

# Prepare Phenotype and Covariate Data

Izel Fourie Sørensen

August 14, 2018

In this script we prepare a phenotype and covariate data frame to be used in downstream genomic analyses. As an example of phenotype data we are using the phenotype “resistance to starvation” from the *Drosophila melanogaster* Genetic Reference Panel (DGRP). The data is available at <http://dgrp2.gnets.ncsu.edu/data.html> under “Phenotype files”, “Mackay, et al., Nature, 2012”. Data for both males and females are used. Inversion status (chromosomal inversions) and Wolbachia (*Wolbachia* infection status) can be found under the heading “Other useful files” at the bottom of the page.

We use the plyr, dplyr and tidyr packages for editing data. The readxl package is used for reading .xlsx files. Install these packages as follows:

```
install.packages("plyr")
install.packages("dplyr")
install.packages("tidyr")
install.packages("readxl")
```

```
library(plyr)
library(dplyr)
library(tidyr)
library(readxl)
```

## Download phenotype and covariate data

```
# Female
download.file("http://dgrp2.gnets.ncsu.edu/data/website/starvation.female.csv",
  destfile = "./data/starvation.female.csv")

# Male
download.file("http://dgrp2.gnets.ncsu.edu/data/website/starvation.male.csv",
  destfile = "./data/starvation.male.csv")

# Inversion status
download.file("http://dgrp2.gnets.ncsu.edu/data/website/inversion.xlsx", mode = "wb",
  destfile = "./data/inversion.xlsx")

# Wolbachia
download.file("http://dgrp2.gnets.ncsu.edu/data/website/wolbachia.xlsx", mode = "wb",
  destfile = "./data/wolbachia.xlsx")
```

## Read and edit phenotype data

Read female data.

```
starF <- read.csv(file="./data/starvation.female_2017.csv", header = FALSE)
head(starF)
```

```
##           V1           V2
## 1 line_100  77.92000
## 2 line_101  57.76000
```

```
## 3 line_105 73.12000
## 4 line_109 53.44000
## 5 line_129 42.77551
## 6 line_136 104.32000
```

```
dim(starF)
```

```
## [1] 203 2
```

Read male data.

```
starM <- read.csv(file="./data/starvation.male_2017.csv", header = FALSE)
head(starM)
```

```
##      V1      V2
## 1 line_100 49.28000
## 2 line_101 47.20000
## 3 line_105 51.04000
## 4 line_109 44.96000
## 5 line_129 33.08475
## 6 line_136 63.04000
```

```
dim(starM)
```

```
## [1] 203 2
```

Give column names. "L" = lines, "F" = female, "M" = male.

```
colnames(starF) <- c("L", "F")
colnames(starM) <- c("L", "M")
```

In dplyr a data frame has to be converted to a tibble (tbl). Convert `starF` and `starM` to tibbles.

```
starF <- tbl_df(starF)
starM <- tbl_df(starM)
```

Look at the tibbles

```
starF
```

```
## # A tibble: 203 x 2
##   L      F
##   <fct> <dbl>
## 1 line_100 77.9
## 2 line_101 57.8
## 3 line_105 73.1
## 4 line_109 53.4
## 5 line_129 42.8
## 6 line_136 104.
## 7 line_138 59.5
## 8 line_142 59.3
## 9 line_149 47
## 10 line_153 59.0
## # ... with 193 more rows
```

```
starM
```

```
## # A tibble: 203 x 2
##   L      M
```

```
##      <fct>      <dbl>
## 1 line_100  49.3
## 2 line_101  47.2
## 3 line_105  51.0
## 4 line_109  45.0
## 5 line_129  33.1
## 6 line_136  63.0
## 7 line_138  47.8
## 8 line_142  38.4
## 9 line_149  35.8
## 10 line_153 40.3
## # ... with 193 more rows
```

Join the tibbles for males and females.

```
starMF <- left_join(starM, starF, by= "L")
starMF
```

```
## # A tibble: 203 x 3
##   L      M      F
##   <fct>  <dbl> <dbl>
## 1 line_100 49.3  77.9
## 2 line_101 47.2  57.8
## 3 line_105 51.0  73.1
## 4 line_109 45.0  53.4
## 5 line_129 33.1  42.8
## 6 line_136 63.0 104.
## 7 line_138 47.8  59.5
## 8 line_142 38.4  59.3
## 9 line_149 35.8   47
## 10 line_153 40.3  59.0
## # ... with 193 more rows
```

Create a column for sex information and a column for the phenotype (y), in this case resistance to starvation.

```
starv <- gather(starMF, sex, y, -L)
head(starv)
```

```
## # A tibble: 6 x 3
##   L      sex      y
##   <fct>  <chr> <dbl>
## 1 line_100 M    49.3
## 2 line_101 M    47.2
## 3 line_105 M    51.0
## 4 line_109 M    45.0
## 5 line_129 M    33.1
## 6 line_136 M    63.0
```

```
head(starv, 3)
```

```
## # A tibble: 3 x 3
##   L      sex      y
##   <fct>  <chr> <dbl>
## 1 line_100 M    49.3
## 2 line_101 M    47.2
```

```
## 3 line_105 M      51.0
```

Remove prefix “line\_” from the contents of the “L” column.

```
starv$L <- gsub("line_", "", starv$L, fixed = TRUE)
head(starv$L)
```

```
## [1] "100" "101" "105" "109" "129" "136"
```

## Read and edit Inversion status

Abbreviations used are: INV = inversion karyotype (homozygous), INV / ST = heterozygote for the inversion and ST = standard configuration in a homozygous form.

```
inv <- read_excel("./data/inversion.xlsx",
                  sheet = 1, col_names = TRUE)
```

```
head(inv)
```

```
## # A tibble: 6 x 17
##   `DGRP Line` `In(2L)t` `In(2R)NS` `In(2R)Y1` `In(2R)Y2` `In(2R)Y3`
##   <chr>      <chr>      <chr>      <chr>      <chr>      <chr>
## 1 DGRP_21    ST          ST          ST          ST          ST
## 2 DGRP_26    INV         ST          ST          ST          ST
## 3 DGRP_28    ST          INV         ST          ST          ST
## 4 DGRP_31    ST          ST          ST          ST          ST
## 5 DGRP_32    INV         ST          ST          ST          ST
## 6 DGRP_38    ST          ST          ST          ST          ST
## # ... with 11 more variables: `In(2R)Y4` <chr>, `In(2R)Y5` <chr>,
## #   `In(2R)Y6` <chr>, `In(2R)Y7` <chr>, `In(3L)P` <chr>, `In(3L)M` <chr>,
## #   `In(3L)Y` <chr>, `In(3R)P` <chr>, `In(3R)K` <chr>, `In(3R)Mo` <chr>,
## #   `In(3R)C` <chr>
```

```
dim(inv)
```

```
## [1] 205 17
```

Remove the “DGRP\_” prefix from the contents of the “DGRP Line” column. Save column names of `inv` as a vector `invcols`. Then edit the column names in the `invcols` vector: “DGRP Line” becomes “L” and the brackets “()” in the inversion names are changed to underscores.

```
inv$`DGRP Line` <- gsub("DGRP_", "", inv$`DGRP Line`, fixed=TRUE)
head(inv, 3)
```

```
## # A tibble: 3 x 17
##   `DGRP Line` `In(2L)t` `In(2R)NS` `In(2R)Y1` `In(2R)Y2` `In(2R)Y3`
##   <chr>      <chr>      <chr>      <chr>      <chr>      <chr>
## 1 21          ST          ST          ST          ST          ST
## 2 26          INV         ST          ST          ST          ST
## 3 28          ST          INV         ST          ST          ST
## # ... with 11 more variables: `In(2R)Y4` <chr>, `In(2R)Y5` <chr>,
## #   `In(2R)Y6` <chr>, `In(2R)Y7` <chr>, `In(3L)P` <chr>, `In(3L)M` <chr>,
## #   `In(3L)Y` <chr>, `In(3R)P` <chr>, `In(3R)K` <chr>, `In(3R)Mo` <chr>,
## #   `In(3R)C` <chr>
```

```
invcols <- colnames(inv)
invcols[1] <- "L"
```

```

invcols[2:17] <- gsub("(", "", invcols[2:17], fixed = TRUE)
invcols[2:17] <- gsub(")", "", invcols[2:17], fixed = TRUE)
colnames(inv) <- invcols
inv

```

```

## # A tibble: 205 x 17
##   L      In2Lt In2RNS In2RY1 In2RY2 In2RY3 In2RY4 In2RY5 In2RY6 In2RY7
##   <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 21     ST    ST    ST    ST    ST    ST    ST    ST    ST
## 2 26     INV   ST    ST    ST    ST    ST    ST    ST    ST
## 3 28     ST    INV   ST    ST    ST    ST    ST    ST    ST
## 4 31     ST    ST    ST    ST    ST    ST    ST    ST    ST
## 5 32     INV   ST    ST    ST    ST    ST    ST    ST    ST
## 6 38     ST    ST    ST    ST    ST    ST    ST    ST    ST
## 7 40     ST    ST    ST    ST    ST    ST    ST    ST    ST
## 8 41     ST    ST    ST    ST    ST    ST    ST    ST    ST
## 9 42     ST    ST    ST    ST    ST    ST    ST    ST    ST
## 10 45     ST    ST    ST    ST    ST    ST    ST    ST    ST
## # ... with 195 more rows, and 7 more variables: In3LP <chr>, In3LM <chr>,
## #   In3LY <chr>, In3RP <chr>, In3RK <chr>, In3RMo <chr>, In3RC <chr>

```

### Read and edit *Wolbachia* status

```

wo <- read_excel("./data/wolbachia.xlsx",
                  sheet = 1, col_names = TRUE)
wo

```

```

## # A tibble: 205 x 2
##   `DGRP Line` `Infection Status`
##   <chr>      <chr>
## 1 DGRP__21   y
## 2 DGRP__26   n
## 3 DGRP__28   n
## 4 DGRP__31   n
## 5 DGRP__32   n
## 6 DGRP__38   n
## 7 DGRP__40   y
## 8 DGRP__41   n
## 9 DGRP__42   n
## 10 DGRP__45  n
## # ... with 195 more rows

```

```
dim(wo)
```

```
## [1] 205 2
```

Change column names of wo

```

colnames(wo) <- c("L", "wo")
wo$L <- gsub("DGRP__", "", wo$L, fixed=TRUE)
wo

```

```

## # A tibble: 205 x 2
##   L      wo
##   <chr> <chr>
## 1 21     y

```

```
## 2 26 n
## 3 28 n
## 4 31 n
## 5 32 n
## 6 38 n
## 7 40 y
## 8 41 n
## 9 42 n
## 10 45 n
## # ... with 195 more rows
```

## Create final data frame

Merge phenotype data with inversion status and *Wolbachia* infection status.

```
starvInv <- left_join(starv, inv, by= "L")
starvInv
```

```
## # A tibble: 406 x 19
##   L      sex      y In2Lt In2RNS In2RY1 In2RY2 In2RY3 In2RY4 In2RY5
##   <chr> <chr> <dbl> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 100 M      49.3 INV/ST ST      ST      ST      ST      ST      ST
## 2 101 M      47.2 INV/ST ST      ST      ST      ST      ST      ST
## 3 105 M      51.0 ST      ST      ST      ST      ST      ST      ST
## 4 109 M      45.0 INV/ST ST      ST      ST      ST      ST      ST
## 5 129 M      33.1 ST      ST      ST      ST      ST      ST      ST
## 6 136 M      63.0 ST      ST      ST      ST      ST      ST      ST
## 7 138 M      47.8 ST      ST      ST      ST      ST      ST      ST
## 8 142 M      38.4 ST      ST      ST      ST      ST      ST      ST
## 9 149 M      35.8 ST      ST      ST      ST      ST      ST      ST
## 10 153 M      40.3 ST      ST      ST      ST      ST      ST      ST
## # ... with 396 more rows, and 9 more variables: In2RY6 <chr>,
## #   In2RY7 <chr>, In3LP <chr>, In3LM <chr>, In3LY <chr>, In3RP <chr>,
## #   In3RK <chr>, In3RMo <chr>, In3RC <chr>
```

```
starvIW <- left_join(starvInv, wo, by="L")
starvIW[1:5,15:20]
```

```
## # A tibble: 5 x 6
##   In3LY In3RP In3RK In3RMo In3RC wo
##   <chr> <chr> <chr> <chr> <chr> <chr>
## 1 ST    ST    INV  ST    ST    y
## 2 ST    ST    ST   ST    ST    n
## 3 ST    ST    INV  ST    ST    n
## 4 ST    ST    ST   ST    ST    n
## 5 ST    ST    ST   ST    ST    n
```

An example of how one can look at the data. Here we show a summary (in table form) of the first three inversions.

```
apply(starvIW[,4:19], 2, table)[1:3]
```

```
## $In2Lt
##
##      INV INV/ST      ST
##      38      50     318
##
```

```
## $In2RNS
##
##      INV INV/ST      ST
##      14    20    372
##
## $In2RY1
##
## INV/ST      ST
##      2    404
```

Convert the tibble to a data frame and save the edited phenotype data.

```
starv <- as.data.frame(starvIW)
save(starv, file="./phenotypes/starv_inv_wo.Rdata")
```