

# RHAPSODY WP3 Pre-Diabetes Analysis Plan

(Version 1.1.0)

*Mickaël Canouil, Ph.D.*

*26 December 2018*

## Files description

At this stage, you were provided with the following files from the (hidden) Gist (don't share outside RHAPSODY):

- `README.Rmd/README.pdf`, the file you are actually reading;
- `RHAPSODY_WP3_PreDiab.Rmd`, a Rmarkdown script which perform:
  - opal node access,
  - phenotype QC,
  - preliminary modelling,
  - VCFs formatting/filtering,
  - variants analysis;
- `opal_credentials.txt`, the server, login and password informations to access phenotype data on your local node;

Utils files, can be downloaded from the following (hidden) Gist:

- `./utils/RHAPSODY_WP3_PreDiab_DEBUG.R`, a small part of the main Rmarkdown script, which includes only node access and phenotype QC;
- `./utils/RunAnalysis.R`, a example of R script to run the whole analysis in bash using `Rscript` (parameters have to be filled properly according to your informations);
- `./utils/handleVCF.sh`, the shell script used to format/filter VCFs files prior to the variant analysis. This shell script is also included within the main script, but you can use it directly as a standalone, if so (or if your already done formatting and don't want to do it again), parameter `format_vcfs` must be set to `FALSE`).
- `./utils/Install_Rpackages.R`, a R script which install a predefined version of all R packages loaded in the Rmarkdown script, only if needed (also available within the main script).

# Run the analysis

## R Setup

Please use R version 3.4.2 (2017-09-28) (if not possible, newer version "3.4.\*" can be used).

If VCFtools is not available on your machine, please install it: VCFtools.

Be sure the needed packages are installed (this is also done within the Rmarkdown script) or copy/paste the following command.

```
# Install R packages and load them
utils::install.packages(
  pkgs = c('udunits2', 'units', 'devtools', 'caTools'),
  configure.args = '--with-udunits2-lib=/usr/local/lib',
  quiet = TRUE,
  repos = c('http://cran.us.r-project.org', 'https://cran.rstudio.com/')
)
utils::install.packages(
  pkgs = 'opal',
  repos = c('https://cran.obiba.org', 'https://cran.rstudio.com/'),
  dependencies = TRUE,
  quiet = TRUE
)
list_packages <- read.table(
  col.names = c('package', 'version'),
  text = "'acepack' '1.4.1'
'assertthat' '0.2.0'
'backports' '1.1.1'
'base64enc' '0.1-3'
'BH' '1.66.0-1'
'bindr' '0.1.1'
'plogr' '0.2.0'
'bindrcpp' '0.2.2'
'bitops' '1.0-6'
'brew' '1.0-6'
'foreign' '0.8-69'
'mnormt' '1.5-5'
'lattice' '0.20-35'
'nlme' '3.1-131'
'psych' '1.7.8'
'plyr' '1.8.4'
'pkgconfig' '2.0.1'
'utf8' '1.1.3'
'pillar' '1.2.2'
'tibble' '1.4.2'
'purrr' '0.2.4'
'tidysselect' '0.2.4'
'dplyr' '0.7.5'
'reshape2' '1.4.3'
'tidyr' '0.8.1'
'broom' '0.4.4'
'callr' '1.0.0'
'rematch' '1.0.1'
'cellranger' '1.1.0'
'checkmate' '1.8.5'
```

```
'cli' '1.0.0'
'cluster' '2.0.6'
'colorspace' '1.3-2'
'compiler' '3.4.2'
'RColorBrewer' '1.1-2'
'dichromat' '2.0-0'
'munsell' '0.4.3'
'labeling' '0.3'
'viridisLite' '0.3.0'
'scales' '0.5.0'
'gtable' '0.2.0'
'lazyeval' '0.2.1'
'ggplot2' '2.2.1'
'cowplot' '0.9.1'
'crayon' '1.3.4'
'curl' '3.0'
'data.table' '1.10.4-3'
'downloader' '0.4'
'htmlwidgets' '0.9'
'Matrix' '1.2-12'
'irlba' '2.3.1'
'igraph' '1.1.2'
'influenceR' '0.1.0'
'Rook' '1.1-1'
'XML' '3.98-1.9'
'rgexf' '0.15.3'
'gridExtra' '2.3'
'viridis' '0.5.1'
'visNetwork' '2.0.1'
'hms' '0.4.2'
'readr' '1.1.1'
'DiagrammeR' '0.9.2'
'data.tree' '0.7.4'
'DBI' '0.7'
'dbplyr' '1.2.1'
'devtools' '1.13.5'
'digest' '0.6.15'
'evaluate' '0.10.1'
'forcats' '0.3.0'
'Formula' '1.2-2'
'git2r' '0.19.0'
'glue' '1.2.0'
'graphics' '3.4.2'
'grDevices' '3.4.2'
'grid' '3.4.2'
'haven' '1.1.1'
'highr' '0.6'
'rpart' '4.1-11'
'latticeExtra' '0.6-28'
'survival' '2.41-3'
'nnet' '7.3-12'
'htmlTable' '1.11.0'
'Hmisc' '4.1-1'
```

```

'htmltools'    '0.3.6'
'htr'          '1.3.1'
'jsonlite'     '1.5'
'xml2'         '1.2.0'
'selectr'     '0.3-1'
'rvest'        '0.3.2'
'kableExtra'   '0.6.1'
'knitr'        '1.20'
'minqa'        '1.2.4'
'nloptr'       '1.0.4'
'RcppEigen'    '0.3.3.3.1'
'lme4'         '1.1-14'
'numDeriv'     '2016.8-1'
'lmerTest'     '3.0-1'
'lubridate'    '1.7.4'
'magrittr'     '1.5'
'markdown'     '0.8'
'MASS'         '7.3-47'
'memoise'      '1.1.0'
'methods'      '3.4.2'
'mgcv'         '1.8-22'
'mime'         '0.5'
'modelr'       '0.1.1'
'openssl'     '0.9.9'
'parallel'     '3.4.2'
'R6'           '2.2.2'
'Rcpp'         '0.12.17'
'RCurl'        '1.95-4.8'
'readxl'       '1.1.0'
'reprex'       '0.1.2'
'rjson'        '0.2.15'
'rlang'        '0.2.1'
'rmarkdown'    '1.9'
'rprojroot'    '1.2'
'rstudioapi'   '0.7'
'splines'      '3.4.2'
'stats'        '3.4.2'
'stringi'      '1.2.3'
'stringr'      '1.3.1'
'tools'        '3.4.2'
'utils'        '3.4.2'
'whisker'      '0.3-2'
'withr'        '2.1.2'
'writexl'      '0.2'
'yaml'         '2.1.15'
'tidyverse'    '1.2.1'
'ggsignif'     '0.4.0'
'ggrepel'      '0.8.0'
'ggsci'        '2.9'
'polynom'      '1.3-9'
'ggpubr'       '0.2'
)
list_packages[, 'local_version'] <- sapply(list_packages[, 'package'], utils::packageDescription, field=

```

```

list_packages[, 'local_version'] <- ifelse(is.na(list_packages[, 'local_version']), '', list_packages[,
list_packages <- list_packages[list_packages[, 'local_version']!=list_packages[, 'version'], ]
maply(
  x = list_packages[, 'package'],
  y = list_packages[, 'version'],
  FUN = function(x, y) {
    devtools::install_version(
      package = x,
      version = as.character(y),
      repos = c('http://cran.us.r-project.org', 'https://cran.rstudio.com/'),
      dependencies = FALSE,
      quiet = FALSE,
      upgrade = "never"
    )
  }
)

list_packages <- c(
  'devtools', 'parallel', 'grid', 'scales',
  'broom', 'viridis', 'readxl', 'writexl',
  'cowplot', 'knitr', 'kableExtra', 'lme4',
  'lmerTest', 'Hmisc', 'data.tree', 'opal'
)
invisible(sapply(X = list_packages, FUN = library, character.only = TRUE, quietly = TRUE, warn.conflict

```

## Using Docker

If you are familiar with Docker, you can find an image which include everything you need to run the analyses.

### Run the docker container with the latest image

To run the image, you can use the following command (need docker to be installed):

```
docker run \  
  --name rhapsody \  
  --hostname rhapsody \  
  --detach \  
  --volume /your_local_volume:/media/remote_volume \  
  --publish 8787:8787 \  
  --publish 22:22 \  
  biostatgood/rhapsody:latest
```

"/your\_local\_volume" is the path to the volume (where your data are located) on the machine when you run the docker software.

"/media/remote\_volume" will be the path within the docker container (keep /media/ as root directory).

### Connect to the docker container

Login and password have been set respectively to `rhapsody` and `wp3` (`rhapsody` is root via `sudo`). You can connect through SSH using standard IP address: `ssh rhapsody@localhost` (equivalent to `ssh rhapsody@127.0.0.1`).

You can also connect through Rstudio using a web browser at `http://localhost:8787/` (equivalent to `http://127.0.0.1:8787/`).

Once you are connected directly by SSH or with Rstudio, you can go to the bash terminal and change the password using `passwd` (just follow the instructions).

### Docker configuration

The (hidden) Gist s cloned in `/home/rhapsody/WP3/scripts`.

Make sure after starting the Docker image to update the scripts, if necessary.

```
cd /home/rhapsody/WP3/scripts  
git pull origin master
```

VCfTools is also installed in its latest version using github repository.

vcftoolsbinary is located in the default directory: `/usr/local/bin` (default path in the Rmarkdown scripts).

## Run the Rmarkdown script

1. First, create (or change) the `opal_credentials.txt` with your informations.
2. Fill in the proper informations.
3. Set the `analysis_step`:
  1. Node access
  2. Phenotype QC
  3. Run Mixed Model without variants
  4. Check VCFs files
  5. Format VCFs files
  6. Run Mixed Model with (imputed) variants
  7. Check output files
4. Run the R code below within interactive R session.
5. a/ If everything worked, you should have a HTML report.  
Increment `analysis_step` and repeat steps 3 to 5.  
b/ If something went wrong, you should have an error message, for example:

```
label: list_vcf (with options)
List of 2
 $ eval  : language params_steps$step_5
 $ engine: chr "sh"

Quitting from lines 1597-1598 (RHAPSODY_WP3_PreDiab.Rmd)
Error in if (options$eval) { : argument is of length zero
```

Either try to solve the issue yourself and/or send the informations to [mickael.canouil@cnrs.fr](mailto:mickael.canouil@cnrs.fr)

```

# set the output directory or leave as is;
# output will be generated where the Rmarkdown file is
working_directory <- './'

analysis_step <- 7
cohort_name <- 'Cohort_Name'
author_name <- 'Firstname LASTNAME'
opal_credentials <- 'opal_credentials.txt'
vcf_directory <- '/media/.../vcf'
imputation_quality_tag <- 'INFO' # To be set according to VCF (could also be 'R2')

# Run the analysis
dir.create(path = working_directory, showWarnings = FALSE, mode = '0777')
rmarkdown::render(
  input = '/home/rhapsody/WP3/scripts/RHAPSODY_WP3_PreDiab.Rmd',
  output_format = 'html_document',
  output_file = paste0(
    'RHAPSODY_WP3_PreDiab_',
    cohort_name, '_step',
    analysis_step, '.html'
  ),
  output_dir = working_directory,
  params = list(
    cohort_name = cohort_name,
    author_name = author_name,
    opal_credentials = opal_credentials,
    vcf_input_directory = vcf_directory,
    imputation_quality_tag = imputation_quality_tag,
    vcftools_binary_path = '/usr/local/bin',
    output_directory = working_directory,
    analysis_step = analysis_step,
    format_vcfs = TRUE,
    variants_analysis = TRUE,
    chunk_size = 1000,
    n_cpu = 2,
    echo = FALSE, # Should R code be printed in the report
    warning = FALSE, # Should warnings be printed in the report
    message = FALSE # Should messages be printed in the report
  ),
  encoding = 'UTF-8'
)

```