# Phenome Wide Association Studies (PheWAS) in R

Robert J. Carroll
Department of Biomedical Informatics
Vanderbilt University School of Medicine
phewas@vanderbilt.edu

March 2, 2018

Packge **PheWAS** provides methods for the creation of PheWAS phenotypes, analysis, and plotting.While these methods are designed primarily for genetics based PheWAS analysis, they can perform GWAS or even phenotype only studies.

## 1 Data Input

There are many potential data sources and types; this necessitates that users handle the basic data i/o and formatting. Below are outlined some methods for importing common data into R.

### 1.1 Preparing plink data

Genome wide data is stored commonly stored in plink formats[1]. The simplest method to import data from plink is the `--recodeA` parameter in plink[2]. Running the following in a terminal will get one started:

```
plink --recodeA --bfile example_data --extract interesting_snps
--out r_genotypes
```

This will recode the binary plink data "example_data", extracting the SNPs under investigation to the file "r_genotypes.raw". This raw data can be loaded into R with a single command:

```
genotypes=read.table("r_genotypes.raw",header=TRUE)
```

Alternatively, assuming FIDs are unique, the following will load the data ready to be put into `phewas`.

```
> genotypes=read.table("r_genotypes.raw",header=TRUE)[,c(-2:-6)]
> names(genotypes)[1]="id"
```

### 1.2 Data from file

R has robust methods for loading data from files[3]. For this section we will consider two examples. The first is loading a csv file containg id, icd9, and count data as appropriate for a classic PheWAS.

*id.icd9.count.csv*:

[1] See `http://pngu.mgh.harvard.edu/~purcell/plink/data.shtml` for plink data format details.
[2] See `http://pngu.mgh.harvard.edu/~purcell/plink/dataman.shtml#recode` for details
[3] See `?read.table` in R for the read methods discussed here.

```
id,icd9,count
1,410,2
1,410.1,1
1,414.0,6
2,250.02,13
...
```

This can be loaded using `csv.phenotypes=read.csv("id.icd9.count.csv",colClasses=c("integer","character","integer"))` Pay special attention to the `colClasses` parameter: we need to ensure that the ICD9 codes are read as character strings so they do not lose trailing or leading zeros. This table is appropriate for use in `createPhewasTable`.

Another example is that the user may have exported their chart review data into a csv from a spreadsheet software.
*example_phenotype.csv*:

```
id,T2D,max.a1c
1,T,10
2,F,NA
3,F,6
...
```

This can be loaded using `csv.phenotypes=read.csv("example_phenotype.csv")`. This table loaded into R is ready to be used in `phewas`-either as covariates or phenotypes (outcomes).

### 1.3 Data from database

The **RODBC** library contains great tools for importing data directly from electronic data warehouses. If one desired to use PheWAS codes in their analysis from an ICD9 billing code table, it might look like the following.

```
> library(RODBC)
> connection=odbcConnect("MyDSN")
> icd9.codes=sqlQuery(connection,"select id, icd9, count(distinct date)
    from icd9_codes group by id, icd9;")
> odbcClose(connection)
```

The `icd9.codes` data frame is ready to be used with the `createPhewasTable` function.

## 2 Data Transformation

The primary data transformation for this package is to convert and aggregate ICD9 codes into PheWAS codes. The function `createPhewasTable` allows for this conversion. Given the database data loaded from the above section, one can use the following code to create PheWAS phenotypes for use in `phewas`:

```
> phenotypes=createPhewasTable(icd9.codes)
```

There are some additional options for PheWAS code translation. Users can opt to forgo exclusions using `add.exclusions=F`; this increases the size of the control population, but at the cost of including potentially similar diagnoses in the control sets. The `min.code.count` parameter allows users to alter the specificity of case selection. It can also be set to `NA` to allow for continuous outcomes, the code count sum by default.

# 3 Phenome Wide Association Studies

The `phewas` function performs the PheWAS itself. Using the examples from above, one can directly pass the parameters.

```
> results=phewas(phenotypes=phenotypes,genotypes=genotypes)
```

If one wishes to speed up the analysis, a multi-threaded approach is available using the base package **parallel**.

```
> results=phewas(phenotypes=phenotypes,genotypes=genotypes,cores=4)
```

One can additionally provide covariates. In this case, we will consider an analysis adjusted by max.a1c.

```
> results=phewas(phenotypes=phenotypes,genotypes=genotypes,
+    covariates=csv.phenotypes[,c("id","max.a1c")])
```

An alternate method is to use the `data` parameter with name vectors in the `phenotype`, `genotype`, and `covariates` parameters.

```
> mydata=merge(phenotypes,genotypes)
> results=phewas(phenotypes=names(phenotypes)[-1],genotypes=c("rs1234","rs5678"),
+    data=mydata)
```

The `phewas` function can be used for more than just generic PheWAS. In the following example, `outcomes` and `predictors` are used for a phenotype only analysis. Note that these parameters are simply alternate names for `phenotypes` and `genotypes`, respectively.

```
> max.a1c.results=phewas(outcomes=phenotypes,
+    predictors=csv.phenotypes[,c("id","max.a1c")])
```

The `phewasMeta` method can assist in meta-analysis of multiple PheWAS, e.g., if one has multiple genotype platforms of data to analyze. It wraps the `metagen` method of the **meta** package.

```
> results.omni1=phewas(phenotypes=phenotypes.omni1,genotypes=genotypes.omni1)
> results.omni1$study="Omni 1"
> results.omni.express=phewas(phenotypes=phenotypes.omni.express,
+    genotypes=genotypes.omni.express)
> results.omni.express$study="Omni Express"
> results.merged=rbind(results.omni1,results.omni.express)
> results.meta=phewasMeta(results.merged)
```

# 4 Plotting

Three methods for plotting data are included, `phewasManhattan`, `phenotypeManhattan`, and `phenotypePlot`, which wrap each other. `phewasManhattan` is the highest level method, and can plot PheWAS results directly from `phewas`.

```
> phewasManhattan(results)
```

This method returns a **ggplot2** object, which can be further manipulated using methods from that package[4]. The `...` parameter will pass further options into `phenotypeManhattan` and `phenotypePlot`. These lower level plot functions can be used in a stand-alone fashion for different types of data. For example, `phenotypePlot` can display information about the count for every individual of each ICD9 code.

[4] See `http://docs.ggplot2.org/current/` for the web documentation of **ggplot2**

```
> id.phenotype.value=icd9.codes
> names(id.phenotype.value)=c("id","phenotype","value")
> phenotypePlot(id.phenotype.value,use.color=F,x.group.labels=F)
```

# 5 Package Example

The following is the complete example from the **PheWAS** package.

```
> library(PheWAS)
> #Set the random seed so it is replicable
> set.seed(1)
> #Generate some example data
> ex=generateExample()
> #Extract the two parts from the returned list
> id.icd9.count=ex$id.icd9.count
> genotypes=ex$genotypes
> #Create the PheWAS code table- translates the icd9s, adds
> #exclusions, and reshapes to a wide format
> phenotypes=createPhewasTable(id.icd9.count)
> #Run the PheWAS
> results=phewas(phenotypes,genotypes,cores=1,
+   significance.threshold=c("bonferroni"))
> #Plot the results
> phewasManhattan(results, annotate.angle=0,
+   title="My Example PheWAS Manhattan Plot")
> #Add PheWAS descriptions
> results_d=addPhecodeInfo(results)
> #List the significant results
> results_d[results_d$bonferroni&!is.na(results_d$p),]

    phecode        description        group       snp adjustment      beta        SE       OR
216     335 Multiple sclerosis neurological rsEXAMPLE       <NA> 0.5194276 0.0661385 1.681065 4.04

> #List the top 10 results
> results_d[order(results_d$p)[1:10],]

     phecode                                                        description
216      335                                                Multiple sclerosis          neurolo
288    512.8                                                             Cough           respir
306    512.9                                                     Other dyspnea           respir
802      151                                                Cancer of stomach             neop
948      981 Toxic effect of (non-ethyl) alcohol and petroleum and other solvents injuries & poiso
165   327.71                                             Restless legs syndrome          neurolo
1279     557                                Intestinal malabsorption (non-celiac)             dige
1482  204.11                                          Lymphoid leukemia, acute             neop
936      809                                    Fracture of unspecified bones injuries & poiso
1465   352.2                                      Facial nerve disorders [CN7]          neurolo
     n_controls HWE_p allele_freq n_no_snp note bonferroni
216        2638     1   0.4969422        0            TRUE
```

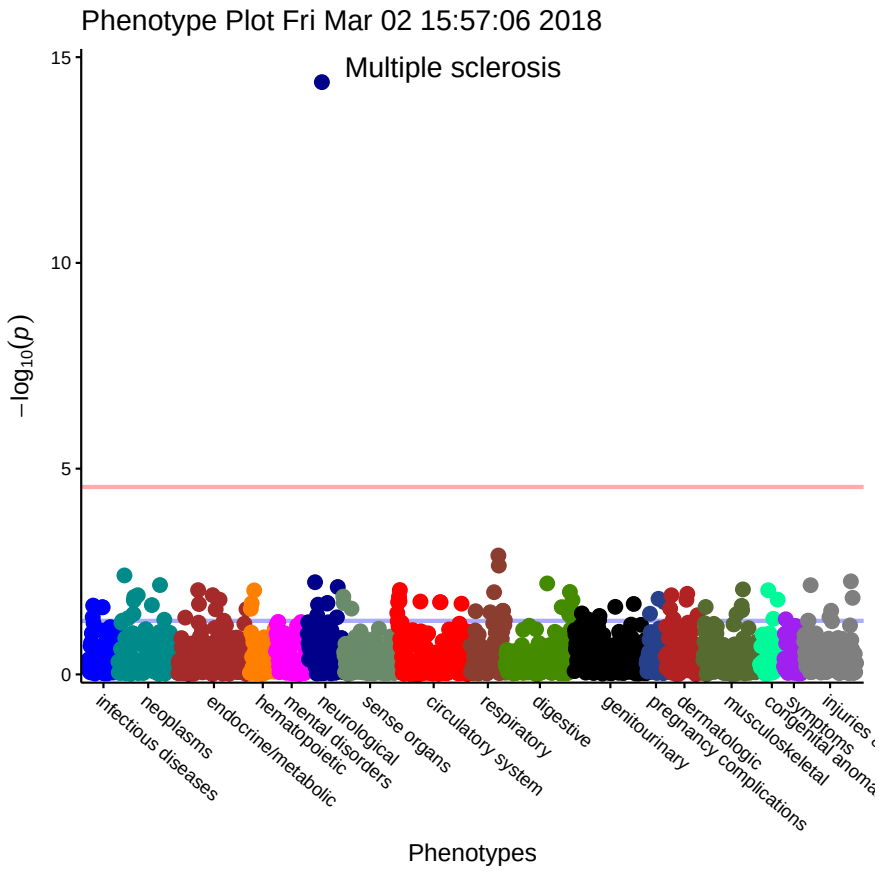


Figure 1: Example PheWAS Manhattan plot

| | | | | | |
|---|---|---|---|---|---|
| 288 | 4828 | 1 | 0.4957029 | 0 | FALSE |
| 306 | 4828 | 1 | 0.4958966 | 0 | FALSE |
| 802 | 4293 | 1 | 0.4960666 | 0 | FALSE |
| 948 | 4801 | 1 | 0.4960614 | 0 | FALSE |
| 165 | 4661 | 1 | 0.4953626 | 0 | FALSE |
| 1279 | 4409 | 1 | 0.4953850 | 0 | FALSE |
| 1482 | 4515 | 1 | 0.4960733 | 0 | FALSE |
| 936 | 4576 | 1 | 0.4943491 | 0 | FALSE |
| 1465 | 4747 | 1 | 0.4958368 | 0 | FALSE |

```
> phewasManhattan(results, annotate.angle=0)
```