Mapping and prediction with `rTASSEL`

Brandon Monier

Cornell University

Overview

- Background info
  - TASSEL 5
  - rTASSEL
- Working with rTASSEL
  - Setup and preamble
  - Data structure
  - Association and relatedness functions
  - Genomic prediction

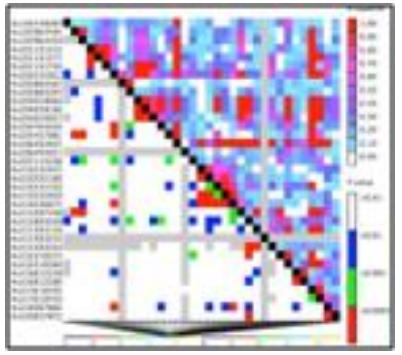
(I) Background

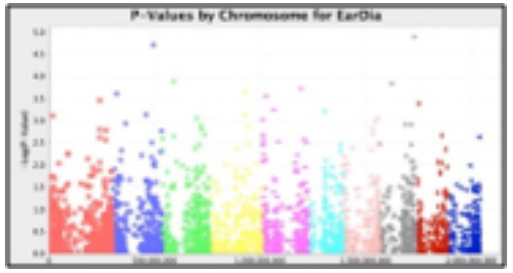
What is TASSEL?

- **T**rait **A**nalysis by a**SS**ociation, **E**volution, and **L**inkage
- TASSEL's primary purpose is to **serve the Buckler Lab's research needs**
- Not originally intended as a community resource
- A collection of useful tools rather than a unified framework

What TASSEL can and can't do

- What TASSEL **can** do:
  - Manipulate genotypes
  - Basic population genetics (MDS, PCA, phylogeny)
  - Association analysis (GLM, MLM, fast association, etc.)
  - Imputation (FILLIN or FSFHap)
- What TASSEL **cannot** do:
  - Other imputation algorithms
  - Advanced population genetics
  - Normal linkage mapping
  - And much more!





Authors of TASSEL

Ed Buckler

Terry Casstevens

Peter Bradbury

Lynn Johnson

Zack Miller

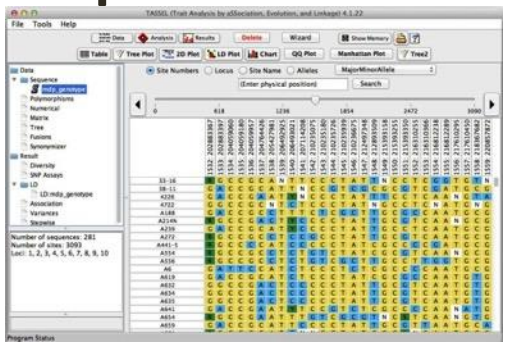
Kelly Swarts

Fei Lu

Jeff Glaubitz

Ways to work with TASSEL

Graphical Interface



Command Line

```
> run_pipeline.pl $TASSEL -fork1 -h
allzea_gbs_2_7.t5.h5 -filterAlign -
filterAlignMinFreq -includeTaxaInFile
my_target_taxa.txt -export
mytaxa_gbs_2_7_filtered.hmp.h5 -
runfork1
```

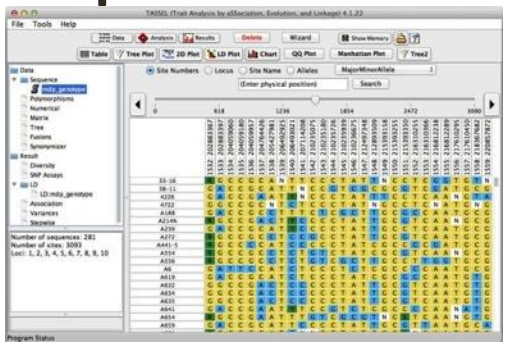
API (Java)

```
Public class FilterMyStuff{

    GenotypeTable myGenos =
        ImportUtils.ReadFromHapmap(
        “test_gbs.hmp.txt.gz”)
    FilterGenotypeTable filtered =
        FilterGenotypeTable.getInstanc
        e(myGenos, myTaxaList);
    …
```

Ways to work with TASSEL

Graphical Interface



API (Java)

```
lic class FilterMyStuff{

GenotypeTable myGenos =
    ImportUtils.ReadFromHapmap(
    “test_gbs.hmp.txt.gz”)
FilterGenotypeTable filtered =
    FilterGenotypeTable.getInstanc
    e(myGenos, myTaxaList);
…
```

RTASSEL

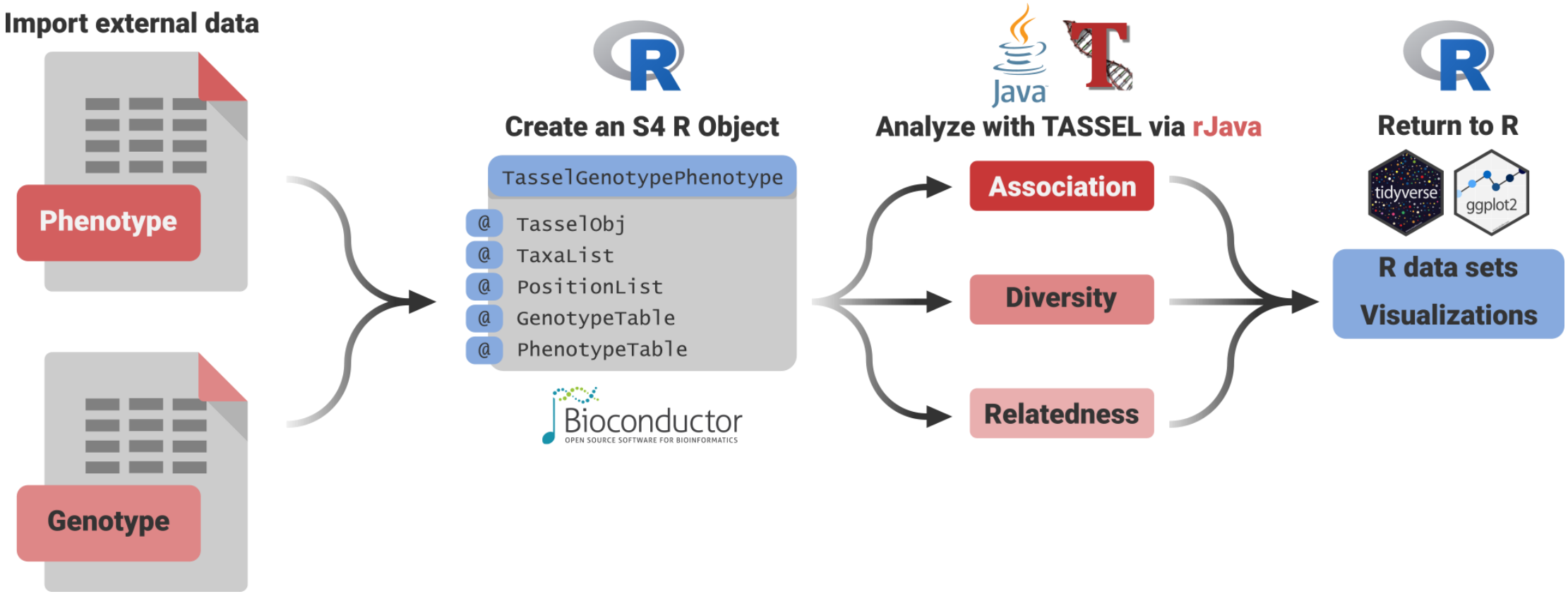
Intuitiveness

Power

What is rTASSEL?

- Provides an R-based front-end for **highly used** TASSEL methods and analytical tools
- Provides a unified workflow between R and TASSEL
  - Analytical power of **TASSEL 5**
  - Data handling and visualization power of **R**
  - **Increase intuitiveness while retaining more power**

rTASSEL generalized workflow



rTASSEL/TASSEL Resources

- TASSEL wiki:
  - https://bitbucket.org/tasseladmin/tassel-5-source/wiki/Home
- TASSEL video tutorials:
  - https://www.youtube.com/channel/UCS1SdXlyMI1OsSf5yA_oFqw
- rTASSEL source code:
  - https://bitbucket.org/bucklerlab/rtassel/src/master/
- rTASSEL vignette:
  - https://bitbucket.org/bucklerlab/rtassel/wiki/Home

rTASSEL / TASSEL forums

- TASSEL / rTASSEL user group:
  - https://groups.google.com/forum/#!forum/tassel
- rTASSEL issues and feature requests:
  - https://bitbucket.org/bucklerlab/rtassel/issues?status=new&status=open

(II) Setup and Preamble

Installation

- Source code pulled and compiled from BitBucket
- If installing yourself, you will need the following:
  1. R v3.5+ / RStudio
  2. Java JDK 11
  3. rJava (R package)
  4. devtools (R package)

Installation

- Additional packages that we will use
  1. `devtools`
  2. `dplyr`
  3. `foreach`
  4. `magrittr`
  5. `readr`
  6. `SummarizedExperiment`
  7. `tibble`

Installation

- Installing and compiling source code

```
# Extract source code from BitBucket
if (!require("devtools")) install.packages("devtools")
devtools::install_bitbucket(
  repo = "bucklerlab/rtassel",
  ref  = "master"
)
```

Obtain data directory

- On Sakai
- Directory layout:

```
cshl_rtassel_2019
|-- doc
    |-- cshl_rtassel_2019.pdf
|-- data
    |-- AGPv4_NAM_subset.recode.vcf.gz
    |-- phenotypic_data.csv
|-- R
    |-- cshl_training_20191015_part_1.R
    |-- cshl_training_20191015_part_2.R
|-- cshl_rtassel_2019.Rproj
```

Loading rTASSEL - memory

- **NOTE:** Before loading rTASSEL, you will need to set Java memory!

```
# Set Java memory parameter
options(java.parameters = c("-Xmx<memory>"))
```


Change `<memory>` to number and space unit (e.g. `10g`)

Loading rTASSEL – logging files

- Exports logging information from TASSEL to external file
- Prevents R console from being overloaded with TASSEL output

```
# Start a logging file
rTASSEL::startLogger(fullPath = NULL, fileName = NULL)
```

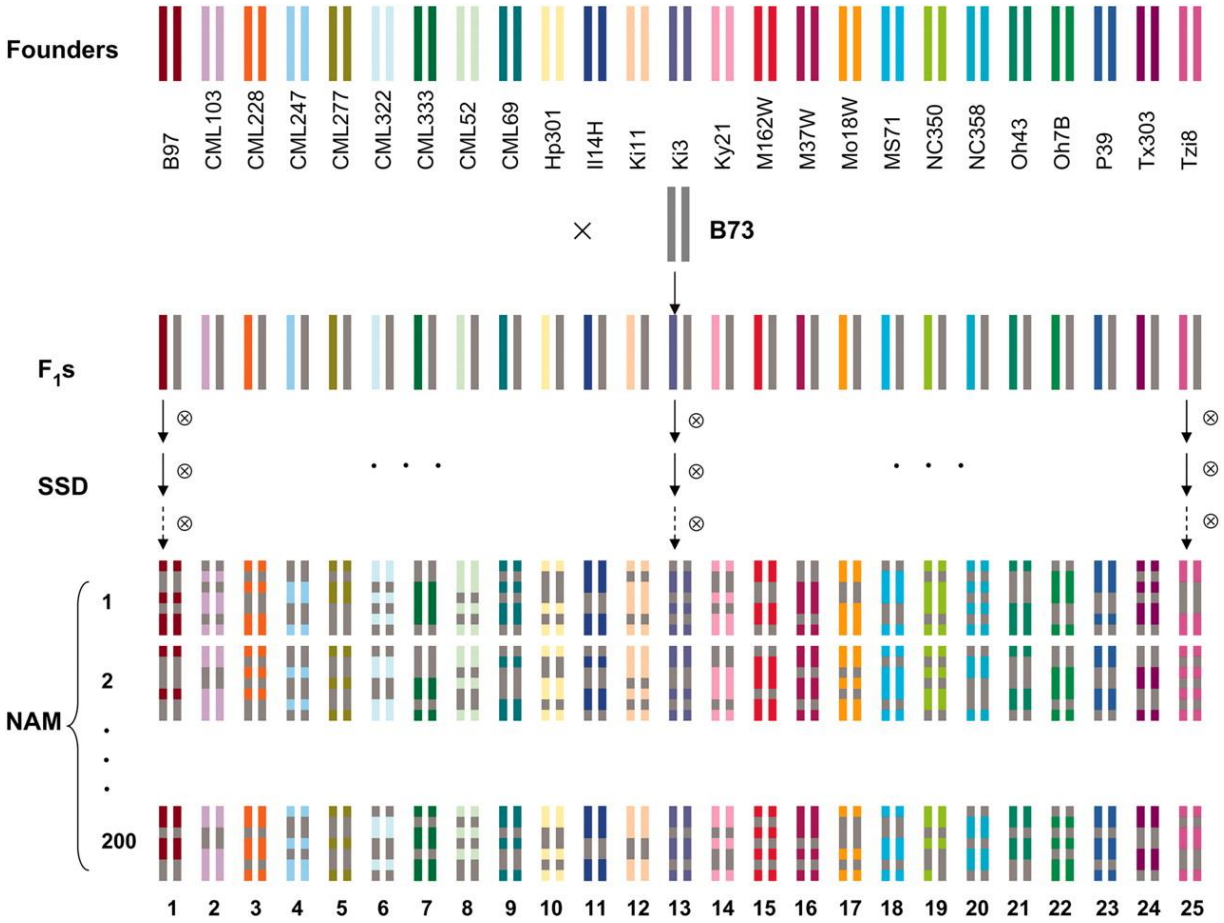
- This will export file (`rTASSEL_log`) to current working directory if both parameters are set to **NULL**

(III) Data Structure

rTASSEL – data types

- rTASSEL can load several genotype file formats
  - HapMap (`.hmp.txt`)
  - VCF (**V**ariant **C**all **F**ormat; `.vcf`)
  - HDF5 (**H**ierarchical **D**ata **F**ormat; `.h5`)

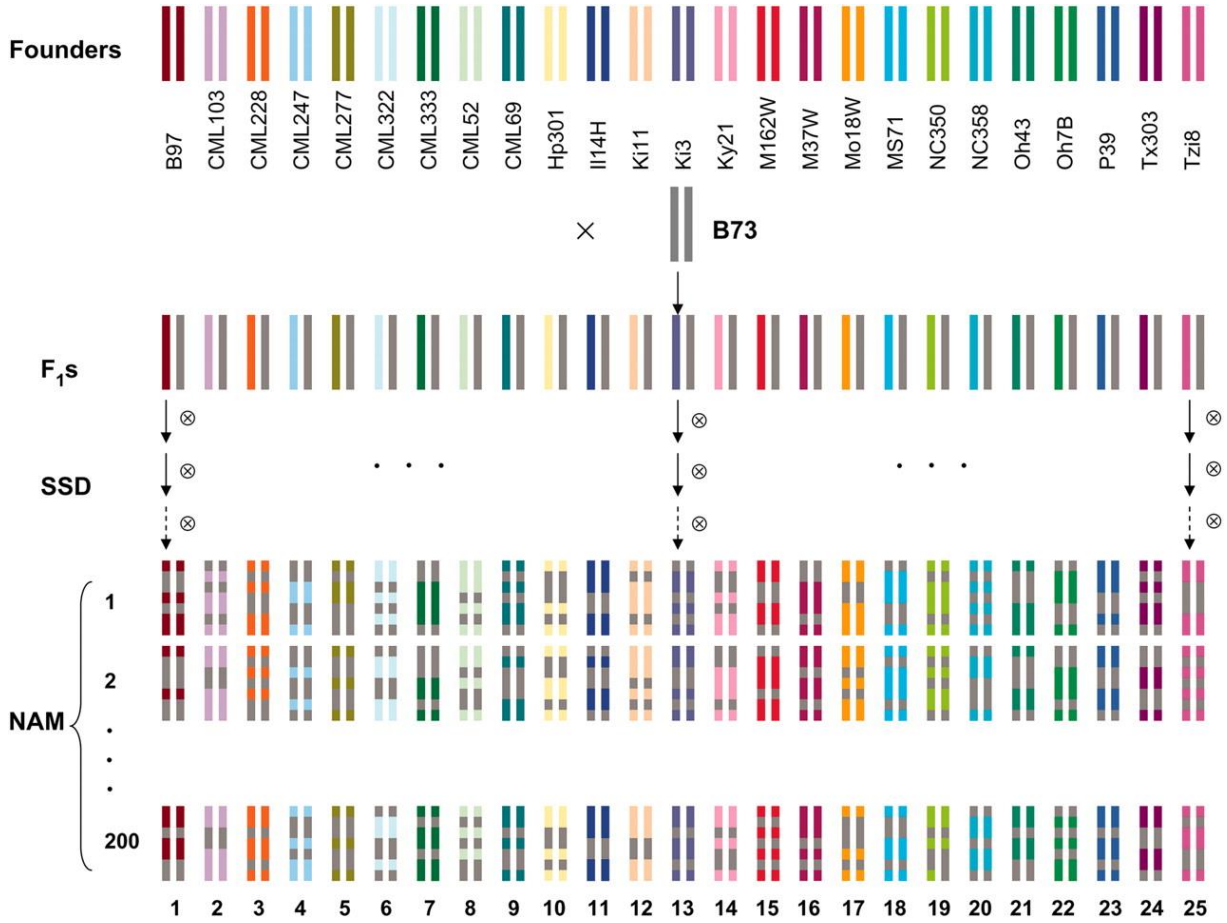
rTASSEL – what data are we using?



Yu et al. 2008 (Genetics)

- US Maize NAM population

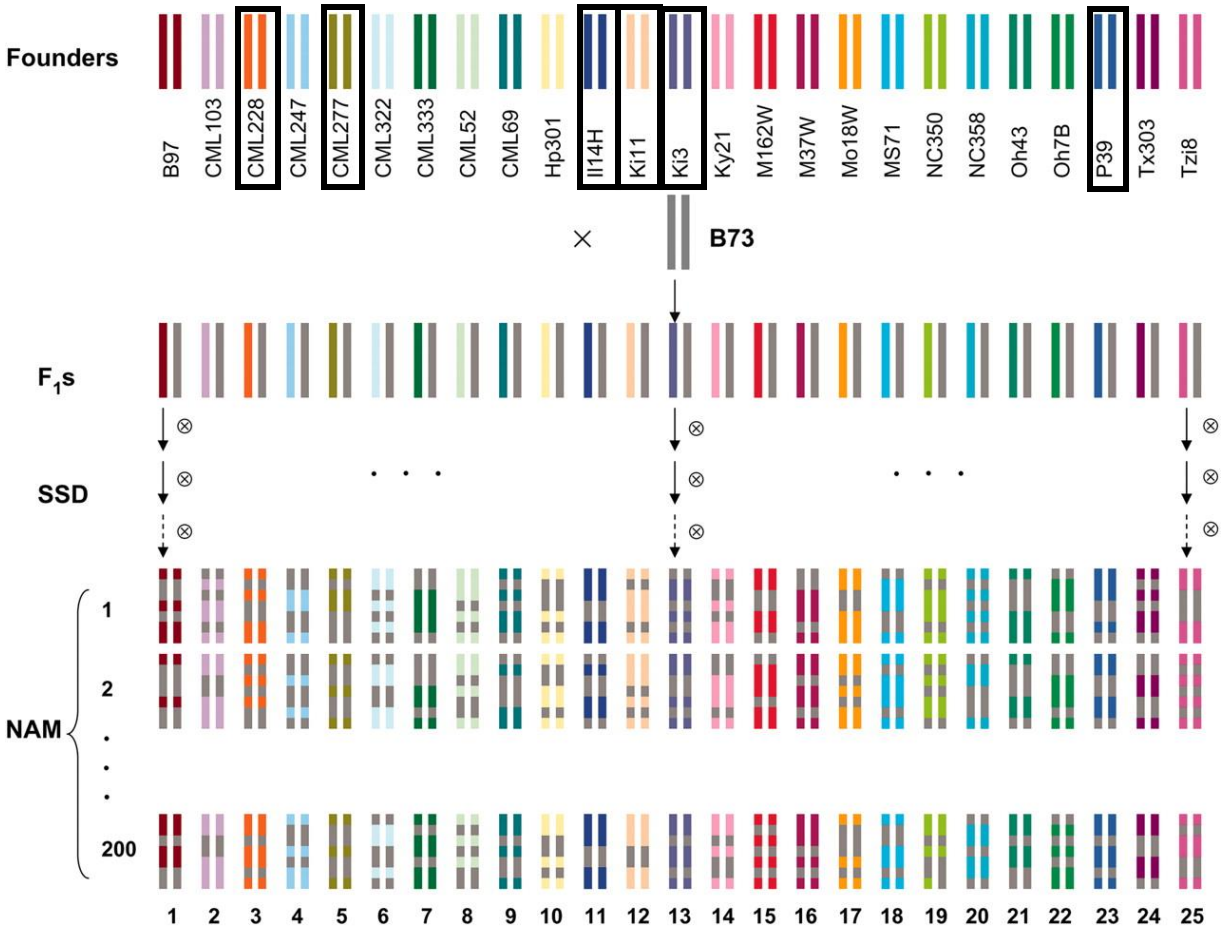
rTASSEL – what data are we using?



Yu et al. 2008 (Genetics)

- US Maize NAM population
  - ~ 1200 RILs (e.g. taxa)

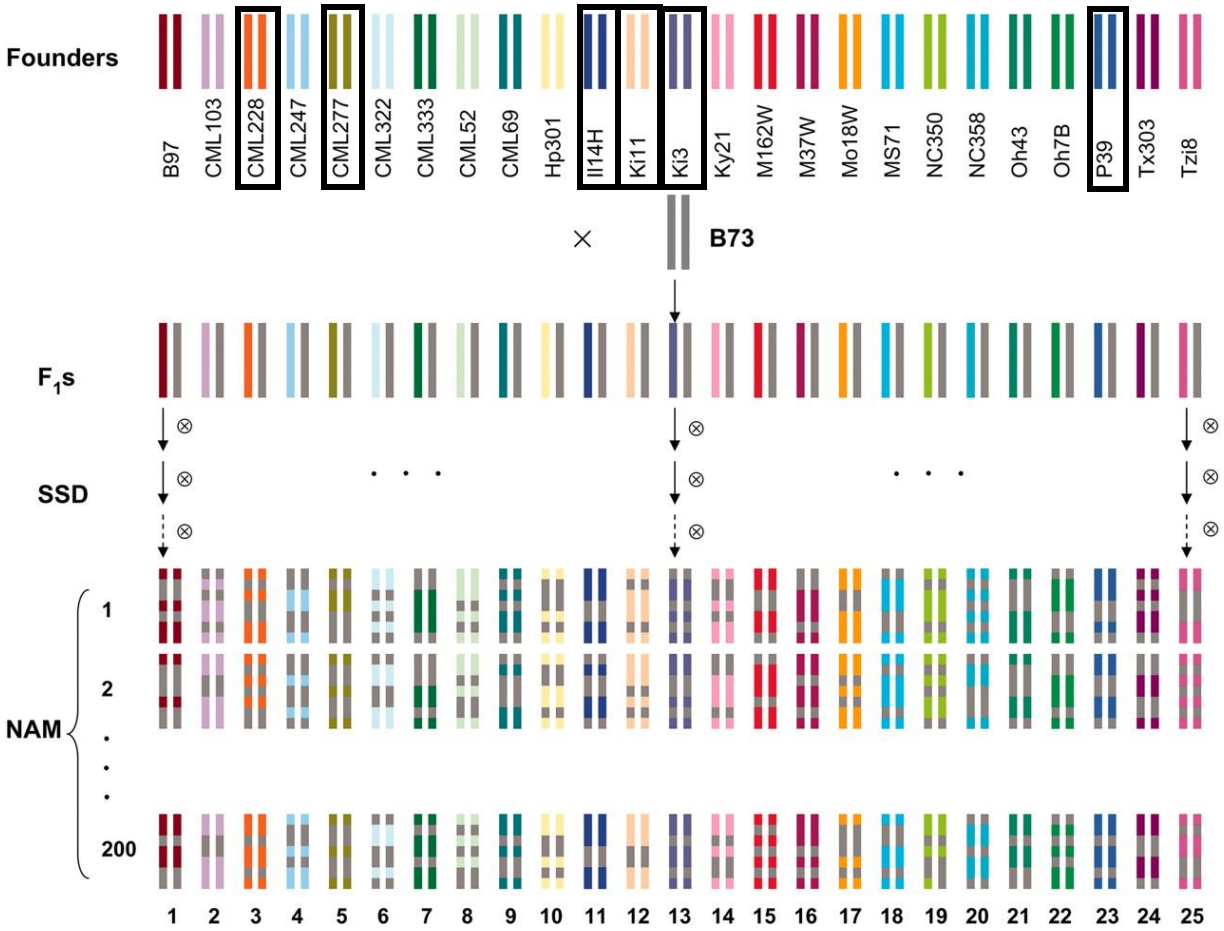
rTASSEL – what data are we using?



Yu et al. 2008 (Genetics)

- US Maize NAM population
  - ~ 1200 RILs (e.g. taxa)
  - 6 families

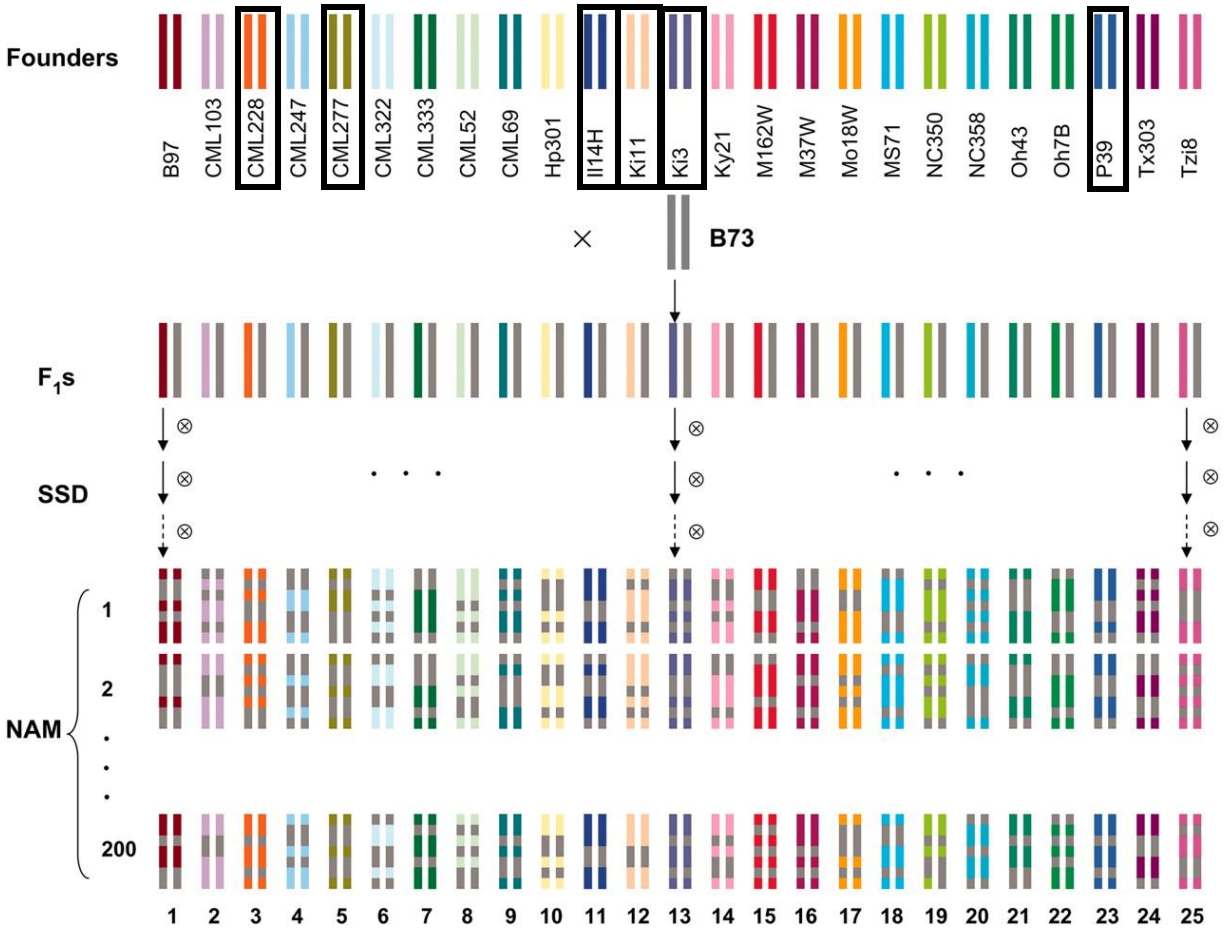
rTASSEL – what data are we using?



Yu et al. 2008 (Genetics)

- US Maize NAM population
  - ~ 1200 RILs (e.g. taxa)
  - 6 families
- Phenotypes
  - Ear weight
  - Days to silk
  - Year and Location

rTASSEL – what data are we using?



Yu et al. 2008 (Genetics)

- US Maize NAM population
  - ~ 1200 RILs (e.g. taxa)
  - 6 families
- Phenotypes
  - Ear weight
  - Days to silk
  - Year and Location
- Genotype Data
  - ~ 9300 SNP locations
  - `.vcf` data

rTASSEL – loading genotype data

```
# Get input variables
genoFile <- paste0(getwd(), "/data/AGPv4_NAM_subset.recode.vcf.gz")

# Get TASSEL genotype object
tasGeno <- rTASSEL::readGenotypeTableFromPath(path = genoFile)
```

rTASSEL – inspect genotype data

```
tasGeno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa............... 1192
##   Positions.......... 9258
##   Taxa x Positions... 11035536
## ---
##   Genotype Table..... [x]
##   Phenotype Table.... [ ]
```

rTASSEL – inspect genotype data

```
tasGeno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa.............. 1192
##   Positions......... 9258
##   Taxa x Positions... 11035536
## ---
##   Genotype Table..... [x]
##   Phenotype Table.... [ ]
```

The number of **taxa** (e.g. genotypes) within the dataset

rTASSEL – inspect genotype data

```
tasGeno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa............... 1192
##   Positions.......... 9258
##   Taxa x Positions... 11035536
## ---
##   Genotype Table..... [x]
##   Phenotype Table.... [ ]
```

The number of **positions** (e.g. SNPs) within the dataset

rTASSEL – inspect genotype data

```
tasGeno %>% object.size()

## 6040 bytes
```

rTASSEL – inspect genotype data

```
tasGeno %>% object.size()

## 6040 bytes
```

- How do we extract various types of data from something so small?

rTASSEL – inspect genotype data

```
methods::slotNames(tasGeno)

## [1] "name"             "jTasselObj"      "jTaxaList"       "jPositionList"
## [5] "jGenotypeTable"   "jPhenotypeTable"
```

rTASSEL – inspect genotype data

```
tasGeno@jGenotypeTable

## [1] "Java-Object{net.maizegenetics.dna.snp.CoreGenotypeTable@4cdbe50f}"
```

- Slots (@) refer to Java objects
- From these Java objects, we can pull TASSEL data into the R environment

rTASSEL – view marker data

```
# Create `SummarizedExperiment()` object
tasSumExp <- rTASSEL::getSumExpFromGenotypeTable(tasGeno)
```

```
# View summary
tasSumExp

## class: RangedSummarizedExperiment
## dim: 11715 996
## metadata(0):
## assays(1): ''
## rownames: NULL
## rowData names(3): tasselIndex refAllele altAllele
## colnames(996): Z004E0016 Z004E0024 ... Z026E0178 Z026E0188
## colData names(2): Sample TasselIndex
```

rTASSEL – view marker data

```
# View genomic ranges (SNPs)
tasSumExp %>% SummarizedExperiment::rowRanges()

## GRanges object with 11715 ranges and 3 metadata columns:
##             seqnames    ranges strand | tasselIndex   refAllele   altAllele
##                <Rle> <IRanges>  <Rle> |   <integer> <character> <character>
##       [1]        1    266478      + |           0           G           T
##       [2]        1    799182      + |           1           C           T
##       [3]        1    804383      + |           2           G           T
##       [4]        1    881302      + |           3           C           A
##       [5]        1   1167934      + |           4           T           C
##       ...      ...       ...    ... .         ...         ...         ...
##   [11711]       10 149994505      + |       11710           C           T
##   [11712]       10 150358428      + |       11711           G           A
##   [11713]       10 150711521      + |       11712           A           G
##   [11714]       10 150849806      + |       11713           A           G
##   [11715]       10 150853928      + |       11714           C           T
##   -------
##   seqinfo: 10 sequences from an unspecified genome; no seqlengths
```

A TASSEL formatted data set (example)

```
<phenotype>
taxa       factor  data              data           data              ...
genotype   family  EarDiameter_NY    EarWeight_NY   KernelWeight_NY   ...
Z001E001   Z001    39.8333           71.0436        6.3500            ...
Z001E002   Z001    41.8333           78.1667        7.1167            ...
Z001E003   Z001    43.0000           80.6667        7.2167            ...
Z001E004   Z001    42.6577           103.5436       5.2507            ...
Z001E005   Z001    42.9077           99.0436        NaN               ...
...        ...     ...               ...            ...               ...
```

A TASSEL formatted data set (example)

```
<phenotype>
taxa        factor  data              data            data              ...
genotype    family  EarDiameter_NY    EarWeight_NY    KernelWeight_NY   ...
Z001E001    Z001    39.8333           71.0436         6.3500            ...
Z001E002    Z001    41.8333           78.1667         7.1167            ...
Z001E003    Z001    43.0000           80.6667         7.2167            ...
Z001E004    Z001    42.6577           103.5436        5.2507            ...
Z001E005    Z001    42.9077           99.0436         NaN               ...
...         ...     ...               ...             ...               ...
```

- Note the `<phenotype>` tag
- Define `taxa`, `data`, `covariate`, and `factor`
- **No spaces** in taxa name
- `NaN` for missing data

A TASSEL formatted data set (example)

```
<phenotype>
taxa        factor   data               data              data                ...
genotype    family   EarDiameter_NY     EarWeight_NY      KernelWeight_NY     ...
Z001E001    Z001     39.8333            71.0436           6.3500              ...
Z001E002    Z001     41.8333            78.1667           7.1167              ...
Z001E003    Z001     43.0000            80.6667           7.2167              ...
Z001E004    Z001     42.6577            103.5436          5.2507              ...
Z001E005    Z001     42.9077            99.0436           NaN                 ...
...         ...      ...                ...               ...                 ...
```

- Note the `<phenotype>`
- Define `taxa`, `data`, `covariate`, and `factor`
- **No spaces** in taxa name
- `NaN` for missing data

A TASSEL formatted data set (example)

```
<phenotype>
taxa        factor  data              data            data              ...
genotype    family  EarDiameter_NY    EarWeight_NY    KernelWeight_NY   ...
Z001E001    Z001    39.8333           71.0436         6.3500            ...
Z001E002    Z001    41.8333           78.1667         7.1167            ...
Z001E003    Z001    43.0000           80.6667         7.2167            ...
Z001E004    Z001    42.6577           103.5436        5.2507            ...
Z001E005    Z001    42.9077           99.0436         NaN               ...
...         ...     ...               ...             ...               ...
```

- Note the `<phenotype>`
- Define `taxa`, `data`, `covariate`, and `factor`
- **No spaces** in taxa name
- `NaN` for missing data

A TASSEL formatted data set (example)

```
<phenotype>
taxa        factor  data              data            data              ...
genotype    family  EarDiameter_NY    EarWeight_NY    KernelWeight_NY   ...
Z001E001    Z001    39.8333           71.0436         6.3500            ...
Z001E002    Z001    41.8333           78.1667         7.1167            ...
Z001E003    Z001    43.0000           80.6667         7.2167            ...
Z001E004    Z001    42.6577           103.5436        5.2507            ...
Z001E005    Z001    42.9077           99.0436         NaN               ...
...         ...     ...               ...             ...               ...
```

- Note the `<phenotype>`
- Define `taxa`, `data`, `covariate`, and `factor`
- **No spaces** in taxa name
- `NaN` for missing data

rTASSEL – phenotype data

TASSEL	rTASSEL / R
taxa	character
data	numeric
covariate	numeric
factor	factor

rTASSEL – loading phenotype data

```
# Get input variables
phenoFile <- paste0(getwd(), "/data/phenotypic_data.csv")


# Load data as tibble / data.frame class
phenoDF <- readr::read_csv(file = phenoFile)


# Convert columns to factor
phenoDF <- phenoDF %>%
    dplyr::mutate(Location = factor(Location)) %>%
    dplyr::mutate(Year = factor(Year))
```

rTASSEL – loading phenotype data

```
# Inspect data set
phenoDF

## # A tibble: 2,546 x 5
##    Taxon     Location Year  EarWeight DaysToSilk
##    <chr>     <fct>    <fct>     <dbl>      <dbl>
##  1 Z003E0001 NY       2006       NA          NA
##  2 Z003E0001 NY       2007       47         856.
##  3 Z003E0002 NY       2006       NA         965.
##  4 Z003E0002 NY       2007       53         973.
##  5 Z003E0003 NY       2006       NA          NA
##  6 Z003E0003 NY       2007       NA        1005.
##  7 Z003E0004 NY       2006       58.5       965.
##  8 Z003E0004 NY       2007       NA         896.
##  9 Z003E0005 NY       2006       60         965.
## 10 Z003E0005 NY       2007       NA         931.
## # … with 2,536 more rows
```

rTASSEL – create rTASSEL data object

```
# Load data frame into rTASSEL data object
tasPheno <- rTASSEL::readPhenotypeFromDataFrame(
    phenotypeDF    = phenoDF,
    taxaID         = "Taxon",
    attributeTypes = c(rep("factor", 2), rep("data", 2))
)
```

rTASSEL – create rTASSEL data object

```
# Load data frame into rTASSEL data object
tasPheno <- rTASSEL::readPhenotypeFromDataFrame(
    phenotypeDF    = phenoDF,
    taxaID         = "Taxon"
    attributeTypes = c(rep("factor", 2), rep("data", 2))
)
```

```
## # A tibble: 2,546 x 5
##    Taxon     Location Year  EarWeight DaysToSilk
##    <chr>     <fct>    <fct>     <dbl>      <dbl>
##  1 Z003E0001 NY       2006       NA          NA
##  2 Z003E0001 NY       2007       47         856.
##  3 Z003E0002 NY       2006       NA         965.
##  4 Z003E0002 NY       2007       53         973.
##  5 Z003E0003 NY       2006       NA          NA
```

rTASSEL – create rTASSEL data object

```
# Load data frame into rTASSEL data object
tasPheno <- rTASSEL::readPhenotypeFromDataFrame(
    phenotypeDF    = phenoDF,
    taxaID         = "Taxon"
    attributeTypes = c(rep("factor", 2), rep("data", 2))
)
```

```
## # A tibble: 2,546 x 5
##    Taxon      Location Year  EarWeight DaysToSilk
##    <chr>      <fct>    <fct>     <dbl>      <dbl>
##  1 Z003E0001 NY       2006       NA          NA
##  2 Z003E0001 NY       2007       47         856.
##  3 Z003E0002 NY       2006       NA         965.
##  4 Z003E0002 NY       2007       53         973.
##  5 Z003E0003 NY       2006       NA          NA
```

rTASSEL – create rTASSEL data object

```
# Load data frame into rTASSEL data object
tasPheno <- rTASSEL::readPhenotypeFromDataFrame(
    phenotypeDF    = phenoDF,
    taxaID         = "Taxon"
    attributeTypes = c(rep("factor", 2), rep("data", 2))
)
```

```
## # A tibble: 2,546 x 5
##    Taxon      Location Year  EarWeight DaysToSilk
##    <chr>      <fct>    <fct>     <dbl>      <dbl>
##  1 Z003E0001 NY       2006       NA          NA
##  2 Z003E0001 NY       2007       47         856.
##  3 Z003E0002 NY       2006       NA         965.
##  4 Z003E0002 NY       2007       53         973.
##  5 Z003E0003 NY       2006       NA          NA
```

rTASSEL – create rTASSEL data object

```
# Load data frame into rTASSEL data object
tasPheno <- rTASSEL::readPhenotypeFromDataFrame(
    phenotypeDF    = phenoDF,
    taxaID         = "Taxon"
    attributeTypes = c(rep("factor", 2), rep("data", 2))
)
```

```
## # A tibble: 2,546 x 5
##    Taxon      Location Year  EarWeight DaysToSilk
##    <chr>      <fct>    <fct>     <dbl>       <dbl>
##  1 Z003E0001 NY       2006       NA          NA
##  2 Z003E0001 NY       2007       47         856.
##  3 Z003E0002 NY       2006       NA         965.
##  4 Z003E0002 NY       2007       53         973.
##  5 Z003E0003 NY       2006       NA          NA
```

rTASSEL – inspect phenotype data

```
tasPheno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa............... 1273
##   Positions.......... NA
##   Taxa x Positions... NA
## ---
##   Genotype Table..... [ ]
##   Phenotype Table.... [x]
## ---
##   Traits: Taxa Location Year EarWeight DaysToSilk
```

rTASSEL – inspect phenotype data

```
tasPheno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa.............. 1273
##   Positions......... NA
##   Taxa x Positions... NA
## ---
##   Genotype Table..... [ ]
##   Phenotype Table.... [x]
## ---
##   Traits: Taxa Location Year EarWeight DaysToSilk
```

The number of **taxa** (e.g. genotypes) within the dataset

rTASSEL – inspect phenotype data

```
tasPheno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa............... 1273
##   Positions.......... NA
##   Taxa x Positions... NA
## ---
##   Genotype Table..... [ ]
##   Phenotype Table.... [x]
## ---
##   Traits: Taxa Location Year EarWeight DaysToSilk
```

Take note that there are **no positions (e.g. SNP data)** in this object

rTASSEL – inspect phenotype data

```
tasPheno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa............... 1273
##   Positions.......... NA
##   Taxa x Positions... NA
## ---
##   Genotype Table..... [ ]
##   Phenotype Table.... [x]
## ---
##   Traits: Taxa Location Year EarWeight DaysToSilk
```

Phenotype columns are displayed below summary

rTASSEL – create rTASSEL data object

```
# Load data frame into rTASSEL data object
tasPheno <- rTASSEL::readPhenotypeFromDataFrame(
    phenotypeDF    = phenoDF,
    taxaID         = "Taxon"
    attributeTypes = c(rep("factor", 2), rep("data", 2))
)
```

- *How do we combine phenotype and genotype data?*

rTASSEL – unify rTASSEL data

An **intersect join** example

rTASSEL – unify rTASSEL data

```
# Load data frame into rTASSEL data object
tasGenoPheno <- rTASSEL::readGenotypePhenotype(
    genoPathOrObj = tasGeno,
    phenoPathDFOrObj = tasPheno
)
```

- **NOTE:** Taxa from genotype and phenotype data **must match!**
- **NOTE:** rTASSEL defaults to intersect join
- This data object will be our main parameter for all subsequent analyses.

rTASSEL – unify rTASSEL data

```
# Load data frame into rTASSEL data object
tasGenoPheno <- rTASSEL::readGenotypePhenotype(
    genoPathOrObj = tasGeno,
    phenoPathDFOrObj = tasPheno
)
```

- We can also run this function beforehand to save some scripting time.
  - Parameters can take paths or objects (e.g. rTASSEL data sets)

rTASSEL – inspect intersect data

```
tasGenoPheno

## A TasselGenotypePhenotype Dataset
##   Class.............. TasselGenotypePhenotype
##   Taxa............... 1144
##   Positions.......... 9258
##   Taxa x Positions... 10591152
## ---
##   Genotype Table..... [x]
##   Phenotype Table.... [x]
## ---
##   Traits: Taxa DaysToSilk EarWeight
```

A **full** `TasselGenotypePhenotype` data object

(IV) Association / Relatedness Functions

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- rTASSEL object that contains phenotype data

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- We **are not** fitting markers (e.g. genotype data) just yet!
- We **are** calculating **B**est **L**inear **U**nbiased **E**stimates for taxa for each trait (with factors)

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

$$phenotype \sim f(...)$$

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

$$phenotype \sim f(...)$$

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

phenotype ~ f(...)

(EarWeight & DaysToSilk)

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

phenotype ~ f(...)

*(. = **both** EarWeight & DaysToSilk)*

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

$$phenotype \sim f(...)$$

$$Location + Year$$

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

$$phenotype \sim f(...)$$

$$(EarWeight \ \& \ DaysToSilk \sim Location + Year)$$

rTASSEL – estimate genotype means (BLUEs)

```
# Calculate BLUEs from initial phenotype data
tasBLUE <- rTASSEL::assocModelFitter(
    tasObj      = tasPheno,
    formula     = . ~ Location + Year,
    fitMarkers = FALSE
)
```

- What in the world does this translate into?
- Remember our basic definition for a model:

$$phenotype \sim f(...)$$

$$(EarWeight \ \& \ DaysToSilk \sim Location + Year)$$

- Use factors to control for field variation

rTASSEL – inspect `tasBLUE` output

```
# What does this BLUE object look like?
tasBLUE %>% class()

## [1] "list"



tasBLUE %>% names()

## [1] "BLUE"       "BLUE_ANOVA"
```

rTASSEL – inspect `tasBLUE` output

```
# What does this BLUE object look like?
tasBLUE$BLUE

## # A tibble: 1,144 x 3
##    Taxa       DaysToSilk EarWeight
##    <chr>           <dbl>     <dbl>
##  1 Z003E0001       884.      47.2
##  2 Z003E0002       969.      53.2
##  3 Z003E0003      1033.     NaN
##  4 Z003E0004       930.      58.3
##  5 Z003E0005       948.      59.8
##  6 Z003E0006       932.      86.5
##  7 Z003E0007       876.      95.8
##  8 Z003E0008       948.      55.8
##  9 Z003E0009       855.     133.
## 10 Z003E0010       892.      34.8
## # … with 1,134 more rows
```

rTASSEL – inspect `tasBLUE` output

```
# What does this BLUE object look like?
tasBLUE$BLUE

## # A tibble: 1,144 x 3
##    Taxa       DaysToSilk EarWeight
##    <chr>           <dbl>     <dbl>
##  1 Z003E0001        884.      47.2
##  2 Z003E0002        969.      53.2
##  3 Z003E0003       1033.     NaN
##  4 Z003E0004        930.      58.3
##  5 Z003E0005        948.      59.8
##  6 Z003E0006        932.      86.5
##  7 Z003E0007        876.      95.8
##  8 Z003E0008        948.      55.8
##  9 Z003E0009        855.     133.
## 10 Z003E0010        892.      34.8
## # … with 1,134 more rows
```

Taxa IDs

rTASSEL – inspect `tasBLUE` output

```
# What does this BLUE object look like?
tasBLUE$BLUE

## # A tibble: 1,144 x 3
##    Taxa       DaysToSilk EarWeight
##    <chr>           <dbl>     <dbl>
##  1 Z003E0001         884.      47.2
##  2 Z003E0002         969.      53.2
##  3 Z003E0003        1033.      NaN
##  4 Z003E0004         930.      58.3
##  5 Z003E0005         948.      59.8
##  6 Z003E0006         932.      86.5
##  7 Z003E0007         876.      95.8
##  8 Z003E0008         948.      55.8
##  9 Z003E0009         855.     133.
## 10 Z003E0010         892.      34.8
## # … with 1,134 more rows
```

Best Linear Unbiased Estimates

rTASSEL – inspect `tasBLUE` output

```
# What does this BLUE object look like?
tasBLUE$BLUE_ANOVA

## # A tibble: 2 x 9
##   Trait           F         p taxaDF taxaMS errorDF errorMS modelDF modelMS
##   <chr>       <dbl>     <dbl>  <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 DaysToSilk  7.82 1.23e-182   1072 10514.     873   1344.    1073  11751.
## 2 EarWeight   1.57 6.36e-  7    908   922.     345    587.     909    922.
```

A summarized **A**nalysis **O**f **VA**riance table

rTASSEL – create new data object

```
# Convert BLUE output to rTASSEL data object
tasGenoPhenoBLUE <- rTASSEL::readGenotypePhenotype(
    genoPathOrObj    = tasGeno,
    phenoPathDFOrObj = tasBLUE$BLUE,
    taxaID           = "Taxa"
)
```

- Use prior genotype data object (`tasGeno`)

rTASSEL – create new data object

```
# Convert BLUE output to rTASSEL data object
tasGenoPhenoBLUE <- rTASSEL::readGenotypePhenotype(
    genoPathOrObj    = tasGeno,
    phenoPathDFOrObj = tasBLUE$BLUE,
    taxaID           = "Taxa"
)
```

- Use `BLUE` output from `tasBLUE` object (`tasBLUE$BLUE`)

rTASSEL – create new data object

```
# Convert BLUE output to rTASSEL data object
tasGenoPhenoBLUE <- rTASSEL::readGenotypePhenotype(
    genoPathOrObj    = tasGeno,
    phenoPathDFOrObj = tasBLUE$BLUE,
    taxaID           = "Taxa"
)
```

- We can pass parameters from other rTASSEL functions (…)
  - e.g. `rTASSEL::readPhenotypeFromDataFrame(`**`taxaID =`** …)

rTASSEL – kinship matrix

```
# Make a kinship matrix
tasKin <- rTASSEL::kinshipMatrix(
    tasObj = tasGenoPhenoBLUE,
    method = "Centered_IBS"
)
```

rTASSEL – kinship matrix

```
# Make a kinship matrix
tasKin <- rTASSEL::kinshipMatrix(
    tasObj = tasGenoPhenoBLUE,
    method = "Centered_IBS"
)
```

- `Centered_IBS` is one of several analytical methods for determinining kinship
  - Endelman and Jannink 2012 (https://www.g3journal.org/content/ggg/2/11/1405.full.pdf)
- More info about other methods can be found here:
  - https://bitbucket.org/tasseladmin/tassel-5-source/wiki/UserManual/Kinship/Kinship

rTASSEL – kinship matrix

```
# Make a kinship matrix
tasKin <- rTASSEL::kinshipMatrix(
    tasObj = tasGenoPhenoBLUE,
    method = "Centered_IBS"
)
```

- **NOTE:** This object can get rather big and is (*currently*) **not advisable** to display it to the console
  - i.e. *N* taxa × *N* taxa matrix
- We can convert this to an R `matrix` class object if we want to observe the kinship metrics

rTASSEL – association analysis (MLM)

```
# Run a mixed linear model
tasMLM <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPhenoBLUE,
    formula     = . ~ 1,
    fitMarkers = TRUE,
    kinship     = tasKin
)
```

- We can now run GWAS using a **M**ixed **L**inear **M**odel:

$$phenotype \sim S + K$$

rTASSEL – association analysis (MLM)

```
# Run a mixed linear model
tasMLM <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPhenoBLUE,
    formula     = . ~ 1,
    fitMarkers = TRUE,
    kinship     = tasKin
)
```

- We can now run GWAS using a **M**ixed **L**inear **M**odel:

$$\textbf{\textit{phenotype}} \sim S + K$$

rTASSEL – association analysis (MLM)

```
# Run a mixed linear model
tasMLM <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPhenoBLUE,
    formula     = . ~ 1,
    fitMarkers = TRUE,
    kinship     = tasKin
)
```

- We can now run GWAS using a **M**ixed **L**inear **M**odel:

$$phenotype \sim S + K$$

rTASSEL – association analysis (MLM)

```
# Run a mixed linear model
tasMLM <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPhenoBLUE,
    formula     = . ~ 1,
    fitMarkers = TRUE,
    kinship     = tasKin
)
```

- We can now run GWAS using a **M**ixed **L**inear **M**odel:

$$phenotype \sim S + K$$

rTASSEL – association analysis (MLM)

```
# Run a mixed linear model
tasMLM <- rTASSEL::assocModelFitter(
    tasObj      = tasGenoPhenoBLUE,
    formula     = . ~ 1,
    fitMarkers = TRUE,
    kinship     = tasKin
)
```

- We can now run GWAS using a **M**ixed **L**inear **M**odel:

$$phenotype \sim S + K$$

- This may take a few minutes to run…

rTASSEL – inspect `tasMLM` output

```
# What does this MLM object look like?
tasMLM %>% class()

## [1] "list"



tasMLM %>% names()

## [1] "MLM_Stats"                    "MLM_Effects"
## [3] "MLM_Residuals_DaysToSilk" "MLM_Residuals_EarWeight"
```

rTASSEL – inspect `tasMLM` output

```
# What does this MLM object look like?
tasMLM$MLM_Stats

## # A tibble: 18,518 x 18
##    Trait Marker Chr       Pos    df         F       p add_effect add_F
##    <fct> <chr>  <fct>   <dbl> <dbl>     <dbl>   <dbl>      <dbl> <dbl>
##  1 Days… 1-226… 1      266478     1   7.57e-1   0.385        NaN   NaN
##  2 Days… 1-769… 1      801190     1   2.02e+0   0.156        NaN   NaN
##  3 Days… 1-772… 1      804383     1   9.45e-1   0.331        NaN   NaN
##  4 Days… 1-156… 1     1614981     1   6.08e-1   0.436        NaN   NaN
##  … …     …      …           …     …         …       …          …     …
```

rTASSEL – inspect `tasMLM` output

```
# What does this MLM object look like?
tasMLM$MLM_Stats

## # A tibble: 18,518 x 18
##    Trait Marker Chr       Pos    df          F       p add_effect add_F
##    <fct> <chr>  <fct>     <dbl> <dbl>      <dbl>   <dbl>      <dbl> <dbl>
##  1 Days… 1-226… 1       266478     1   7.57e-1   0.385        NaN   NaN
##  2 Days… 1-769… 1       801190     1   2.02e+0   0.156        NaN   NaN
##  3 Days… 1-772… 1       804383     1   9.45e-1   0.331        NaN   NaN
##  4 Days… 1-156… 1      1614981     1   6.08e-1   0.436        NaN   NaN
##  … …     …      …            …     …         …       …          …     …
```

- Trait data
  - `DaysToSilk`
  - `EarWeight`

rTASSEL – inspect `tasMLM` output

```
# What does this MLM object look like?
tasMLM$MLM_Stats

## # A tibble: 18,518 x 18
##    Trait  Marker Chr      Pos    df         F       p add_effect add_F
##    <fct>  <chr>  <fct>    <dbl> <dbl>     <dbl>   <dbl>      <dbl> <dbl>
##  1 Days…  1-226… 1       266478     1   7.57e-1   0.385        NaN   NaN
##  2 Days…  1-769… 1       801190     1   2.02e+0   0.156        NaN   NaN
##  3 Days…  1-772… 1       804383     1   9.45e-1   0.331        NaN   NaN
##  4 Days…  1-156… 1      1614981     1   6.08e-1   0.436        NaN   NaN
##  … …      …      …            …     …         …       …          …     …
```

- SNP data and genomic coordinates

rTASSEL – inspect `tasMLM` output

```
# What does this MLM object look like?
tasMLM$MLM_Stats

## # A tibble: 18,518 x 18
##    Trait Marker Chr       Pos    df          F       p add_effect add_F
##    <fct> <chr>  <fct>     <dbl> <dbl>      <dbl>   <dbl>      <dbl> <dbl>
##  1 Days… 1-226… 1       266478     1    7.57e-1   0.385        NaN   NaN
##  2 Days… 1-769… 1       801190     1    2.02e+0   0.156        NaN   NaN
##  3 Days… 1-772… 1       804383     1    9.45e-1   0.331        NaN   NaN
##  4 Days… 1-156… 1      1614981     1    6.08e-1   0.436        NaN   NaN
##  … …     …      …            …     …          …       …          …     …
```

- Association statistics

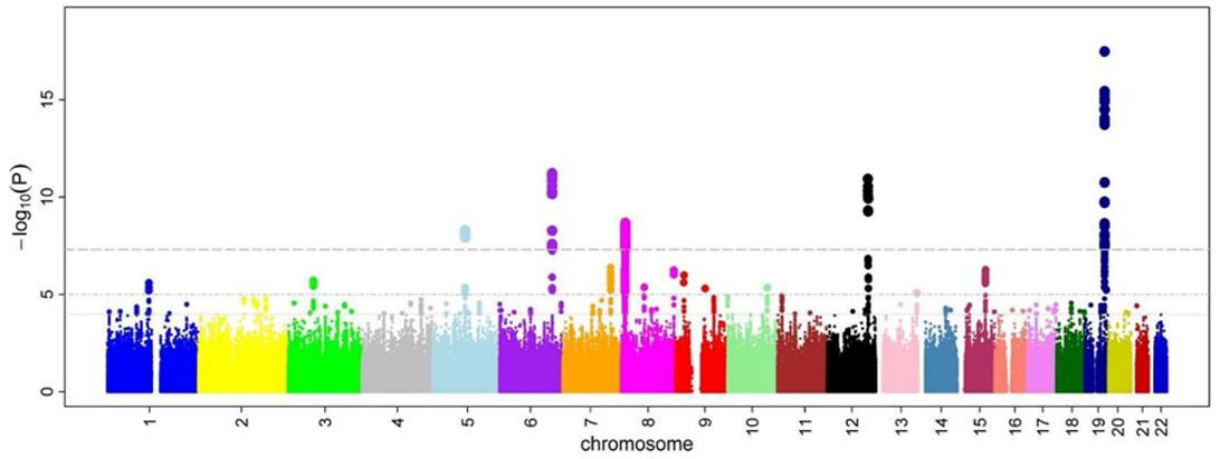
rTASSEL – inspect `tasMLM` output

```
# What does this MLM object look like?
tasMLM$MLM_Stats

## # A tibble: 18,518 x 18
##    Trait Marker Chr       Pos    df         F       p add_effect add_F
##    <fct> <chr>  <fct>   <dbl> <dbl>     <dbl>   <dbl>      <dbl> <dbl>
##  1 Days… 1-226… 1      266478     1   7.57e-1   0.385        NaN   NaN
##  2 Days… 1-769… 1      801190     1   2.02e+0   0.156        NaN   NaN
##  3 Days… 1-772… 1      804383     1   9.45e-1   0.331        NaN   NaN
##  4 Days… 1-156… 1     1614981     1   6.08e-1   0.436        NaN   NaN
##  … …     …      …           …     …         …       …          …     …
```

- How do we analyze such data?

Manhattan plots

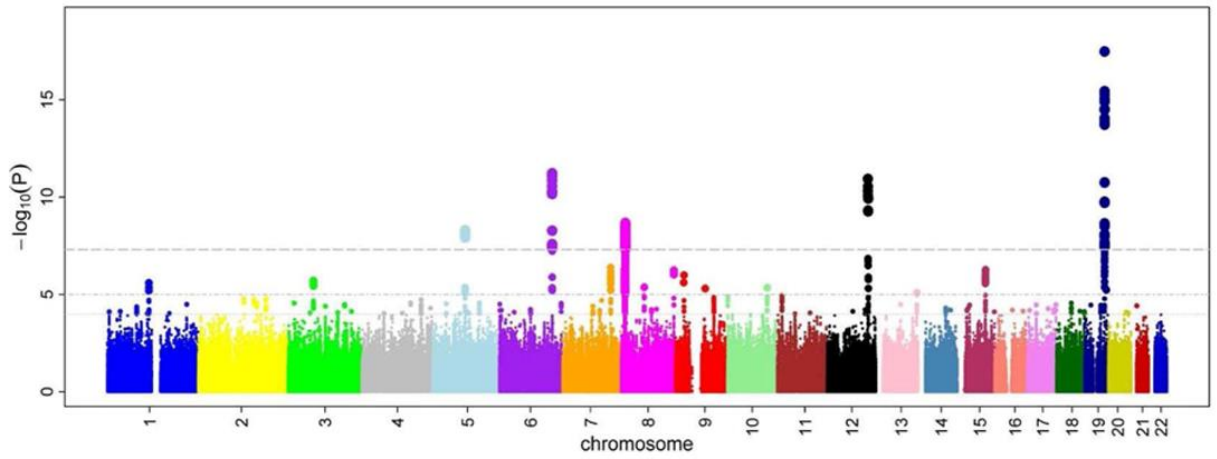


- Genomic coordinates are displayed along the X-axis
- Negative logarithm of the association *p*-value for each SNP displayed on the Y-axis
- Each dot on the Manhattan plot signifies a SNP

Manhattan plots

- Genomic coordinates are displayed along the X-axis
- Negative logarithm ($-\log_{10}$) of the association *p*-value for each SNP displayed on the Y-axis
- Each dot on the Manhattan plot signifies a SNP
- ...and it kind of looks like the skyline of Manhattan

Your mission…



- Using R, create a Manhattan plot with the `tasMLM$MLM_Stats` data that you have generated for both traits.
- **BONUS**: create a new column called (`FDR`), populate it with adjusted *p*-values using false discovery rate, and filter significant markers
  - *Hint*: use the R function `stats::p.adjust()` for this task

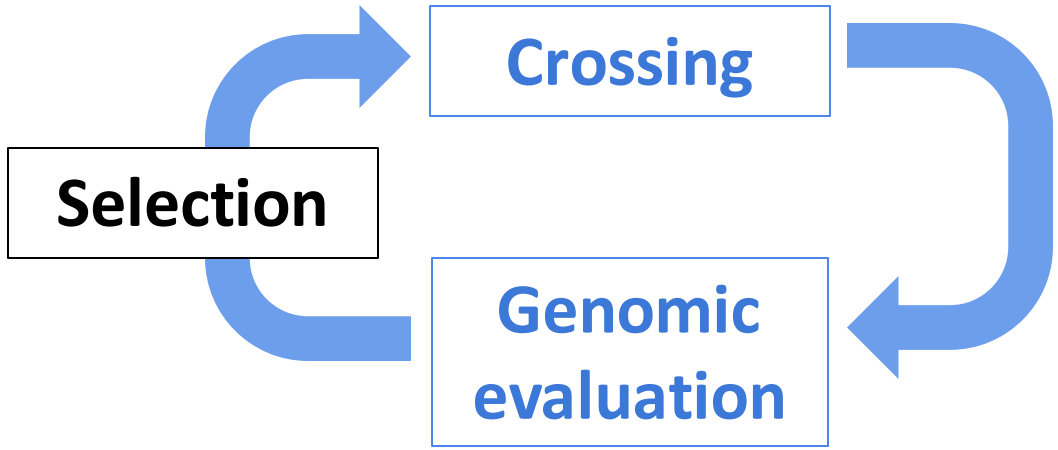
(VI) Genomic Prediction

Actual breeding populations

Selection gains

=

Average increase in phenotypic value

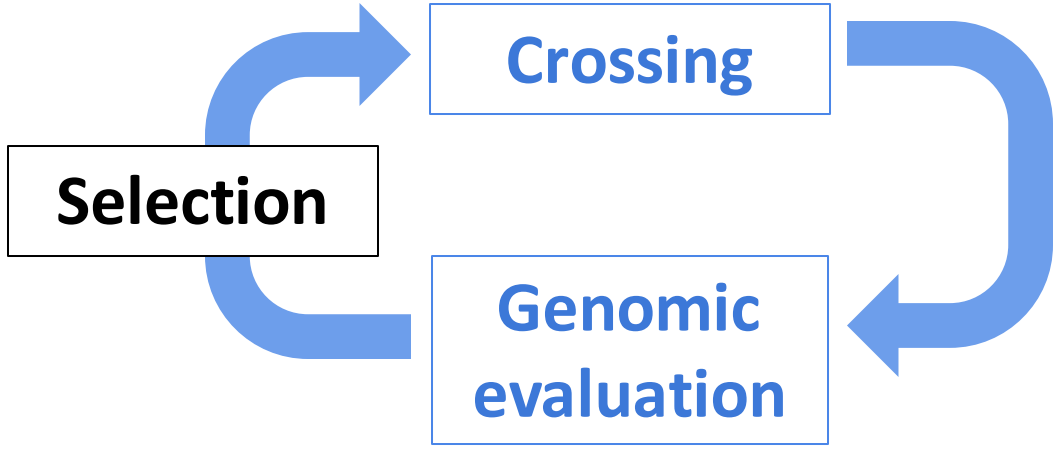


Actual breeding populations

Selection gains

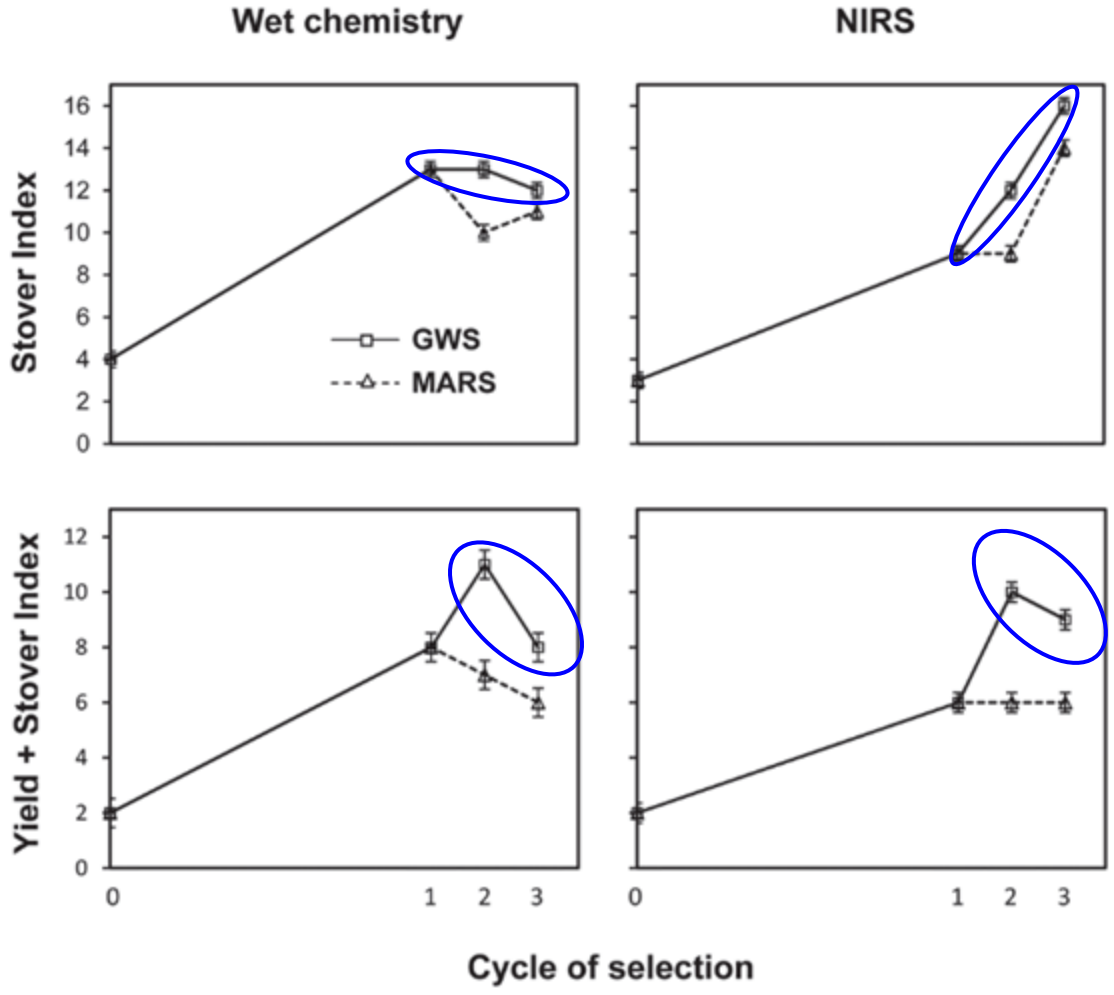
=

Average increase in phenotypic value



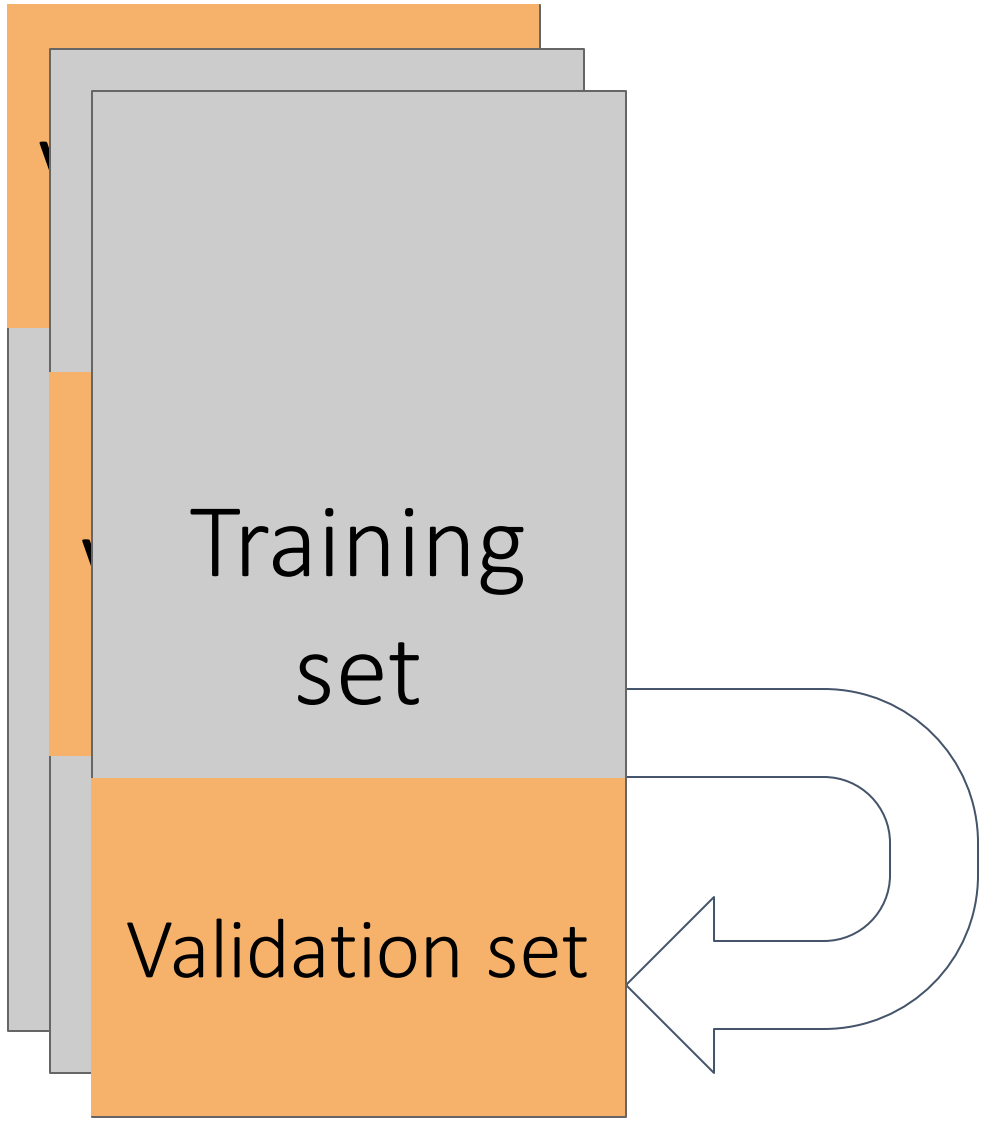
➲ Selected marker effects: MARS

➲ **Regularized marker effects: GWS**



Massman et al. 2013 (Crop Sci.)

Simulated breeding populations

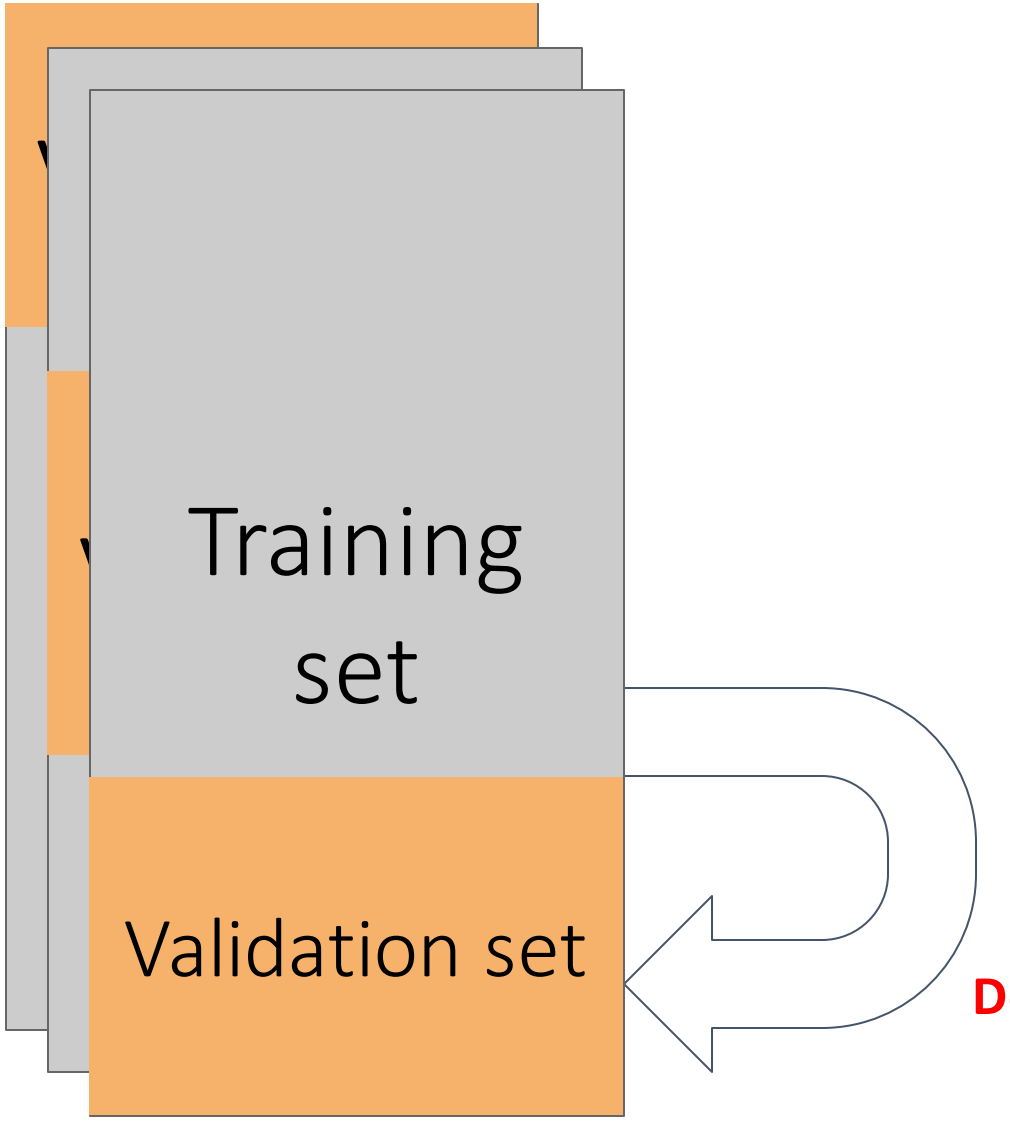


Prediction accuracy

=

Cor(predicted, observed)

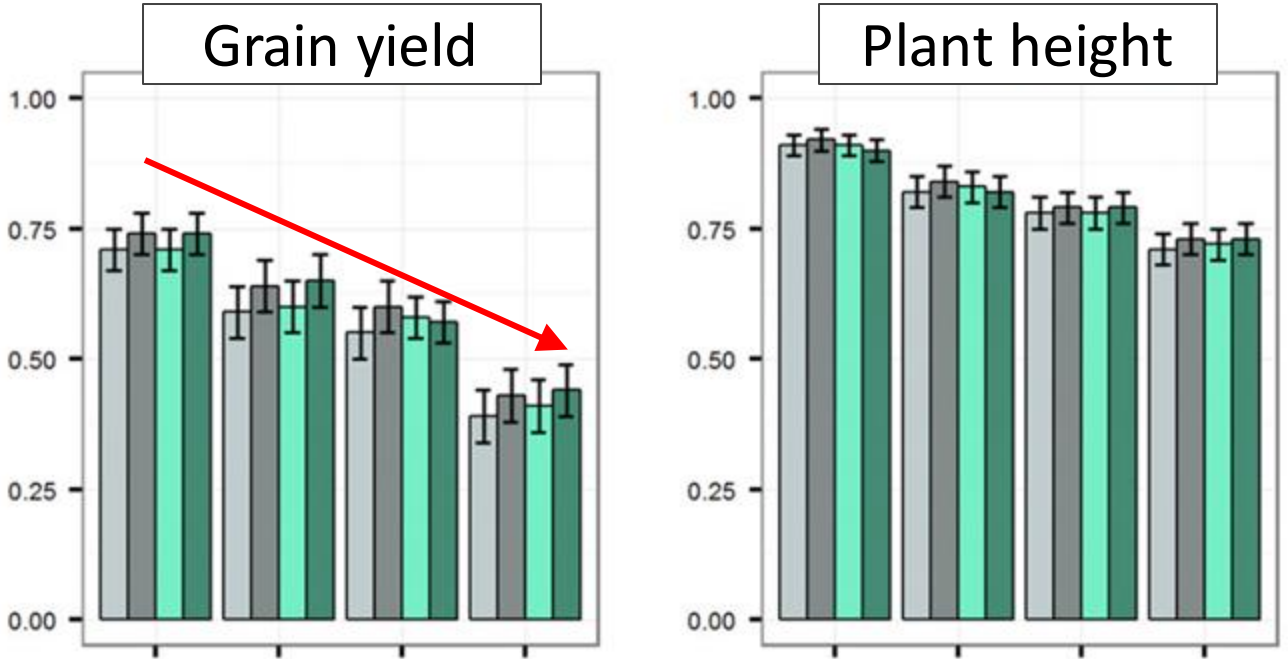
Simulated breeding populations



Prediction accuracy

=

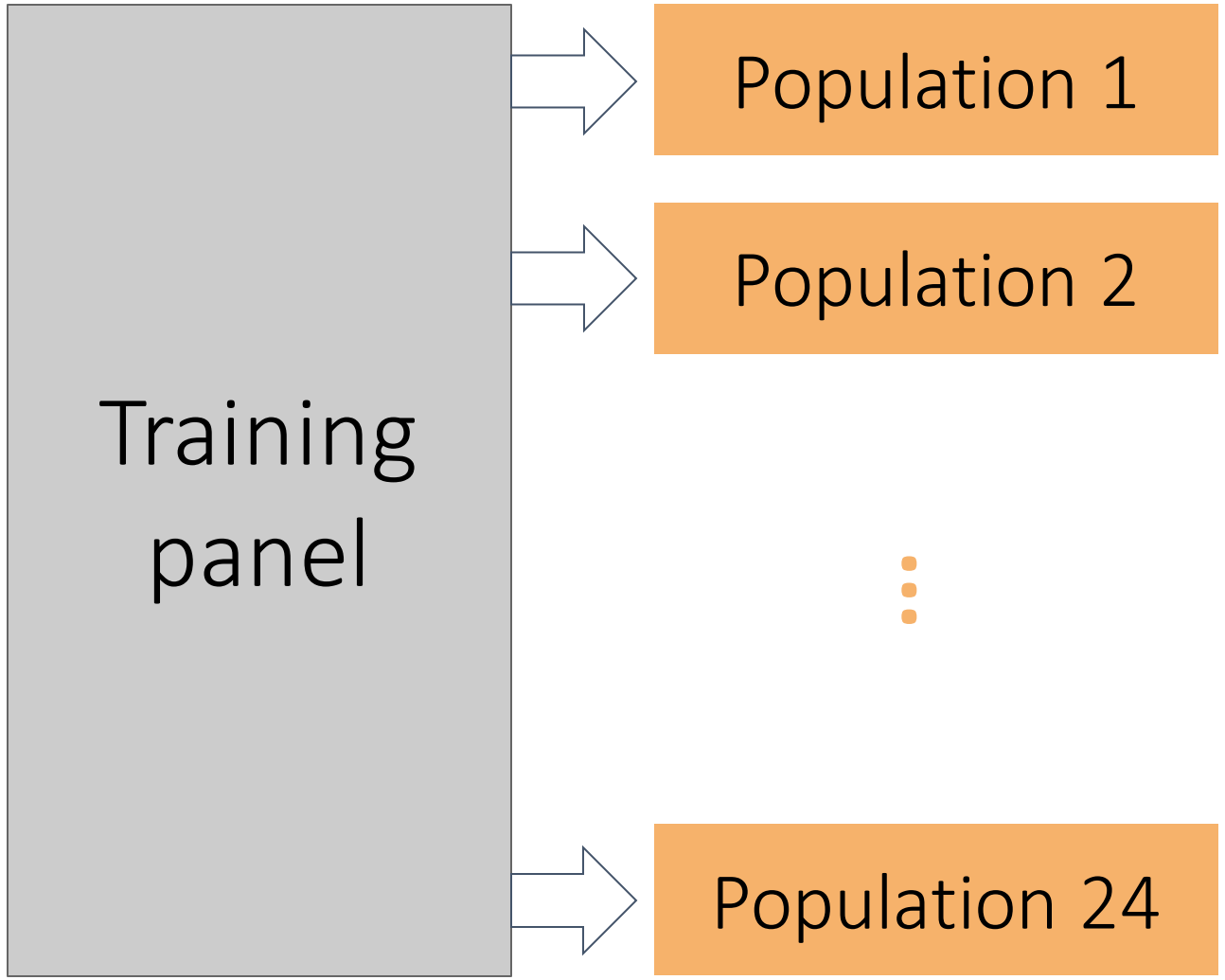
Cor(predicted, observed)



Decreasing relatedness between training set and validation set

Kadam et al. 2016 (G3)

Simulated breeding populations



Ramstein et al. 2019 (bioRxiv)

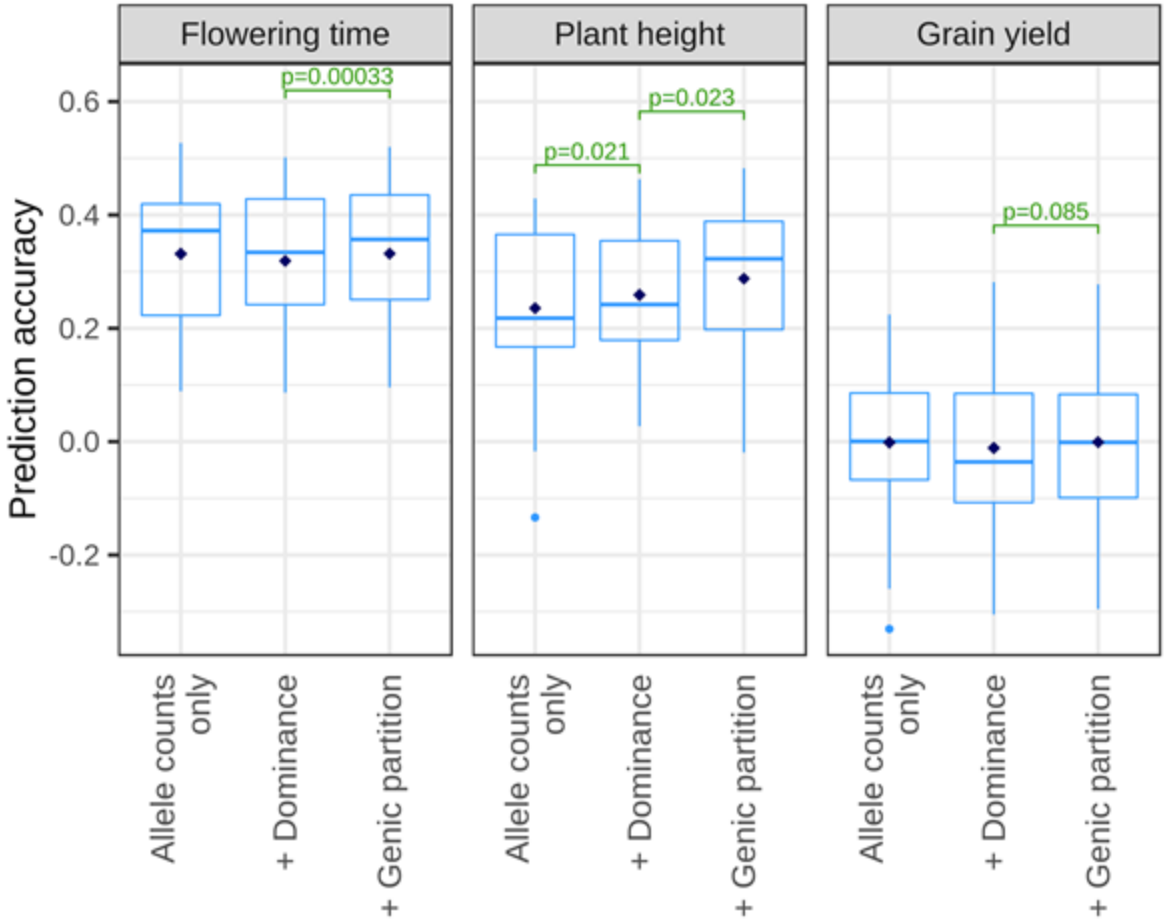
Simulated breeding populations

Training panel

Population 1

Population 2

Population 24



Ramstein et al. 2019 (bioRxiv)

rTASSEL – run cross validation (preamble)

```
# Add family column to BLUEs data
phenoFamilyDF <- tasBLUE$BLUE %>%
    dplyr::mutate(
        family = gsub(
            pattern = "E.*",
            replacement = "",
            x = .$Taxa
        )
    ) %>%
    dplyr::mutate(family = factor(family)) %>%
    dplyr::select(Taxa, family, dplyr::everything())
```

rTASSEL – run cross validation (preamble)

```
# View new data frame
phenoFamilyDF

## # A tibble: 1,144 x 4
##    Taxa       family DaysToSilk EarWeight
##    <chr>      <fct>       <dbl>     <dbl>
##  1 Z003E0001 Z003         884.      47.2
##  2 Z003E0002 Z003         969.      53.2
##  3 Z003E0003 Z003        1033.     NaN
##  4 Z003E0004 Z003         930.      58.3
##  5 Z003E0005 Z003         948.      59.8
##  6 Z003E0006 Z003         932.      86.5
##  7 Z003E0007 Z003         876.      95.8
##  8 Z003E0008 Z003         948.      55.8
##  9 Z003E0009 Z003         855.     133.
## 10 Z003E0010 Z003         892.      34.8
## # … with 1,134 more rows
```

rTASSEL – run cross validation (preamble)

```
# Convert to rTASSEL object
tasFamilyGenoPhenoBLUE <- rTASSEL::readGenotypePhenotype(
    genoPathOrObj    = tasGeno,
    phenoPathDFOrObj = phenoFamilyDF,
    taxaID           = "Taxa",
    attributeTypes   = c("factor", rep("data", 2))
)
```

rTASSEL – cross validation comparisons

```
# Run and create cross validation object report
tasCV <- rTASSEL::genomicPrediction(
    tasPhenoObj = tasFamilyGenoPhenoBLUE,
    kinship     = tasKin,
    doCV        = TRUE,
    kFolds      = 5,
    nIter       = 1
)

# Leave one family out cross-validation
tasLOFO <- rTASSEL::leaveOneFamilyOut(
    phenoFamilyDF = phenoFamilyDF,
    tasKin        = tasKin
)
```