QQPlots in GWAS

Julin Maloof

April 30, 2020

QQ Plots and GWAS

- **Purpose:** Determine if there are a likely a large number of false positive in the GWAS
- **Method:** Compare the p-values from the GWAS to those expected from doing the same number of SNP tests if there were no true associations.
- **Logic:** Only a very, very small number of SNPs should be associated with our trait.
- Therefore, the *distribution of p-values obtained from the true GWAS should mostly match those from random data*.

What is QQ anyway?

- Q-Q stands for quantile-quantile
- OK so what is a quantile?
- You are probably familiar with percentiles.
- Quantiles are similar but are cut points that divide a set of observations (or a distribution) into evenly sized groups.
- For example, quartiles divide a set of observations into 4 groups, split at the 25th, 50th, and 75th percentile.
- An example may help:

Quantiles example

```
somedata <- rnorm(n=20, mean=10, sd=3) %>% sort()
somedata
```

```
##  [1]  3.355900  7.493114  7.538595  8.120639  8.136278  9.083835  9.865199
##  [8]  9.951429 10.550930 10.988523 11.169530 11.462287 11.727344 11.781704
## [15] 12.214974 12.463664 12.831509 13.374793 14.535344 14.785842
```

```
quantile(somedata, c(.25,.5,.75))
```

```
##       25%       50%       75%
##  8.846946 11.079027 12.277146
```

- the 25% quantile is between the 5th and 6th data point
- the 50% is between the 10th and 11th
- and the 75% is between 15th and 16th

What are expected p-values?

- Expected p-values are those we would expect to see in a randomized data set.
- These are simple to calculate. If we had a 100 randomized data sets (or SNPs), then:
- We expect 1 test (1%) to have a p-value of <= 0.01
- We expect 5 tests (5%) to have a p-value of <= 0.05
- We expect 75 tests (75%) to have a p-value of <= 0.75
- *Thus if you know the quantile, you know the expected p-value!*

But how does this work in practice?

GWAS with 10 SNPs:

```
## # A tibble: 100 x 12
##    plantID height SNP1  SNP2  SNP3  SNP4  SNP5  SNP6  SNP7  SNP8  SNP9  SNP10
##      <int>  <dbl> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
##  1       1  10.8  G     C     A     A     C     G     G     T     C     C
##  2       2  11.6  T     C     G     T     C     C     C     T     T     C
##  3       3   9.15 G     T     A     T     G     C     C     T     C     G
##  4       4   6.02 T     T     G     A     C     C     C     C     T     G
##  5       5  10.2  G     C     A     A     C     C     C     T     C     G
##  6       6   9.89 T     C     G     T     C     G     G     C     T     C
##  7       7   8.69 G     T     A     T     G     C     G     C     C     C
##  8       8   7.06 T     C     G     T     C     G     G     C     T     C
##  9       9   8.04 G     T     A     A     G     C     C     C     C     C
## 10      10  10.8  T     C     G     A     G     C     C     C     T     C
## # … with 90 more rows
```

Test association for each SNP and get pvalue

First: pivot to longer

```
snplong <- snpdata %>%
  pivot_longer(c(-plantID, -height), names_to="SNP_name", values_to = "genotype") %>%
  arrange(SNP_name)
snplong
```

```
## # A tibble: 1,000 x 4
##    plantID height SNP_name genotype
##      <int>  <dbl> <chr>    <chr>
##  1       1  10.8  SNP1     G
##  2       2  11.6  SNP1     T
##  3       3   9.15 SNP1     G
##  4       4   6.02 SNP1     T
##  5       5  10.2  SNP1     G
##  6       6   9.89 SNP1     T
##  7       7   8.69 SNP1     G
##  8       8   7.06 SNP1     T
##  9       9   8.04 SNP1     G
## 10      10  10.8  SNP1     T
## # … with 990 more rows
```

Next do a t-test on each SNP and extract p value

```
 pvals <- snplong %>% group_by(SNP_name) %>%
  summarize(gwas_pval=t.test(height ~ genotype)$p.value)
pvals
```

```
## # A tibble: 10 x 2
##    SNP_name gwas_pval
##    <chr>        <dbl>
##  1 SNP1        0.862
##  2 SNP10       0.455
##  3 SNP2        0.379
##  4 SNP3        0.0195
##  5 SNP4        0.170
##  6 SNP5        0.691
##  7 SNP6        0.610
##  8 SNP7        0.859
##  9 SNP8        0.483
## 10 SNP9        0.0238
```

These are the **observed** pvalues

sort based on p-value, add quantile

For these type of plots often we calculate a quantile for every test.

```
pvals <- pvals %>%
  arrange(gwas_pval) %>%
  mutate(quantile=1:nrow(pvals)/nrow(pvals))
pvals
```

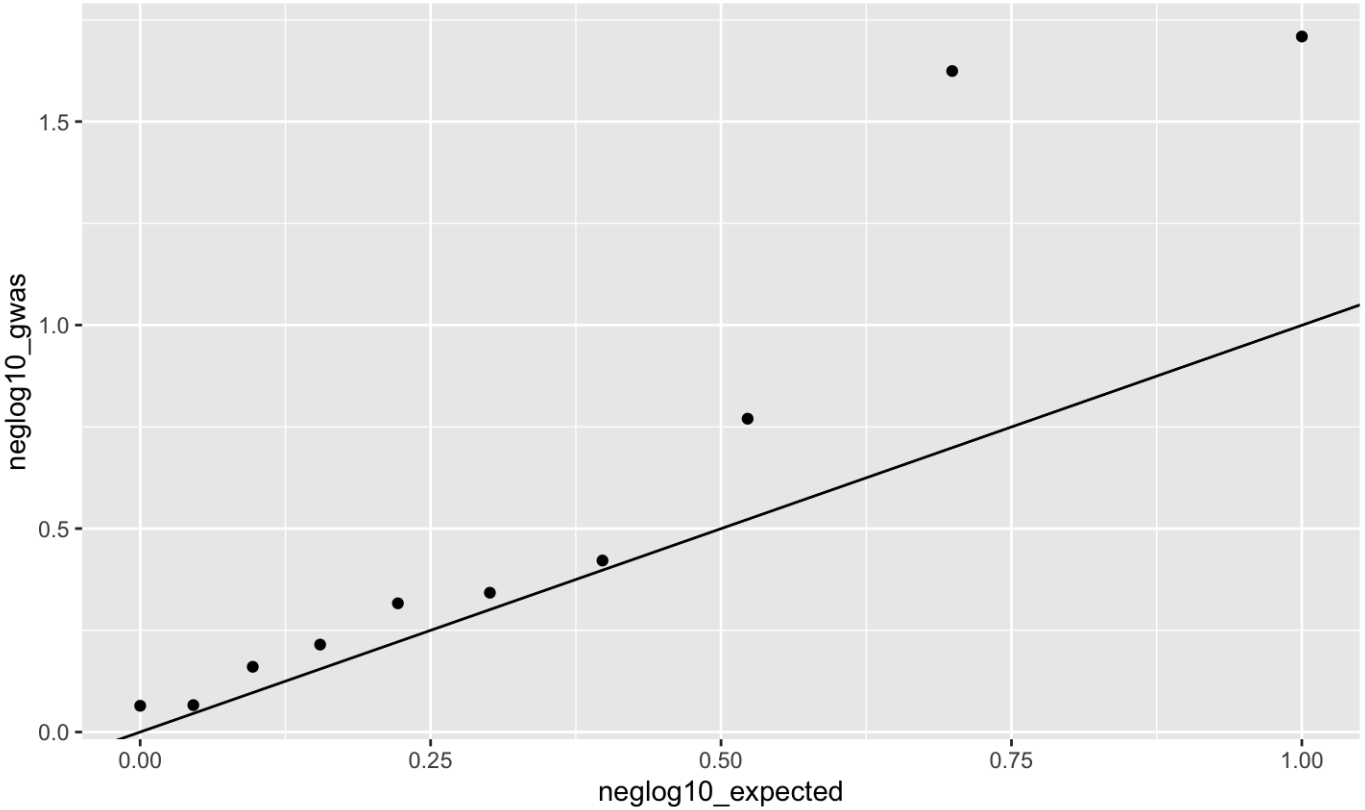
```
## # A tibble: 10 x 3
##    SNP_name gwas_pval quantile
##    <chr>        <dbl>    <dbl>
##  1 SNP3        0.0195      0.1
##  2 SNP9        0.0238      0.2
##  3 SNP4        0.170       0.3
##  4 SNP2        0.379       0.4
##  5 SNP10       0.455       0.5
##  6 SNP8        0.483       0.6
##  7 SNP6        0.610       0.7
##  8 SNP5        0.691       0.8
##  9 SNP7        0.859       0.9
## 10 SNP1        0.862       1
```

calculate expected p values, convert to -log10(p)

```
pvals <- pvals %>%
  mutate(exp_pval = quantile,
         neglog10_gwas = -log10(gwas_pval),
         neglog10_expected = -log10(exp_pval))
pvals
```

```
## # A tibble: 10 x 6
##    SNP_name gwas_pval quantile exp_pval neglog10_gwas neglog10_expected
##    <chr>        <dbl>    <dbl>    <dbl>         <dbl>             <dbl>
##  1 SNP3        0.0195      0.1      0.1         1.71              1
##  2 SNP9        0.0238      0.2      0.2         1.62              0.699
##  3 SNP4        0.170       0.3      0.3         0.770             0.523
##  4 SNP2        0.379       0.4      0.4         0.421             0.398
##  5 SNP10       0.455       0.5      0.5         0.342             0.301
##  6 SNP8        0.483       0.6      0.6         0.316             0.222
##  7 SNP6        0.610       0.7      0.7         0.215             0.155
##  8 SNP5        0.691       0.8      0.8         0.160             0.0969
##  9 SNP7        0.859       0.9      0.9         0.0662            0.0458
## 10 SNP1        0.862       1        1           0.0646            0
```

plot it


neglog10_gwas
1.5
1.0
0.5
0.0
0.00
0.25
0.50
0.75
1.00
neglog10_expected