# Vignette: Using GWASpoly

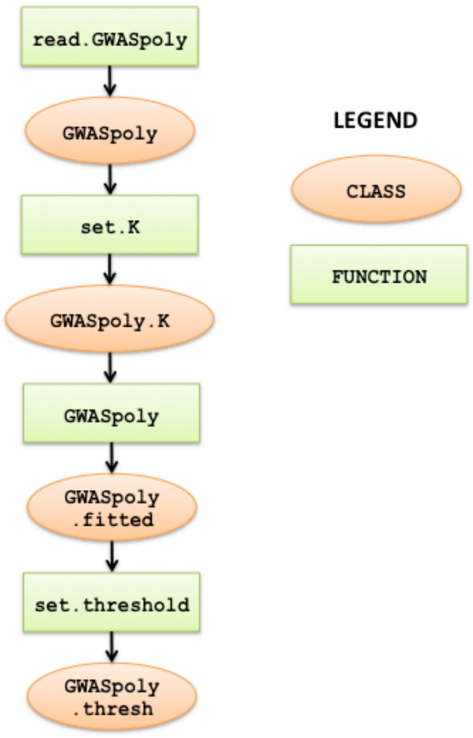
December 3, 2015

The following table outlines the steps involved in performing a mixed model, genome-wide association study with GWASpoly.

| Function | Purpose |
|---|---|
| `read.GWASpoly` | Read in the marker and phenotype data |
| `set.K` | Calculate the kinship matrix |
| `set.params` | Specify additional model parameters |
| `GWASpoly` | Perform the mixed model association test for all markers |
| `qq.plot` | Check the inflation of the p-values |
| `set.threshold` | Establish the QTL detection threshold |
| `manhattan.plot` | Visualize the results |
| `get.QTL` | Extract a list of significant markers |
| `write.GWASpoly` | Write all results to file |

The workflow is supported by a set of four, nested S4 classes:
`GWASpoly ⊃ GWASpoly.K ⊃ GWASpoly.fitted ⊃ GWASpoly.thresh`

The connection between the classes and functions is illustrated in the following figure:



The workflow will be illustrated using the potato dataset analyzed by Rosyara et al.

## 1. Read in the data

The marker (TableS1.csv) and phenotype (TableS2.csv) data were published as Supplemental Tables by Rosyara et al. These tables can also be downloaded with the package at http://potatobreeding.cals.wisc.edu/software.

```
> library(GWASpoly)
Loading required package: rrBLUP
> data <- read.GWASpoly(ploidy=4,pheno.file="TableS2.csv",
geno.file="TableS1.csv",format="ACGT",n.traits=13,delim=",")
Number of polymorphic markers: 3521
Missing marker data imputed with population mode
N = 187 individuals with phenotypic and genotypic information
Detected following fixed effects:
Grp1
Grp2
Grp3
Grp4
Detected following traits:
total_yield
chip_color
tuber_eye_depth
tuber_shape
tuber_size
tuber_length
tuber_width
sucrose
log10_glucose
log10_fructose
malic_acid
vine_maturity_95d
vine_maturity_120d
```

The first column of the geno.file contains the marker name, the second and third columns contain the genetic or physical map, and subsequent columns contain the marker data. Marker data can be coded using one of three formats. The potato dataset is coded in "ACGT", or nucleotide, format. The other possible formats are "numeric" (0,1,2,..ploidy) and "AB". Missing marker data are imputed with the population mode (most frequent genotype).

The first column of the pheno.file contains the genotype identifier (i.e., the name of the individual or line), followed by columns of trait data and then any columns with fixed effect information. The `n.traits` parameter tells the software where to find the break between the trait and fixed effect columns. In the potato example, there are 13 traits and 4 columns to be used as population structure covariates.

## 2. Set the kinship matrix and other parameters

```
> data2 <- set.K(data)
```

```
> params <- set.params(fixed=c("Grp1","Grp2","Grp3","Grp4"),
fixed.type=rep("numeric",4))
```

By default, the software calculates kinship using the realized relationship matrix. To include the four population structure covariates in the model, the `set.params` function is used. The argument `fixed` contains the column names of the effects from the input file, while the argument `fixed.type` indicates whether they are of type "numeric" (covariates) or "factor" (e.g., referring to different environments).

## 3. Mixed model analysis

The following code tells the software to analyze two traits using four different marker-effect models.

```
> data3 <- GWASpoly(data2,models=c("general","additive","1-
dom","2-dom"),traits=c("tuber_shape","tuber_eye_depth"),
params=params)
```

```
Analyzing trait: tuber_shape
P3D approach: Estimating variance components...Completed
Testing markers for model: additive
Testing markers for model: general
Testing markers for model: 1-dom-alt
Testing markers for model: 1-dom-ref
Testing markers for model: 2-dom-alt
Testing markers for model: 2-dom-ref
Analyzing trait: tuber_eye_depth
P3D approach: Estimating variance components...Completed
Testing markers for model: additive
Testing markers for model: general
Testing markers for model: 1-dom-alt
Testing markers for model: 1-dom-ref
Testing markers for model: 2-dom-alt
Testing markers for model: 2-dom-ref
```

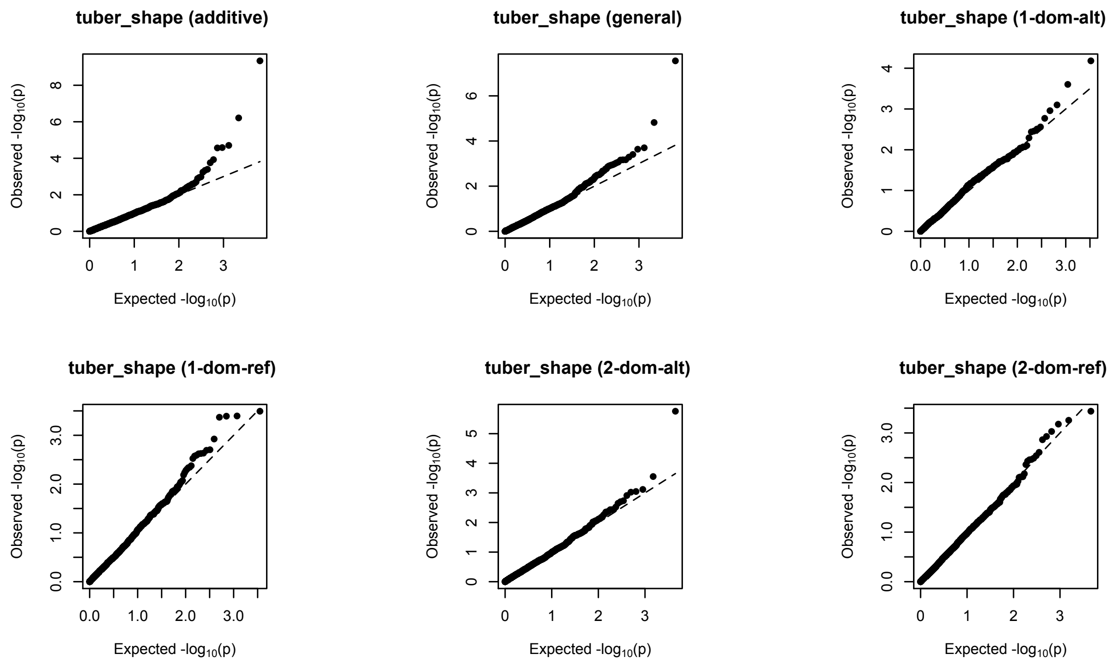
As the output illustrates, for each dominance model specified ("1-dom" = simplex dominant, "2-dom" = duplex dominant) the software generates results for two models, corresponding to whether the reference or alternate allele is dominant. For "AB" marker data, the "A" allele is the reference. For "ACGT" marker data, the software chooses the reference allele for each marker, which is included when the results are written to file.

With any GWAS, a quantile-quantile of the –log(p) values (or "scores") should be consulted to check whether they are improperly "inflated" compared to the null hypothesis. The `qq.plot` function was designed for one trait and one model, but multiple plots can be easily combined into a single panel:

```
> par(mfrow=c(2,3))  #specifies a 2 x 3 panel

> models <- c("additive","general","1-dom-alt","1-dom-ref","2-
dom-alt","2-dom-ref")

> for (i in 1:6) {
     qq.plot(data3,trait="tuber_shape",model=models[i])
}
```



The figure shows that for every model, the majority of p-values lie close to the 1:1 dashed line, as expected for the null hypothesis. Any significant departures from the null hypothesis will appear at the top right of the qq-plot. For tuber shape, the strongest signal was observed with the additive (tetraploid) model.

## 4.  QTL detection

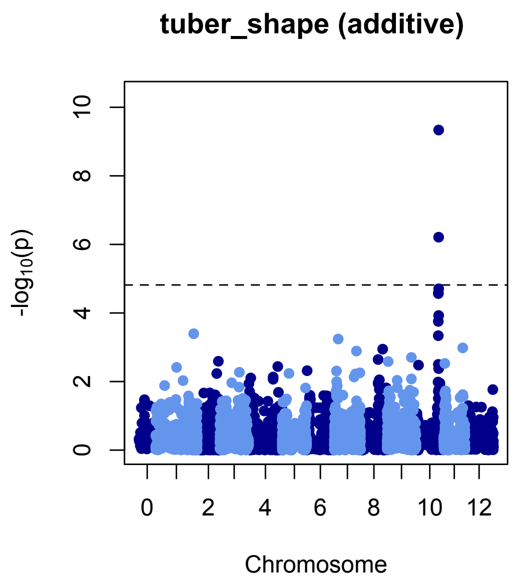
There are three options for setting the QTL significance threshold. To control the genome-wide false positive rate, one can use either the Bonferroni method or a permutation test. Alternatively, one can control the false discovery rate (FDR).  The following code illustrates use of the Bonferroni method.

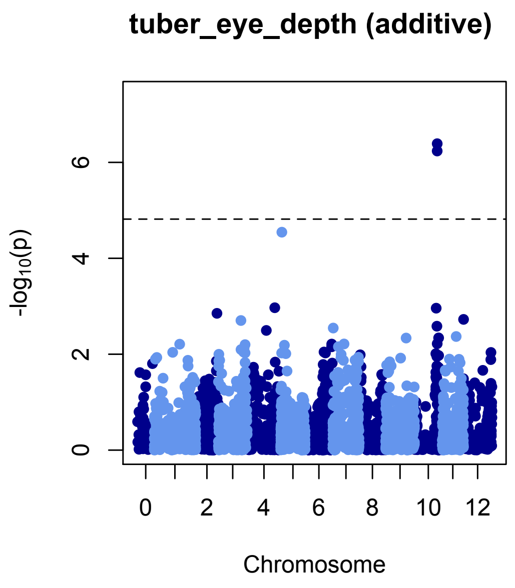
```
> data4 <- set.threshold(data3,method="Bonferroni",level=0.05)
> get.QTL(data4)
```

```
      Trait     Model   Threshold  Marker Chrom Position Ref Alt Score  Effect
tuber_shape      additive    4.82 c2_25471    10 48808404   C   T  6.21   0.33
tuber_shape      additive    4.82  c1_8019    10 48863165   A   G  9.34  -0.40
tuber_shape       general    4.82 c2_25471    10 48808404   C   T  4.82     NA
tuber_shape       general    4.82  c1_8019    10 48863165   A   G  7.55     NA
tuber_shape     2-dom-alt    4.65  c1_8019    10 48863165   A   G  5.76  -0.56
tuber_eye_depth  additive    4.82 c2_25471    10 48808404   C   T  6.24  -0.30
tuber_eye_depth  additive    4.82  c1_8019    10 48863165   A   G  6.39   0.30
```

For compactness, the estimated marker effect is only returned for the single degree-of-freedom models (additive and dominance), not the general model (which has multiple estimated effects). As with the `qq.plot` function, `manhattan.plot` was designed for a single trait and model combination, but multiple traits (or models) can be easily combined into one panel:

```
> par(mfrow=c(1,2))
> manhattan.plot(data4,trait="tuber_shape",model="additive")
> manhattan.plot(data4,trait="tuber_eye_depth",model="additive")
```





Please consult the reference manual for more information about the syntax and additional features of the functions in this package.