.24
3	11_21354	1	1.68
4	11_21067	1	1.88
5	11_10460	1	3.21
6	11_10419	1	4.71
7	11_21174	1	8.96
8	11_21226	1	9.37
9	11_10332	1	11.66
10	11_10775	1	15.91
11	11_20749	1	15.91
12	11_10030	1	17.40
13	11_20371	1	17.40
14	11 10873	1	20.33

Showing 1 to 14 of 742 entries, 3 total columns

- **Both SelectionTools and PopVar can use the « mcp » format.**
  - 3 columns = Marker, Chromosome, Position
    - Position unit: cM
- **However, they have different header requirements.**
  - In PopVar, the first row must contain column names.
  - In SelectionTools, use skip = 1 to remove this row if it is present.
- **Other formats are available for SelectionTools.**
  - Please see the manual
- **If no genetic map is available, a physical map may be converted to an “approximate” genetic map by dividing positions by 100,000.**

Using the SoyaGen training set for genomic mating

- **Genotypic dataset**
  - The SoyaGen training set genotypic dataset was created with the FastGBS pipeline and filtered using vcftools and Tassel.
    - It can be exported from TASSEL in the diploid format required by SelectionTools.
    - The standard hapmap output from TASSEL can be easily phased and converted to the format required by PopVar using UNIX or R commands.
  - Warning. When using real datasets with PopVar, they first need to be read into RStudio with the "read.table" or "read.cvs" command.
    - Set header=**FALSE** when importing a tab-delimited **genotypic dataset** file.
    - Set header=**TRUE** when importing a tab-delimited **phenotypic dataset** or **genetic map** files.

- **Phenotypic dataset**
  - The original multi-environment phenotypic data for the training set was converted to EBV using a BLUP (Yan and Rajcan 2003, Crop Sci. 43:549-555).
- **Genetic map**
  - During most of the SoyaGen project, no genetic maps were available.
    - The physical map was converted to an "approximate" genetic map by dividing positions by 100,000 and used for genomic mating.
  - However, a consensus genetic map is now available and could be used in future analyses.

- **Read in options**
  - Genotypes
  - Phenotypes
  - Map
- **Preprocessing options**
  - Filtering, subsetting and duplicating
  - Imputing

Preprocessing datasets

SelectionTools

genotypic dataset filtration

```
st.restrict.marker.data(NoAll.MAX = 2,
                          ExHet.MIN = 0.1,
                         MaMis.MAX = 0.1,
                          InMis.MAX = 0.1,
                         data.set = "default")
```

genotypic dataset subsetting and duplicating

```
st.restrict.marker.data(ind.list = c(x,y,z),
                          ind.file = filename,
                         mar.list = c(a,b,c),
                         mar.file = filename,
                         data.set = "default")
```

```
st.copy.marker.data (target.data.set = "newfile",
                      source.data.set = "default")
```

PopVar

```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
            min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
            remove.dups = TRUE, impute = "EM", map.plot = TRUE,
            models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
            "BL", "BRR"), nIter = 12000, burnIn = 3000,
            frac.train = 0.6, nCV.iter = 100,
            nFold = NULL, nFold.reps = 1,
            parents = NULL, crossing.table = NULL,
            nInd = 200, nSim = 25, tail.p = 0.1)
```

Genotypic dataset filtration options

	SelectionTools	PopVar
Maximum missing data		
- Per individual	Yes	Yes
- Per marker	Yes	Yes
Missing data imputation	No	Yes
Minimum minor allele frequency	**No**	Yes
Minimun expected heterozygocity	Yes	No
Maximum number of alleles	Yes	No
Duplicate entry removal	No	Yes

- **Warning: there is no maf filtering option in SelectionTools**
- **However, the expected heterozygosity (ExHet) is also a measure of the allelic diversity**
  - It can be used to filter both biallelic and multiallelic markers
    - For biallelic markers, ExHet = 0.095 is equal to maf = 0.05

$$\mathrm{ExHet} = 1 - \sum_{a \in \mathcal{A}} f_a^2$$

Example of ExHet calculation:
freq A: 0,05; freq T: 0,95
ExHet = 1 – [(0,05*0,05)+(0,95*0,95)]
ExHet = 1 – [0,0025+0,9025]
ExHet = 1 – 0,905
ExHet = 0,095

Subsetting and duplicating datasets with SelectionTools

```
st.restrict.marker.data(ind.list = c(x,y,z),
                         ind.file = filename,
                        mar.list = c(a,b,c),
                        mar.file = filename,
                        data.set = "default")
```

```
st.copy.marker.data (target.data.set = "newfile",
                     source.data.set = "default")
```

- **The "st.restrict.marker.data" function can both filter datasets and create subsets.**
- **This function should mainly be used to create subsets and re-processing them.**
  - I highly recommend using already filtered and imputed datasets when working with SelectionTools
- **Subsets can be created by selecting individuals (ind) or markers (mar) specified in a list or a file.**
- **Warning: By default, this function will modify the « default » dataset.**
  - If you wish to keep the original dataset intact:
    - make a copy of it under a new name
    - use this name in the « data.set » option

Section 4. Selecting crosses using conventional approaches with SelectionTools

Evaluate crossing partners using known performance

Evaluate crossing partners using genetic distances

Evaluate crossing partners using allelic composition at specific loci

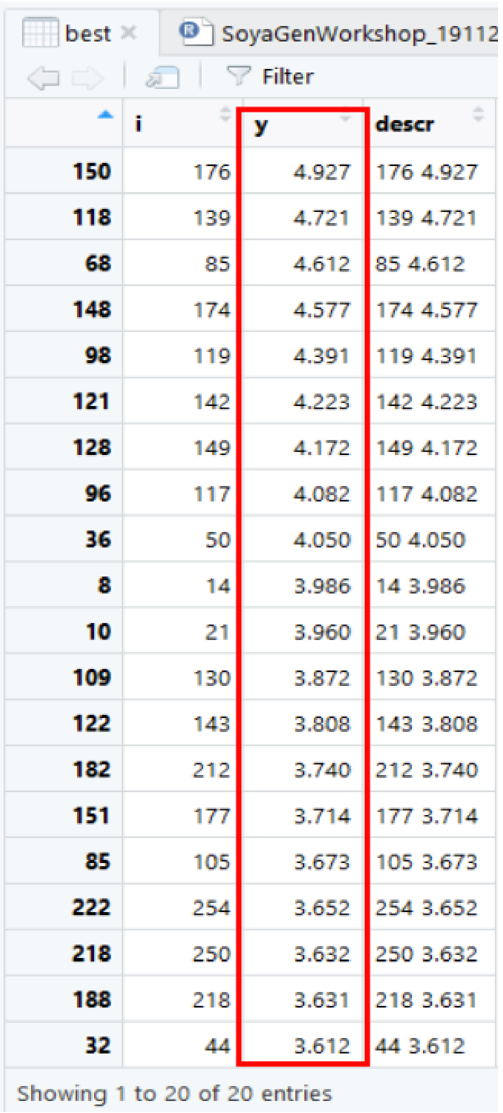
Selecting crosses using conventional approaches

Line A with good performance X Line B with good performance = Progeny line with better performance

- **A breeder’s goal**
  - Select crosses between elite parents with complementary genetic information that could be combined to generate superior progeny.
- **Strategy**
  - Step 1. Identify the highest performing lines.
  - Step 2. Evaluate their genetic relationship to avoid crosses between highly related lines.
    - These are more likely to carry identical genetic information at critical loci.
  - Step 3. Evaluate allele distribution at specific loci in the best lines to favour crosses that will maintain or segregate specific allele combinations.

- **SelectionTools can help breeders perform these 3 steps.**

Evaluate crossing partners using known performance



	i	y	descr
150	176	4.927	176 4.927
118	139	4.721	139 4.721
68	85	4.612	85 4.612
148	174	4.577	174 4.577
98	119	4.391	119 4.391
121	142	4.223	142 4.223
128	149	4.172	149 4.172
96	117	4.082	117 4.082
36	50	4.050	50 4.050
8	14	3.986	14 3.986
10	21	3.960	21 3.960
109	130	3.872	130 3.872
122	143	3.808	143 3.808
182	212	3.740	212 3.740
151	177	3.714	177 3.714
85	105	3.673	105 3.673
222	254	3.652	254 3.652
218	250	3.632	250 3.632
188	218	3.631	218 3.631
32	44	3.612	44 3.612

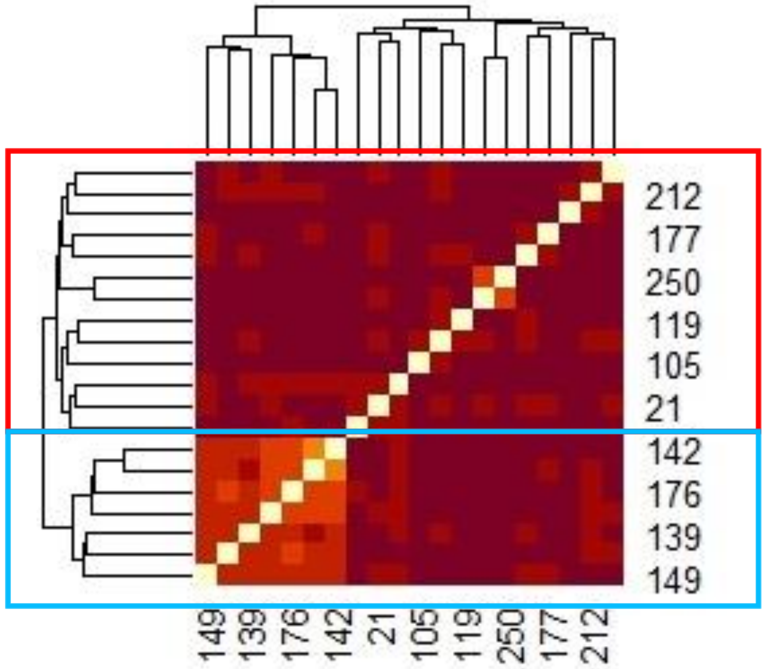
Showing 1 to 20 of 20 entries

```
st.select.phen (pheno,
                n = 20,
         decreasing = TRUE)
```

- **Step 1. Identify the 20 highest performing lines with the "st.select.phen" function.**
  - This function will automatically sort lines by phenotype and create a subset of a specified number of top lines.
    - Options:
      - n = Number of lines to select
      - decreasing = Select lines with the highest (TRUE) or lowest (FALSE) values

Evaluate crossing partners using genetic distances

```
st.genetic.distances(measure = "mrd",
                     format = "l",
                     data.set = "default")
dm <- (as.matrix(dist.mat))
heatmap (dm, scale = "none")
```

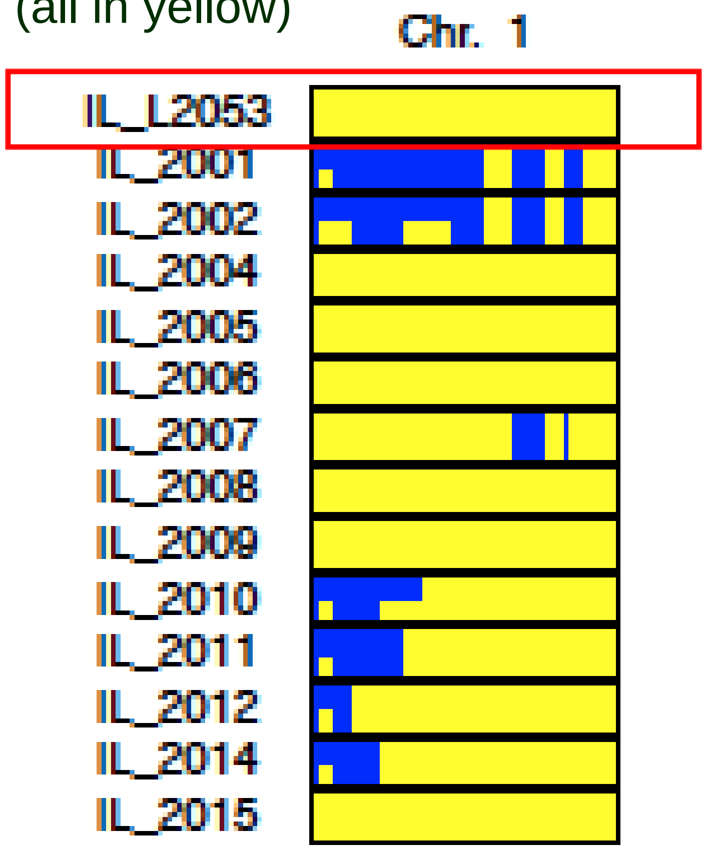


- **Step 2. Visualize the genetic relationships of the best 20 lines with a heatmap**
  - Step 2.1. Calculate genetic distances between the lines using the “st.genetic.distances” function.
    - Options for measure
      - "**mrd**” = modified Roger distance, "rd” = Rogers distance and "euc” = Euclidean distance
    - Options for format
      - "**l**" = long and "m” = matrix
  - Step 2.2. Convert distance matrix to standard R matrix
  - Step 2.3. Create a heatmap using a R command

Crosses should be done between best lines from different clusters to avoid crosses between highly related lines

Compare allele distribution at specific loci in the best lines

Reference line
(all in yellow)



```
st.def.hblocks (hap = 1 ,      # number of units
                    hap.unit = 1,      # type of units: 0, 1, 2
                    data.set = "default",
st.recode.hbc (reference = 1,
                              data.set = "default" )
st.plot.ggt(data.set = "default",
                  ifilename = ”")
```

- **Step 3. Compare the distribution of alleles at specific loci using graphical genotypes**
  - Step 3.1. Define haplotypes using the “st.def.hblocks” function.
  - Step 3.2. Recode the genotypes using the “st.recode.hbc » function.
  - Step 3.3. Plot graphical genotypes using the “st.plot.ggt” function.
    - ifilename = file containing a list of individuals to plot
- **Check the manual for further details**

Section 5. Genomic Mating : Selecting crosses using genome-wide predictions

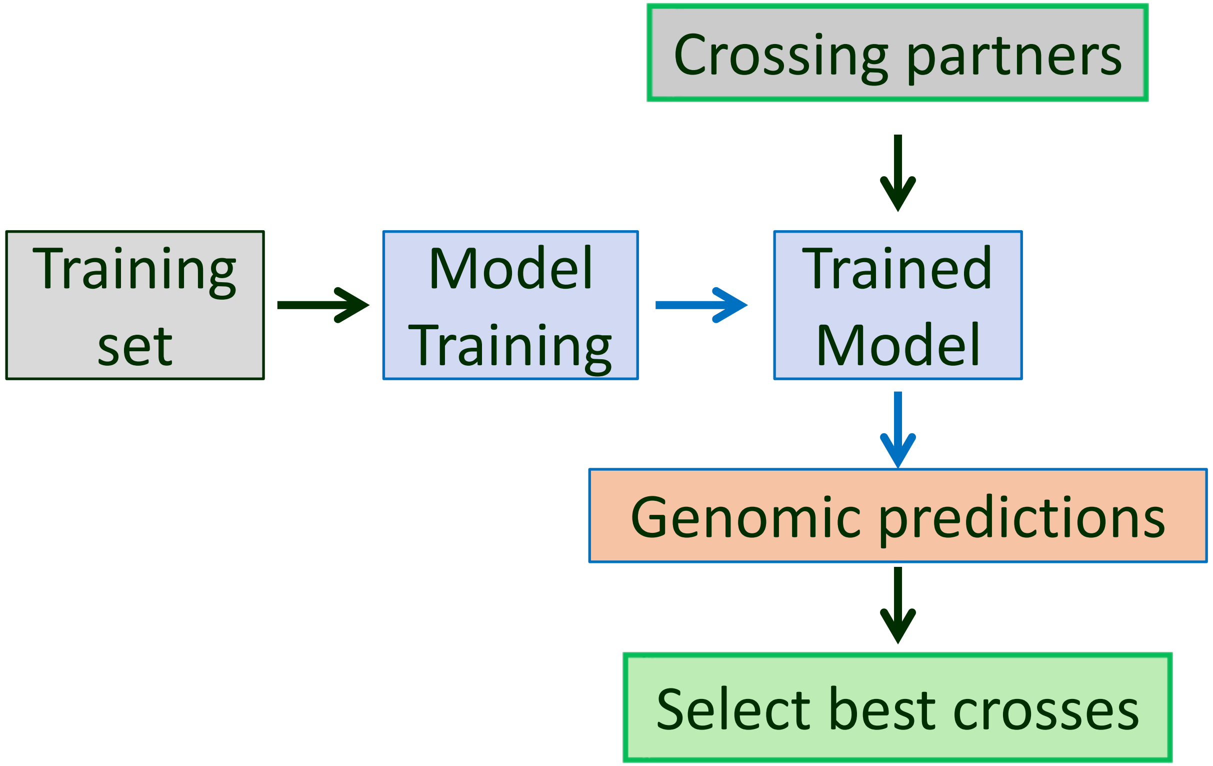
Model training and selection

Predicting progeny phenotypes

Selecting crosses with SelectionTools

Selecting crosses with PopVar

Main steps in genomic mating

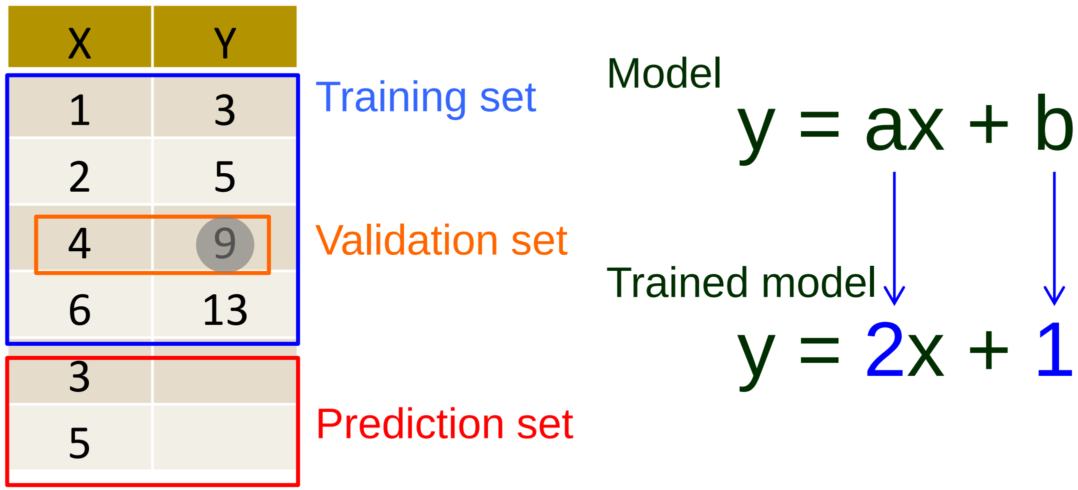


- Step 1: Reading in and preprocessing datasets from the training set
- **Step 2: Model selection**
- **Step 3: Prediction of progeny phenotypes**
- **Step 4: Cross selection**

- Most of these steps are done automatically with PopVar.
- With SelectionTools, the user must run these steps manually and sequentially.

Using genome-wide marker effects to predict phenotypes

Using phenotypes and genotypes of training set to estimate allelic effects



X	Y
1	3
2	5
4	9
6	13
3	
5	

Training set

Validation set

Prediction set

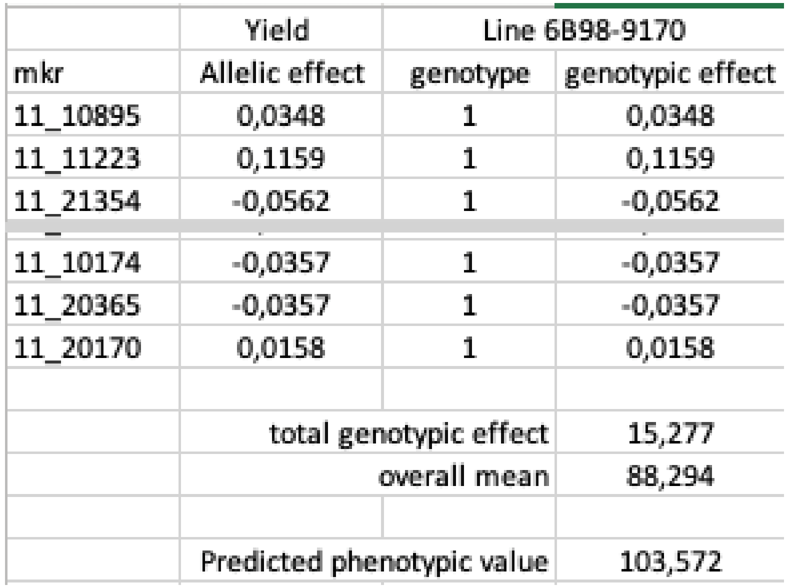
Model

$$y = ax + b$$

Trained model

$$y = 2x + 1$$

Using allelic effects and overall mean to calculate predicted phenotypes



	Yield	Line 6B98-9170	
mkr	Allelic effect	genotype	genotypic effect
11_10895	0,0348	1	0,0348
11_11223	0,1159	1	0,1159
11_21354	-0,0562	1	-0,0562
11_10174	-0,0357	1	-0,0357
11_20365	-0,0357	1	-0,0357
11_20170	0,0158	1	0,0158
		total genotypic effect	15,277
		overall mean	88,294
		Predicted phenotypic value	103,572

Model training and selection

Predicting progeny phenotypes

Selecting crosses with SelectionTools

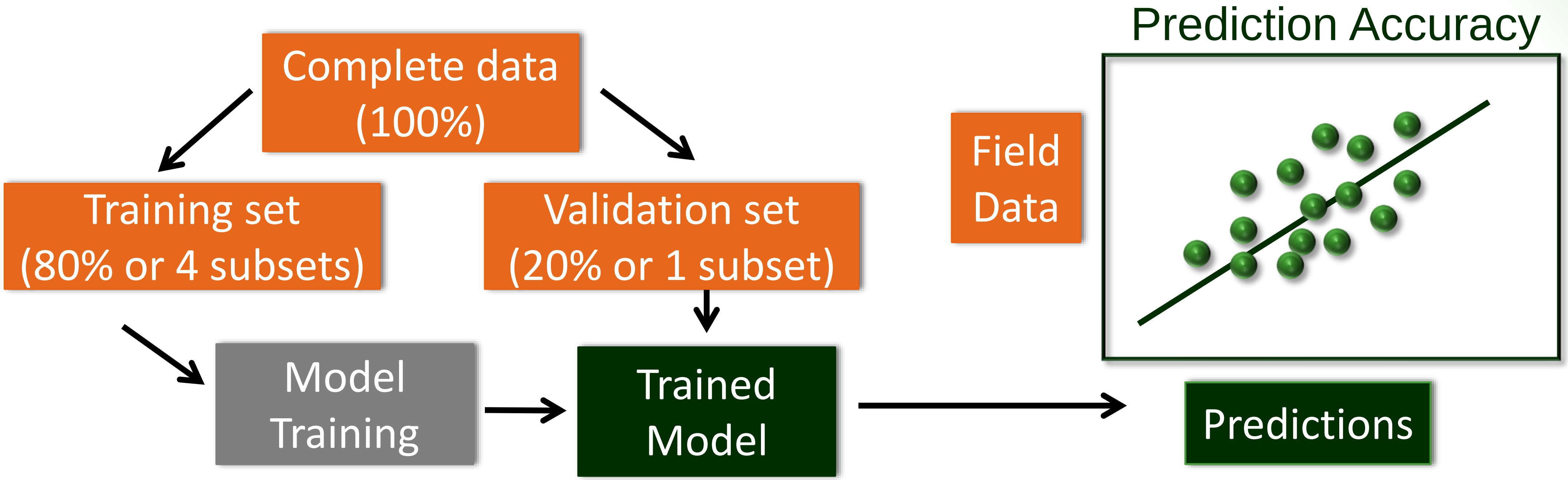
Selecting crosses with PopVar

Model selection

	SelectionTools	PopVar
Models available	rrBLUP, RMLA	rrBLUP, BayesA, BayesB, BayesC, BL, BRR
Prediction accuracy assessment		
- Cross-Validation		
- random sampling	Yes	Yes
- fold sampling	No	Yes
- External validation	Yes	No
Model selection	Manual	Automatic

- **Cross-validation by random sampling or fold sampling?**
  - Because random sampling is computationally more efficient, it is often used for cross-validation even though fold sampling is, in theory, a statistically better approach.
  - The main drawback of random sampling is that some lines may be included in the validation set in more than one rep while others may never be included in it.

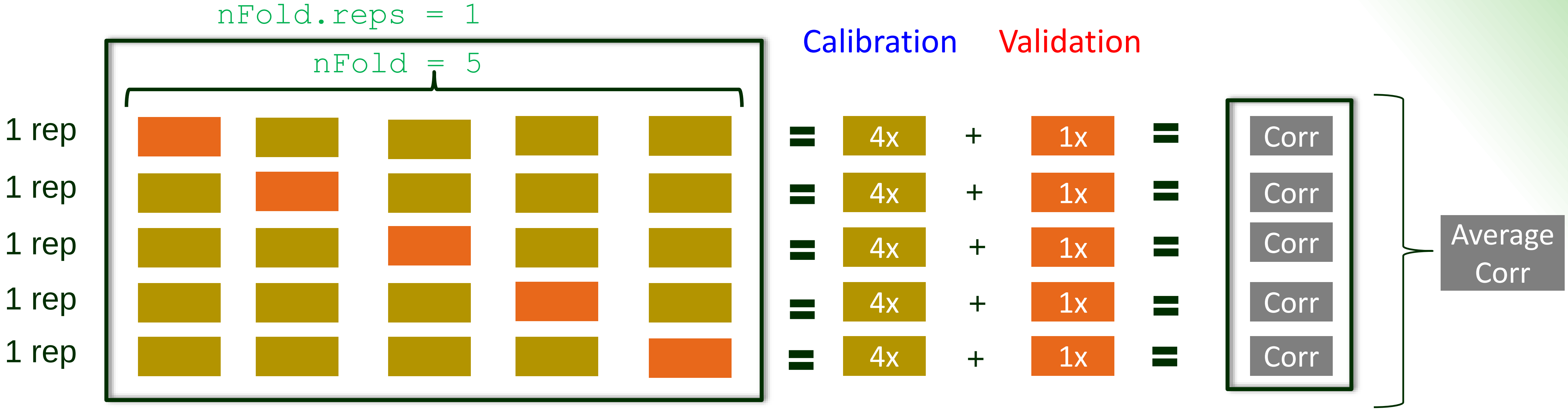
Assessing prediction accuracy by internal validation



- Advantage: Accuracy is estimated with already available field data
- Inconvenient: Internal validation usually overestimates accuracy

Cross-validation with PopVar: random vs fold sampling

Fold sampling (ex. 5-folds)



nFold.reps = 1

nFold = 5

Calibration Validation

1 rep = 4x + 1x = Corr

1 rep = 4x + 1x = Corr

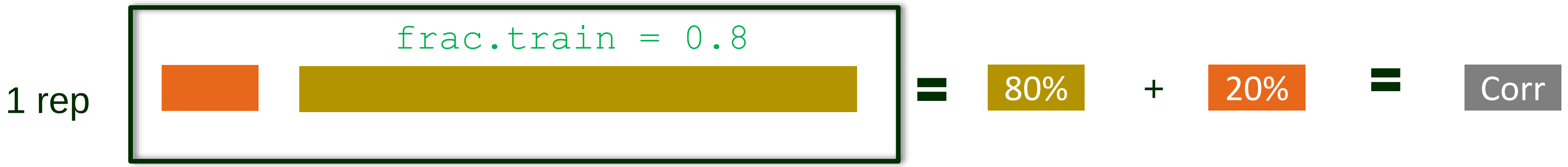
1 rep = 4x + 1x = Corr

1 rep = 4x + 1x = Corr

1 rep = 4x + 1x = Corr

Average Corr

Random sampling



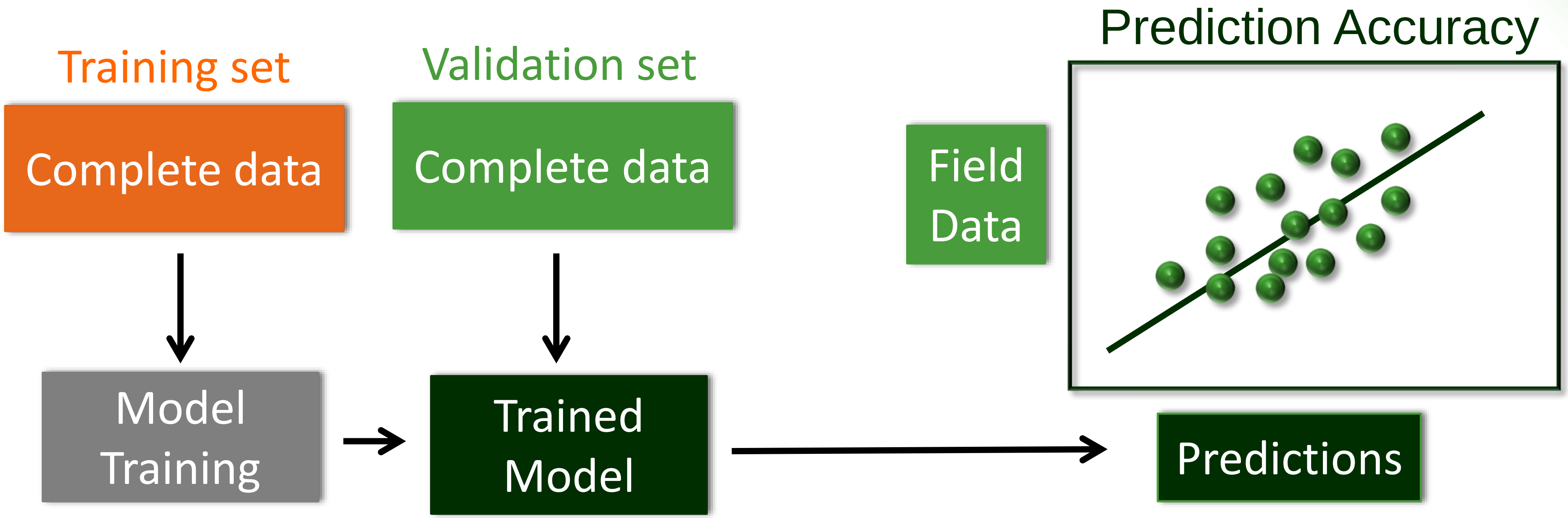
nCV.iter = 1

frac.train = 0.8

1 rep = 80% + 20% = Corr

Warning: With SelectionTools, only random sampling is available

Assessing prediction accuracy by external validation



- Advantage: External validation sets are usually more similar to the real prediction sets so estimates of prediction accuracy are more reliable
- Inconvenient: Generating good field data for validation sets takes time

Model selection

SelectionTools

Estimating genome-wide marker effects

```
gs.esteff.rr (method = "BLUP",
              data.set = "default")

gs.esteff.external (method = "rrBLUP",
                    data.set = "t")
```

Assessing prediction accuracy

```
gs.cross.validation (estimation.method,
                     n.ts, n.runs,
                     data.set = "default" )

gs.plot.validation(estimation.set,
                   validation.set)
```

PopVar

```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
            min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
            remove.dups = TRUE, impute = "EM", map.plot = TRUE,
            models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
            "BL", "BRR"), nIter = 12000, burnIn = 3000,
            frac.train = 0.6, nCV.iter = 100,
            nFold = NULL, nFold.reps = 1,
            parents = NULL, crossing.table = NULL,
            nInd = 200, nSim = 25, tail.p = 0.1)
```

Model selection with SelectionTools

```
gs.esteff.rr (method = "BLUP",
              data.set = "default")
```

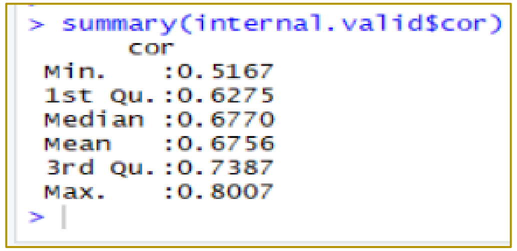
```
gs.esteff.external (method = "rrBLUP",
                    data.set = "t")
```

```
gs.predict.genotypes (training.set = "default",
                      prediction.set = "default")
```

- **Step 1. Estimating genome-wide marker effects**
  - The gs.esteff.rr function can be used to estimate maker effects with two main models:
    - BLUP (default, = rrBLUP) : constant shrinkage
    - RMLA : marker-specific shrinkage
  - SelectionTools can also use models from the R packages rrBLUP (default), regress and sommer.
    - Warning: these packages must be loaded with the R command “library” before being used
- **Step 2. Predicting phenotypes of the validation set**
  - Prediction of phenotypes for the validation set is done automatically when using one of the validation functions
    - The gs.predict.genotypes function can, however, be used to predict phenotypes of genotyped lines manually.
    - This function is used for predicting phenotypes of selection candidates during **genomic selection**.

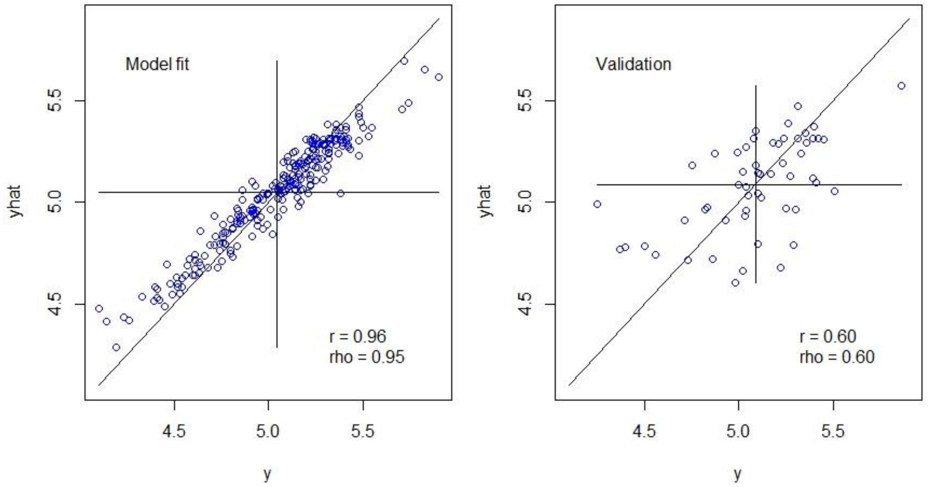
Model selection with SelectionTools

gs.cross.validation (estimation.method,
n.ts, n.runs,
data.set = "default")



```
> summary(internal.valid$cor)
      cor
 Min.   :0.5167
 1st Qu.:0.6275
 Median :0.6770
 Mean   :0.6756
 3rd Qu.:0.7387
 Max.   :0.8007
>
```

gs.plot.validation(estimation.set, validation.set)



Accuracy measures :
r = Pearson's correlation,
rho = Spearsman rank correlation

- **Step 3. Assessing prediction accuracy**
  - The gs.cross.validation function estimates prediction accuracy by cross-validation.
    - Option:
      - n.ts = number of individuals in the training set (the rest will be the validation set)
      - n.runs: number of replications to run
    - The R function "summary" can be used to visualize the mean prediction accuracy.
  - The gs.plot.validation function estimates prediction accuracy by external validation.
    - It automatically creates plots that make it easy to assess prediction accuracy.
- **Step 4. Model selection**
  - There is no function to automatically identify the best model for a given trait.
    - The user must manually test several models, compare their accuracy and select one (usually the most accurate).

Model selection with PopVar

```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
            min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
            remove.dups = TRUE, impute = "EM", map.plot = TRUE,
            models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
            "BL", "BRR"), nIter = 12000, burnIn = 3000,
            frac.train = 0.6, nCV.iter = 100,
            nFold = NULL, nFold.reps = 1,
                           ossing.table = NUL
                           25, tail
```

nIter and burnIn options are used when fitting Bayesian models

nFold : number of subsets (folds)
nFold.reps: number of times to repeat folding

frac.train: fraction of the training set used to train the model (the rest will be used as validation set)
nCV.iter: number of iterations (repetitions)

- **Model selection is done automatically if the user indicates more than one model in the "models" option.**
  - Up to 6 models can be tested.
    - BL = Bayesian LASSO ; BRR = Bayesian ridge regression
  - Accuracy is assessed by two cross-validation methods:
    - **Random sampling** (default) is implemented if nFold = NULL.
    - Fold sampling is implemented when nFold is set to a number.

- The best model is selected automatically.
- Predicted line phenotypes are then calculated automatically for all genotyped lines, even those not in the training set.
  - This is why genotyped lines that have no phenotype can be used as parents by PopVar.

Model training and selection

Predicting progeny phenotypes

Selecting crosses with SelectionTools

Selecting crosses with PopVar

Predicting progeny phenotypes

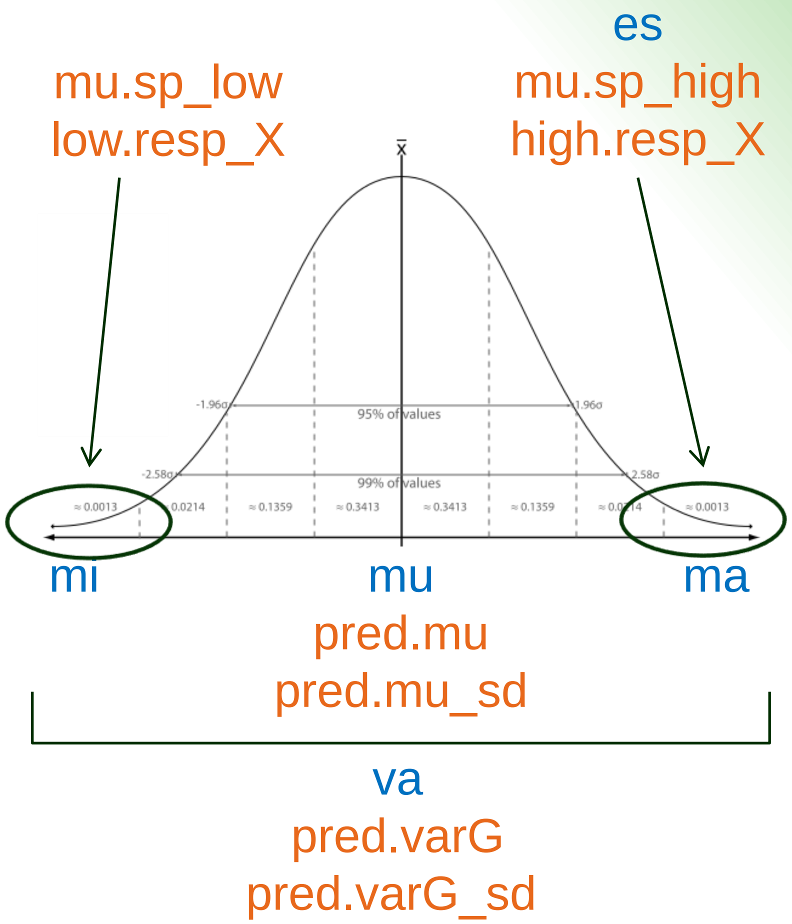
	SelectionTools	PopVar
Crosses evaluated	All TP lines (with geno+pheno)	All TP lines, all genotyped lines, list of parents, list of crosses
Specialized predictions generated		
- unphenotyped parents	No	Yes
- multiple traits	No	Yes
- correlations between traits	No	Yes

- **Because of its simulation approach, PopVar can calculate a larger set of phenotypes for each cross progeny.**
- **However, a simulation approach is very slow.**
  - It took more than a month to generate predictions for the SoyaGen TS on Manitou…
- **It is therefore suggested to use a 2-step strategy for cross selection :**
  - Step 1. Use SelectionTools to do a first scan of all possible crosses.
  - Step 2. Use PopVar to get a more in-depth evaluation of preselected, targeted, subsets of crosses.

Predicted progeny phenotypes

The fraction of the progeny to use as superior progeny is set by the options “alpha” in SelectionTools and “tail.p” in PopVar

	SelectionTools	PopVar
Parental genetic distances	**gd**	
Mid-parental values calculated from		**midPar.Pheno**
- observed phenotypes of the parents		**midPar.GEBV**
- predicted phenotypes of the parents		
Predicted progeny phenotypic values	**mi**	
- minimum value	**ma**	
- maximum value	**mu**	**pred.mu**
- mean		**pred.mu_sd**
- standard deviation of the mean	**va**	**pred.varG**
- variance		**pred.varG_sd**
- standard deviation of the variance		
Predicted mean of the expected superior progeny		**mu.sp_low**
- when favorable values are low values		**mu.sp_high**
- when favorable values are low values		
Predicted mean of the expected superior progeny		
for a secondary trait		**low.resp_X**
- when favorable values are low values	**es**	**high.resp_X**
- when favorable values are low values		**cor_w/_X**
Predicted correlation between primary and		
secondary traits		



Modified from:
https://commons.wikimedia.org/wiki/File:
The_Normal_Distribution.svg

Predicting progeny phenotypes

SelectionTools

```
gs.cross.eval.gd (dist = "rd")
gs.cross.eval.mi ()
gs.cross.eval.ma ()
gs.cross.eval.mu ()
gs.cross.eval.va (pop.type = "DH")
gs.cross.eval.es (alpha = 0.1)
```

PopVar

```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
            min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
            remove.dups = TRUE, impute = "EM", map.plot = TRUE,
            models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
            "BL", "BRR"), nIter = 12000, burnIn = 3000,
            frac.train = 0.6, nCV.iter = 100,
            nFold = NULL, nFold.reps = 1,
            parents = NULL, crossing.table = NULL,
            nInd = 200, nSim = 25, tail.p = 0.1)
```

Predicting progeny phenotypes with SelectionTools

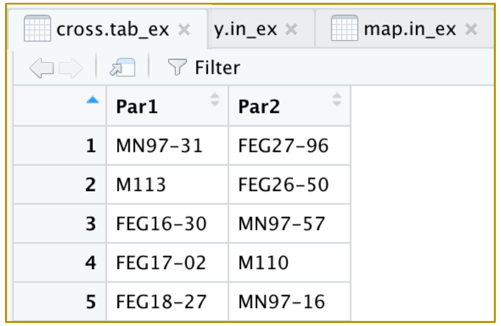
```
gs.cross.eval.gd (dist = "rd")    # calculates genetic distances of parents
                                   # dist = “euc”, “rd”, ”mrd”
gs.cross.eval.mi ()               # predicts minimum progeny values
gs.cross.eval.ma ()               # predicts maximum progeny values
gs.cross.eval.mu ()               # predicts progeny means
gs.cross.eval.va (pop.type = "DH")# predicts genetic variances
                                    # pop.type = "DH" or "SSD"
gs.cross.eval.es (alpha = 0.25)   # predicts expected superior progeny means
                                    # for the selected fraction alpha
```

- **SelectionTools automatically tests all possible combinations of lines from the training set.**
    - There is no way to test only a subset of crosses of interest.
- **All progeny phenotypes are calculated separately.**
    - It calculates the phenotype of a genotype predicted to carry all bad (mi) or good (ma) alleles.
    - Models for two population types (DH: double haploids, SSD: single seed descents) are available to predict variances.

Predicting progeny phenotypes with PopVar

```
pop.predict(G.in = filename, y.in = filename, map.in = filename,
 min.maf = 0.01, mkr.cutoff = 0.5, entry.cutoff = 0.5,
 remove.dups = TRUE, impute = "EM", map.plot = TRUE,
 models = c("rrBLUP", "BayesA", "BayesB", "BayesC",
 "BL", "BRR"), nIter = 12000, burnIn = 3000,
 frac.train = 0.6, nCV.iter = 100,
 nFold = NULL, nFold.reps = 1,
 parents = NULL, crossing.table = filename,
 nInd = 200, nSim = 25, tail.p = 0.1)
```

Format of the list of crosses



	Par1	Par2
1	MN97-31	FEG27-96
2	M113	FEG26-50
3	FEG16-30	MN97-57
4	FEG17-02	M110
5	FEG18-27	MN97-16

- **Target cross options**
  - parents : Testing all combination of a parental list
    - Parental list options
      - NULL (default) (= all Geno lines)
      - TP (training pop) (= Geno+Pheno)
      - User-defined list of parental lines
  - Crossing.table : Testing only a user-defined list of crosses

- **Progeny simulation**
  - Progeny simulations are performed using the R/qtl package
    - nInd: number of individual to simulate
    - nSim : number of simulation to run

Model training and selection

Predicting progeny phenotypes

Selecting crosses with SelectionTools

Selecting crosses with PopVar

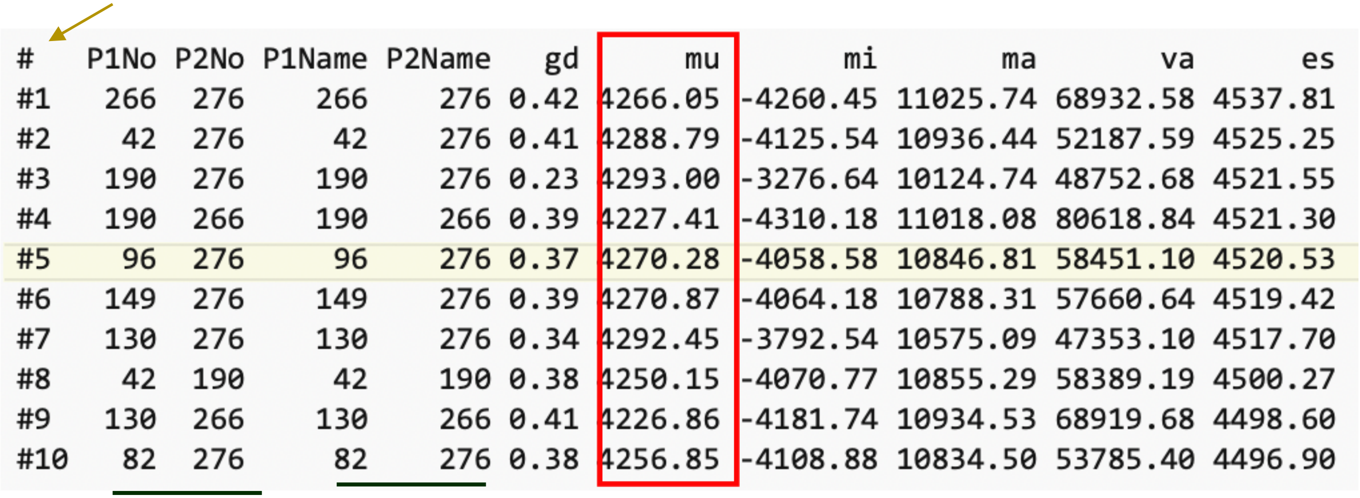
Running the various functions used to predict progeny phenotypic values

```
gs.cross.eval.gd (dist = "rd")    # calculates genetic distances of parents
                                   # dist = “euc”, “rd”, ”mrd”
gs.cross.eval.mi ()               # predicts minimum progeny values
gs.cross.eval.ma ()               # predicts maximum progeny values
gs.cross.eval.mu ()               # predicts progeny means (mid parental value)
gs.cross.eval.va (pop.type = "DH")# predicts genetic variances
                                    # pop.type = "DH" or "SSD"
gs.cross.eval.es (alpha = 0.25)   # predicts expected superior progeny value
                                    # for the selected fraction alpha
```

- **Warning. Marker effects must have been calculated before using those functions.**

Visualizing the predictions and selecting the best 10 crosses

Use sortby = "index”
to sort by #



```
#    P1No P2No P1Name P2Name   gd      mu       mi       ma       va      es
#1    266  276    266    276 0.42 4266.05 -4260.45 11025.74 68932.58 4537.81
#2     42  276     42    276 0.41 4288.79 -4125.54 10936.44 52187.59 4525.25
#3    190  276    190    276 0.23 4293.00 -3276.64 10124.74 48752.68 4521.55
#4    190  266    190    266 0.39 4227.41 -4310.18 11018.08 80618.84 4521.30
#5     96  276     96    276 0.37 4270.28 -4058.58 10846.81 58451.10 4520.53
#6    149  276    149    276 0.39 4270.87 -4064.18 10788.31 57660.64 4519.42
#7    130  276    130    276 0.34 4292.45 -3792.54 10575.09 47353.10 4517.70
#8     42  190     42    190 0.38 4250.15 -4070.77 10855.29 58389.19 4500.27
#9    130  266    130    266 0.41 4226.86 -4181.74 10934.53 68919.68 4498.60
#10    82  276     82    276 0.38 4256.85 -4108.88 10834.50 53785.40 4496.90
```

Parents order in the input file

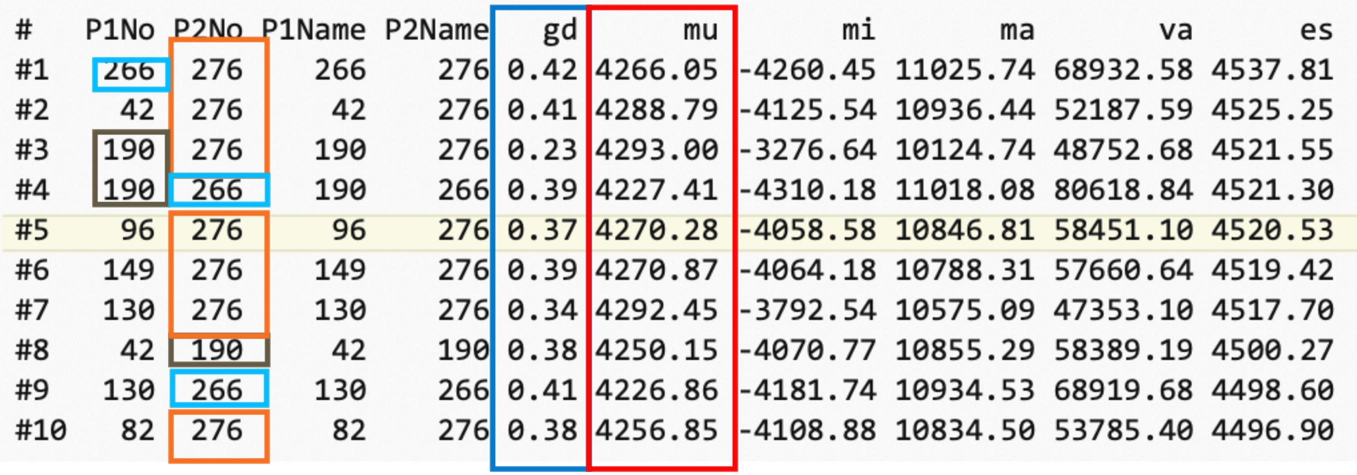
Parent names

In this example, the line ”names” were increasing numbers.

gs.cross.info (bestn = 10,
sortby = ”mu",
data.set = "default")

- This function will sort crosses according to the predicted phenotypes specified by the “sortby” option and will automatically create a subset of the size specified by the “bestn” option.

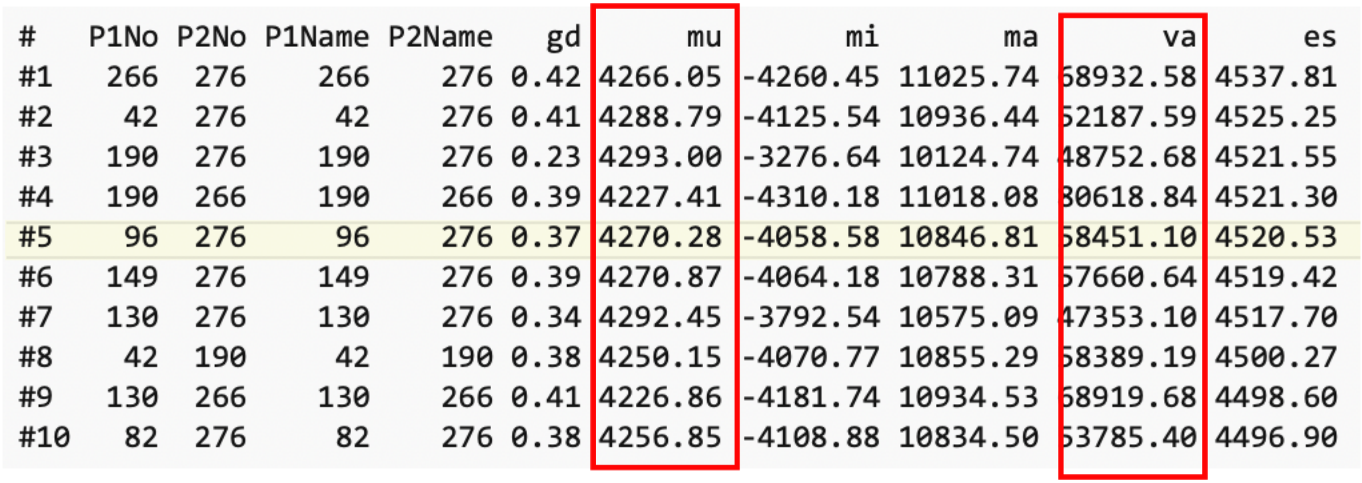
Visualizing the predictions and selecting the best 10 crosses



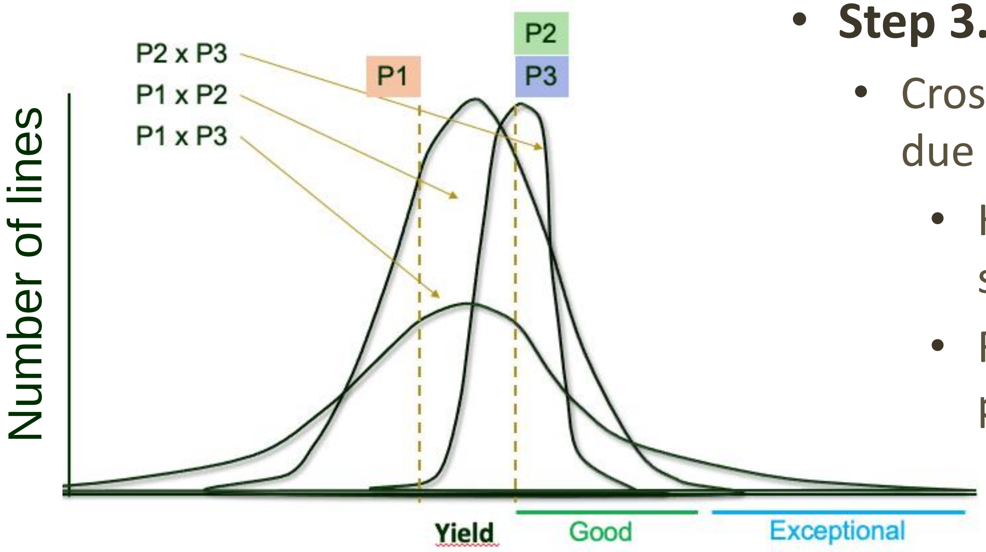
```
#    P1No P2No P1Name P2Name   gd      mu       mi       ma       va      es
#1    266  276    266    276 0.42 4266.05 -4260.45 11025.74 68932.58 4537.81
#2     42  276     42    276 0.41 4288.79 -4125.54 10936.44 52187.59 4525.25
#3    190  276    190    276 0.23 4293.00 -3276.64 10124.74 48752.68 4521.55
#4    190  266    190    266 0.39 4227.41 -4310.18 11018.08 80618.84 4521.30
#5     96  276     96    276 0.37 4270.28 -4058.58 10846.81 58451.10 4520.53
#6    149  276    149    276 0.39 4270.87 -4064.18 10788.31 57660.64 4519.42
#7    130  276    130    276 0.34 4292.45 -3792.54 10575.09 47353.10 4517.70
#8     42  190     42    190 0.38 4250.15 -4070.77 10855.29 58389.19 4500.27
#9    130  266    130    266 0.41 4226.86 -4181.74 10934.53 68919.68 4498.60
#10    82  276     82    276 0.38 4256.85 -4108.88 10834.50 53785.40 4496.90
```

- **Step 1. Identify the best crosses based on progeny means (mu).**
- **Step 2. Check the genetic distance (gd) between the parents of these crosses and avoid crosses with very small genetic distances.**
  - Warning. When crosses are selected by highest progeny mean, a small number of lines are found to be used repeatedly as parents of the best crosses (= lines with the highest trait values).
    - Care should be taken to avoid a reduction in genetic diversity.

Visualizing the predictions and selecting the best 10 crosses



```
#    P1No P2No P1Name P2Name   gd      mu       mi       ma       va      es
#1    266  276    266    276 0.42 4266.05 -4260.45 11025.74 68932.58 4537.81
#2     42  276     42    276 0.41 4288.79 -4125.54 10936.44 52187.59 4525.25
#3    190  276    190    276 0.23 4293.00 -3276.64 10124.74 48752.68 4521.55
#4    190  266    190    266 0.39 4227.41 -4310.18 11018.08 80618.84 4521.30
#5     96  276     96    276 0.37 4270.28 -4058.58 10846.81 58451.10 4520.53
#6    149  276    149    276 0.39 4270.87 -4064.18 10788.31 57660.64 4519.42
#7    130  276    130    276 0.34 4292.45 -3792.54 10575.09 47353.10 4517.70
#8     42  190     42    190 0.38 4250.15 -4070.77 10855.29 58389.19 4500.27
#9    130  266    130    266 0.41 4226.86 -4181.74 10934.53 68919.68 4498.60
#10    82  276     82    276 0.38 4256.85 -4108.88 10834.50 53785.40 4496.90
```



- **Step 3.Check the predicted variance (va) of the best crosses.**
  - Crosses with high progeny variance may contain exceptional lines due to transgressive segregation.
    - However, screening a larger number of progeny than a standard trial size may be needed to find them.
    - Furthermore, the variance is the hardest progeny trait to predict and its accuracy would need to be validated.

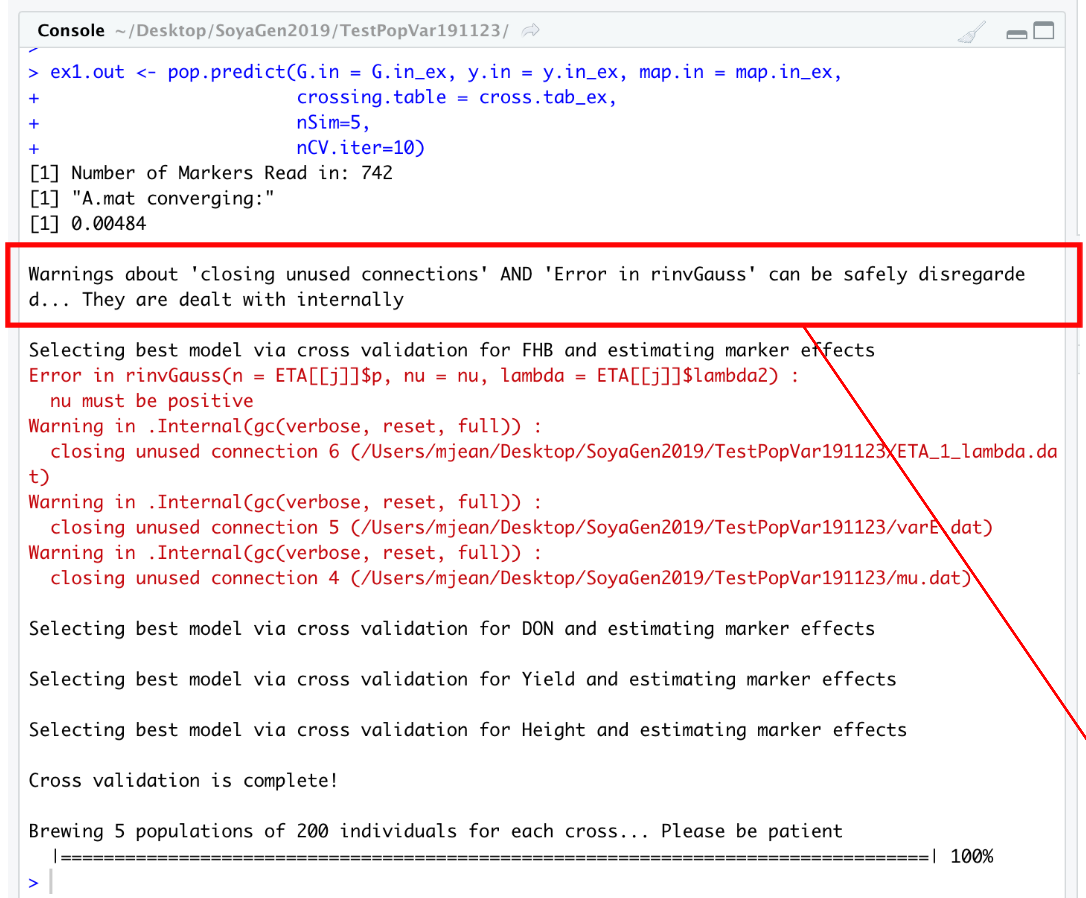
Model training and selection

Predicting progeny phenotypes

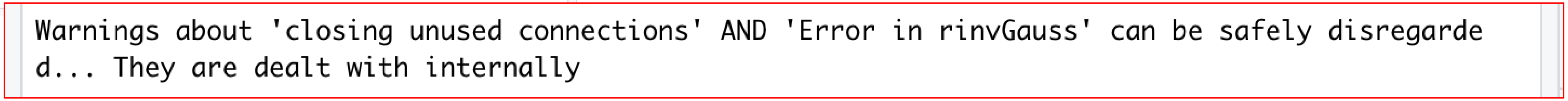
Selecting crosses with SelectionTools

Selecting crosses with PopVar

Running the pop.predict function



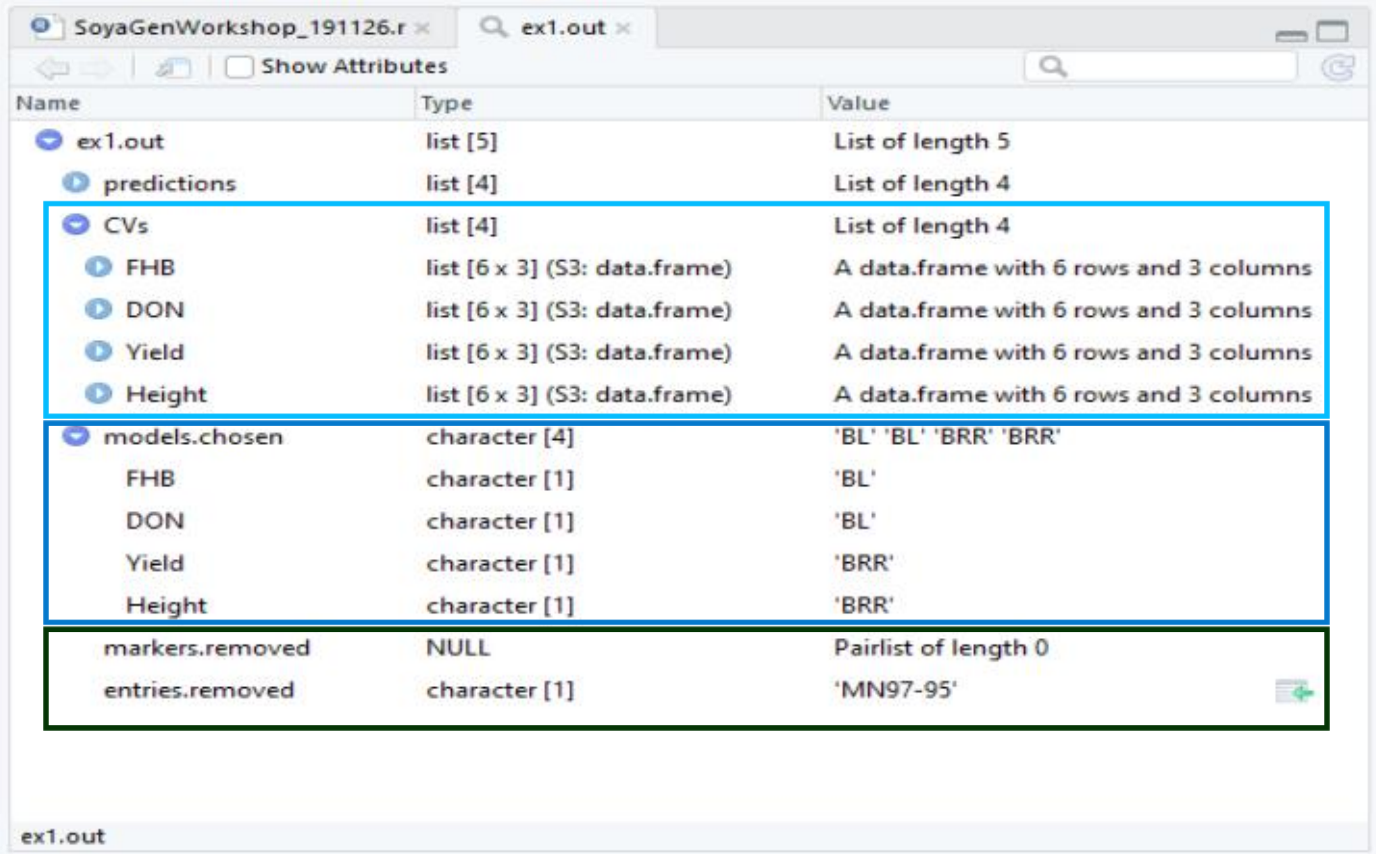
```
> ex1.out <- pop.predict(G.in = G.in_ex, y.in = y.in_ex, map.in = map.in_ex,
+                        crossing.table = cross.tab_ex,
+                        nSim=5,
+                        nCV.iter=10)
[1] Number of Markers Read in: 742
[1] "A.mat converging:"
[1] 0.00484

Warnings about 'closing unused connections' AND 'Error in rinvGauss' can be safely disregarde
d... They are dealt with internally

Selecting best model via cross validation for FHB and estimating marker effects
Error in rinvGauss(n = ETA[[j]]$p, nu = nu, lambda = ETA[[j]]$lambda2) :
  nu must be positive
Warning in .Internal(gc(verbose, reset, full)) :
  closing unused connection 6 (/Users/mjean/Desktop/SoyaGen2019/TestPopVar191123/ETA_1_lambda.da
t)
Warning in .Internal(gc(verbose, reset, full)) :
  closing unused connection 5 (/Users/mjean/Desktop/SoyaGen2019/TestPopVar191123/varE.dat)
Warning in .Internal(gc(verbose, reset, full)) :
  closing unused connection 4 (/Users/mjean/Desktop/SoyaGen2019/TestPopVar191123/mu.dat)

Selecting best model via cross validation for DON and estimating marker effects

Selecting best model via cross validation for Yield and estimating marker effects

Selecting best model via cross validation for Height and estimating marker effects

Cross validation is complete!

Brewing 5 populations of 200 individuals for each cross... Please be patient
  |==================================================================================| 100%
>
```



```
Warnings about 'closing unused connections' AND 'Error in rinvGauss' can be safely disregarde
d... They are dealt with internally
```

- **Warning 1. Error messages are often output by PopVar**
  - According to the authors, they can be be safely disregarded.
  - However, carefully read them to detect possible “true” ones…
- **Warning 2. Runtime can grow significantly when using large values for some options such as nCV.iter and nSim.**
  - Smaller values can be used for tests but larger values such as the default values should always be used in real analyses.

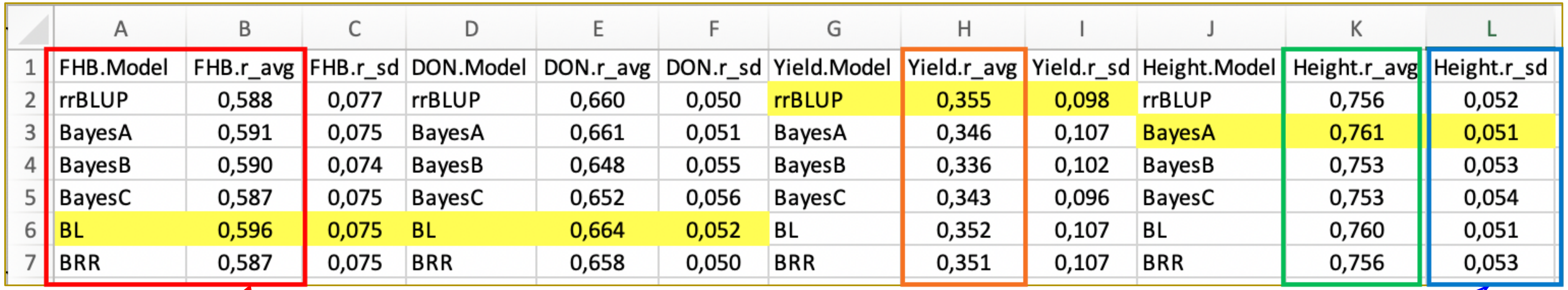
Visualizing the results of the various steps that were automatically carried out by PopVar



```
Name                Type                              Value
ex1.out             list [5]                          List of length 5
  predictions       list [4]                          List of length 4
  CVs               list [4]                          List of length 4
    FHB             list [6 x 3] (S3: data.frame)     A data.frame with 6 rows and 3 columns
    DON             list [6 x 3] (S3: data.frame)     A data.frame with 6 rows and 3 columns
    Yield           list [6 x 3] (S3: data.frame)     A data.frame with 6 rows and 3 columns
    Height          list [6 x 3] (S3: data.frame)     A data.frame with 6 rows and 3 columns
  models.chosen     character [4]                     'BL' 'BL' 'BRR' 'BRR'
    FHB             character [1]                     'BL'
    DON             character [1]                     'BL'
    Yield           character [1]                     'BRR'
    Height          character [1]                     'BRR'
  markers.removed   NULL                              Pairlist of length 0
  entries.removed   character [1]                     'MN97-95'
ex1.out
```

- Preprocessing results
  - **markers.removed :** List of markers removed during filtering for MAF and missing data.
  - **entries.removed :** List of entries removed during filtering for missing data and duplicate entries.
- Model selection results
  - **models.chosen :** List of the statistical model chosen for each trait.
  - **CVs** : CV results for each trait/model combination specified.
    - Can be exported to disk in text format to be imported in Excel

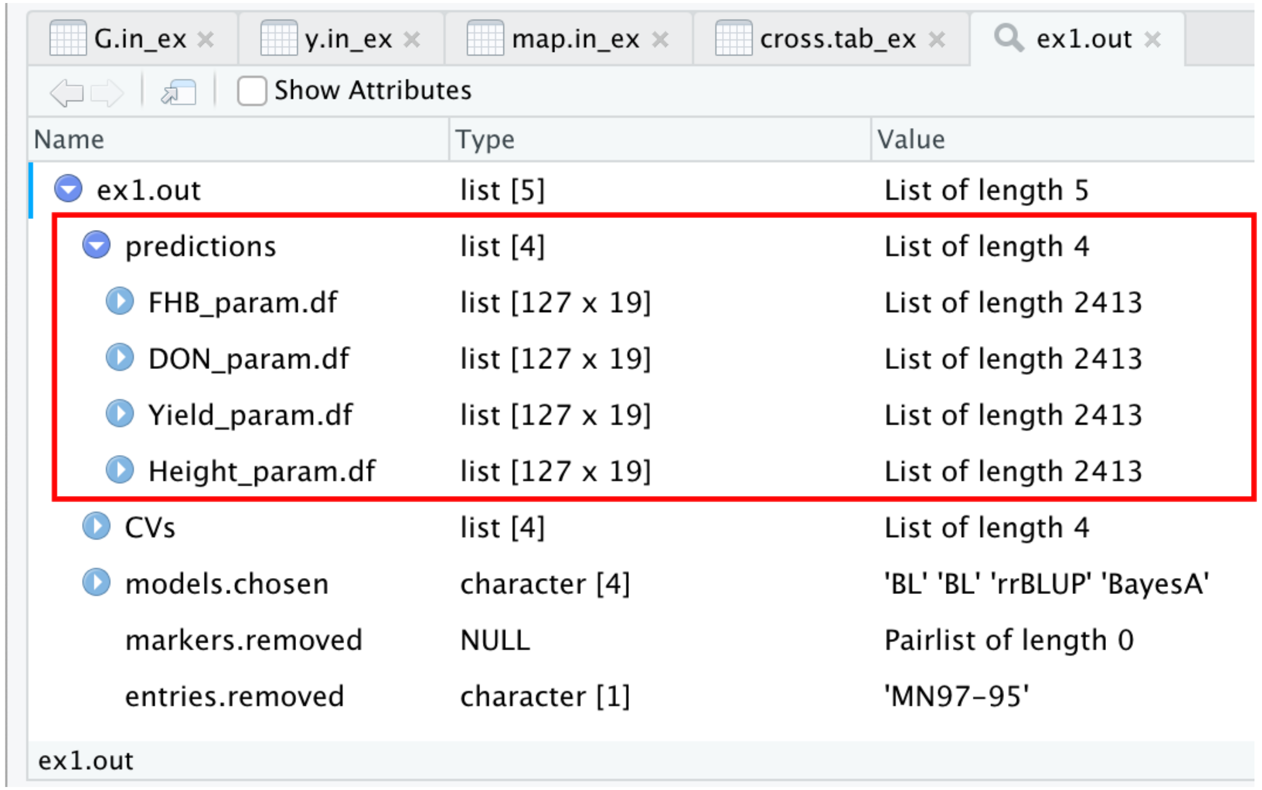
Visualizing the model selection results



	A	B	C	D	E	F	G	H	I	J	K	L
1	FHB.Model	FHB.r_avg	FHB.r_sd	DON.Model	DON.r_avg	DON.r_sd	Yield.Model	Yield.r_avg	Yield.r_sd	Height.Model	Height.r_avg	Height.r_sd
2	rrBLUP	0,588	0,077	rrBLUP	0,660	0,050	rrBLUP	0,355	0,098	rrBLUP	0,756	0,052
3	BayesA	0,591	0,075	BayesA	0,661	0,051	BayesA	0,346	0,107	BayesA	0,761	0,051
4	BayesB	0,590	0,074	BayesB	0,648	0,055	BayesB	0,336	0,102	BayesB	0,753	0,053
5	BayesC	0,587	0,075	BayesC	0,652	0,056	BayesC	0,343	0,096	BayesC	0,753	0,054
6	BL	0,596	0,075	BL	0,664	0,052	BL	0,352	0,107	BL	0,760	0,051
7	BRR	0,587	0,075	BRR	0,658	0,050	BRR	0,351	0,107	BRR	0,756	0,053

- **PopVar automatically tests 6 models that should cover most underlying trait architectures and select the one that achieve the highest accuracy.**
  - Major differences in accuracy are observed between traits.
    - These are reproducible differences (low variance).
  - Very small differences in accuracy are observed between models.
    - In theory, the best model for a given trait is related to it genetic architecture.

Visualizing the prediction results



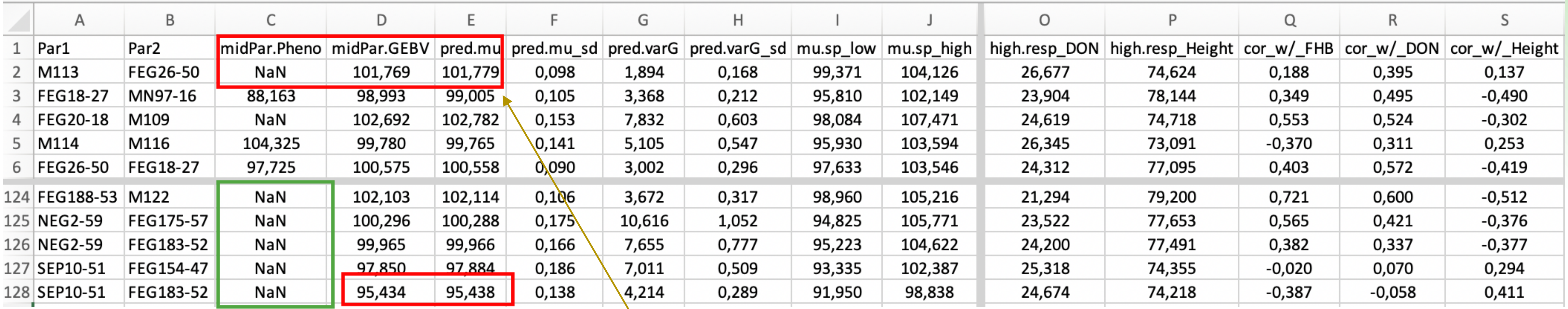
Name	Type	Value
ex1.out	list [5]	List of length 5
predictions	list [4]	List of length 4
FHB_param.df	list [127 x 19]	List of length 2413
DON_param.df	list [127 x 19]	List of length 2413
Yield_param.df	list [127 x 19]	List of length 2413
Height_param.df	list [127 x 19]	List of length 2413
CVs	list [4]	List of length 4
models.chosen	character [4]	'BL' 'BL' 'rrBLUP' 'BayesA'
markers.removed	NULL	Pairlist of length 0
entries.removed	character [1]	'MN97-95'

ex1.out

- **Prediction results**
  - **predictions** : contains variables storing predictions for each trait for each parental combination specified.
    - They can be accessed with R commands by using the $ symbol and can be exported in text format to be visualized in Excel.
- **Warning:**
  - There is no function in PopVar to sort crosses according to one of the progeny phenotypes and create subsets of the best crosses.
    - To do so, you need to use standard R commands or export the results and do so in Excel or another software.

Selecting crosses based on mid-parental values

Example: Cross predictions for yield



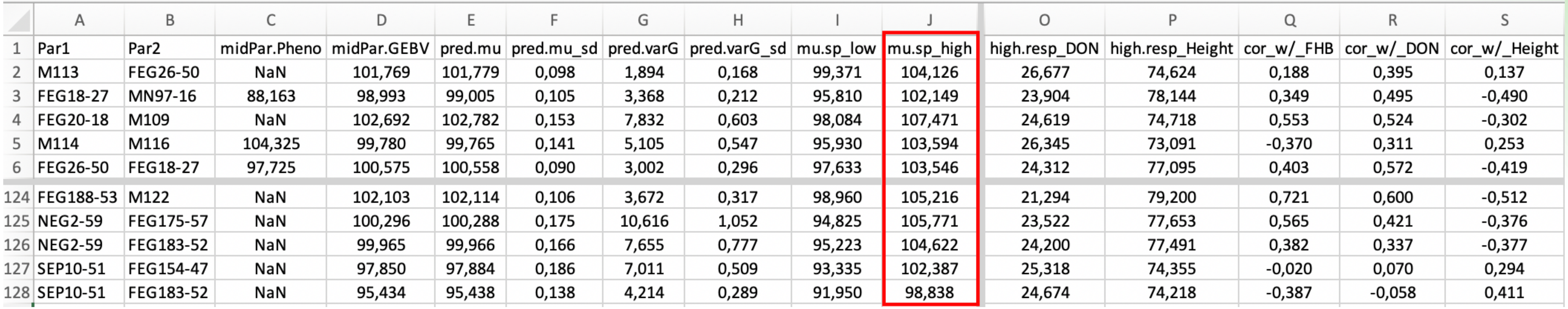
	A	B	C	D	E	F	G	H	I	J	O	P	Q	R	S
1	Par1	Par2	midPar.Pheno	midPar.GEBV	pred.mu	pred.mu_sd	pred.varG	pred.varG_sd	mu.sp_low	mu.sp_high	high.resp_DON	high.resp_Height	cor_w/_FHB	cor_w/_DON	cor_w/_Height
2	M113	FEG26-50	NaN	101,769	101,779	0,098	1,894	0,168	99,371	104,126	26,677	74,624	0,188	0,395	0,137
3	FEG18-27	MN97-16	88,163	98,993	99,005	0,105	3,368	0,212	95,810	102,149	23,904	78,144	0,349	0,495	-0,490
4	FEG20-18	M109	NaN	102,692	102,782	0,153	7,832	0,603	98,084	107,471	24,619	74,718	0,553	0,524	-0,302
5	M114	M116	104,325	99,780	99,765	0,141	5,105	0,547	95,930	103,594	26,345	73,091	-0,370	0,311	0,253
6	FEG26-50	FEG18-27	97,725	100,575	100,558	0,090	3,002	0,296	97,633	103,546	24,312	77,095	0,403	0,572	-0,419
124	FEG188-53	M122	NaN	102,103	102,114	0,106	3,672	0,317	98,960	105,216	21,294	79,200	0,721	0,600	-0,512
125	NEG2-59	FEG175-57	NaN	100,296	100,288	0,175	10,616	1,052	94,825	105,771	23,522	77,653	0,565	0,421	-0,376
126	NEG2-59	FEG183-52	NaN	99,965	99,966	0,166	7,655	0,777	95,223	104,622	24,200	77,491	0,382	0,337	-0,377
127	SEP10-51	FEG154-47	NaN	97,850	97,884	0,186	7,011	0,509	93,335	102,387	25,318	74,355	-0,020	0,070	0,294
128	SEP10-51	FEG183-52	NaN	95,434	95,438	0,138	4,214	0,289	91,950	98,838	24,674	74,218	-0,387	-0,058	0,411

No mid-parental values = one or both parents without phenotype

- **midPar.Pheno, midPar.GEBV and pred.mu are basically identical.**
  - When phenotypes are available for both parents, midPar.Pheno could be easily calculated by breeders and used in conventional selection.
  - When phenotypes for one or both parents are missing, PopVar could be used to predict this statistics.

Selecting crosses based on superior progeny values

Example: Cross predictions for yield



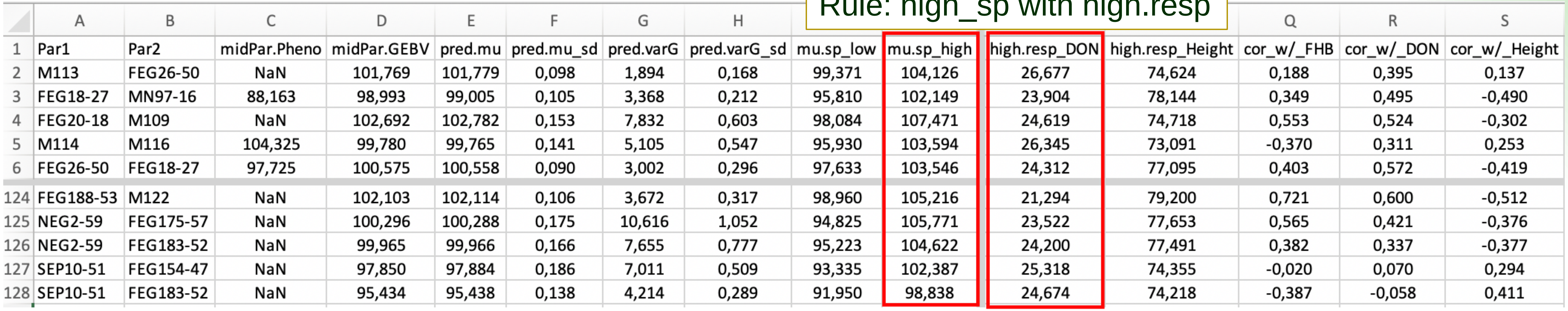
	A	B	C	D	E	F	G	H	I	J	O	P	Q	R	S
1	Par1	Par2	midPar.Pheno	midPar.GEBV	pred.mu	pred.mu_sd	pred.varG	pred.varG_sd	mu.sp_low	mu.sp_high	high.resp_DON	high.resp_Height	cor_w/_FHB	cor_w/_DON	cor_w/_Height
2	M113	FEG26-50	NaN	101,769	101,779	0,098	1,894	0,168	99,371	104,126	26,677	74,624	0,188	0,395	0,137
3	FEG18-27	MN97-16	88,163	98,993	99,005	0,105	3,368	0,212	95,810	102,149	23,904	78,144	0,349	0,495	-0,490
4	FEG20-18	M109	NaN	102,692	102,782	0,153	7,832	0,603	98,084	107,471	24,619	74,718	0,553	0,524	-0,302
5	M114	M116	104,325	99,780	99,765	0,141	5,105	0,547	95,930	103,594	26,345	73,091	-0,370	0,311	0,253
6	FEG26-50	FEG18-27	97,725	100,575	100,558	0,090	3,002	0,296	97,633	103,546	24,312	77,095	0,403	0,572	-0,419
124	FEG188-53	M122	NaN	102,103	102,114	0,106	3,672	0,317	98,960	105,216	21,294	79,200	0,721	0,600	-0,512
125	NEG2-59	FEG175-57	NaN	100,296	100,288	0,175	10,616	1,052	94,825	105,771	23,522	77,653	0,565	0,421	-0,376
126	NEG2-59	FEG183-52	NaN	99,965	99,966	0,166	7,655	0,777	95,223	104,622	24,200	77,491	0,382	0,337	-0,377
127	SEP10-51	FEG154-47	NaN	97,850	97,884	0,186	7,011	0,509	93,335	102,387	25,318	74,355	-0,020	0,070	0,294
128	SEP10-51	FEG183-52	NaN	95,434	95,438	0,138	4,214	0,289	91,950	98,838	24,674	74,218	-0,387	-0,058	0,411

- **The superior progeny mean correspond to the mean value of the subset of lines that will persist through selection.**
- **When selecting for more than one traits :**
  - **Step 1. Crosses should be ordered by the mu_sp value of the primary target to identify the best crosses.**
    - mu_sp_**high** should be used when selecting for the highest trait value like yield
    - mu_sp_**low** should be used when selecting for the lowest trait value like for maturity.

Selecting crosses based on the value of correlated traits in the superior progeny

Example: Cross predictions for yield

Rule: high_sp with high.resp



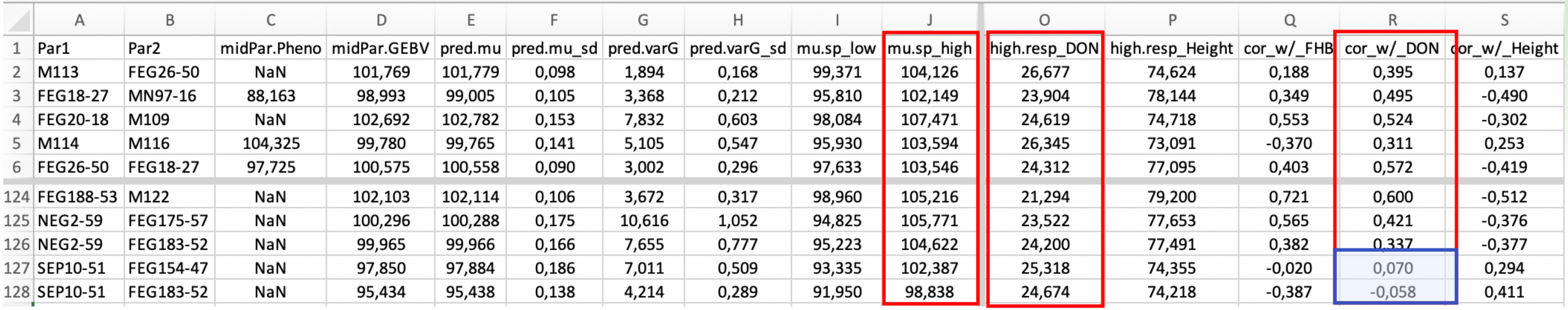
	Par1	Par2	midPar.Pheno	midPar.GEBV	pred.mu	pred.mu_sd	pred.varG	pred.varG_sd	mu.sp_low	mu.sp_high	high.resp_DON	high.resp_Height	cor_w/_FHB	cor_w/_DON	cor_w/_Height
2	M113	FEG26-50	NaN	101,769	101,779	0,098	1,894	0,168	99,371	104,126	26,677	74,624	0,188	0,395	0,137
3	FEG18-27	MN97-16	88,163	98,993	99,005	0,105	3,368	0,212	95,810	102,149	23,904	78,144	0,349	0,495	-0,490
4	FEG20-18	M109	NaN	102,692	102,782	0,153	7,832	0,603	98,084	107,471	24,619	74,718	0,553	0,524	-0,302
5	M114	M116	104,325	99,780	99,765	0,141	5,105	0,547	95,930	103,594	26,345	73,091	-0,370	0,311	0,253
6	FEG26-50	FEG18-27	97,725	100,575	100,558	0,090	3,002	0,296	97,633	103,546	24,312	77,095	0,403	0,572	-0,419
124	FEG188-53	M122	NaN	102,103	102,114	0,106	3,672	0,317	98,960	105,216	21,294	79,200	0,721	0,600	-0,512
125	NEG2-59	FEG175-57	NaN	100,296	100,288	0,175	10,616	1,052	94,825	105,771	23,522	77,653	0,565	0,421	-0,376
126	NEG2-59	FEG183-52	NaN	99,965	99,966	0,166	7,655	0,777	95,223	104,622	24,200	77,491	0,382	0,337	-0,377
127	SEP10-51	FEG154-47	NaN	97,850	97,884	0,186	7,011	0,509	93,335	102,387	25,318	74,355	-0,020	0,070	0,294
128	SEP10-51	FEG183-52	NaN	95,434	95,438	0,138	4,214	0,289	91,950	98,838	24,674	74,218	-0,387	-0,058	0,411

- **Step 2. The best crosses should then be ordered by the mean value of the correlated secondary target in the superior progeny to identify crosses with progeny improved for both traits.**
  - Warning. The correlated trait value of the secondary target should be examined in the same slice of the progeny as the main target
    - For example, when **high** yield is the main target with maturity being a secondary target, one would look at the maturity value from the **high**.resp column.
    - By contrast, if **early** maturity is the main target with yield being a secondary target, one would look at the yield value from the **low**.resp column.

This step can be repeated as often as needed if there are more than one secondary targets

Selecting crosses based on the value of correlated traits in the superior progeny

Example: Cross predictions for yield



	A	B	C	D	E	F	G	H	I	J	O	P	Q	R	S
1	Par1	Par2	midPar.Pheno	midPar.GEBV	pred.mu	pred.mu_sd	pred.varG	pred.varG_sd	mu.sp_low	mu.sp_high	high.resp_DON	high.resp_Height	cor_w/_FHB	cor_w/_DON	cor_w/_Height
2	M113	FEG26-50	NaN	101,769	101,779	0,098	1,894	0,168	99,371	104,126	26,677	74,624	0,188	0,395	0,137
3	FEG18-27	MN97-16	88,163	98,993	99,005	0,105	3,368	0,212	95,810	102,149	23,904	78,144	0,349	0,495	-0,490
4	FEG20-18	M109	NaN	102,692	102,782	0,153	7,832	0,603	98,084	107,471	24,619	74,718	0,553	0,524	-0,302
5	M114	M116	104,325	99,780	99,765	0,141	5,105	0,547	95,930	103,594	26,345	73,091	-0,370	0,311	0,253
6	FEG26-50	FEG18-27	97,725	100,575	100,558	0,090	3,002	0,296	97,633	103,546	24,312	77,095	0,403	0,572	-0,419
124	FEG188-53	M122	NaN	102,103	102,114	0,106	3,672	0,317	98,960	105,216	21,294	79,200	0,721	0,600	-0,512
125	NEG2-59	FEG175-57	NaN	100,296	100,288	0,175	10,616	1,052	94,825	105,771	23,522	77,653	0,565	0,421	-0,376
126	NEG2-59	FEG183-52	NaN	99,965	99,966	0,166	7,655	0,777	95,223	104,622	24,200	77,491	0,382	0,337	-0,377
127	SEP10-51	FEG154-47	NaN	97,850	97,884	0,186	7,011	0,509	93,335	102,387	25,318	74,355	-0,020	0,070	0,294
128	SEP10-51	FEG183-52	NaN	95,434	95,438	0,138	4,214	0,289	91,950	98,838	24,674	74,218	-0,387	-0,058	0,411

- **Step 3. Check the predicted correlation between the main and secondary target to identify crosses where the correlation is weaker than usual.**
  - It might be easier to break the unfavorable correlation and find individual combining improvement for both traits in these crosses.

Thank you

GenomeCanada

GenomeQuébec

Western Grains Research Foundation

UNIVERSITÉ LAVAL

CÉROM Centre de recherche sur les grains inc.

Prograin® A world of Soybean

UNIVERSITY OF SASKATCHEWAN

UNIVERSITY of GUELPH

Agriculture and Agri-Food Canada

La Coop fédérée

SEVITA GENETICS

syngenta

Canadian Field Crop Research Alliance (CFCRA)

SASKATCHEWAN pulse Growers

MANITOBA Pulse Soybean GROWERS

GRAIN FARMERS OF ONTARIO

PRODUCTEURS DE GRAINS DU QUÉBEC

INSTITUT DE BIOLOGIE INTÉGRATIVE ET DES SYSTÈMES IBIS

Embrapa

University of Missouri

Questions…