Selection detection between populations

K. Schmid, University of Hohenheim

Summer Term 2017

Introduction

In this computer lab a population genetics-based is demonstrated to identify useful new genetic variation during in a prebreeding project that is based on recurrent selection of a given phenotypic trait.

Assume the following scenario:

- The goal is to identify and introgress new resistance alleles of a disease resistance genes
- The resistance is mediated by a single, well characterized gene, but it is hoped that genetic resources provide new resistance genes
- New resistance genes are identified by allele frequency change in response to recurrent phenotypic selection

This abstract scenario is translated easily into a real-life example:

- New resistance genes against a viral disease should be present in wild ancestors of a crop
- Year 1 : A modern elite variety is taken; **Genotyping** 1 individual
- Year 2 : F1 generation Using handcrosses and open pollination, many different genetic resources (mostly wil d relatives) are intercrossed with the elite variety; **Genotyping**
- Year 5 : Buildup of pool for selection completed; **Genotyping**
- Year 6 : Selection cycle 1; 2 soil types
- Year 7 : Selection cycle 2
- Year 8 : Selection cycle 3
- Year 9 : Selection cycle 4;
- Year 10 : Selection cycle 5; **Genotyping**
- Years 11 - 14: No selection
- Year 15: Selection cycle 6; 1 soil type only; **Genotyping**

The goal is to test whether the selection enriches for certain alleles in the well known resistance genes and whether new genes can be identified based on their allele frequency changes.

Setup of R environment

We use the `tidyverse` libraries to analyse the data. The `tidyverse` consists of several R packages that implement the concept of *Tidy data*, which is supposed to be a better and more approach to data analysis than the classical `Base R` tools (Wickham and others 2014).

```
library(tidyverse)
```

For plotting, we use the package `ggman`. This package is not available from the CRAN repository but from GitHub. To install a package from GitHub, the library `devtools` has to be installed first.

```
library(devtools)
install_github("drveera/ggman")
library(ggman)
```

Technical note: CRAN is the official repository for R packages. All packages hosted there have to fulfill certain formal standards, and each package is peer-reviewed before it is accepted by CRAN. In contrast, R

packages downloadable from GitHub do not undergo this formal procedure, which is useful for packages that are still under development.

Load datasets

Two datasets are needed: 1. A matrix with the SNP data (rows) and for each genotype (columns) 2. A "mapping" file which contains the name of the genotype and the year it was selected

```
snpdata <- read_csv("markerdata_public.csv")
dim(snpdata)
```

The SNP data are encoded as follows: - Only biallelic SNPs are included - The homozygous genotype of the major allele is `0` - The heterozygous genotype is `1` - The homozygous genotype of the minor allele is `2`

Since we are only interested in allele or genotype frequencies, it is not necessary to know the actual base (adenosin, thymin, etc) of each allele.

The first three columns of the SNP file are:

- SNP ID
- Chromosome
- Position on chromosome (in centiMorgan)

The remaining columns are the genotypes (272 in total)

```
yeardata <- read_csv("entry_data.csv")
dim(yeardata)
```

This dataframe has only two columns: - the ID of the genotype (`entryID`) - the year in which it was selected (`year`)

Count the size of the different populations in each year

```
count(yeardata, year)
```

This shows that the largest population size was in year 5 with 180 individuals genotyped.

Calculation of allele differences

There are many methods to calculate allele frequencies between populations. Some of them are described in (Weir 1996), Chapter 5.

We use the simplest one and just count the alleles of each SNP in each population and compare them in a 2x2 table for differences using a χ^2 distribution.

For this, the genotypes need to be grouped by year and the frequencies of the minor and major alleles need to be counted. We first select all genotype IDs that belong to a certain year and then select those from the SNP data.

```
count_frequencies <- function(df, entryData, selyear) {
  # get genotype names from a year
  entryIDs <- unlist(filter(entryData, year == selyear)["entryID"])
  print(selyear)
  print(length(entryIDs))

  # select columns from SNP data frame with
  snpdataSubset <- select(df, one_of(entryIDs))
```

```
  # calculate the allele counts
  totalcount <- rowSums(!is.na(snpdataSubset)) * 2
  mincount <- rowSums(snpdataSubset, na.rm = TRUE) # frequency of minor alleles
  majcount <- totalcount - mincount # freq of major alleles
  return (data.frame(totalcount, majcount, mincount))
}
```

Allele frequency differences are analysed using simple χ^2 test of homogeneity using allele counts in a 2x2 table. This test essentially asks whether the ratio of minor to major allele in two populations are the same and uses the χ^2 distribution to calculate a p-value.

Allele type	Population 1	Population 2
Minor allele	20	81
Major allele	80	19

This translates into R code:

```
x <- matrix(c(20, 81, 80, 19), byrow = TRUE, 2, 2)
x
```

```
##      [,1] [,2]
## [1,]   20   81
## [2,]   80   19
```

Statistical test:

```
  chisq.test(x)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 72.007, df = 1, p-value < 2.2e-16
```

We include the test in a function to make it easily reusable. The `broom` package of `tidyverse` is used to store the output of the statistical test in a data frame for easier access.

```
chisquaretest <- function(x1, x2, x3, x4) {
  x <- matrix(c(x1, x2, x3, x4), byrow = TRUE, 2, 2)
  if (sum(x) == 0) {
    return (NA)
  }
  output <- chisq.test(x)
  output <- broom::tidy(output)
  return(output$p.value)
}
```

Another function compares two populations. The corresponding genotypes from the different years are taken from the data set and combined in dataframes.

```
comparepops <- function(snpdata, yeardata, year1, year2) {
  y1 <- count_frequencies(snpdata, yeardata, year1)
  y2 <- count_frequencies(snpdata, yeardata, year2)
  print(y1)
  df <- data.frame(y1$mincount,y1$majcount,y2$mincount,y2$majcount)
```

```
  colnames(df) <- c('min1', 'maj1', 'min2', 'maj2')
  result <- df %>% rowwise() %>% mutate(pvalue = chisquaretest(min1, min2, maj1, maj2))
  newdf <- data_frame(snpdata$markerid, snpdata$chromosome, snpdata$position, result$pvalue)
  colnames(newdf) <- c('snp', 'chrom', 'bp', 'pvalue')
  return(newdf)
}
```

Now run the analysis with a certain combination of years.

```
results <- comparepops(snpdata, yeardata, 1, 15)
ggman(results, relative.positions = TRUE, pointSize = 0.5)
```

Exercises:

Compare different years

Use the above code chunk to compare different years (e.g., 1 vs. 2, 2 vs. 5, 5 vs. 10, 10 vs. 15)

```
results <- comparepops(snpdata, yeardata, 1, 2)
ggman(results, relative.positions = TRUE, pointSize = 0.5)
```

```
results <- comparepops(snpdata, yeardata, 2, 5)
ggman(results, relative.positions = TRUE, pointSize = 0.5)
```

Create a composite plot

Plot the results from the pairwise comparisons of the different years and combine them into a single plot:

- year 1 vs 2
- year 2 vs 5
- year 5 vs 10
- year 10 vs 15

Name the figures `p1` to `p4`.

Produce a publication-ready PDF file from the composite figure.

```
library(cowplot)
grid <- plot_grid(p1, p2, p3, p4, labels=c("A", "B", "C", "D"), ncol = 1, nrow = 4)
save_plot("selection_cycles_combined.pdf", grid, base_height = 15)
```

Discussion questions

- Compare the allele changes in allele frequencies in the different cycles of selection: What can you learn about the strenght and type of selection? Is it possible to identify new disease resistance genes?
- How is it possible to map QTLs with this approach to specific genomic regions?
- Would it be possible to use other information than allele frequencies to identify genomic regions that were affected by the selection?

References

Weir, BS. 1996. "Genetic Data Analysis Ii: Methods for Discrete Population Genetic Data. Sinauer Assoc." *Inc., Sunderland, MA, USA*.

Wickham, Hadley, and others. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10). Foundation for Open Access Statistics: 1–23.