

# GWAS\_vis\_vignette

*Arcadio*

*12/13/2018*

Loading up packages and functions

```
## Loading required package: stats4
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:parallel':
##
##   clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##   clusterExport, clusterMap, parApply, parCapply, parLapply,
##   parLapplyLB, parRapply, parSapply, parSapplyLB
## The following objects are masked from 'package:rJava':
##
##   anyDuplicated, duplicated, sort, unique
## The following objects are masked from 'package:stats':
##
##   IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##   anyDuplicated, append, as.data.frame, basename, cbind,
##   colMeans, colnames, colSums, dirname, do.call, duplicated,
##   eval, evalq, Filter, Find, get, grep, grepl, intersect,
##   is.unsorted, lapply, lengths, Map, mapply, match, mget, order,
##   paste, pmax, pmax.int, pmin, pmin.int, Position, rank, rbind,
##   Reduce, rowMeans, rownames, rowSums, sapply, setdiff, sort,
##   table, tapply, union, unique, unsplit, which, which.max,
##   which.min
## Loading required package: S4Vectors
##
## Attaching package: 'S4Vectors'
## The following object is masked from 'package:base':
##
##   expand.grid
## Loading required package: IRanges
## Loading required package: GenomeInfoDb
## Loading required package: Biobase
## Welcome to Bioconductor
##
##   Vignettes contain introductory material; view with
```

```

##      'browseVignettes()'. To cite Bioconductor, see
##      'citation("Biobase")', and for packages 'citation("pkgname)".

## Loading required package: DelayedArray
## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##      anyMissing, rowMedians

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##      colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##      aperm, apply

Set directory, loading up start functions and rTassel

is_experimental <- TRUE

#set workdir for rtassel
setwd("~/myBins/bucklerlabBitbucket/rtassel/")

path_exp_tassel <- paste0(getwd(), "/inst/java/sTASSEL.jar")
path_exp_tassel_libs <- paste0(getwd(), "/inst/java/lib")

## jinit
rJava::.jinit(parameters="-Xmx6g")
.jcall(.jnew("java/lang/Runtime"), "J", "totalMemory")

## [1] 257425408

.jcall(.jnew("java/lang/Runtime"), "J", "maxMemory")

## [1] 5726797824

## Add class paths
if(is_experimental == TRUE) {
  tasselPath <- path_exp_tassel
  tasselLibs <- path_exp_tassel_libs
}

rJava::.jaddClassPath(tasselPath)
rJava::.jaddClassPath(tasselLibs)
print(.jclassPath())

## [1] "/Users/jav246/myBins/R-packages/rJava/java"
## [2] "/Users/jav246/myBins/bucklerlabBitbucket/rtassel/inst/java/sTASSEL.jar"
## [3] "/Users/jav246/myBins/bucklerlabBitbucket/rtassel/inst/java/lib"

```

```

## Source files
source("R/AllGenerics.R")
source("R/AllClasses.R")
source("R/TasselPluginWrappers.R")
source("R/PullFunctions.R")
source("R/gwasPolyObjectCreator.R")

Load up genotypes implementing Tassel code through rJava

geno_fileName <- "/Users/jav246/myBins/bucklerlabBitbucket/rtassel/data/mdp_genotype.hmp.txt"

## Make genotype table from tassels sample data
tasGenoTable <- readGenotypeTable(geno_fileName)

## Make summarized experiment from genotypetable
tas_se <- summarizeExperimentFromGenotypeTable(tasGenoTable)

tas_se

## class: RangedSummarizedExperiment
## dim: 3093 281
## metadata(0):
## assays(1): ''
## rownames: NULL
## rowData names(3): tasselsIndex refAllele altAllele
## colnames(281): 33-16 38-11 ... WF9 YU796NS
## colData names(3): Sample TasselIndex
## matrix.unlist.fourNewCols...nrow...length.fourNewCols...byrow...T.

genoDF <- GWASpolyGenoFromSummarizedExperiment(tas_se)

dim(genoDF)

## [1] 3093 284
genoDF[1:4, 1:8]

##   markerName chr    pos 33-16 38-11 4226 4722 A188
## 1 dummy-1    1 157104     0     0     0     0     0
## 2 dummy-2    1 1947984     0     2     0     2     0
## 3 dummy-3    1 2914066     0     0     0     0     0
## 4 dummy-4    1 2914171     0     0     0     0     0

markers_rrblup_mat <- apply(genoDF[, -(1:3)], 1, convert.snp)

dim(markers_rrblup_mat)

## [1] 281 3093
markers_rrblup <- data.frame(genoDF[, c(1:3)], t(markers_rrblup_mat))

colnames(markers_rrblup) <- colnames(genoDF)

dim(markers_rrblup)

## [1] 3093 284

```

```
markers_rrblup[1:4, 1:8]
```

```
## markerName chr      pos 33-16 38-11 4226 4722 A188
## 1 dummy-1 1 157104 -1 -1 -1 -1 -1
## 2 dummy-2 1 1947984 -1 1 -1 1 -1
## 3 dummy-3 1 2914066 -1 -1 -1 -1 -1
## 4 dummy-4 1 2914171 -1 -1 -1 -1 -1
```

Load phenotype data

```
####straight load as dataframe, skipping first two rows on tassel specific phenotype table format
pheno_fileName <- "/Users/jav246/myBins/bucklerlabBitbucket/rtassel/data/mdp_phenotype.txt"
phenos <- read.table(file = pheno_fileName, skip = 2, header = T, sep = "\t", na.strings = "NaN")
summary(phenos)
```

```
##      Taxa      location      EarHT      dpoll      EarDia
## 33-16 : 2 A:283 Min. : 6.40 Min. :52.60 Min. :23.72
## 38-11 : 2 B:280 1st Qu.: 48.50 1st Qu.:63.50 1st Qu.:34.35
## 4226 : 2      Median : 60.20 Median :67.50 Median :37.00
## 4722 : 2      Mean : 61.58 Mean :67.78 Mean :37.06
## A188 : 2      3rd Qu.: 72.50 3rd Qu.:71.50 3rd Qu.:40.09
## A214N : 2      Max. :138.80 Max. :85.80 Max. :49.30
## (Other):551      NA's :4      NA's :7      NA's :37
##      Q1      Q2      Q3
## Min. :0.0010 Min. :0.0010 Min. :0.0000
## 1st Qu.:0.0020 1st Qu.:0.0050 1st Qu.:0.0020
## Median :0.0100 Median :0.5700 Median :0.0190
## Mean :0.1744 Mean :0.5011 Mean :0.3245
## 3rd Qu.:0.1205 3rd Qu.:0.9680 3rd Qu.:0.7940
## Max. :0.9990 Max. :0.9980 Max. :0.9980
##
```

```
#### select single location, as GWASpoly requires single entries for taxa.
phenosOneLoc <- phenos[phenos$location == "A",]
rownames(phenosOneLoc) <- phenosOneLoc$Taxa
####remove location as it is now redundant.
####Also, GWASpoly expects all traits as initial columns, and fixed effect covariates last
phenosOneLoc <- phenosOneLoc[,-c(2)]
summary(phenosOneLoc)
```

```
##      Taxa      EarHT      dpoll      EarDia
## 33-16 : 1 Min. : 8.00 Min. :54.50 Min. :23.72
## 38-11 : 1 1st Qu.: 48.12 1st Qu.:64.00 1st Qu.:34.86
## 4226 : 1 Median : 60.50 Median :67.50 Median :37.32
## 4722 : 1 Mean : 61.75 Mean :67.75 Mean :37.20
## A188 : 1 3rd Qu.: 73.00 3rd Qu.:71.50 3rd Qu.:40.02
## A214N : 1 Max. :136.00 Max. :85.00 Max. :46.35
## (Other):277      NA's :1      NA's :3      NA's :33
##      Q1      Q2      Q3
## Min. :0.0010 Min. :0.001 Min. :0.0000
## 1st Qu.:0.0020 1st Qu.:0.005 1st Qu.:0.0020
## Median :0.0090 Median :0.579 Median :0.0230
## Mean :0.1728 Mean :0.502 Mean :0.3253
## 3rd Qu.:0.1160 3rd Qu.:0.968 3rd Qu.:0.7940
## Max. :0.9990 Max. :0.998 Max. :0.9980
```

```

##
Run GWAS
## create GWASpoly object with coopted read.GWASpoly function
#uses tassel created summarizedExperiment for genotypes
data_gwasPoly <- se_createGWASpolyObject(ploidy = 2, phenoDF = phenosOneLoc,
                                         SummarizedExperimentObject = tas_se,
                                         format = "numeric", n.traits = 3)

## Number of polymorphic markers: 3093
## Missing marker data imputed with population mode
## N = 264 individuals with phenotypic and genotypic information
## Detected following fixed effects:
## Q1
## Q2
## Q3
## Detected following traits:
## EarHT
## dpoll
## EarDia

## add kinship information to object
data_gwasPoly <- set.K(data_gwasPoly)

## set parameters for mixed model
params <- set.params(fixed=unlist(strsplit("Q1,Q2,Q3", ",")),
                    fixed.type=rep("numeric",3))

## run gwas with GWASpoly
data_gwasPoly_res <- GWASpoly(data = data_gwasPoly, models = "additive",
                              params = params)

## Analyzing trait: EarHT
## P3D approach: Estimating variance components...Completed
## Testing markers for model: additive
## Analyzing trait: dpoll
## P3D approach: Estimating variance components...Completed
## Testing markers for model: additive
## Analyzing trait: EarDia
## P3D approach: Estimating variance components...Completed
## Testing markers for model: additive

#run gwas with rrBLUP
k_rrblup <- A.mat(markers_rrblup_mat)

phenosOneLoc_rrblup <- phenosOneLoc[phenosOneLoc$Taxa %in% colnames(markers_rrblup), ]

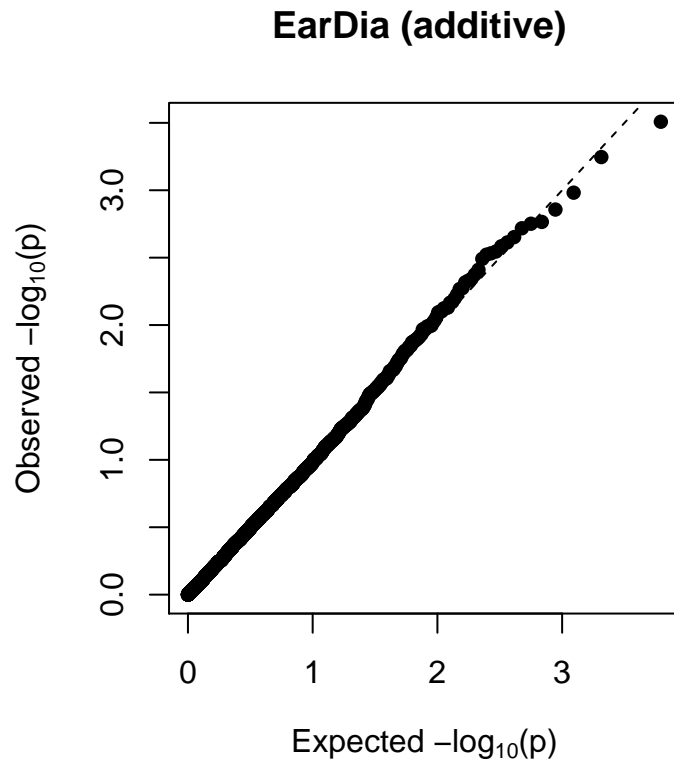
gwas_rrblup <- GWAS(pheno = phenosOneLoc_rrblup, geno = markers_rrblup, K = k_rrblup,
                  fixed = unlist(strsplit("Q1,Q2,Q3", ",")), P3D = T, n.core=6, plot = F)

## [1] "GWAS for trait: EarHT"
## [1] "Variance components estimated. Testing markers."
## [1] "GWAS for trait: dpoll"
## [1] "Variance components estimated. Testing markers."
## [1] "GWAS for trait: EarDia"
## [1] "Variance components estimated. Testing markers."

```

Create GWASpoly plots and set thresholds to get QTLs

```
qq.plot(data_gwasPoly_res, trait = "EarDia", model = "additive")
```



```
## NULL
```

```
#can set Bonferroni and own pvalue
```

```
data_gwasPoly_res <- set.threshold(data_gwasPoly_res, method = "FDR", level=0.05)
```

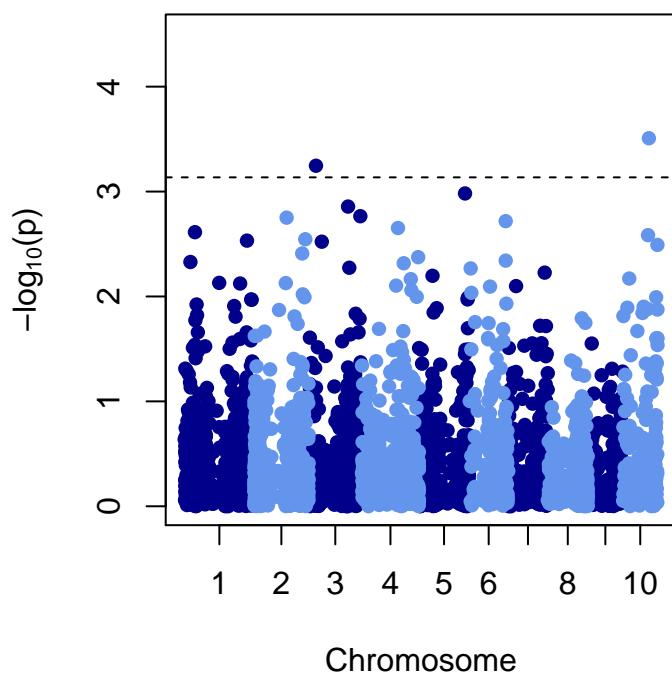
```
get.QTL(data = data_gwasPoly_res)
```

```
##      Trait    Model Threshold      Marker Chrom  Position Ref Alt Score
## 987 EarDia additive      3.14 dummy-987     3  31695534  0  1  3.25
## 2987 EarDia additive      3.14 dummy-2987    10 111608788  0  1  3.51
##      Effect
## 987    -1.08
## 2987    -1.15
```

```
#can set any of the 3 traits
```

```
manhattan.plot(data_gwasPoly_res, trait = "EarDia", model = "additive")
```

## EarDia (additive)



```
## NULL
```

```
Unwrap gwaspoly results class object
```

```
traitGWASresults <- gwasPolyToDF(data_gwasPoly_res)
```

```
## getting results for: EarHT
```

```
## getting results for: dpoll
```

```
## getting results for: EarDia
```

```
traitGWASresults[traitGWASresults$markerLogPVal > traitGWASresults$sigTreshold,]
```

```
##           Marker markerpVal markerLogPVal markerEffect trait sigTreshold
## 4036 dummy-2209 0.02995077      3.508200      -1.145989 EarDia      3.13549
## 6942 dummy-3080 0.03892190      3.246198      -1.083329 EarDia      3.13549
##      Chrom  Position Ref Alt
## 4036      7  14349767   0   1
## 6942     10 144548839   0   1
```

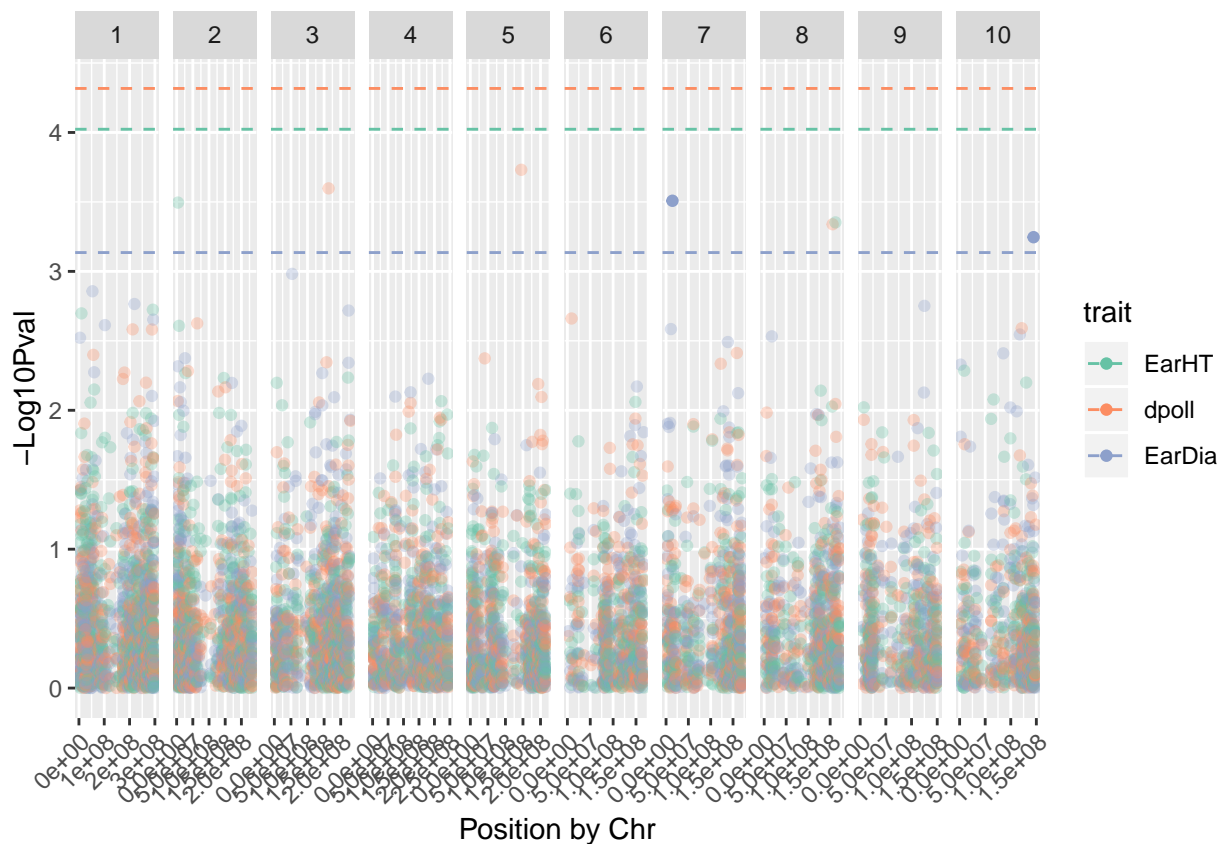
```
summary(traitGWASresults)
```

```
##           Marker      markerpVal      markerLogPVal      markerEffect
## dummy-1      : 3      Min.      :0.02396      Min.      :0.000007      Min.      : -4.37836
## dummy-10     : 3      1st Qu.:0.54946      1st Qu.:0.127346      1st Qu.: -0.29295
## dummy-100    : 3      Median :0.73557      Median :0.307111      Median :  0.02879
## dummy-1000   : 3      Mean    :0.69642      Mean    :0.434249      Mean     :  0.07314
## dummy-1001   : 3      3rd Qu.:0.88043      3rd Qu.:0.598824      3rd Qu.:  0.36503
## dummy-1002   : 3      Max.    :0.99999      Max.    :3.731254      Max.     :  5.34599
## (Other)      :9261
##      trait      sigTreshold      Chrom      Position
## EarHT :3093      Min.      :3.135      1      :1620      Min.      : 139753
```

```
## dpoll :3093 1st Qu.:3.135 2 :1179 1st Qu.: 43868122
## EarDia:3093 Median :4.023 5 :1071 Median :128402775
## Mean :3.825 3 :1065 Mean :119893324
## 3rd Qu.:4.317 4 : 957 3rd Qu.:175159119
## Max. :4.317 8 : 768 Max. :299170077
## (Other):2619
## Ref Alt
## Min. :0 Min. :1
## 1st Qu.:0 1st Qu.:1
## Median :0 Median :1
## Mean :0 Mean :1
## 3rd Qu.:0 3rd Qu.:1
## Max. :0 Max. :1
##
```

Create simple manhattan like plot for all traits

```
manhattan_trait_plot(traitGWASresults = traitGWASresults, traitIDcol = "trait",
                    positionIDcol = "Position", chromIDcol = "Chrom", pValIDcol = "markerpVal")
```



Parse GFF file to get genes and create GenomicRanges object

```
maizeGFFgenesGR <- gffToGeneGR(gffFile = "~/Box/projectMaize/PHG/cimmyt_assemblies_analy/b73/Zea_mays.A
```

Annotating SNPs with their closest gene. Best for annotating purposes.

```
traitGWASresults_annotated <- annotate_gwasRes_byNearest(gwasResDF = traitGWASresults,
                annotationGR = maizeGFFgenesGR, positionIDcol_gwas = "Position",
                chromIDcol_gwas = "Chrom", outFmt = "data.frame")
```

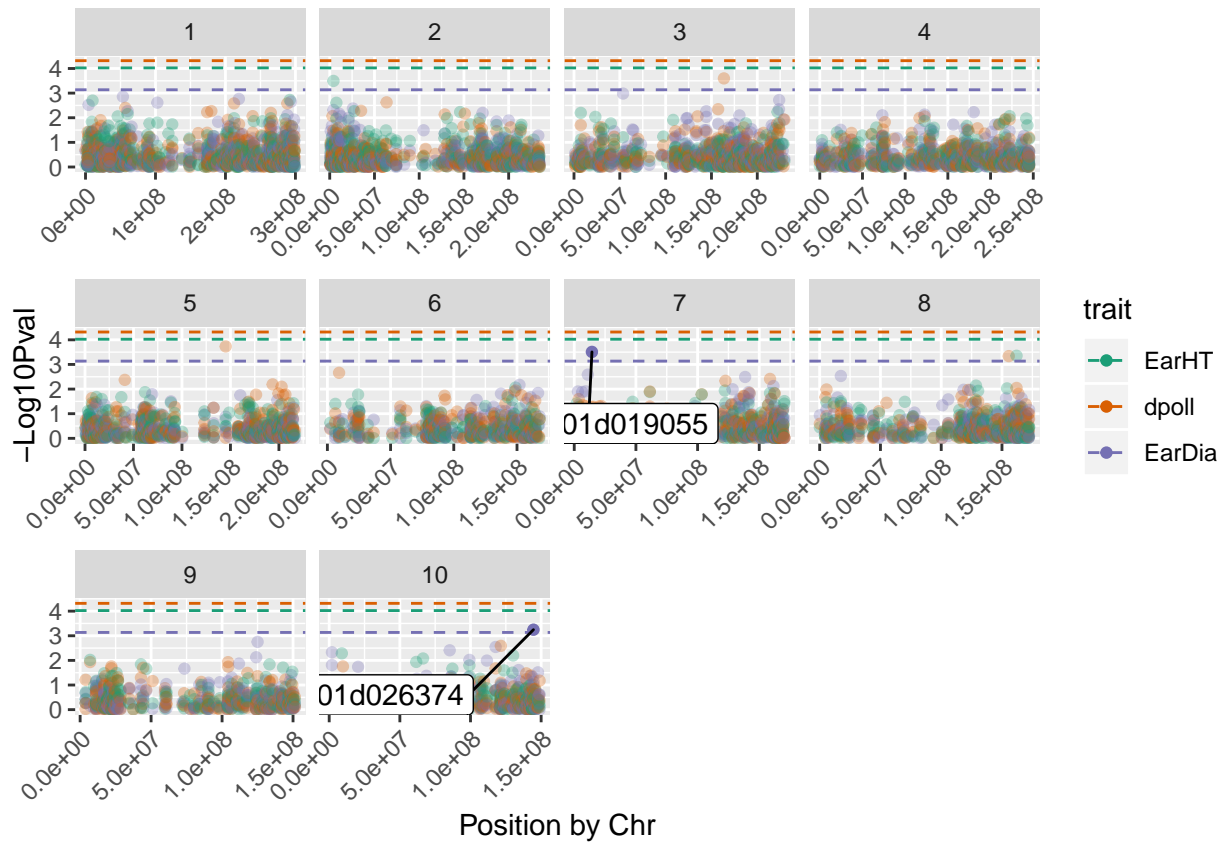


```
## Returning data.frame
```

Plot manhattan with nearest annotation on significant SNPs or by genomicRange set

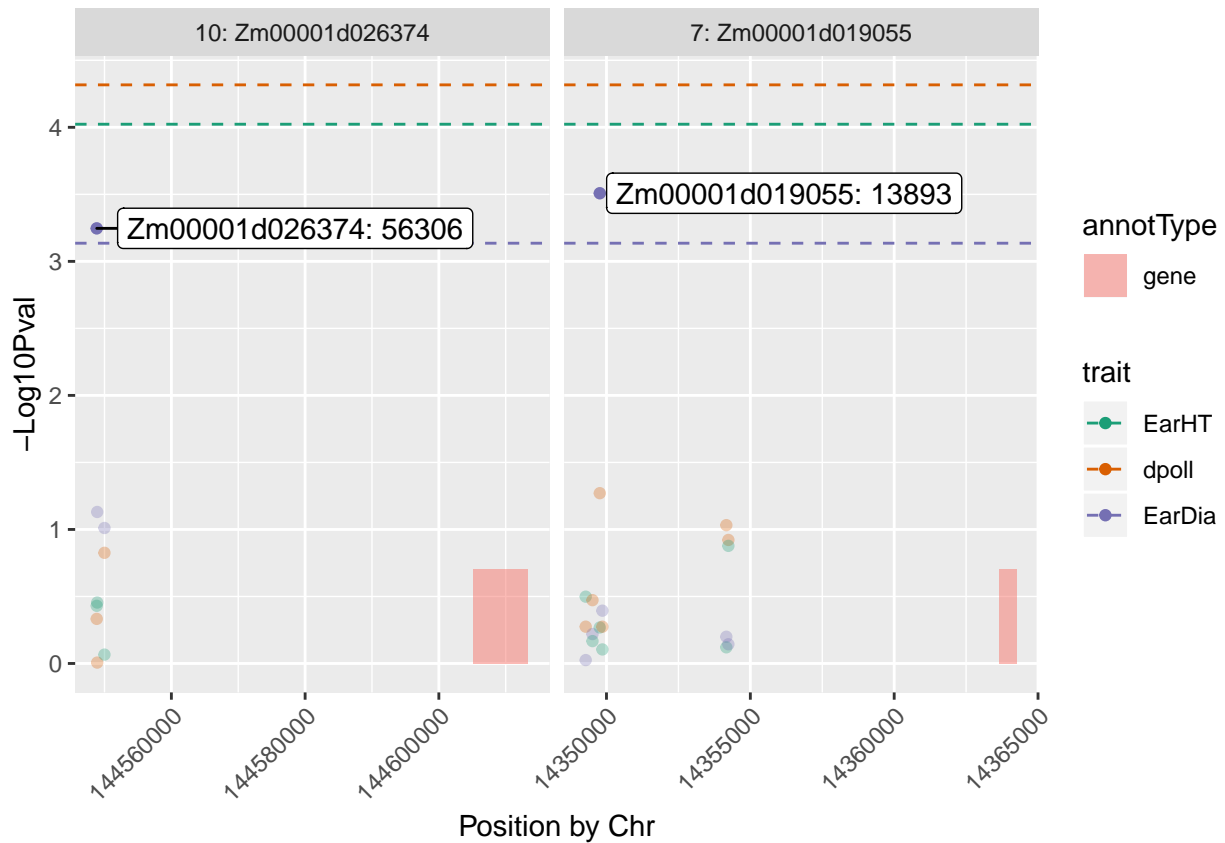
```
#plotting by chromosome
```

```
manhattan_annot_plot(annotatedGWASresults = traitGWASresults_annotated,  
  labelType = "annotationName", zoomGR = "none")
```

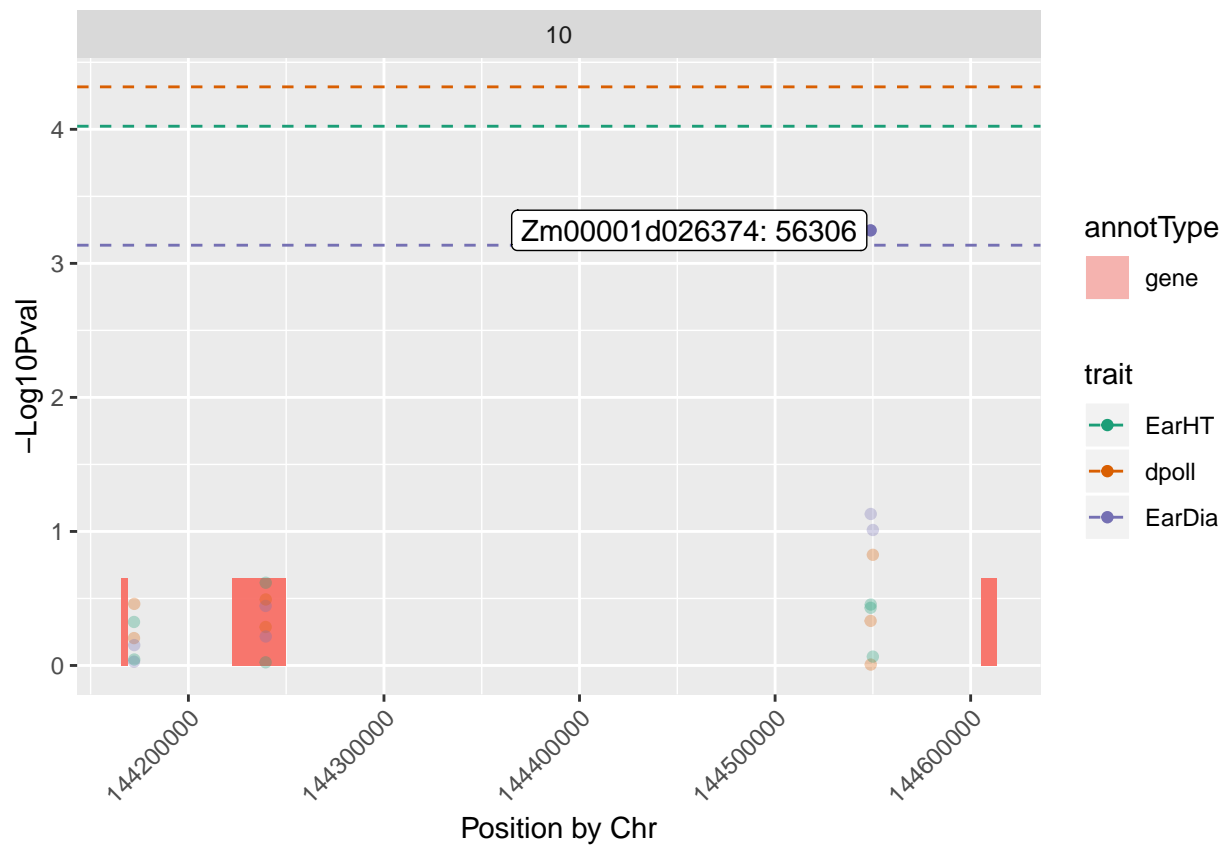


```
#plotting with zoom over each closest significant annotations
```

```
manhattan_annot_plot(annotatedGWASresults = traitGWASresults_annotated,  
  labelType = "composite", zoomGR = "auto")
```



```
#plotting with zoom over genomicRanges
myGRegions <- GRanges(seqnames = 10, ranges = IRanges(start=144605146-5e5, end=144605146+5e5))
manhattan_annot_plot(annotatedGWASresults = traitGWASresults_annotated,
  labelType = "composite", zoomGR = myGRegions)
```



Subset GWAS plot by regions

Make GenomicRanges object out of gwas results

Subset geneRanges with gwasRanges. Finding genes with SNPs in them

Subset GWAS hits by significance for plotting with genes.