PhenoMan Documentation and Tutorial

Biao Li

Last updated: November 7, 2013

Contents

Chapter 1

Getting Started with PhenoMan

1.1 Citing PhenoMan

If you use `PhenoMan` in any published work, please cite both the software (as an electronic resource/URL) and the manuscript that describes the methods.

- Package: PhenoMan
- Authors: Biao Li, Gao Wang and Suzanne M. Leal
- License: GNU General Public License (http://www.gnu.org/license/)
- URL: http://code.google.com/p/phenoman/
- Manuscript: Bioinformatics ...

1.1.1 Reporting problems, bugs and questions

If you have any problems using `PhenoMan` or would like to report a bug, please follow these steps:

When `PhenoMan` does not properly generate figures or intermediate/final cleaned datasets, or when `PhenoMan` is applied on the same dataset but seemingly gives different answers at different times, etc, please feel free to contact me:

`biaol` AT `bcm` DOT `edu`

but also please consider the following before doing so:

- Please go through cookbook examples (found in the Chapter - **A Quick PhenoMan Tutorial**) to get more familiar with `PhenoMan` commands if you have not done so yet.
- Please check the screen output or the LOG file, which may contain important ERROR information. Frequently, it need re-specify command arguments and/or options according to specific `PhenoMan` requirements.
- Please check the format of your input data file: Is it plain text and in **dbGaP** phenotype format? Does each row have correct number of values, which should be equal to number of columns? Are columns delimited by **tabs**? Are missing values coded appropriately by `nan` (or `NA`, or `none`) as required? etc..

- Please check PhenoMan website if PhenoMan documentation or the program itself has been updated, sometimes the syntax of an option may change.
- If the above steps do not resolve your problem, please send an email preferably with the following specific information:
  - The complete LOG file or screen ERROR messages
  - The type of machine you were using
  - Versions of Python and R installed
  - Ideally, please try to make some reduced phenotype dataset that replicates the problem, which can be zipped and sent as an attachment; any data sent to me for the purpose of debugging will be immediately deleted after that problem is resolved.

Important

We are willing and able to advise on the use of specific features implemented in PhenoMan, to diagnose whether they are working as intended and to give a generic description of a procedure or method, if it is unclear from reading the documentation and tutorial.

The remaining contents of this chapter contains important information regarding how to set up and use PhenoMan. Individuals familiar with using command line programs can probably skip them.

1.2 Prerequisites of Using PhenoMan

PhenoMan does require installation of dependency software and possibly need pre-handling of initial phenotype datasets to meet the requirement of input data format.

- Make sure that **Python 2.7 or above** [1] (**Python 3.2 or above** [2] preferred), **R** [3] and **ggplot2 0.9 or above** (a R package) have been installed.
- Initial input data must be in **dbGaP** phenotype data format [4], where each row represents an individual and each column represents a trait.

Tip

Values of each trait can be quantitative, qualitative or text-based strings.

- The input data must be **Tab** delimited and starts with header (first row) including names of all traits/phenotypes.
- The first column must be sample names of individuals and its header info must be named by `sample_name`.
- Missing values must be coded/represented by **nan** (or **NA** or **none**) and cannot be left blank in the data file.

1.3 Download

PhenoMan is available for free download. Below is the link to ZIP file containing Python/R source code, http://phenoman.googlecc
src-0.1.0.tar.gz Linux/Mac/Windows users should download the source code and compile (see notes below).

Note

- This release is considered a stable release, although please remember that we cannot guarantee that it, just like most computer programs, does not contain bugs...
- If you download PhenoMan please drop an email libiaospe AT gmail DOT com letting me know that you have downloaded a copy.

1.4 Installation

PhenoMan is distributed as Python/R source code, which user can compile for your particular system using standard Python compiler.

1.4.1 Dependencies

PhenoMan requires installing **Python 2.7+** (**Python 3.2+** preferred), **R** and **ggplot2 0.9+** (a R package) first.

For example, in Ubuntu-like system run the following commands:

```
sudo apt-get install python3
sudo apt-get install r-base r-base-dev python3
sudo R
install.packages("ggplot2")
```

Important

For any other type of operating system, please refer to online tutorial or your system administrator to have dependency software properly installed and make sure that **Python** and **R** are in your system **PATH**.

1.4.2 Compilation

Download the .tar.gz file and perform the following steps (for Linux-like systems):

```
tar -xvzf phenoman-src.tar.gz
cd phenoman-src
sudo python3 setup.py install
```

Note

For those who may not have Admin privileges please refer to the following steps to install PhenoMan locally:

- Install PhenoMan on a local directory

BASH
```
python3 setup.py install --prefix=/path/to/your/local/folder
```

- Set environmental variables by opening ~/.bash_profile (if it does not exist, you can create it) and add the following:

BASH
```
PATH=/path/to/your/local/folder/bin:${PATH}
export PATH
PYTHONPATH=/path/to/your/local/folder/lib/pythonx.x/site-packages:$PYTHONPATH
export PYTHONPATH
```

1.4.3 Running PhenoMan from the command line

A typical session might involve running several commands, e.g. to view help info of each `PhenoMan` command, to produce summary statistics on phenotypes, to view distributions of quantitative traits, to remove/winsorize outliers, to select/remove samples based on given criteria, to control for covariates, etc. Each command involves a separate instantiation of `PhenoMan`.

Note

`PhenoMan` does not remember any parameter settings between different runs or store any other information.

Chapter 2

A Quick PhenoMan Tutorial

In this tutorial, we will consider using PhenoMan to clean example data: phenotypes of randomly selected individuals. As examples, we will walk through applying PhenoMan on three primary traits of interest, one quantitative, one case-control (qualitative), and one extreme quantitative trait using a variety of features: data exploration, sampling, management, summary statistics, quality control, covariates selection, etc., where, of course, quality control is the most important feature of PhenoMan.

Note

These data do not represent any realistic study design or realistic collection of phenotype information. The point of this tutorial is simply to get used to running PhenoMan.

2.1 Preparation

The first step is to obtain a copy of PhenoMan.

- Make sure you have PhenoMan successfully installed on your machine (See **Installation** in the previous chapter).
- Download the example data 'example_data.tar.gz', which is in **dbGaP** format and contains phenotype information of randomly generated individuals.
- Create a new folder/directory on your machine, and unzip the file you downloaded into this folder. There you should see 'AA.txt', 'EA.txt', 'AAdups.txt' and 'EAdups.txt'.

2.2 Use PhenoMan on a Quantitative Trait

Note

This tutorial is intended to introduce some of PhenoMan's features rather than provide exhaustive coverage of them. Furthermore, it is not intended as a study/analysis plan for quality control procedures of any primary trait of interest, or to represent anything close to 'best practice'.

2.2.1 Getting started

Just typing `phenoman -h` and specifying no further options is to list available modules and brief description of each module's functionality.

To view detailed help info of each module, also type the name of the module after `phenoman`, e.g. `phenoman show -h`

As an example for a quantitative trait, we choose 'Adiponectin'. It's corresponded field name in the example dataset is `adiponectin`.

Run >> `phenoman show fields --samples AA.txt | grep adiponectin` to check if the primary trait 'adiponectin' is contained in the data.

Warning

All the following data cleaning steps need to be applied separately on different ethnic groups, e.g. **AA** (African Americans) and **EA** (European Americans) as the example shows. We use 'AA.txt' in this tutorial. In practice, if the original data set contains individuals from multiple ethnic backgrounds users should have them separated into different data sets to avoid population stratification problem.

Tip

Use command `phenoman select` to sample subpopulations from the original data set. See details about `--removethese` and `--keepthese` by running `phenoman select -h`.

2.2.2 Remove missingness and duplicates

First run `phenoman cook` command to remove individuals that have missing values on the primary trait and are duplicates or related individuals. Save remained ones into file 'AA_adi.txt'.

```
phenoman cook adiponectin --samples AA.txt --duplicates AAdups.txt --output AA_adi.txt
```

OUTPUT

```
There are 1575 individuals that are missing values on phenotype adiponectin
There are 10 individuals that are duplicates and have been removed
```

The `--samples` option takes a single file as the input data. The `--duplicates` option may take one or multiple files. Each of them must contain two columns of sample names, where on each row the first individual is to be removed if it co-exists with the second one in the data. In practice, users should create file(s) of duplicates in advance and be able to use them at this step. The `--output` option takes a single file name as the output data file name.

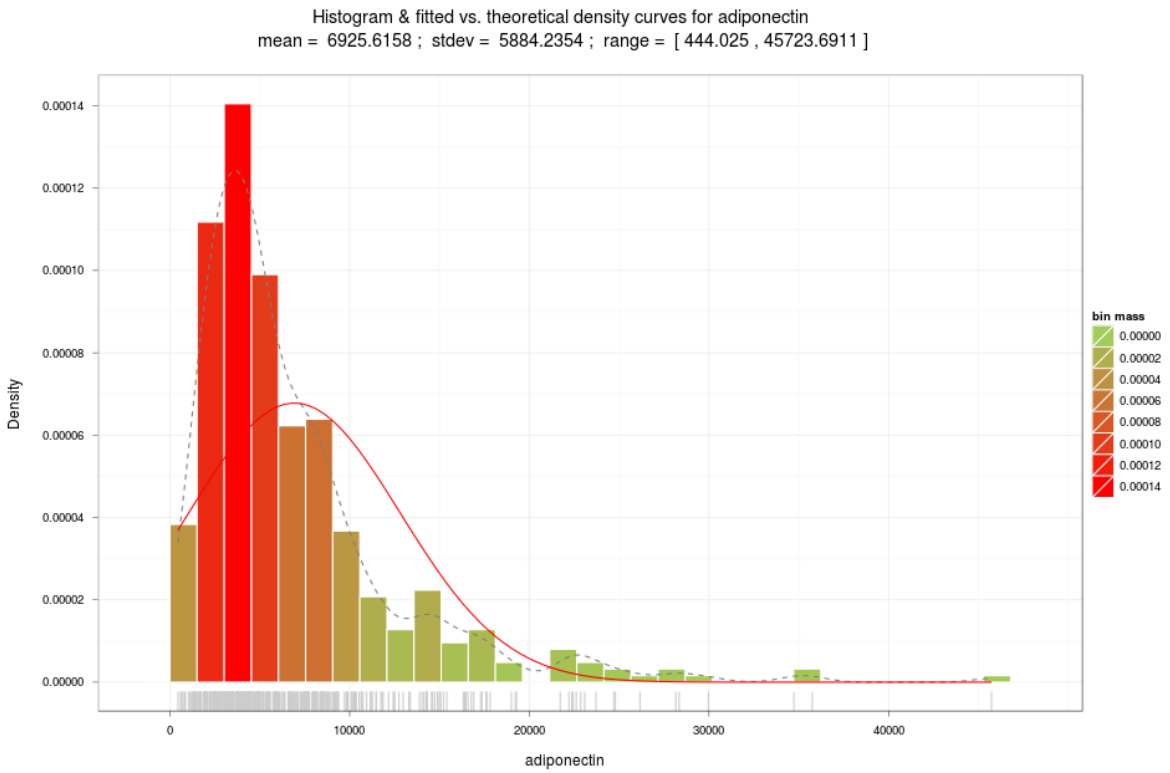
Tip

If you wish to remove N individuals while keeping other M individuals (as if N individuals are duplicates of the M individuals) regardless of whether those individuals co-exist in the dataset or not, you may use `phenoman select --keepthese --removethese` (see `phenoman select -h` or chapters of **Reference** and **Command Examples** for details) other than `phenoman cook --duplicates` so as to end efforts to create the two-columned file(s) required by the `--duplicates` option.

2.2.3 Detect outliers

1. Draw histogram of phenotype data using `phenoman view` to check its normality.

```
phenoman view adiponectin --samples AA_adi.txt
eog adiponectin_histogram_AA_adi.png
```

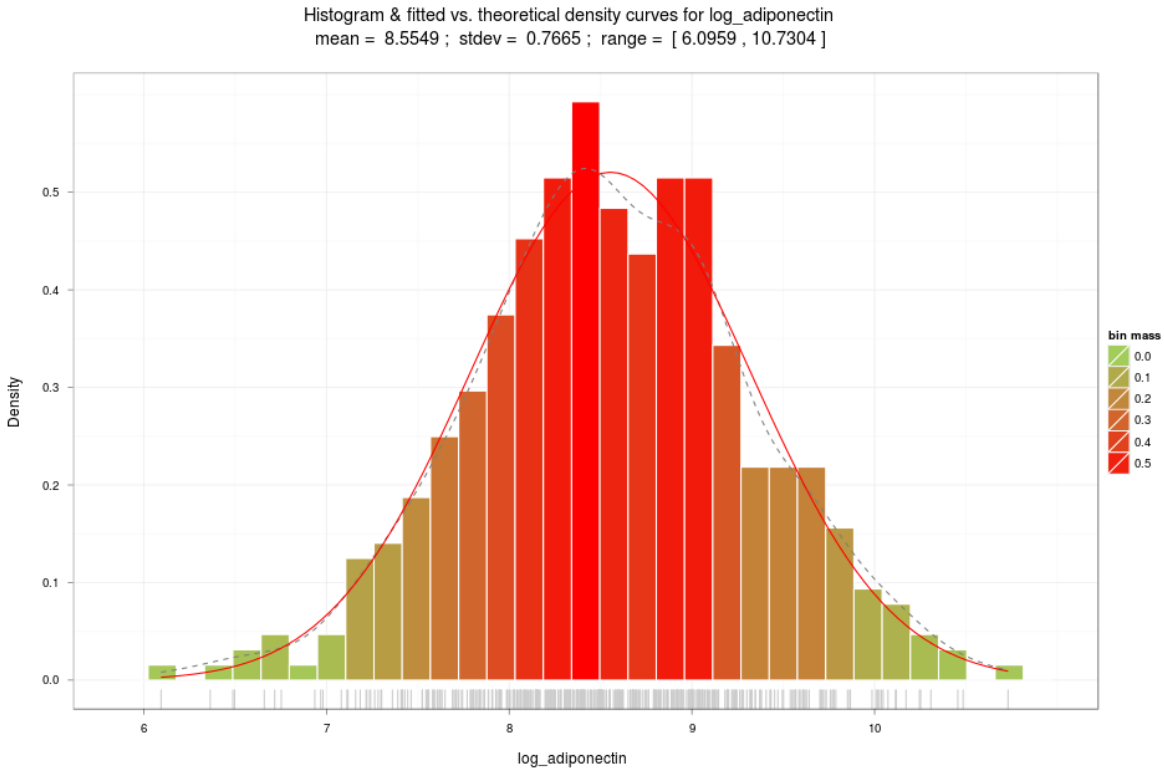


2. Perform Log transformation and check if log transformed data is normally distributed.

Tip

Use `log` for the `--transform` option for natural log transformation, use `log10` for *log* transformation with base 10. PhenoMan can also perform other types of data transformations, such as scaling, standardization, normalization and Gaussian quantile normalization.

```
phenoman view adiponectin --samples AA_adi.txt --transform log
eog log_adiponectin_histogram_AA_adi.png
```



3. Store info of `log_esp_adiponectin` into data file.

```
phenoman view adiponectin --samples AA_adi.txt --transform log --savedata AA_log_adi.txt
```

OUTPUT

```
...
Write intermediate dataset to AA_log_adi.txt
```

Now, the data file 'AA_log_adi.txt' contains trait `log_adiponectin`. The option `--savedata` takes a single output file name to save an intermediate dataset.

Note

Output file (`--savedata FILE2`) must have a different name from the input file (`--samples FILE1`).

4. Set criteria for determining outliers based on the primary trait `log_adiponectin` distribution: by viewing the histogram we decide those that have values < (mean-3*stdev) are outliers. Since the remained sample size is small we will winsorize outliers using `phenoman massage` instead of having them removed.

mean = 8.5549; stdev = .7665; range = [6.0959, 10.7304] –> (mean-3*stdev) = 6.2554

```
phenoman massage log_adiponectin -s AA_log_adi.txt --lower 6.2554 --savedata AA_logwin_adi.txt
```

OUTPUT

```
sample_name        log_adiponectin        winsorized_log_adiponectin
sample:807         6.096          6.255
There are 1 individuals that have been winsorized on phenotype log_adiponectin
```

The options `--lower` and `--upper` specify lower and upper bounds, respectively. See `phenoman massage -h` for more details.

5. Set criteria for determining outliers based on other traits

After individuals with extreme values on the primary trait being removed or winsorized, it may also require to check if within the remaining ones some may have extreme values on other critical factors (e.g. BMI, triglyceride, etc.). If those are necessarily to be removed, it requires to construct intermediate datasets from which more additional individuals can be removed based on other traits.

Note

Skip this step if only outlier detection on the primary trait is needed.

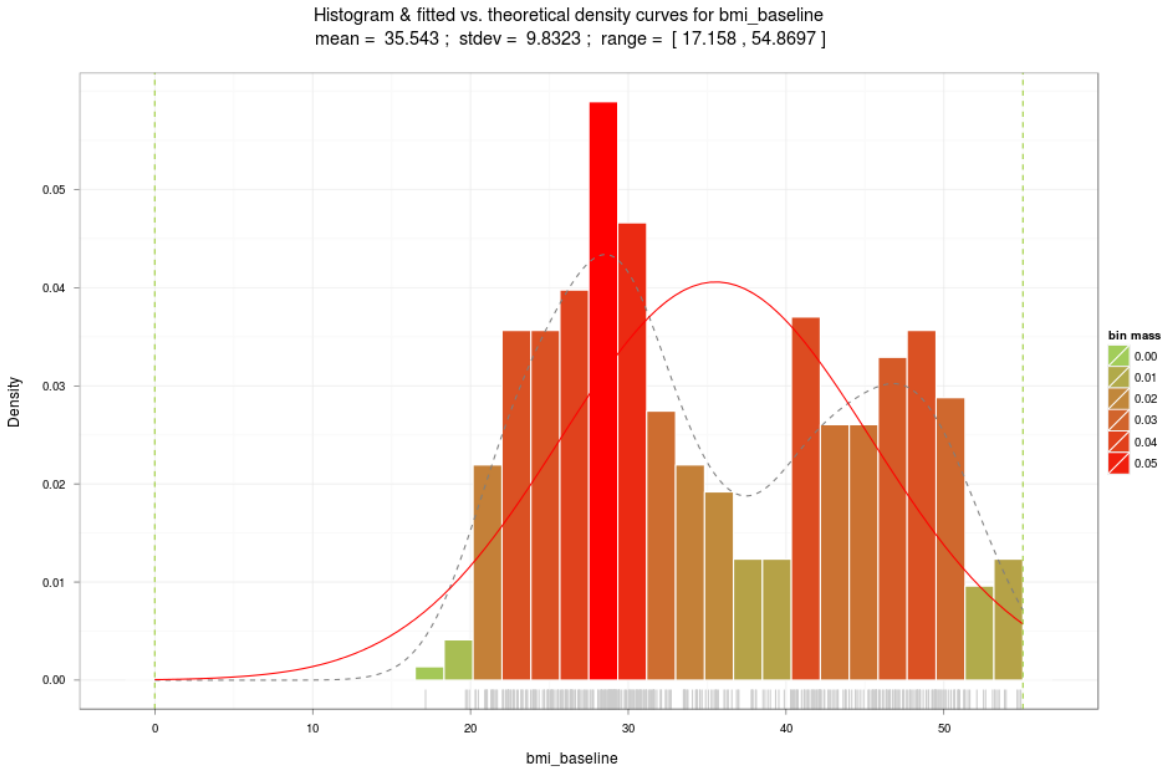
For example, check BMI(`bmi_baseline`) for AA dataset in addition to the primary trait (`log_adiponectin`).

Generate BMI distribution by reading the intermediate dataset 'AA_logwin_adi.txt' and view the BMI distribution

```
phenoman view bmi_baseline -s AA_logwin_adi.txt
eog bmi_baseline_histogram_AA_logwin_adi.png
```

OUTPUT

```
There are 2 individuals that are missing values on phenotype esp_bmi_baseline
Generating histogram ...
```

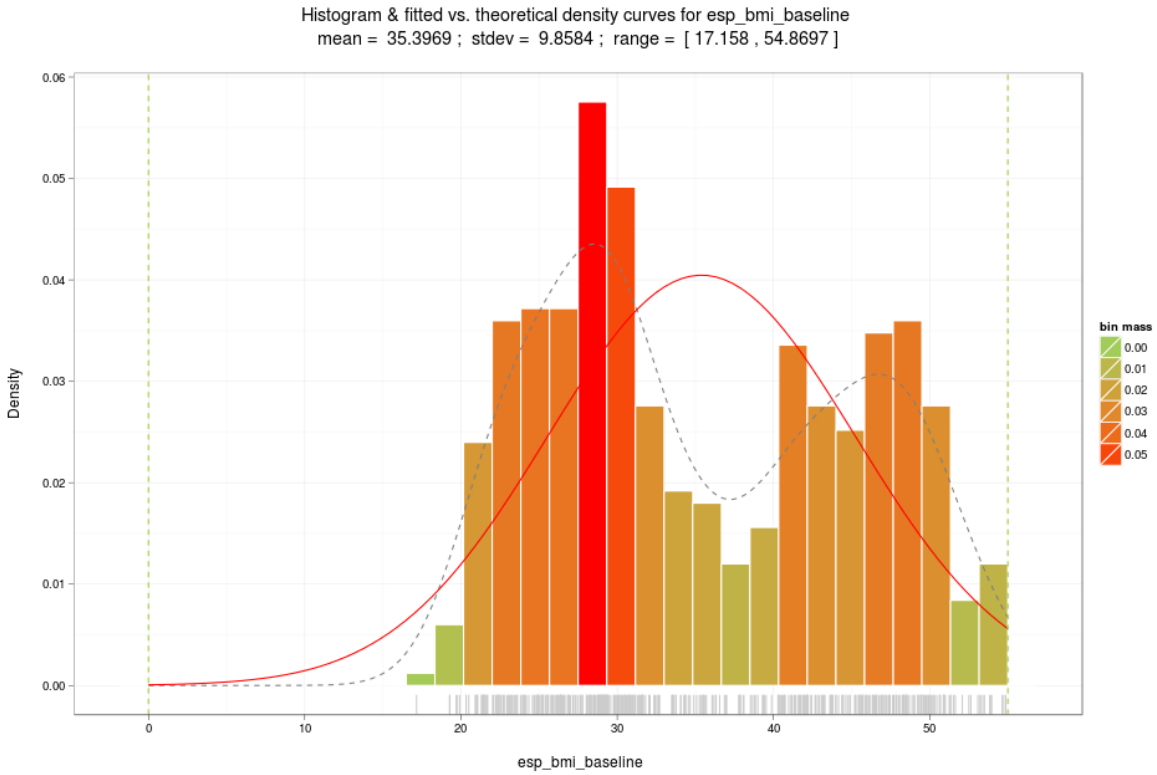


We will remove individuals that have BMI > 55 for AA; save another intermediate dataset to 'AA_logwinbmi_adi.txt' after removing outliers based on the secondary information, **BMI**, and save namelist of outliers to 'outliers_AA_bmi_adi.txt'.

```
phenoman view bmi_baseline -s AA_logwin_adi.txt --critical_values 0 55 --savedata AA_logwinbmi_adi.txt --keepmissin\
g > outliers_AA_bmi_adi.txt
cat outliers_AA_bmi_adi.txt
eog bmi_baseline_histogram_AA_logwin_adi.png
```

OUTPUT

```
There are 2 individuals that are missing values on phenotype esp_bmi_baseline
Generating histogram ...
Done!
OUTLIERS detected!
sample:622                  55.84041019
sample:605                  59.57042838
...
Write intermediate dataset to AA_logwinbmi_adi.txt
```



Tip

You may notice that by specifying `--savedata AA_logwinbmi_adi.txt` the histogram drawn already has the outliers removed.

The option `--critical_values` takes two numbers to specify the lower and upper bounds that define extreme phenotypes. Declaring the True/False option `--keepmissing` keeps instead of removing all missing values on trait `bmi_baseline`.

Note

Use `--keepmissing` with `--savedata` in `phenoman view` command to keep individuals that have missing values on secondary traits. Since BMI is not the primary trait it does not necessarily have to remove those that are missing BMI values.

Exercise

Please use PhenoMan to view the distribution of triglyceride (`trigs_baseline`) of the dataset 'AA_logwinbmi_adi.txt' and determine whether to do transformation, if there are any outliers and if there is need to remove/winsorize outliers if detected.

2.2.4 Select covariates

Normally for quantitative traits, we detect phenotype-genotype associations by linear regression. Thus, we need to control for covariates in the regression framework to avoid false positive results in establishing such associations. We will use `PhenoMan`'s model selection feature.

Important

It is recommended that a number of covariates as candidates to include in the regression have already been selected in previously determined analysis plan.

For Adiponectin, we select the following covariates and use `PhenoMan` to check if they are significantly co-related with the primary trait:

- age at adiponectin (`age_at_adiponectin`)
- BMI (`bmi_baseline`)
- sex (`sex`)
- smoking (current_smoker_baseline)

1. Check significance of covariates individually by running `phenoman show model` commands

```
phenoman show model --y log_adiponectin --covariates age_at_adiponectin < AA_logwinbmi_adi.txt
phenoman show model --y log_adiponectin --covariates bmi_baseline < AA_logwinbmi_adi.txt
phenoman show model --y log_adiponectin --covariates sex < AA_logwinbmi_adi.txt
phenoman show model --y log_adiponectin --covariates current_smoker_baseline < AA_logwinbmi_adi.txt
```

OUTPUT

```
INPUT MODEL:
Predictor                                   Estimate        t value         Pr(>|t|)
(Intercept)                                 7.74            3.77E+01        1.68E-133
age_at_adiponectin                          1.41E-02        4.03            6.71E-05
p-value:                                    6.71E-05
...
```

- age at adiponectin (p-value: 6.71E-05)
- BMI (p-value: 7.54E-01)
- sex (p-value: 9.55E-14)
- smoking (p-value: 8.94E-01)

The option `--y` takes the main phenotype name (primary trait) and the option `--covariates` takes a list of phenotype names as covariates.

Important

To use `phenoman show model` command, the input dataset should be specified in the end by `< AA_logwinbmi_adi.txt` instead of `--samples AA_logwinbmi_adi.txt`

2. Check significance of covariates by regressing them together

```
phenoman show model --y log_adiponectin --covariates age_at_adiponectin bmi_baseline sex current_smoker_baseline < \
AA_logwinbmi_adi.txt
```

OUTPUT

```
INPUT MODEL:
Predictor                                   Estimate        t value         Pr(>|t|)
(Intercept)                                 2.45E+02        1.25E-01        9.01E-01
age_at_adiponectin                          6.07E+01        2.30            2.22E-02
bmi_baseline                                -1.02E+02       -3.23           1.35E-03
sex                                         3.92E+03        5.74            1.93E-08
current_smoker_baseline                     6.12E+02        6.94E-01        4.88E-01
p-value:                                    1.26E-08
OPTIMIZED MODEL:
Predictor                                   Estimate        t value         Pr(>|t|)
(Intercept)                                 5.00E+02        2.60E-01        7.95E-01
age_at_adiponectin                          5.91E+01        2.25            2.52E-02
bmi_baseline                                -1.05E+02       -3.35           8.99E-04
sex                                         3.92E+03        5.75            1.83E-08
MODEL SELECTION:
AIC=6806.24       age_at_adiponectin+bmi_baseline+sex+current_smoker_baseline
AIC=6804.72       age_at_adiponectin+bmi_baseline+sex
```

For adiponectin (log transformed: `log_adiponectin`), we decide to select the following covariates for AA:

- age at adiponectin (age_at_adiponectin)
- BMI (bmi_baseline)
- sex (sex)

Important

- In real-world practice, you may want to calculate the first 2 MDS components based on genotype information and include them as two additional covariates. Please consider using `PLINK` and `KING` to generate MDS components and then using `phenoman merge --addcolumns` to add the first 2 MDS as new fields/covariates in the phenotype dataset.
- In addition, you may also treat text-based fields, such as target/cohort information, as candidate covariates and check their significance. Please refer to `phenoman comdummy` to combine and dummy code text-based covariates. The dummy coded covariates can be included in the regression framework and model selection.

2.2.5 Fill missingness in covariates

`PhenoMan` will automatically fill missing values for the selected covariates with **mean** (quantitative traits) or **baseline value** (binary traits). We have to provide the criteria to discard any selected covariate if its proportion of missingness exceeds a given threshold (use `--filter_missing` option in `phenoman cook` command). To disable this feature, use `--filter_missing -1` or leave the option out. See the next section for details.

2.2.6 Perform final cleaning

We use `phenoman cook` to fill in missing values in covariates and to finalize the phenotype data quality control for this example.

```
phenoman cook log_adiponectin --samples AA_logwinbmi_adi.txt --include age_at_adiponectin bmi_baseline sex --filter\
_missing 0.9 --output adiponectin_AA_cleaned.txt > adiponectin_AA_summary.txt
cat adiponectin_AA_summary.txt
```

OUTPUT

```
There are 0 individuals that are missing values on phenotype log_adiponectin
There are 0 individuals that are duplicates and have been removed
FIELD        MISSING_RATE        #BELOW_AVG        #ABOVE_AVG
log_adiponectin        0.0000        206        194
bmi_baseline        0.0050        222        178
age_at_adiponectin        0.0000        186        214
sex        0.0000        116        284
```

The `--include` option takes a list of phenotypes which will be extracted from the input data and saved in the output file, if left unspecified it will dump out all phenotypes as covariates. Normally, one should include at least all the selected covariates. The `--filter_missing` option takes a percentage (between 0 and 1) as the threshold of maximum allowed proportion of missingness for covariates included. Any covariate that has missingness greater than or equal to the threshold will be excluded.

```
head -10 adiponectin_AA_cleaned.txt
```

sample_name	log_adiponectin	age_at_adiponectin	bmi_baseline	sex
sample:591	8.333383327642693	62.0	50.0	2
sample:592	8.395053579648275	72.0	24.57786715	1
sample:593	8.229936628422465	59.0	21.06674327	1
sample:594	9.642304590056616	66.0	50.87325749	2
sample:596	9.135382099958434	71.0	47.38292011	2
sample:597	8.718986274946996	67.0	31.55365915	2
sample:598	8.371797408723952	62.0	54.8696845	2
sample:599	8.578907290125372	72.0	29.00176254	2
sample:600	7.112723454308319	60.0	33.76825311	2
...	...	...	...	...

Properties of cleaned dataset

```
phenoman show summary < adiponectin_AA_cleaned.txt
```

OUTPUT

```
 log_adiponectin  age_at_adiponectin  bmi_baseline        sex
 Min.   : 6.255   Min.   :26.00      Min.   :17.16   Min.   :1.00
 1st Qu.: 8.060   1st Qu.:51.00      1st Qu.:27.44   1st Qu.:1.00
 Median : 8.515   Median :58.50      Median :33.53   Median :2.00
 Mean   : 8.554   Mean   :57.75      Mean   :35.54   Mean   :1.71
 3rd Qu.: 9.055   3rd Qu.:66.00      3rd Qu.:44.98   3rd Qu.:2.00
 Max.   :10.730   Max.   :81.00      Max.   :54.87   Max.   :2.00
```

Warning

By default the input format for `phenoman show summary` accepts `phenoman cook` output. If data are coming from other sources you will have to specify `--y` and `--covariates` options.

2.2.7 Use residuals to account for covariates

To control for covariates, PhenoMan can also calculate the residuals in the regression model, add residuals as a new field phenotype_residuals to data and keep only the residuals in the cleaned dataset instead of all selected covariates.

For example, we can calculate residuals in the regression framework with selected covariates using the option --add_residuals:

```
phenoman show model --y log_adiponectin --covariates age_at_adiponectin bmi_baseline sex --add_residuals --savedata\
AA_logwinbmi_resid_adi.txt < AA_logwinbmi_adi.txt
```

OUTPUT

```
Add residuals to 'log_adiponectin_residuals' and write new dataset to AA_logwinbmi_resid_adi.txt
```

Note

The option --add_residuals can only be used on quantitative traits.

Now we may perform final cleaning by including log_esp_adiponectin_residuals as the the covariate:

```
phenoman cook log_adiponectin --samples AA_logwinbmi_resid_adi.txt --include log_adiponectin_residuals --filter_mis\
sing 1.0 --output adiponectin_AA_cleaned_resid.txt > adiponectin_AA_summary_resid.txt
cat adiponectin_AA_summary_resid.txt
```

OUTPUT

```
...
FIELD        MISSING_RATE        #BELOW_AVG        #ABOVE_AVG
log_adiponectin_residuals        0.0000        219        181
log_adiponectin        0.0000        206        194
```

```
head -10 adiponectin_AA_cleaned_resid.txt
```

sample_name	log_adiponectin	log_adiponectin_residuals
sample:591	8.333383327642693	-0.267429216490079
sample:592	8.395053579648275	0.0445802543654406
sample:593	8.229936628422465	-0.0370795071895057
sample:594	9.642304590056616	1.01303585146902
sample:596	9.135382099958434	0.409099608922191
sample:597	8.718986274946996	-0.179400375596134
sample:598	8.371797408723952	-0.1637125531406
sample:599	8.578907290125372	-0.403908600340659
sample:600	7.112723454308319	-1.68567389384316
...	...	...

Properties of cleaned dataset

```
phenoman show summary < adiponectin_AA_cleaned_resid.txt
```

OUTPUT

```
 log_adiponectin  log_adiponectin_residuals
 Min.   : 6.255   Min.   :-2.02430
```

```
 1st Qu.: 8.060   1st Qu.:-0.47150
 Median : 8.515   Median :-0.01607
 Mean   : 8.554   Mean   : 0.04893
 3rd Qu.: 9.055   3rd Qu.: 0.47268
 Max.   :10.730   Max.   :10.11832
```

2.2.8 Generate PhenoMan report

PhenoMan can generate a report file, which contains a list successfully executed `phenoman ...` commands during a specified time frame and/or applied on a specific data set. Such report can be used as input reference to process another data set using the same protocol to ensure consistency (run `phenoman report -h` for details)

Here we generate a report, phenoman_report_AA.log that includes all `phenoman` commands applied on data set 'AA_txt'.

```
phenoman report --samples AA.txt --filename phenoman_report_AA.log
```

2.3 Use PhenoMan on a Qualitative Trait

Note

This tutorial is intended to introduce some of `PhenoMan`'s features rather than provide exhaustive coverage of them. Furthermore, it is not intended as a study/analysis plan for quality control procedures of any primary trait of interest, or to represent anything close to 'best practice'.

2.3.1 Getting started

Just typing `phenoman -h` and specifying no further options is to list available modules and brief description of each module's functionality. To view detailed help info of each module, also type the name of the module after phenoman, e.g. `phenoman show -h`.

As an example for a qualitative/case-control trait, we choose **Gout**. Its corresponded field name in the example dataset is `gout`.

Run

```
>> @@phenoman show fields --samples EA.txt | grep gout@@
to check if the primary trait @@gout@@ is contained in the data.
```

Warning

All the following data cleaning steps need to be applied separately on different ethnic groups, e.g. AA (African Americans) and EA (European Americans) as the example data shows. We use 'EA.txt' in this tutorial. In practice, if the original data set contains individuals from multiple ethnic backgrounds, users should have them separated into different data sets to avoid population stratification problem.

Tip

Use command `phenoman select` to sample subpopulations from the original data set. See details about `--removethese` and `--keepthese` by running `phenoman select -h`.

2.3.2 Remove missingness and duplicates

First run `phenoman cook` command to remove individuals that have missing values on the primary trait and are duplicates or related individuals. Save remained ones into file 'EA_gout.txt'.

```
phenoman cook gout --samples EA.txt --duplicates EAdups.txt --output EA_gout.txt
```

OUTPUT

```
There are 2912 individuals that are missing values on phenotype gout
There are 43 individuals that are duplicates and have been removed
```

The `--samples` option takes a single file as the input data. The `--duplicates` option may take one or multiple files. Each of them must contain two columns of sample names, where on each row the first individual is to be removed if it co-exists with the second one in the data. In practice, users should create file(s) of duplicates in advance and be able to use them at this step. The `--output` option takes a single file name as the output data file name.

2.3.3 Choose case samples

Select all case samples using `phenoman select` command

```
phenoman select --samples EA_gout.txt --traits gout --criteria 1 --savedata EA_cases_gout.txt
```

OUTPUT

```
The following individuals (N = 995) have been removed:
sample:500
sample:501
sample:502
sample:503
sample:504
...
(990 sample names omitted)
Number of individuals removed: 995
Number of individuals remained/selected: 50
Write intermediate dataset to EA_cases_gout.txt
```

There are 50 case samples in EA.

The option `--traits` takes a list of phenotypes/traits based on which samples will be selected. The option `--criteria` takes a list of criteria. See `phenoman select -h` for more details. Only individuals that have given traits fit into specified criteria will be selected.

2.3.4 Choose control samples

As an example, we will use `PhenoMan` to select 150 = 3 x 50 (number of cases) control samples following a series of restrictions.

First, select all control samples

```
phenoman select --samples EA_gout.txt --traits gout --criteria 0 --savedata EA_controls_gout_tmp1.txt
```

OUTPUT

```
The following individuals (N = 50) have been removed:
sample:506
sample:525
sample:530
sample:592
sample:615
...
(45 sample names omitted)
Number of individuals removed: 50
Number of individuals remained/selected: 995
Write intermediate dataset to EA_controls_gout_tmp1.txt
```

1. Select those who have `gout == 0 AND phenotype == 'DPR'`

```
phenoman select -s EA_controls_gout_tmp1.txt --traits gout phenotype --criteria 0 DPR --savedata EA_controls_gout_t\
mp2.txt
```

OUTPUT

```
Number of individuals removed: 708
Number of individuals remained/selected: 287
...
```

Note

`DPR` is the name of a cohort which contains deeply sequenced controls.

2. Further select controls individuals that have `mi_baseline == 0 AND mi_during_followup == 0`

```
phenoman select --samples EA_controls_gout_tmp2.txt --traits mi_baseline mi_during_followup --criteria 0 0 --saveda\
ta EA_controls_gout_tmp3.txt
```

OUTPUT

```
Number of individuals removed: 42
Number of individuals remained/selected: 245
...
```

Note

`mi_baseline == 0 AND mi_during_followup == 0` indicates that no medical inspection occurred either at the beginning or during the follow up of the cohort study.

3. Select controls who have `t2diabetes_baseline == 0`

```
phenoman select -s EA_controls_gout_tmp3.txt --traits t2diabetes_baseline --criteria 0 --savedata EA_controls_gout_\
tmp4.txt
```

OUTPUT

```
Number of individuals removed: 10
Number of individuals remained/selected: 235
...
```

Note

`t2diabetes_baseline == 0` represents individuals that did not have type II diabetes.

4. Remove controls that have `t2d_med_baseline == 0`

```
phenoman select --samples EA_controls_gout_tmp4.txt --traits t2d_med_baseline --criteria 0 --removeselected --saved\
ata EA_controls_gout_tmp5.txt
```

OUTPUT

```
Number of individuals removed: 185
Number of individuals remained/selected: 50
...
```

The option `--removeselected` will remove selected individuals and keep unselected ones.

Note

5. There left two few control samples (50) compared to the required number (150). We rerun last command with the option `--samplesize 150` to randomly choose a number of individuals that ought to be removed but are added back in the remained control samples in order to meet the sample size requirement.

Tip

Except restoring removed individuals, using the option `--samplesize` can also delete remained ones to meet the sample size requirement. See chapters **Reference** and **Command Examples** for more details.

```
phenoman select --samples EA_controls_gout_tmp4.txt --traits t2d_med_baseline --criteria 0 --removeselected --sampl\
esize 150 --savedata EA_controls_gout_final.txt
```

OUTPUT

```
WARNING: The following individuals ought to be removed according to the criteria but are randomly chosen from removed
 ones to be kept in the remained dataset in order to meet the sample size requirement (150). If you still want them t
o be removed, please rerun the last command WITHOUT specifying --samplesize argument
WARNING: these individuals (N = 100) are restored:
sample:1654
sample:779
sample:818
sample:1746
sample:1824
...
(95 sample names omitted)
The following individuals (N = 85) have been removed:
sample:557
sample:570
sample:579
sample:587
sample:606
...
(80 sample names omitted)
Number of individuals removed: 85
Number of individuals remained/selected: 150
```

2.3.5 Merge selected cases and controls

We use `phenoman merge` to combine multiple data files into a single file.

```
phenoman merge --byrows EA_cases_gout.txt EA_controls_gout_final.txt --output EA_gout_selected.txt
```

The option `--byrows` takes a list of files to be merged together by row.

2.3.6 Select covariates

Normally in logistic regression framework for detecting phenotype-genotype associations in qualitative traits, we also control for covariates as we do to quantitative traits in linear regression to avoid false positive results.

Important

It is recommended that a number of covariates as candidates to include in the regression have already been selected in previously determined analysis plan.

For Gout, we select the following covariates and use `PhenoMan` model selection feature to check if they are significantly correlated with the primary trait:

- sex (`sex`)
- BMI (`bmi_baseline`)
- age (`age_baseline`)
- age^2 (`ageE2`)

1. Check significance of covariates individually by running `phenoman show model` commands

```
phenoman show model --y gout --covariates sex <  EA_gout_selected.txt
phenoman show model --y gout --covariates bmi_baseline <  EA_gout_selected.txt
phenoman show model --y gout --covariates age_baseline <  EA_gout_selected.txt
phenoman show model --y gout --covariates ageE2 <  EA_gout_selected.txt
```

- sex (p-value: 5.95E-01)
- BMI (p-value: 2.34E-05)
- age (p-value: 8.78E-05)
- age^2 (p-value: 2.06E-04)

2. Check significance of covariates by regressing them together

```
phenoman show model --y gout --covariates sex bmi_baseline age_baseline ageE2 < EA_gout_selected.txt
```

OUTPUT

```
INPUT MODEL:
Predictor                                                Estimate          z value           Pr(>|z|)
(Intercept)                                              5.98E-01          1.73E-01          8.63E-01
bmi_baseline                                             1.63E-01          3.72              1.98E-04
sex                                                      4.17E-01          8.73E-01          3.82E-01
age_baseline                                             -2.00E-01         -1.62             1.04E-01
ageE2                                                    1.30E-03          1.14              2.53E-01
p-value:                                                 6.91E-07
OPTIMIZED MODEL:
Predictor                                                Estimate          z value           Pr(>|z|)
(Intercept)                                              -2.25             -1.54             1.24E-01
bmi_baseline                                             1.59E-01          3.74              1.84E-04
age_baseline                                             -6.33E-02         -3.48             4.98E-04
MODEL SELECTION:
AIC=200.77        bmi_baseline+sex+age_baseline+ageE2
AIC=199.52        bmi_baseline+age_baseline+ageE2
AIC=198.67        bmi_baseline+age_baseline
```

Thus, we decide to select age (`age_baseline`) and BMI (`bmi_baseline`) as covariates for EA_gout

2.3.7 Perform final cleaning

We use `phenoman cook` to fill in missing values in covariates and to finalize the phenotype data quality control for this example.

```
phenoman cook gout --samples EA_gout_selected.txt --include age_baseline bmi_baseline --filter_missing 0.8 --output\
gout_EA_cleaned.txt > gout_EA_summary.txt
cat gout_EA_summary.txt
```

OUTPUT

```
There are 0 individuals that are missing values on phenotype gout
There are 0 individuals that are duplicates and have been removed
FIELD        MISSING_RATE        #BELOW_AVG        #ABOVE_AVG
gout        0.0000        150        50
bmi_baseline        0.0000        118        82
age_baseline        0.0000        109        91
```

```
head -10 gout_EA_cleaned.txt
```

sample_name	gout	bmi_baseline	age_baseline
sample:506	1.0	28.80597	58.0
sample:525	1.0	44.86731	49.0
sample:530	1.0	31.35358	55.0
sample:592	1.0	31.75244	62.0
sample:615	1.0	32.49632	59.0
sample:616	1.0	24.67549	56.0
sample:619	1.0	27.40127	48.0
sample:635	1.0	28.13609	48.0
sample:659	1.0	30.95366	45.0
...	...	...	...

Properties of cleaned dataset

```
phenoman show summary < gout_EA_cleaned.txt
```

OUTPUT

```
      gout        bmi_baseline     age_baseline
 Min.   :0.00   Min.   :17.36   Min.   :22.00
 1st Qu.:0.00   1st Qu.:24.62   1st Qu.:46.00
 Median :0.00   Median :26.96   Median :52.00
 Mean   :0.25   Mean   :27.43   Mean   :53.70
 3rd Qu.:0.25   3rd Qu.:29.16   3rd Qu.:60.25
 Max.   :1.00   Max.   :49.50   Max.   :83.00
```

2.3.8 Generate PhenoMan report

To ensure consistency before using the same phenotype QC protocal to process another data set, here we generate a report, phenoman_report_EA.log that includes all `phenoman` commands applied on data set 'EA_txt' (run `phenoman report -h` for details).

```
phenoman report --samples EA.txt --filename phenoman_report_EA.log
```

2.4 Use PhenoMan on an Extreme Quantitative Trait

Warning

This example is provided for a quick demonstration purpose only without much detailed explanations. Please walk through the other two examples above first if you are new to PhenoMan.

2.4.1 Getting started

We will select AA (African American) individuals that have their BMI on two extreme ends and perform several quality control steps based on hypothetical restrictions to create a 'cleaned' case-control phenotype dataset.

check if phenotype **BMI** is contained in the input data

```
phenoman show fields --samples AA.txt | grep bmi
```

OUTPUT

```
bmi_baseline ...
```

2.4.2 Remove missingness and duplicates

```
phenoman cook bmi_baseline --samples AA.txt --duplicates AAdups.txt --output AA_bmi.txt
```

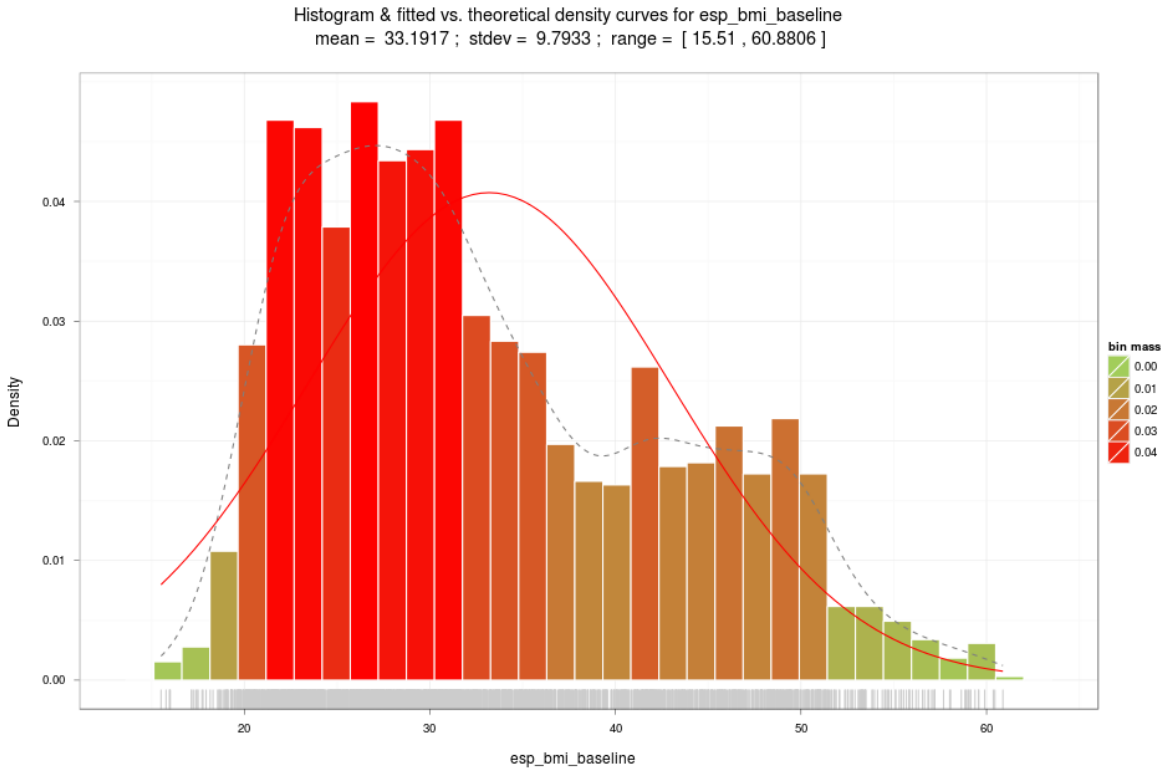
OUTPUT

```
There are 28 individuals that are missing values on phenotype esp_bmi_baseline
There are 88 individuals that are duplicates and have been removed
```

2.4.3 Detect outliers

1. Draw histogram of bmi_baseline

```
phenoman view bmi_baseline --samples AA_bmi.txt
eog bmi_baseline_histogram_AA_bmi.png
```



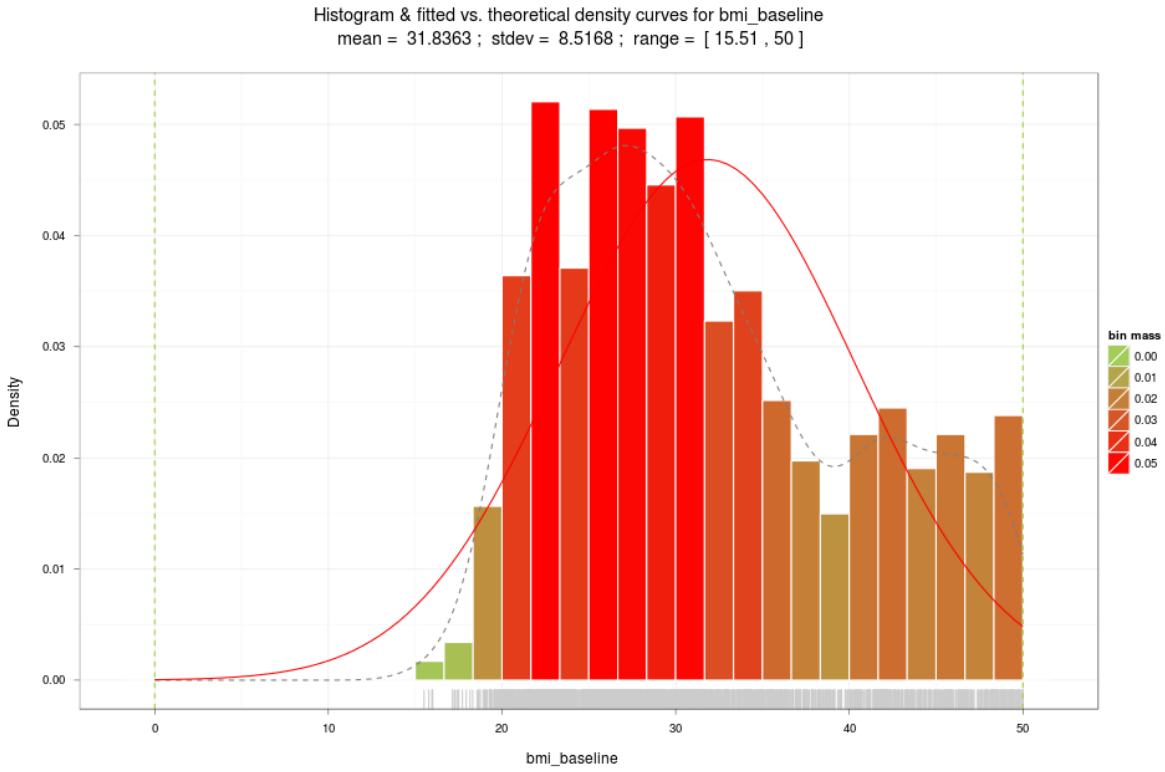
2. Remove those that have BMI > 50

```
phenoman view bmi_baseline -s AA_bmi.txt --critical_values 0 50 --savedata AA_rmo_bmi.txt > outliers_AA_bmi.txt
head -10 outliers_AA_bmi.txt
```

OUTPUT

```
OUTLIERS detected!
sample:97                   55.83103765
sample:1893                 51.79688
sample:1418                 55.29934
sample:1266                 58.98692
```

```
eog bmi_baseline_histogram_AA_bmi.png
```



3. Perform Gaussian quantile normalization and view histogram of the transformed data

Important

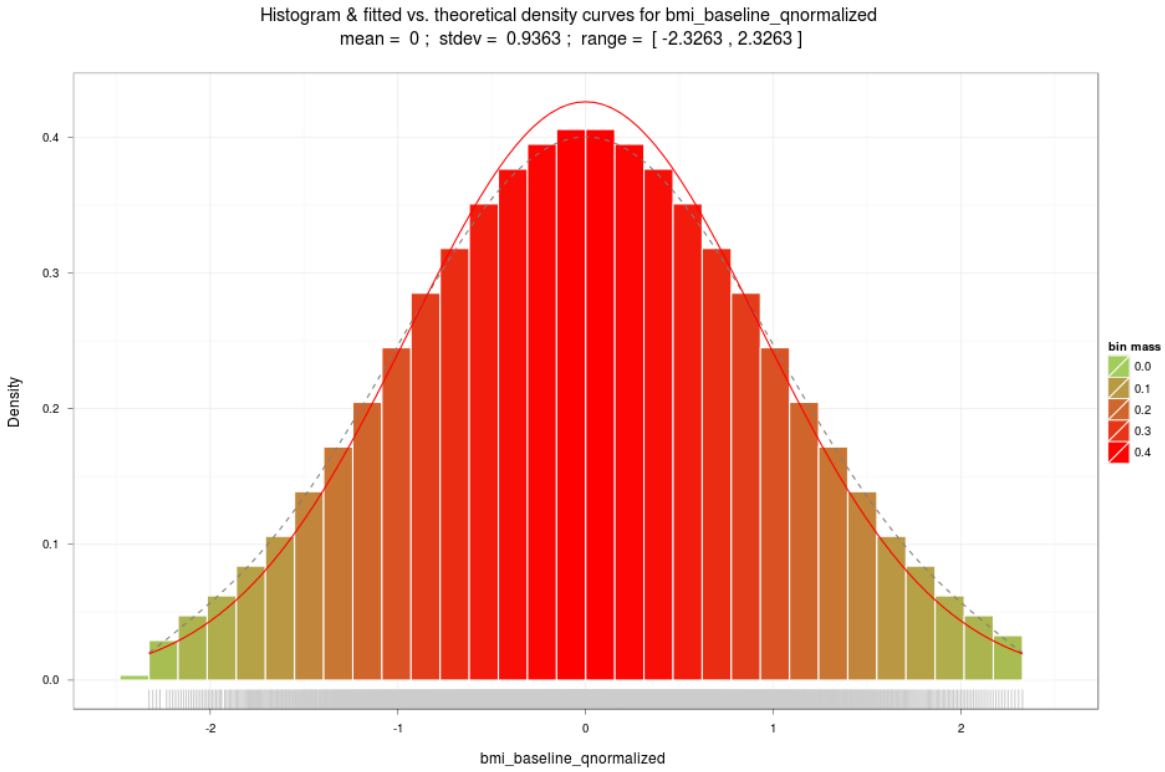
All missing values must be removed (by running `phenoman cook ...` shown above) before normalization.

```
phenoman view bmi_baseline -s AA_rmo_bmi.txt --qnormalize 0.01 0.99 --savedata AA_qnorm_bmi.txt
```

OUTPUT

```
Add a new column bmi_baseline_qnormalized to the dataset
...
```

```
eog bmi_baseline_qnormalized_histogram_AA_rmo_bmi.png
```



2.4.4 Select case-control samples

mean = 0; stdev = 0.936; range = [-2.326, 2.326]

Set (mean - 0.75*stdev) and (mean + 0.75*stdev) as thresholds for sampling cases and controls.

```
phenoman select --samples AA_qnorm_bmi.txt --traits bmi_baseline_qnormalized --criteria "-2.5, -0.7" --tobecases --\
savedata AA_cases_bmi.txt
```

OUTPUT

```
The following individuals (N = 1346) have been removed:
sample:1
sample:2
sample:4
sample:5
sample:6
...
(1341 sample names omitted)
Number of individuals removed: 1346
Number of individuals remained/selected: 418
Write intermediate dataset to AA_cases_bmi.txt
```

```
phenoman select --samples AA_qnorm_bmi.txt --traits bmi_baseline_qnormalized --criteria "0.7, 2.5" --tobecontrols -\
-savedata AA_controls_bmi.txt
```

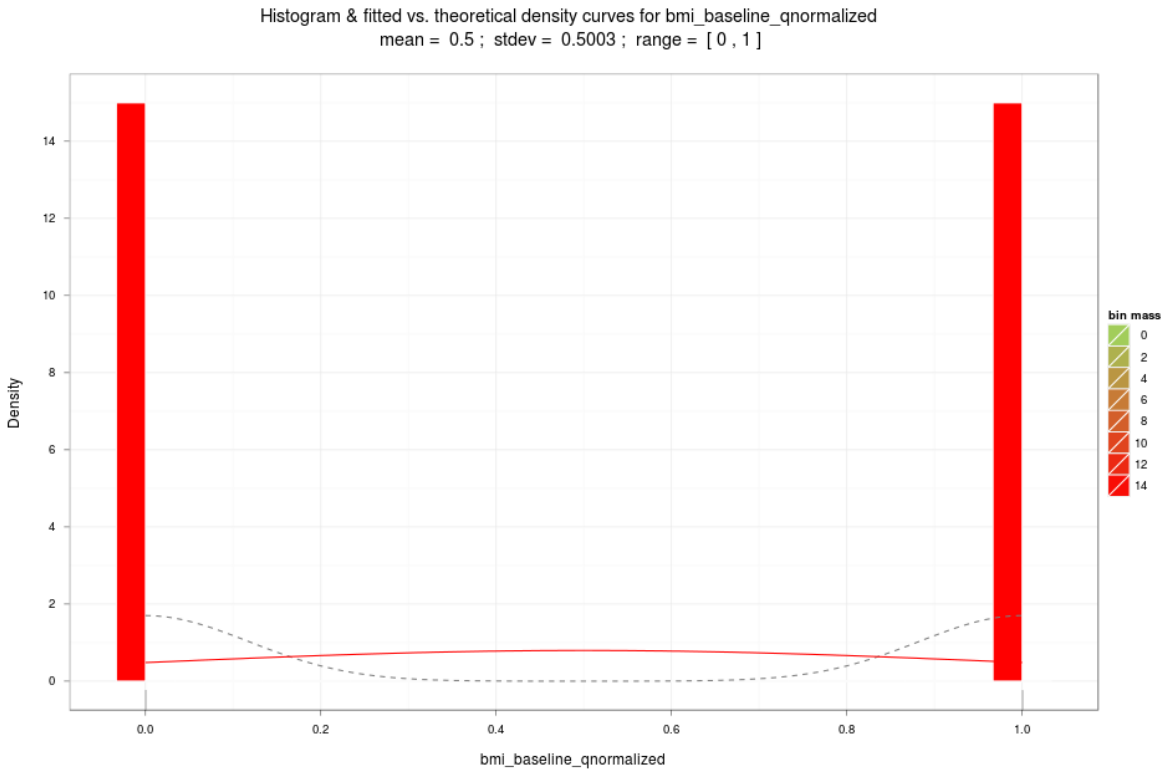
OUTPUT

```
The following individuals (N = 1346) have been removed:
sample:1
sample:5
sample:6
sample:8
sample:9
```

```
...
(1341 sample names omitted)
Number of individuals removed: 1346
Number of individuals remained/selected: 418
Write intermediate dataset to AA_controls_bmi.txt
```

2.4.5 Merge selected cases and controls

```
phenoman merge --byrows AA_cases_bmi.txt AA_controls_bmi.txt --output AA_casectrl_bmi.txt
phenoman view bmi_baseline_qnormalized -s AA_casectrl_bmi.txt
eog bmi_baseline_qnormalized_histogram_AA_casectrl_bmi.png
```



2.4.6 Select covariates

For BMI extreme quantitative trait, we choose the following covariates to check if any of them has significant correlation with the primary trait.

- age (`age_baseline`)
- sex (`sex`)
- smoking (`current_smoker_baseline`)

1. Check significance of covariates individually

```
phenoman show model --y bmi_baseline_qnormalized --covariates age_baseline < AA_casectrl_bmi.txt
phenoman show model --y bmi_baseline_qnormalized --covariates sex < AA_casectrl_bmi.txt
phenoman show model --y bmi_baseline_qnormalized --covariates current_smoker_baseline < AA_casectrl_bmi.txt
```

- age (p-value: 2.18E-04)
- sex (p-value: 0)
- smoking (p-value: 7.77E-16)

2. Check significance of covariates by regressing them together

```
phenoman show model --y bmi_baseline_qnormalized --covariates age_baseline sex current_smoker_baseline < AA_casectr\
l_bmi.txt
```

OUTPUT

```
INPUT MODEL:
Predictor                                          Estimate          z value           Pr(>|z|)
(Intercept)                                        2.62              4.20              2.62E-05
sex                                                -1.85             -6.62             3.64E-11
current_smoker_baseline                            1.29              6.11              9.75E-10
age_baseline                                       9.26E-03          1.31              1.91E-01
p-value:                                           0
OPTIMIZED MODEL:
Predictor                                          Estimate          z value           Pr(>|z|)
(Intercept)                                        3.05              5.69              1.25E-08
sex                                                -1.80             -6.53             6.49E-11
current_smoker_baseline                            1.24              6.00              2.00E-09
MODEL SELECTION:
AIC=932.23        sex+current_smoker_baseline+age_baseline
AIC=931.95        sex+current_smoker_baseline
```

For BMI extreme quantitative trait (`bmi_baseline_qnormalized`), select age, sex and smoking as covariates.

2.4.7 Perform final cleaning

We use `phenoman cook` to fill in missing values in covariates and to finalize the phenotype data quality control for this example.

```
phenoman cook bmi_baseline_qnormalized --samples AA_casectrl_bmi.txt --include age_baseline sex current_smoker_base\
line --filter_missing 0.8 --output bmi_AA_casectrl_cleaned.txt > bmi_AA_casectrl_summary.txt
cat bmi_AA_casectrl_summary.txt
```

OUTPUT

```
FIELD        MISSING_RATE        #BELOW_AVG        #ABOVE_AVG
bmi_baseline_qnormalized        0.0000        418        418
current_smoker_baseline        0.0981        668        168
age_baseline        0.0000        334        502
sex        0.0000        146        690
```

```
head -10 bmi_AA_casectrl_cleaned.txt
```

sample_name	bmi_baseline_qnormalized	sex	current_smoker_baseline	age_baseline
sample:9	1	2	0.0	55.0
sample:16	1	2	1.0	57.0
sample:25	1	1	1.0	48.0
sample:42	1	2	1.0	47.0
sample:46	1	1	1.0	27.0
sample:50	1	1	1.0	47.0
sample:55	1	1	1.0	48.0
sample:70	1	1	0.0	46.0
sample:92	1	2	0.0	60.0
...	...	...	...	...

Properties of cleaned dataset

```
phenoman show summary < bmi_AA_casectrl_cleaned.txt
```

OUTPUT

```
bmi_baseline_qnormalized      sex        current_smoker_baseline   ...
Min.   :0.0              Min.   :1.000   Min.   :0.000 ...
1st Qu.:0.0              1st Qu.:2.000   1st Qu.:0.000 ...
Median :0.5              Median :2.000   Median :0.000 ...
Mean   :0.5              Mean   :1.825   Mean   :0.201 ...
3rd Qu.:1.0              3rd Qu.:2.000   3rd Qu.:0.000 ...
Max.   :1.0              Max.   :2.000   Max.   :1.000 ...
```

2.4.8 Generate PhenoMan report

run `phenoman report -h` for more details.

```
phenoman report --samples AA.txt --filename phenoman_report_AA.log
```

Chapter 3

References

PhenoMan is a command line program written in Python. All commands involve typing phenoman at the command prompt (e.g. Linux/Mac terminal or DOS window) followed by a number of options (all starting with --option) to specify the data files/methods to be used. All results can be written to files with various extensions. The input phenotype data file is by default of following the **dbGaP** format with one header row and a fixed number of columns per line. A complete list of all options is given in the reference section.

Important

Before moving along, please make sure that all prerequisites of using PhenoMan have been met (including proper installation of PhenoMan and its dependency programs, and correct format and coding of the input data)

3.1 Running PhenoMan

Open up a command prompt or terminal window and perform all analyses by typing commands as:

```
phenoman module (phenotype) --option1 (argument1) --option2 (argument2) ...
```

Terms in '()' are optional or not required depending on which module and what options are called.

Note

By default, running phenoman commands will print error and warning messages to the standard output, which is the screen most of the time. To redirect those messages and save them into a text file, add > FILENAME.txt to the end of any phenoman command.

3.2 Modules

PhenoMan consists of the following modules:

- Help - show detailed information of each module and its options

- Show - show raw data, basic statistical summary, check data fields (column names), perform model selection (regression framework), control for covariates, add residuals
- View - view trait distribution, data transformation (log, scaling, standardization, normalization, Gaussian quantile normalization), detect and determine outliers based on both primary trait and secondary traits
- Cook - remove missingness on primary trait, remove duplicates, fill missingness in covariates, include/exclude selected/removed covariates, finalize phenotype data cleaning
- Select - sample individuals randomly or based on given criteria of specified phenotypes, select/remove individuals by required sample size
- Massage - winsorize outliers
- Comdummy - combine and/or dummy code covariates
- Merge - merge separated datasets into single one (either by row or by column)

3.2.1 `phenoman show`

Show raw data, show data summary for data from standard input, show model selection result.

Usage:

```
phenoman show object(data/fields) --samples dataset.txt --other_options args
```

or

```
phenoman show object(summary/model) --options args < dataset.txt
```

Choose an object first to decide what to show then select to use options.

List of options:

- `--y` - name of the primary phenotype (usable for `show summary` and `show model`)
- `--samples` - takes an input data file (/path/to/dataset.txt)
- `--covariates` - takes a list of covariates names (usable for `show summary` and `show model`)
- `--add_residuals` - takes no value, if specified it will calculate residuals between model prediction and data, and add a column `phenotype_residuals` to the dataset
- `--direction` - choose from both, forward or backward, to specify the mode of stepwise search for model selection
- `--savedata` - takes a file name (/path/to/datafilename.txt) to which the intermediate dataset with remained individuals will be saved
- `--reference_group` - takes a string to set reference group for dummy coded covariates (usable for `show model`)

3.2.2 `phenoman view`

For specified phenotype/trait in specified population, generate histogram to help determine transformation of trait and outliers (duplicates implicitly excluded).

Usage:

```
phenoman view phenotype --samples dataset.txt --other_options args
```

Choose a 'phenotype' to be processed first then select to use options. Use `phenoman show fields` to view all available phenotypes

List of options:

- `--samples` - takes an input data file (/path/to/dataset.txt)
- `--output` - takes an output graph(histogram) filename (default: 'phenotype_histogram_dataset.png' where 'phenotype' and 'dataset' are replaced with name of the trait and prefix of the input data file). Output is in png format.
- `--pdf` - if specified generate pdf instead of png graph
- `--savedata` - takes a file name (/path/to/datafilename.txt) to which the intermediate dataset with remained individuals will be saved
- `--keepmissing` - keep instead of removing individuals that have missing values on the 'phenotype' (use with `--savedata` option)
- `--transform` - takes 'log' or 'log10', log normal or log 10 transformation of 'phenotype'
- `--critical_values` - specify the lower (the first input value) and upper (the second input value) bounds that define extreme phenotypes. If '–percentile' option is specified then the input upper and lower bound values are taken as percentiles (e.g., '0.01 0.99' means to include phenotypes having values within 1st and 99th percentiles)
- `--percentile` - treat input lower and upper bounds from `--critical_values` as percentiles (only usable with `--critical_values`)
- `--scale` - shift 'phenotype' by mean (x-mean)
- `--standardize` - standardize 'phenotype' by (x-mean)/variance
- `--normalize` - apply normalization on 'phenotype', which scales all numeric values in the range [0,1] by (x-min)/(max-min)
- `--qnormalize` - apply Gaussian Quantile normalization on 'phenotype' between the specified lower and upper bounds that define the range of probabilities, e.g. 0.025 0.975

3.2.3 `phenoman cook`

For specified phenotype/trait in specified population, remove missing values, remove duplicates, remove outliers, transform trait if necessary, fill-up missing covariates and output cleaned phenotypes and selected covariates.

Usage:

```
phenoman cook phenotype --samples dataset.txt --other_options args
```

Choose a 'phenotype' to be processed first then select other options.

Option list:

- `--samples` - takes an input data file (/path/to/dataset.txt)
- `--duplicates` - takes a number of files that contains names of lists of duplicates and/or related individuals
- `--output` - takes an output data file name (/path/to/cleaned_dataset.txt)
- `--include` - specify particular covariates to output (default set to output all covariates).
- `--exclude` - specify particular covariates NOT to output (default set to exclude no covariate)
- `--filter_missing` - specify threshold of maximum allowed proportion of missingness for selected covariates, e.g. `--filter_missing 0.8` fill-in missing values in covariates that have percentage of missingness less than 80%, discard covariates that do otherwise.
- The following options are inherited from `phenoman view` and work exactly the same here in `phenoman cook`.
- `--transform`, `--critical_values`, `--percentile`, `--scale`, `--standardize`, `--normalize`, `--qnormalize`

3.2.4 `phenoman select`

Select individuals according to specified traits with given criteria, add individuals to or remove individuals from the dataset.

Usage:

```
phenoman select --samples dataset.txt --other_options args
```

Option list:

- `--samples` - takes an input data file (/path/to/dataset.txt)
- `--traits` - specify a lists of traits/phenotypes (usable only with `--criteria`)
- `--criteria` - specify selection criteria for 'traits' (usable with `--traits`. Only individuals that have specified 'traits' following given 'criteria' will be selected. Each criterion for each trait can be a single value/string(0 or NA), a range of numerical values (e.g. `20, 80` or `-2, -1` including boundary values) or a list of strings (separated by '|') to choose from (e.g. A|B|C)'")
- `--savedata` - takes a file name (/path/to/datafilename.txt) to which the intermediate dataset with remained individuals will be saved
- `--removeselected` - Remove selected individuals from the dataset instead of keeping them (otherwise if the option is not used)
- `--samplesize` - Desired number of individuals remained in the dataset after selection 'criteria' have been applied (restore or remove some individuals to meet the sample size requirement)
- `--removethese` - file names that contain lists of sample names to be removed from the dataset under any circumstance

- `--keepthese` - file names that contain lists of sample names to be kept in the dataset regardless of `--criteria`
- `--tobecases` - recode trait #1 (`--traits 1 2 ..`) in all remained individuals to 1 (1 for case, only usable with `--traits`)
- `--tobecontrols` - recode trait #1 (`--traits 1 2..`) in all remained individuals to 0 (0 for control, only usable with `--traits`)
- `--keeporiginal` - keep original values of trait #1 by adding a new column named by 'trait1_original' to the dataset (only usable with `--tobecases` or `--tobecontrols`)

3.2.5 `phenoman massage`

Reset trait values that are out of specified bounds with boundary values (winsorization).

Usage:

```
phenoman massage phenotype --samples dataset.txt --other_options args
```

List of options:

- `--samples` - takes an input data file (/path/to/dataset.txt)
- `--lower` - specify lower bound (e.g. 0 or 0.001 if `--percentile` is used)
- `--upper` - specify upper bound (e.g. 400 or 0.999 if `--percentile` is used)
- `--savedata` - specify the output data file name
- `--percentile` - treat input lower and upper bounds as percentile (only usable with `--lower` and/or `--upper`)

3.2.6 `phenoman comdummy`

Combine and/or dummy code phenotypes/covariates

Usage:

```
phenoman comdummy --samples dataset.txt --other_options args
```

Option List:

- `--samples` - takes an input data file (/path/to/dataset.txt)
- `--sample_col` - specify column index for sample names (default set to 1)
- `--style` - takes ''diagonal' or 'triangle'
- ...

3.2.7 phenoman merge

Merge separated datasets into single dataset (either by row or by column)

Usage:

```
phenoman merge --byrows data1.txt data2.txt ... --output data.txt
```

or

```
phenoman merge --bycolumns data1.txt data2.txt ... --output data.txt
```

List of options:

- `--byrows` - data files to be merged by rows (need to make sure that all files contain equivalent number of columns and column names in these files are arranged in the same order)
- `--bycolumns` - data files to be merged by columns, where file #1 is the primary file into which other files will be merged (add columns that are contained in files #2, #3, ..., but not in file #1 to file #1; missing values in newly added columns will be marked by 'nan')
- `--output` - specify the merged data file name

3.2.8 phenoman report

generate a list of successfully executed PhenoMan commands during a specified time frame and/or applied on a specific data set.

Usage:

```
phenoman report --year ... --month ... --day ... --hour ... --samples data.txt --filename report.txt
```

List of options:

- `--year` - specify which year, an integer, e.g. 13, or year range, e.g. 13-14
- `--month` - specify which month or range of months, e.g. 10 or 6-9
- `--day` - specify which day or days, e.g. 15 or 10-20
- `--hour` - specify which hour or hours, e.g. 14 or 10-16
- `--samples` - specify which data set, e.g. AA.txt
- `--filename` - specify output file name to save retrieved PhenoMan commands, e.g. phenoman_report.txt

Chapter 4

Command Examples

This section contains a rough overview of main operations in PhenoMan. In particular, it is written to indicate which certain operations are performed to clean raw phenotype datasets. Typically, we undergo data exploration, management and quality control steps to determine phenotype data cleaning rules, order/re-order datasets and create the 'cleaned' data that can be directly used in association studies of quantitative and qualitative traits.

Note

By providing command examples listed in the following several subsections we do NOT suggest any particular order of performing PhenoMan operations. The combination of operations should be appropriately designed to be data-centric and user-specific while PhenoMan is applied on real data. See the 'Tutorial' chapter for examples of applying PhenoMan on a quantitative, a case-control and an extreme quantitative traits.

4.1 Phenotype Data Exploration

4.1.1 header information

```
phenoman show fields --samples xxx
```

4.1.2 basic statistical summary

show selected phenotype and covariates only

```
phenoman show summary --y phenotype --covariates phenotypes < data.txt
```

show all phenotypes

```
phenoman show summary < data.txt
```

data.txt can be either raw data or cleaned data

4.1.3 view distribution

```
phenoman view phenotype --samples data.txt
phenoman view phenotype --samples data.txt --output phenotype_hist.png
phenoman view phenotype --samples data.txt --pdf --output phenotype_hist.pdf
```

4.1.4 determine outliers

primary trait (remove individuals with missing values)

```
phenoman view phenotype --samples data.txt --critical_values lower_bound upper_bound
```

secondary traits (keep individuals with missingness)

```
phenoman view phenotype_non_primary --samples data.txt --critical_values ower_bound upper_bound --keepmissing --sav\
edata data_tmp.txt > outliers.txt
phenoman view phenotype --samples data.txt --critical_values lower_bound_in% upper_bound_in% --percentile
```

4.2 Phenotype Data Management

4.2.1 log transformation

```
phenoman view phenotype --samples data.txt --transform log
phenoman view phenotype --samples data.txt --transform log10
```

4.2.2 scaling

```
phenoman view phenotype --samples data.txt --scale --savedata data_scaled.txt
```

4.2.3 standardization

```
phenoman view phenotype --samples data.txt --standardize --savedata data_standardized.txt
```

4.2.4 normalization

```
phenoman view phenotype --samples data.txt --normalize --savedata data_normalized.txt
```

4.2.5 Gaussian Quantile normalization

```
phenoman view phenotype --samples data.txt --qnormalize lowest_probability highest_probability --savedata data_qnor\
malized.txt
```

4.2.6 winsorising

```
phenoman massage phenotype --samples data.txt --lower lower_bound --upper upper_bound --savedata data_win.txt
phenoman massage phenotype --samples data.txt --lower lower_bound_in% --upper upper_bound_in% --percentile --saveda\
ta data_win.txt > summary_winsorising.txt
```

4.2.7 dummy coding

combined dummy coding

```
phenoman comdummy --samples data.txt --sample_col 1 --cols 2 3 --header TACO > cdummy_pheno.txt
```

4.2.8 merging datasets

```
phenoman merge --byrows data1.txt data2.txt ... --output data_merged.txt
phenoman merge --bycolumns data1.txt data2.txt ... --output data_merged.txt
```

4.3 Phenotype Data Quality Control

4.3.1 remove missingness on primary phenotype

```
phenoman cook phenotype --samples data.txt --output data_no_missing.txt
```

4.3.2 remove duplicates

```
phenoman cook phenotype --samples data.txt --duplicates dups.txt --output data_no_dups.txt
```

4.3.3 fill missingness in covariates

```
phenoman cook phenotype --samples data.txt --filter_missing missing_ratio --output data_cleaned.txt > summary.txt
```

4.3.4 model selection (control for covariates)

```
phenoman show model --y phenotype --covariates phenotypes < data.txt
```

4.3.5 add residuals

```
phenoman show model --y phenotype --covariates phenotypes --add_residuals --savedata data_with_residuals.txt < data\
.txt
```

4.3.6 select individuals

unbiased

```
phenoman select --samples data.txt --traits phenotype1 phenotype2 phenotype3 --criteria 1 "0.1, 0.9" 'A|B|C' --save\
data data_selected_individuals.txt
```

biased (with `--keepthese` and/or `--removethese`)

```
phenoman select --samples data.txt --traits phenotypes --criteria criteria --keepthese sample_names_to_keep.txt --r\
emovethese sample_names_to_remove.txt
```

4.3.7 remove individuals

unbiased

```
phenoman select --samples data.txt --traits phenotypes --criteria criteria --removeselected --savedata data_remaine\
d_individuals.txt
```

biased (with `--keepthese` and/or –removethese)

```
phenoman select --samples data.txt --traits phenotypes --criteria criteria
--removeselected --keepthese sample_names_to_keep.txt --removethese sample_names_to_remove.txt
```

4.3.8 select extreme quantitative trait

cases (recode 'phenotype')

```
phenoman select --samples data.txt --traits phenotype --criteria @@lower_bound, upper_bound@@ --tobecases --savedat\
a data_cases.txt
```

controls (recode 'phenotype')

```
phenoman select --samples data.txt --traits phenotype --criteria @@lower_bound, upper_bound@@ --tobecontrols --save\
data data_controls.txt
```

keep original 'phenotype'

```
phenoman select --samples data.txt --traits phenotype --criteria @@lower_bound, upper_bound@@ --tobecases --savedat\
a data_cases_ori.txt --keeporiginal
```

4.3.9 draw sample

```
phenoman select --samples data.txt --traits phenotypes --criteria criteria --samplesize N --savedata data_sample.tx\
t
```

References

1. [**Python 2.7 or above**]
   http://www.python.org/download/releases/2.7/
2. [**Python 3.2 or above**]
   http://www.python.org/download/releases/3.2/
3. [**R**]
   http://www.r-project.org/
4. [**dbGaP** phenotype data format]
   http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2031016/