**Quick Start Guide for QCEWAS**

Package version 1.2-3

Peter J. van der Most & Leanne Küpers, September-December 2016  
Updated January-February 2023

Welcome to QCEWAS, an R package that allows you to run easy and fast, automated quality control (QC) over epigenone-wide association study (EWAS) results. This Quick Start Guide will show you how to run a QC, and what you should be paying attention to when examining the results. It does not require any familiarity with R itself, beyond being able to install the package.

**The absolute basics: installing R and QCEWAS.**

R is both an environment and language for statistical computing and graphics. It’s freely available from the [Comprehensive R Archive Network](https://cran.r-project.org/). The base R program is very barebones: it consists of a command prompt, and a text editor to pass scripts to the command prompt. Installing an additional UI program such as [RStudio](https://www.rstudio.com/) provides a greater ease of use, but this is optional.

After installing R on a 64-bit computer (the example analyses in this guide is small enough to run on a 32-bit computer, but the sheer size of most real EWAS results files necessitates 64-bit), open the 64-bit version of R (R x64 4.x.x). Then select Packages > Install Package(s) from the menu. You will be prompted to select a download mirror, and then provided with a very long list of available packages. Select QCEWAS from this list (it’s ordered alphabetically), and you are set.

As noted, base R is very barebones. It’s operated by instructions typed into the command prompt. The R language is large and flexible, but you need only a few commands to use QCEWAS. One handy tip is to store the commands in a script file. You do this by selecting File > New script. This will open a text editor where you can type the commands. The commands are passed to the command prompt by selecting the text and pressing CTRL + R. Scripts can also be saved, so that you don’t have to retype the same commands over and over. (R saves script files with the .R extension; but they are simple text files and can be opened and edited by Windows Notepad and other basic text editors.)

**Running QCEWAS: a simple QC**

To use QCEWAS, open R and run the following command (the most convenient way to do this is by opening a new script file in R and copy & paste the commands to it. Then select the commands and press CTRL + R to run them):

library("QCEWAS")

This tells R to use the QCEWAS package. The next command will set a working directory, from which R will load the input files and where it will save the output. Change the command to point to an existing folder on your computer (preferably an empty one). Also note that R uses forward slash ( / ) where Windows uses backslash ( \ ).

setwd("C:/EWAS\_files/Example")

The next five command look very complex, but they are really just for copying 5 sample files from the package to your new working directory.

file.copy(from = file.path(system.file("extdata", package = "QCEWAS"), "sample1.txt.gz"),  
to = getwd(), overwrite = FALSE, recursive = FALSE)

file.copy(from = file.path(system.file("extdata", package = "QCEWAS"), "sample2.txt.gz"),  
to = getwd(), overwrite = FALSE, recursive = FALSE)

file.copy(from = file.path(system.file("extdata", package = "QCEWAS"), "sample\_map.txt.gz"), to = getwd(), overwrite = FALSE, recursive = FALSE)

file.copy(from = file.path(system.file("extdata", package = "QCEWAS"), "translation\_table.txt"), to = getwd(), overwrite = FALSE, recursive = FALSE)

file.copy(from = file.path(system.file("extdata", package = "QCEWAS"), "sample\_markers\_to\_exclude.txt "), to = getwd(), overwrite = FALSE, recursive = FALSE)

These five files are, respectively, two EWAS results files, a map containing the genetic position of the 3000 CpG locations in the sample files, a table for translating column names to the QCEWAS standard, and a list of markers. To run a QC over the first EWAS file, use the following command:

QC\_results <- EWAS\_QC(data = "sample1.txt.gz",  
outputname = "output\_A",  
save\_final\_dataset = FALSE)

This tells R to run the function EWAS\_QC() with 3 arguments. The argument ‘data’ tells R the name of the file to be QC’ed (enclosed in quotation marks), ‘outputname’ is the name of the resulting QC log and image files, and ‘save\_final\_dataset’ tells EWAS\_QC whether or not to save a copy of the cleaned data.

After running this command, there will be five new files in the working directory: ‘output\_A.log’, a histogram, a volcano plot, a QQ plot and a p plot. Let’s start by inspecting the log file. The log file is best viewed inside a spreadsheet program like Microsoft Excel, but it can be opened in any basic text editor like Windows Notepad. The log is divided in 3 sections. The top section provides administrative details about the QC. The middle section counts the number of markers inside the file, the number of problematic (duplicated markers; missing, invalid or extreme values, etc.) and the number of markers that passed QC. The right column is only of interest if you use a map or are filtering the dataset (which we haven’t done in this command). However, if there are many missing or invalid values in the left column (for example: standard errors of 0 or lower), it suggests something went wrong during the analysis.

The bottom section contains the meat of the QC. The bottom table gives a detailed description of the beta, standard error and p-values reported in the results file. Normally, you’d expect the beta to centre around zero, while the p-values are evenly distributed between 0 and 1. That is not the case here, but then we are using a very small dataset so deviations are to be expected. Extreme beta and standard error values are also to be expected in a large dataset, so don’t be too concerned about those, but do check whether the 25% and 75% values are reasonable for the phenotype.

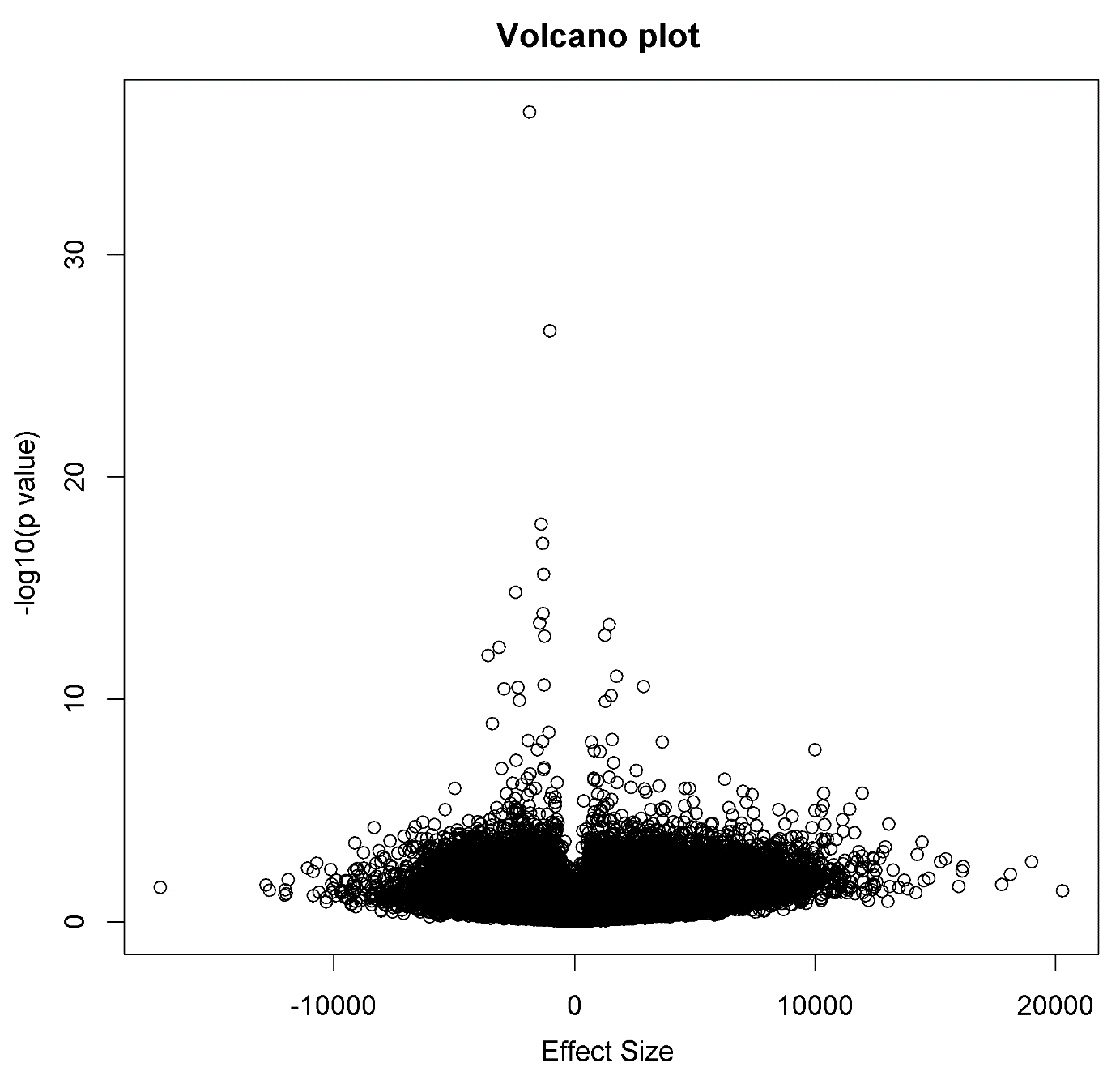
Just above the bottom table, the log file reports 3 values that summarize the quality of the data. The p-value correlation is an internal check to see if the reported p equals the p-value calculated from beta and standard error. For our dataset, the value is 1, as it should be. If it isn’t exactly 1, it suggests either a data-mix up or a formatting error occurred previously, and you need to check how the results file was created. (If the p-value correlation is close to 1, it may indicate that the reported values have been rounded after calculation of the p-value. This is not a serious problem, but it should be checked.)

The lambda value is a measurement of p-value inflation. In a genome-wide dataset (or any large dataset that isn’t enriched), lambda should be close to 1 (between 0.9 and 1.1). In our dataset, lambda is 1.3 - but we’ll explain what this means and how to interpret it below.

Finally, the SE median isn’t very informative by itself, but it’s used by QCEWAS to compare the results of multiple analysis; this will be explained below.

Now let’s examine the graphs produced by the QC. output\_A.histo.png is a histogram of the distribution of effect-sizes and standard errors, which conveys roughly the same information as the bottom table of the log file. output\_A.graph\_volcano.png is a volcano plot, which plots the significance (y axis) against the effect size (x axis). It is named as such because the expected distribution looks roughly like an erupting volcano. Unfortunately, our dataset is too small to generate a proper volcano plot, but an example is provided on the next page. Any extreme effect-size values (far left and far right of the plot) are almost certainly fluke outcomes with equally huge standard errors, and hence low significance. As effect-sizes become less extreme, significance rises, forming the left and right slope of the volcano. The crater at the centre is formed by markers with effect-sizes close to 0, and hence low significance. Finally, the significant or suggestive markers (if any) form the eruption plume.

*This is what a Volcano plot of a full dataset looks like*



As you can see from the graph, our sample dataset looks more like a geyser than a volcano. This means either that the dataset has been filtered (which should show on the other graphs), or the p-values don’t belong to the effect-sizes (in which case the p-value correlation should be below 1). In our case it’s the small size of the dataset that causes there to be no extreme values, hence the slopes of the volcano don’t show.

output\_A.graph\_QQ.png is a QQ plot of the p-values. Like the lambda value in the QC log, this is a test of p-value inflation. If the p-values are random (which we expect to be the case for the majority of markers), the p-values should follow the diagonal 1:1 line. Markers lying above the line are more significant than expected based on random chance. If this was a genome-wide dataset, we’d expect only a subset of markers (the significant hits we found) to be above the line. If most of the dataset is above the line, our p-values are inflated and something is amiss with our model. As our lambda value of 1.3 suggested, this is the case for our dataset as well. However, since we only have 2990 CpG sites, we don’t expect a perfectly random distribution of p, so for our data the QQ plot looks well enough.

Finally output\_A.graph\_p.png plots the correlation between reported p-values and p-values calculated from the reported effect size and standard error. It conveys the same information of the p-value correlation statistic in the log file, just in graphical form.

**Advanced options for running a single QC**

Now that we’ve explained the basics, let’s look at a few more advanced options.

QC\_results <- EWAS\_QC(data = "sample1.txt.gz",  
map = "sample\_map.txt.gz",  
outputname = "output\_B",  
threshold\_outliers = c(-20, 20),  
exclude\_outliers = FALSE,  
exclude\_X = TRUE, exclude\_Y = FALSE,  
save\_final\_dataset = TRUE,  
gzip\_final\_dataset = FALSE)

The above command executes the function EWAS\_QC with 9 arguments. It QC’s the same sample file, but this time uses ‘output\_B’ as the name for the output files. It also loads a map file for generating a Manhattan plot and for identifying allosomal markers. The ‘threshold\_outliers’ argument tells EWAS\_QC that markers with a beta value below -20 or above 20 are considered outliers. The ‘exclude\_outliers’ indicates whether or not such outliers should be excluded. There are also options for excluding X-chromosomal and Y-chromosomal markers. ‘save\_final\_dataset’ is now changed to TRUE, while the additional argument ‘gzip\_final\_dataset’ tells EWAS\_QC whether this final dataset is to be compressed with the gzip format.

Running this command produces the same output as the earlier QC, with two additional files: a Manhattan plot (output\_B.graph\_M.png) and a post-QC (‘final’) dataset (output\_B.txt). Let’s start by looking at the log file first. As it processed the same results file, the bottom table is not going to differ. However, the right column of the middle table now includes counts of allosomal markers (0) and outliers (309). However, since ‘exclude\_ouliers’ was set to FALSE, none of these have been removed. The function was told to remove X-chromosomal markers, but since there are none present nothing has happened.

Note that the choice of -20 / + 20 for outliers was arbitrary. It will depend on what phenotype you are analysing; and is mostly intended to filter out results that are clearly false. If large numbers of markers are removed as outliers, it suggests that your unit of measurement isn’t what you think it is.

Now take a look at ‘output\_B.graph\_M.png’. This is a Manhattan plot, made using the map file, showing where the significant hits, if any, are. Our sample dataset only contains chromosome 2 markers, so that plot is as it should be. The horizontal dotted red line indicates the Bonferonni-corrected significance threshold. As you can see, there are no markers exceeding it, though one gets close.

**Generating a map file**

The Manhattan plot and allosomal exclusion options mentioned in the above section require a map file: a file containing the probe ID of the included samples, as well as their chromosome and genetic (basepair) location. As there are various methylation arrays available, and we wanted to keep the size of the package down, we did not include a full map. The map file is entirely optional – but if you do want to generate a Manhattan plot (or exclude allosomal variants), then this section will explain how.

Firstly, you will need a manifest file from the particular chipset that you are using. These are usually included with the documentation of the chipset. Alternatively, Illumina lists many of their legacy arrays on their website (for example: [Infinium MethylationEPIC v1.0 Product Files (illumina.com)](https://emea.support.illumina.com/downloads/infinium-methylationepic-v1-0-product-files.html)).

Download the relevant manifest file in a format that can be read with R. In this example, I will be using the EPIC v1.0 B5 Manifest (infinium-methylationepic-v-1-0-b5-manifest-file.csv) as an example. Because the manifest is in .csv format, the fastest way to import it is using the read.csv() function. Plain text files can be read using the read.table() command.

This particular manifest has 7 lines of unnecessary information (I found this out through using the head command in UNIX), so we’re skipping 7 lines when loading.

manifest <- read.csv("infinium-methylationepic-v-1-0-b5-manifest-file.csv", skip = 7)

This will take a minute or so even on a cluster computer. Next, inspect the dataset using:

str(manifest)

The information we need is the probeID and the chromosome and position of said ID. For this file, it means we only need the following columns:

manifest <- manifest[ , c("IlmnID", "CHR", "MAPINFO")]

QCEWAS requires that the mapfile uses 3 specific column names: TARGETID, CHR and MAPINFO. To change the columnnames in R, use:

colnames(manifest) <- c("TARGETID", "CHR", "MAPINFO")

Next, we do a check of the column contents

table(manifest$CHR, useNA = "always")

This shows that there are 695 probes without a chromosome value. There is no point in including these in a map file, so let’s remove them:

manifest <- manifest[manifest$CHR != "", ]

Finally, save the dataset so that it can be used for later analyses:

write.table(manifest, gzfile("mapfile\_EPIC\_v1-0-b5.txt.gz"),  
row.names = F, quote = F, sep = "\t")

The gzfile function will compress the file into .gz format, making it of a more manageable size. To use the map file in the QC, enter the filename into the map argument.

In the event that you are meta-analysing EWAS results generated using multiple platforms, generating a map file that will cover everything will be a bit more complicated. Briefly, you will have to generate separate maps (as above) for each chipset (named e.g. manifest1, manifest2, manifest3), and then merge these in such a way that there are no duplicate probe IDs. In the above example, and assuming that manifest1 is the most recently updated file (i.e. its position information is the most reliable):

manifest2 <- manifest2[!manifest2$TARGETID %in% manifest1$TARGETID, ]

This will remove probes from manifest 2 if they are also in manifest 1.

manifestMerged <- rbind(manifest1, manifest2)

Now repeat the process for manifest 3

manifest3 <- manifest3[!manifest3$TARGETID %in% manifestMerged$TARGETID, ]

manifestMerged <- rbind(manifestMerged, manifest3)

And save the result:

write.table(manifestMerged, gzfile("mapfile\_MergedChipsets.txt.gz"),  
row.names = F, quote = F, sep = "\t")

**Additional options**

There are a couple of additional options for EWAS\_QC that should be highlighted.

QC\_results <- EWAS\_QC(data = "sample1.txt.gz",  
outputname = "output\_C",  
save\_final\_dataset = FALSE,  
markers\_to\_exclude = "sample\_markers\_to\_exclude.txt",  
high\_quality\_plots = TRUE)

The above command has 2 new arguments. ‘markers\_to\_exclude’ reads in a file (it also accepts data frames and vectors from inside R) with marker IDs that will be removed from the dataset. This is useful for removing markers whose probes are (suspected) aspecific. The other option, ‘high\_quality\_plots’, will produce high resolution (350 dpi) tiff files (instead of the standard .png files) for use in publications.

One restriction of QCEWAS is that requires the EWAS results file to use standard column names (PROBEID, BETA, SE, P\_VAL). However, if file uses a different names, you can use a translation table to allow QCEWAS to process it anyway. For example, the second sample file (‘sample2.txt’) uses id, effect, stderr, and p as column names. The sample data also includes a ‘translation\_table.txt’, which contains 2 columns. On the left are the standard column names, and on the right are the alternatives (in capital letters). The names in the alternative column *must* be capitalized. However, you can specify multiple alternatives for each standard name, as shown for PROBEID.

If EWAS\_QC saves the final, post-QC dataset, it will use the standardized column names rather than the originals. However, you can use the ‘header\_final\_dataset’ argument to return them to the original:

QC\_results <- EWAS\_QC(data = "sample2.txt.gz",  
outputname = "output\_D",  
save\_final\_dataset = TRUE,  
gzip\_final\_dataset = FALSE,  
header\_translations = "translation\_table.txt",  
header\_final\_dataset = "original")

Other options for ‘header\_final\_dataset’ are ‘PLINK’, ‘GWAMA’ and ‘META’, for column names used in those programs.

**QC’ing and comparing multiple files**

Now that we’ve analysed a single file, we are going to try QC’ing multiple files at once. First run these two commands:

sample\_files <- c("sample1.txt.gz", "sample2.txt.gz")

sample\_N <- data.frame(file = sample\_files,  
N = c(77, 79),  
stringsAsFactors = FALSE)

The first command saves the names of the files in the R memory under the name ‘sample\_files’. The second command creates a table (technically: a data frame) that tells QCEWAS the sample size (the number of subjects) of the results files that we will QC. This argument is optional, but adding sample sizes allows QCEWAS to generate an additional graph. To run the actual QC, use the EWAS\_series command:

QC\_results <- EWAS\_series(EWAS\_files = sample\_files,  
output\_files = c("output\_E1", "output\_E2"),  
map = "sample\_map.txt.gz",  
N = sample\_N,  
header\_translations = "translation\_table.txt",  
save\_final\_dataset = FALSE,  
threshold\_outliers = c(-20, 20),  
exclude\_outliers = TRUE)

EWAS\_series uses most of the arguments that EWAS\_QC does, with 2 exceptions. The data argument is replaced by EWAS\_files, and outputname is replaced by output\_files, because EWAS\_series accepts multiple file names. There’s also a new argument N for the sample-size table.

The EWAS\_series command essentially runs EWAS\_QC for each file name passed to it. It produces the same QC output files as did our previous QC for both sample files. However, it also generates two additional plots and a legend file.

The EWAS\_QC\_graph\_effectsize.png is a box-and-whiskers plot of the effect-size distribution in the datasets. The distribution of effect-sizes should be similar between cohorts. If it is not, this indicates that one of the results actually measured a different phenotype (this could results from a different unit of measurement, for example mg/dl instead of mmol/l, or because the phenotype was transformed during the analysis). That said, the range of effect-sizes will shrink with increasing sample size because the increased power will result in fewer larger values. Cohorts are ordered by sample size, with the largest cohorts (and hence the smallest expected range in effect size) on the right (for our dataset the difference in size is too small to show). To save space, the datasets are numbered rather than named. You can find the numbers in the legend file.

The other graph is a precision plot: a plot of the square root of the sample size against the inverse of the median standard error. The precision should increase along with sample size. As we only have 2 datasets with near identical sample sizes, this plot is not very informative. However, if you are analysing 8 or more datasets with a wider spread of sample sizes, this plot can identify outliers: cohorts that have far lower or higher variance than expected. One cannot determine from the plot what causes an outlier, or whether one should exclude it (it could be due to a specific population or differences in measurement), but you should pay attention to outlying values if these are really out of line.

**Final notes**

This explains the basic functionality of EWAS\_QC and EWAS\_series. The package includes a few other functions, but this is all you need to run a QC. In case you want more information on a function of R (and this applies not just to functions of the QCEWAS package), you can obtain it by typing a question mark followed by the name of the function into the command prompt:

?EWAS\_QC

This will open the internal R documentation, which gives a detailed overview of the input (arguments), output and related functions.

We hope you found this helpful, and wish you best of luck with your analysis.