Manual for ProbABEL v0.5.0

Current Programmers: Lennart Karssen[1], Maarten Kooyman[2], Yurii Aulchenko[1,3]
Former Programmers: Maksim Struchalin

[1]PolyOmica, Groningen, The Netherlands
[2]Erasmus MC, Rotterdam, The Netherlands
[3]Institute of Cytology and Genetics SD RAS, Novosibirsk

May 4, 2016

Contents

1 Motivation

Many statistical and experimental techniques, such as imputations and high-throughput sequencing, generate data which are informative for genome-wide association analysis and are probabilistic in the nature.

When we work with directly genotyped markers using such techniques as SNP or microsatellite typing, we would normally know the genotype of a particular person at a particular locus with very high degree of confidence, and, in case of biallelic marker, can state whether the genotype is AA, AB or BB.

On the other hand, when dealing with imputed or high-throughput sequencing data, the genotypic status of the person is known with a much lower confidence. Instead of dealing with known genotypes we work with a probability distribution that is based on observed information, and we have estimates that the true underlying genotype is either AA, AB or BB. The degree of confidence about the real status is measured with the probability distribution $\{P(AA), P(AB), P(BB)\}$.

Several techniques may be applied to analyse such data. The most simplistic approach would be to pick up the genotype with highest probability, i.e. $\max_g[P(g = AA), P(g = AB), P(g = BB)]$ and then analyse the data

as if directly typed markers were used. The disadvantage of this approach is that it does not take into account the probability distribution – i.e. the uncertainty about the true genotypic status. Such analysis is statistically wrong: the estimates of association parameters (regression coefficients, odds or hazard ratios, etc.) are biased, and the bias becomes more pronounced with greater probability distribution uncertainty (entropy).

One of the solutions that generate unbiased estimates of association parameters and takes the probability distribution into account is achieved by performing association analysis by means of regression of the outcome of interest onto estimated genotypic probabilities.

The `ProbABEL` package was designed to perform such regression in a fast, memory-efficient and, consequently, genome-wide feasible manner. Currently, `ProbABEL` implements linear and logistic regression, as well as the Cox proportional hazards model. The corresponding analysis programs are called `palinear`, `palogist`, and `pacoxph`.

For more information, please have a look at the GenABEL project website at http://www.genabel.org. The `ProbABEL`-specific home page is located at http://www.genabel.org/packages/ProbABEL. For user support questions, please turn to our forum at http://forum.genabel.org. Bugs in `ProbABEL` can be reported in the GenABEL project bug tracker at https://r-forge.r-project.org/tracker/index.php?group_id=505&atid=2058.

2 Obtaining and installing `ProbABEL`

`ProbABEL` is a tool that is mostly used on computers running the Linux operating system. We try to publish binary packages for Windows as well, but these aren't tested. We strongly suggest using `ProbABEL` on Linux.

2.1 Precompiled packages

`ProbABEL` can be obtained in several ways:

- If you are using Ubuntu Linux and have administrative rights on the machine you can add the GenABEL PPA to your APT configuration and install it from there. The PPA can be found at https://launchpad.net/~l.c.karssen/+archive/genabel-ppa. Instructions on how to add the PPA can also be found there.

- If your computer runs Debian Linux (and you have administrative rights on it), you can install ProbABEL like this:

```
user@server:~$ sudo apt install probabel ↩
   probabel-examples
```

Note that since Debian has a relatively slow release cycle the ProbABEL version in the Debian repositories may be relatively old[a].

- Zip files with pre-compiled binaries (if available) can be found on the ProbABEL web page (http://www.genabel.org/packages/ProbABEL).
- If you don't fall in any of the aforementioned categories[b], you can install `ProbABEL` manually by downloading the source code of the latest version from the website and compiling it yourself. This will be explained in section 2.2.

2.2 Obtaining the source code and compiling it yourself

If you can't use any of the aforementioned pre-compiled packages, you can download the source code of `ProbABEL` yourself, compile it and run it from your own home directory. This section details the steps you need to take. More information can be found in the `doc/INSTALL`.

On the ProbABEL website you can find the link to the latest version of the source code of `ProbABEL` in a `tar.gz` file[c]. A `.asc` file with the same base name as the source code archive is also provided. This file contains a so-called GPG signature of the `tar.gz` file. Using this file and the `gpg` tool you can verify the authenticity of the source code by typing this command on the command line of a Linux shell[d]:

```
user@server:~$ gpg --verify probabel-0.4.3.tar.gz.asc
gpg: Signature made Thu Jan  2 02:38:25 2014 CET using DSA key ID DA9CD509
gpg: Good signature from "L.C. Karssen (GPG key for personal stuff) ↩
    <lennart@karssen.org>"
gpg:                 aka "L.C. Karssen (My GMail address) ↩
    <l.c.karssen@gmail.com>"
```

Notice the "Good signature" message and the fact that the package was signed by Lennart Karssen, the ProbABEL maintainer. If a malicious hacker would have replaced the source code file (for example with one including a

[a] The Debian package tracker (https://tracker.debian.org/pkg/probabel) lists the `ProbABEL` versions currently in the Debian stable, testing and unstable releases.

[b] We know that many people have use Red Hat Linux, CentOS, Scientific Linux or any other Red Hat derivative. Unfortunately we haven't got `rpm` files yet. Any help in creating those will be highly appreciated.

[c] The `tar.gz` file archive format is the most commonly used format for distributing source code on Linux/UNIX systems. These are compressed files, similar to `zip` files.

[d] The $ sign indicates the end of the command line prompt. You don't need to type it.

virus), he won't be able to sign the package using the same key (with key ID DA9CD509). If, for some reason, the `tar.gz` file has changed (e.g. by such a hacker or because the file didn't get downloaded correctly) you will see output like this (notice the "BAD signature" message):

```
user@server:~$ gpg --verify probabel-0.4.2.tar.gz.asc
gpg: Signature made Thu Jan  2 02:38:25 2014 CET using DSA key ID DA9CD509
gpg: BAD signature from "L.C. Karssen (GPG key for personal stuff) ↩
    <lennart@karssen.org>"
```

Before continuing, it is important to mention that `ProbABEL` needs the Eigen library[e]. The required source code for Eigen v3.2.1 is included in the ProbABEL `.tar.gz` file in the `src/eigen-3.2.1` directory.

Now it's time to extract the `ProbABEL` source code and move into the directory that is created:

```
user@server:~$ tar -xzf probabel-0.5.0.tar.gz
user@server:~$ cd probabel-0.5.0
```

With the following command we will indicate where we want to install `ProbABEL`. Let's install in a subdirectory of your home directory, e.g. `/home/yourusername/ProbAB`

```
user@server:~$ ./configure \
   --prefix=/home/yourusername/ProbABEL/
```

This will be followed by a series of checks to see if all tools required for compilation and installation are present on your system. If you don't see any warnings you can continue to compile[f] the code using the `make` command[g] The next step will check the compiled code, after wich you install the program, documentation and examples to the directory you specified previously with the `--prefix` argument to the `./configure` command.

```
user@server:~$ make
user@server:~$ make check
user@server:~$ make install
```

Note that each of these steps will scroll a lot of output on the screen. Please watch it for any warnings or errors. Please ask any questions on our support forum.

[e]Eigen is a library for fast matrix multiplication. The Eigen website can be found at http://eigen.tuxfamily.org. In ProbABEL versions before v0.5.0 Eigen was not required (but still recommended).

[f]Compilation is the process of converting the source files containing human readable program code to a files with machine readable instructions.

[g]If you work on a machine with multiple processors (or processor cores), which should be the case on modern servers, but also on most PCs, you can speed up the process by adding this number to the `-j` option. For example for four cores run `make -j4`.

If all went well you will find the executable programs (`palinear`, `palogist`, and `pacoxph`) in the directory `/home/yourusername/ProbABEL/bin/`. You are now ready to analyse your data!

3 Input files

`ProbABEL` takes three files as input: a file containing SNP information (e.g. the MLINFO file of MaCH), a file with genome- or chromosome-wide predictor information (e.g. the MLDOSE or MLPROB file of MaCH or `minimac`), and a file containing the phenotype of interest and covariates.

Optionally, the map information can be supplied (e.g. the "legend" files of HapMap).

The dose/probability file may be supplied in filevector format in which case `ProbABEL` will operate much faster, and in low-RAM mode (approx. 128 MB). See the R libraries `GenABEL` and `DatABEL` on how to convert MaCH and IMPUTE2 files to filevector format (functions: `mach2databel()` and `impute2databel()`, respectively).

3.1 SNP information file

In the simplest scenario, the SNP information file is an MLINFO file generated by MaCH/`minimac`. This must be a space or tab-delimited file containing SNP name, coding for allele 1 and 2 (e.g. A, T, G or C), frequency of allele 1, minor allele frequency and two quality metrics ("Quality", the average maximum posterior probability and "Rsq", the proportion of variance decrease after imputations).

Actually, for `ProbABEL`, it (almost) does not matter what is written in this file – this information is simply copied to the output. However, **it is critical** that the number of columns is seven[h] and the number of lines in the file is equal to the number of SNPs in the corresponding DOSE file (plus one for the header line). Also make sure that the "Rsq" column contains values $> 1 \cdot 10^{-16}$, otherwise you will end up with β's set to `nan`.

The example of SNP information file content follows here (also to be found in `ProbABEL/examples/test.mlinfo`)

```
SNP Al1 Al2 Freq1 MAF Quality Rsq
rs7247199 G A 0.5847 0.4150 0.9299 0.8666
rs8102643 C T 0.5847 0.4150 0.9308 0.8685
```

[h] This means that for `minimac` output files the number of columns needs to be reduced. This can be done using e.g. GAWK or `cut`.

```
rs8102615 T A 0.5006 0.4702 0.9375 0.8932
rs8105536 G A 0.5783 0.4213 0.9353 0.8832
rs2312724 T C 0.9122 0.0877 0.9841 0.9232
```

Note that a header line is present in the file. The file describes five SNPs.

3.2 Genomic predictor file

Again, in the simplest scenario this is an MLDOSE or MLPROB file generated by MaCH and/or `minimac`. Such file starts with two special columns plus, for each of the SNPs under consideration, a column containing the estimated allele 1 dose (MLDOSE). In an MLPROB file, two columns for each SNP correspond to posterior probability that person has two ($P_{A_1A_1}$) or one ($P_{A_1A_2}$) copies of allele 1. The first "special" column is made of the sequential id, followed by an arrow followed by study ID (the one specified in the MaCH input files). The second column contains the method keyword (e.g. "MLDOSE").

An example of the few first lines of an MLDOSE file for five SNPs described in SNP information file follows here (also to be found in the file `ProbABEL/examples/test.mldose`)

```
1->id636728 MLDOSE 0.974 0.974 0.968 0.971 2
2->id890314 MLDOSE 0.947 0.947 0.113 0.944 1.094
3->id102874 MLDOSE 1.005 1.004 NaN 1.002 2
4->id200949 MLDOSE 1.968 1.969 1.973 1.977 2
5->id336491 MLDOSE 1.007 1.006 1.001 1.004 2
6->id988766 MLDOSE 1.006 1.006 1 1.003 2
7->id21999 MLDOSE 1.968 1.969 1.973 1.977 2
8->id433893 MLDOSE 1.006 1.006 1.001 1.004 2
9->id688932 MLDOSE 1.006 1.006 1.001 1.004 2
10->id394203 MLDOSE 1.967 1.968 1.972 1.976 1.999
11->id995678 MLDOSE 1.014 1.014 1.006 1.009 2
```

The order of SNPs in the SNP information file and DOSE or PROB file must be the same. This should be the case if you just used MaCH/`minimac` outputs. Consequently, the number of columns in the genomic predictor file must be the same as the number of lines in the SNP information file plus one in the case of a DOSE file. Similarly, for a PROB file the number of columns must be equal to two times the number of SNPs plus 1.

The dose/probability file may be supplied in filevector format (`.fvi` and `.fvd` files) in which case `ProbABEL` will operate much faster, and in low-RAM

mode (approx. 128 MB). On the command line simply specify the `.fvi` file as argument for the `--dose` option (cf. section 4 for more information on the options accepted by `ProbABEL`). See the R libraries `GenABEL` and `DatABEL` on how to convert MaCH and IMPUTE files to filevector format (functions: `mach2databel()` and `impute2databel()`, respectively).

3.3 Phenotypic file

The phenotypic data file contains phenotypic data, but also specifies the analysis model. There is a header line, specifying the variable names. The first column should contain personal study IDs. It is assumed that **both the total number and the order of these IDs are exactly the same as in the genomic predictor (DOSE/PROB) file described in previous section**. This is not difficult to arrange using e.g. `R`; an example is given in the `examples` directory.

Missing data should be coded with 'NA', 'N' or 'NaN' codes. Any other coding will be converted to some number which will be used in analysis! E.g. coding missing as '-999.9' will result in an analysis which will consider -999.9 as indeed a true measurements of the trait/covariates.

In the case of linear or logistic regression (programs `palinear` and `palogist`, respectively), the second column specifies the trait under analysis, while the third, fourth, etc. provide information on covariates to be included into analysis. As an example, a few lines of a phenotypic information file designed for linear regression analysis follow here (also to be found in `examples/height.txt`)

```
id height sex age
id636728 174.429795159687 0 56.5664877162697
id890314 168.176943059097 0 74.8311971509938
id102874 178.612190619767 1 45.2478051768211
id200949 171.770230117638 0 46.7362651142108
id336491 185.941629656499 1 61.2743318817997
id988766 173.159286450017 1 43.9794924518567
id21999 167.478282481124 0 64.842094190157
id433893 168.33178468379 1 49.2526444099125
id688932 171.691587811178 0 50.3954417563357
id394203 173.491495887183 1 71.6498502881161
```

Note again that the order of IDs is the same in the MLDOSE file and the phenotypic data file. The model specified by this file is

$$\text{height} \sim \mu + \text{sex} + \text{age},$$

where μ is the intercept.

Clearly, you can for example include `sex` $\times$ `age` interaction terms by specifying another column having a product of sex and age here.

For logistic regression, it is assumed that in the second column cases are coded as "1" and controls as "0". An couple of example lines of a phenotypic information file designed for logistic regression analysis follow here (also to be found in `examples/logist_data.txt`)

```
id chd sex age othercov
id636728 0 0 56.5664877162697 -0.616649220436139
id890314 0 0 74.8311971509938 0.695315865158652
id102874 1 1 45.2478051768211 -0.919192364890525
id200949 0 0 46.7362651142108 -0.623212536893650
id336491 0 1 61.2743318817997 -0.0835744351009496
id988766 0 1 43.9794924518567 -0.360419162609288
id21999 1 0 64.842094190157 -0.180940346913155
id433893 0 1 49.2526444099125 0.126374731789777
id688932 0 0 50.3954417563357 1.06437576032067
id394203 1 1 71.6498502881161 -1.18226498491599
```

You can see that in the first 10 people, there are three cases, as indicated by "chd" equal to one. The model specified by this file is

$$\text{chd} \sim \mu + \text{sex} + \text{age} + \text{other cov.}$$

In case of the Cox proportional hazards model, the composition of the phenotypic input file is a bit different. In the second column and third column, you need to specify the outcome in terms of follow-up time (column two) and event (column three, "1" if an event occurred and zero if censored). Columns starting from four (inclusive) specify covariates to be included into the analysis. An example few lines of a phenotypic information file designed for the Cox proportional hazards model analysis follow here (also to be found in `examples/coxph_data.txt`)

```
id fupt_chd chd sex age othercov
id636728 3.187930645 0 0 56.56648772 -0.61664922
id890314 2.099691952 0 0 74.83119715 0.695315865
id102874 9.133488079 1 1 45.24780518 -0.919192365
id200949 7.525406804 0 0 46.73626511 -0.623212537
id336491 6.798229522 0 1 61.27433188 -0.083574435
id988766 6.149545358 0 1 43.97949245 -0.360419163
id21999 1.013546103 1 0 64.84209419 -0.180940347
```

```
id433893 1.282853098 0 1 49.25264441 0.126374732
id688932 8.340206657 0 0 50.39544176 1.06437576
id394203 3.392345681 1 1 71.64985029 -1.182264985
```

You can see that for the first ten people, the event occurs for three of them, while for the other seven there is no event during the follow-up time, as indicated by the "chd" column. Follow-up time is specified in the preceding column. The covariates included into the model are age (presumably at baseline), sex and "othercov"; thus the model, in terms of `R/survival` is

$$\text{Surv}(\text{fuptime\_chd}, \text{chd}) \sim \text{sex} + \text{age} + \text{other cov.}$$

3.4 Optional map file

If you would like map information (e.g. base pair position) to be included in your outputs, you can supply a map file. These follow HapMap "legend" file format. For example, for the five SNPs we considered the map-file may look like (example can be found in `examples/test.map`)

```
rs position 0 1
rs7247199 204938 A G
rs8102643 207859 C T
rs8102615 211970 A T
rs8105536 212033 A G
rs2312724 217034 C T
```

The order of the SNPs in the map file should follow that in the SNP information file. Only information from the second column – the SNP location – is actually used to generate the output.

4 Running an analysis

To run linear regression, you should use the program called `palinear`; for logistic analysis use `palogist`, and for the Cox proportional hazards model use `pacoxph` (all are found in the `bin/` directory after you have compiled the program).

There are in total 11 command line options you can specify to the `ProbABEL` analysis functions `palinear` or `palogist`. If you run either program with the `--help` option, you will get a short explanation to the command line options:

```
user@server:~$ palogist --help
probabel v. 0.5.0
(C) Yurii Aulchenko, Lennart C. Karssen, Maarten Kooyman, Maksim ↩
    Struchalin, The GenABEL team, EMC Rotterdam

Using EIGEN version 3.2.0 for matrix operations

Usage: src/palogist options
Options:
     --pheno    : phenotype file name
     --info     : information (e.g. MLINFO) file name
     --dose     : predictor (e.g. MLDOSE/MLPROB) file name
     --map      : [optional] map file name
     --nids     : [optional] number of people to analyse
     --chrom    : [optional] chromosome (to be passed to output)
     --out      : [optional] output file name (default is regression.out.txt)
     --skipd    : [optional] how many columns to skip in the predictor
                  (dose/prob) file (default 2)
     --ntraits  : [optional] how many traits are analysed (default 1)
     --ngpreds  : [optional] how many predictor columns per marker
                   (default 1 = MLDOSE; else use 2 for MLPROB)
     --separat  : [optional] character to separate fields (default is space)
     --flipmaf  : [optional] swap/flip reference and effect allele based ↩
         on MAF
     --score    : use score test
     --no-head  : do not report header line
     --allcov   : report estimates for all covariates (large outputs!)
     --interaction: Which covariate to use for interaction with SNP ↩
         analysis (default is no interaction, 0)
     --mmscore  : score test in samples of related individuals. File with ↩
         inverse of variance-covariance matrix (for palinear) or inverse ↩
         correlation (for palogist) as input parameter
     --robust   : report robust (aka sandwich, aka Hubert-White) standard ↩
         errors
     --help     : print help
```

More information on the options can also be found in the manual page of each of the programs. For example, to access the manual page for `palinear` on a Linux system run[i]:

```
user@server$ man palinear
```

4.1 Basic analysis options

However, for a simple run only three options are mandatory, which specify the necessary files needed to run the regression analysis.

These options are `--dose` (or `-d`), specifying the genomic predictor/MLDOSE file described in section 3.2; `--pheno` (or `-p`), specifying the phenotypic data file described in section 3.3; and `--info` (or `-i`), specifying the SNP information file described in section 3.1.

[i] Use the `q` key to quit the man page viewer.

If you change to the `examples` directory you can run an analysis of height by running

```
palinear -p height.txt -d gtdata/test.mldose -i gtdata/test.mlinfo
```

Output from the analysis will be stored in the `regression.out.csv` file. The analysis of a binary trait (e.g. chd) can be run with

```
palogist -p logist_data.txt -d gtdata/test.mldose \
    -i gtdata/test.mlinfo
```

To run a Cox proportional hazards model, try

```
pacoxph -p coxph_data.txt -d gtdata/test.mldose \
    -i gtdata/test.mlinfo
```

Please have a look at the shell script files `example_qt.sh`, `example_bt.sh` and `example_all.sh` to have a better overview of the analysis options.

To run an analysis with MLPROB files, you need specify the MLPROB file with the `-d` option and also specify that there are two genetic predictors per SNP, e.g. you can run a linear model with

```
palinear -p height.txt -d gtdata/test.mlprob -i gtdata/test.mlinfo \
    --ngpreds=2
```

When using genomic predictor files (dosages or probabilities) stored in filevector (a.k.a. DatABEL) format (i.e. a combination of `.fvi` and `.fvd` files) you can specify these like you would with ordinary text files. This is how the previous example would change:

```
palinear -p height.txt -d gtdata/test.mlprob.fvi -i gtdata/test.mlinfo \
    --ngpreds=2
```

4.2 Advanced analysis options

4.2.1 `--mmscore`

With the option `--mmscore` a score test for association between a trait and genetic polymorphisms in samples of related individuals is performed. For quantitative traits (`palinear`) a file with the inverse of the variance-covariance matrix goes as input parameter with that option, e.g. `--mmscore` ↩ `<filename>`. The file has to contain the first column with id names exactly like in phenotype file, BUT OMITTING people with no measured phenotype. The rest is a matrix. The phenotype file in case of using the `--mmscore` argument may contain any amount of covariates (this is different from previous

versions). The first column contains id names, the second the trait. The others are covariates.

For binary traits (`palogist`) the file should contain the inverse of the correlation matrix. **Note: this is an experimental feature!** This matrix can be obtained through (in GenABEL notation):

```
h2.obj$InvSigma * h2.obj$h2an$estimate[length(h2.obj$h2an$estimate)]
```

where `h2.obj` is the object returned by GenABEL's `polygenic()`. The GenABEL documentation explains this procedure in more detail.

An example of how a polygenic object estimated by `GenABEL` can be used with ProbABEL is provided in `examples/mmscore.R`

Though technically `--mmscore` allows for inclusion of multiple covariates, these should be kept to minimum as this is a score test. We suggest that any covariates explaining an essential proportion of variance should be fit as part of `GenABEL`'s `polygenic` procedure.

4.2.2 `--flipmaf`

The `--flipmaf` option flips the reference and effect alleles according to Minor Allele Frequency (MAF) such that the minor allele is the effect/predictor allele. If this option is set, ProbABEL will check for each genetic variant whether the `Freq1` column in the info file (see the `--info` option) is > 0.5 and if that is indeed the case the probabilities/dosage of that variant will be flipped such that `A1` and `A2` are interchanged. In order to easily identify the variants that were flipped[j] enabling the `flipmaf` option will automatically add a column called `AllelesFlipped` to the output file(s), indicating whether the alleles were flipped (`1`) or not (`0`).

Using this option may help in case or moderately rare variants where one of the homozygote classes has only a few observations. In this case (particularly with the Cox proportional hazards model), using the rare homozygote class as reference can lead to numeric problems while performing the regression leading to `NaN`s in the output. Using the `--flipmaf` option can help to mitigate this issue.

Assuming alleles A and B, flipping the allele coding has the following effect on the following genetic models:

- additive model: $\beta_{\text{SNP}} \rightarrow -\beta_{\text{SNP}}$
- dominant model: $\beta_{\text{SNP, dominant}} \rightarrow -\beta_{\text{SNP, recessive}}$

[j] One could of course simply do the comparison on the `Freq1` column oneself.

- recessive model: $\beta_{\text{SNP, recessive}} \rightarrow -\beta_{\text{SNP, dominant}}$
- 2df/genotypic model: One of the βs will be flipped, the other one will be raised/lowered.

To better observe what happens, one could use the `--allcov` option (cf. 4.2.3) to output the effect sizes and corresponding standard errors of all covariates. This will show how the flipping of the alleles changes the value of the offset (mean, μ) term of the regression equation.

4.2.3 `--allcov`

Specifying the `--allcov` option will output the effect size (β) and corresponding standard error for each of the covariates (in addition to those for the SNP term that are output by default). For example, a linear regression, additive model with the equation

$$Y \sim \mu + \text{sex} + \text{age} + \text{SNP},$$

the header of the output file will be:

```
name A1 A2 Freq1 MAF Quality Rsq n Mean_predictor_allele chrom position ↩
    beta_mu sebeta_mu beta_sex sebeta_sex beta_age sebeta_age ↩
    beta_SNP_addA1 sebeta_SNP_addA1 chi2_SNP_add
```

Note the `beta_mu`, `sebeta_mu` etc. terms. Also note that using this option can lead to very large outputs.

4.2.4 `--interaction`

The option `--interaction` allows you to include interaction between SNPs and any covariate. If for example your model is

$$\text{trait} \sim \text{sex} + \text{age} + \text{SNP},$$

running the program with the option `--interaction=2` will model

$$\text{trait} \sim \text{sex} + \text{age} + \text{SNP} + \text{age} \times \text{SNP}.$$

In this case, the `chi2_SNP` column in the output file contains the χ^2_2 statistic taken from the Likelihood ratio test. This LRT statistic is calculated based on a full model that contains all terms (sex, age, SNP and age×SNP). The null model does not contain the terms that include the SNP.

4.2.5 `--robust`

The option `--robust` allows you to compute so-called "robust" (a.k.a. "sandwich", a.k.a. Hubert-White) standard errors (cf. section 8 "Methodology" for details).

4.3 Running multiple analyses at once: `probabel`

The `bin/probabel` script is a handy wraper for the `ProbABEL` functions. To start using it the configuration file `etc/probabel_config.cfg.example` needs to be edited and renamed to `etc/probabel_config.cfg`. The configuration file consists of five columns, separated by commas. Each column except the first is a pattern for files produced by MaCH or `minimac` (imputation tools). The column named "cohort" is an identifying name of a population ("STUDY_1" in the example), the column "info_path" is the full path to "info" files, including a pattern where the chromosome number has been replaced by `_._chr_._`. In case the imputations were run on chunks of chromosomes, the pattern `_._chunk_._` will be replaced with the corresponding chunk number. Chunk numbers should start at 1 for each chromosome. The columns "dose_path", "prob_path" and "legend_path" are paths and patterns for "dose", "prob" and "legend" files, respectively. These also need to include the pattern for the chromosome as used in the column for the "info" files. Empty lines and lines starting with a # are ignored.

The `make install` installation procedure should have set all paths in the `probabel` script correctly. If that is not the case you will have to change the variable `$config` in the script to point to the full path of the configuration file and the variables `$base_path` and `@anprog` to point the full path to the `ProbABEL` scripts.

5 Output file format

Let us consider what comes out of the linear regression analysis described in the previous section. After the analysis has run, in the output file you will find something like

```
name A1 A2 Freq1 MAF Quality Rsq n Mean_predictor_allele chrom
+>  position beta_SNP_add sebeta_SNP_add chi2_SNP
rs7247199 G A 0.5847 0.415 0.9299 0.8666 182 0.564439 19
+>  204938 -0.218693 0.734966 0.0905063
rs8102643 C T 0.5847 0.415 0.9308 0.8685 182 0.564412 19
+>  207859 -0.218352 0.734214 0.0904094
```

Here, only the first three lines of output have been shown. Note that lines starting with `+>` are actually the ones continuing the previous line – they have just been wrapped so we can see these long lines.

The header provides a short description of what can be found in a specific column. The first column provides the SNP name and next six are descriptions which were taken directly from the SNP information file. Therefore, these describe allele frequencies and the quality in your total imputations, not necessarily in the data under analysis.

In contrast, starting with the next column, named `n`, the output concerns the data analysed. Column 8 (`n`) tells the number of subjects for whom complete phenotypic information was available. At this point, unless you have complete measurements on all subjects, you should feel alarmed if the number here is exactly the number of people in the file – this may indicate you did not code missing values according to `ProbABEL` format ('NA', 'NaN', or 'N').

The next column, nine ("Mean_predictor_allele"), gives the estimated frequency of the predictor allele (`A1`) in subjects with complete phenotypic data.

If the `--chrom` option was used, in the next column you will find the value specified by this option. If `--map` option was used, in the subsequent column you will find map location taken from the map-file. The subsequent columns provide coefficients of regression of the phenotype onto the genotype (β), corresponding standard errors (SE_β), and the χ^2 test value based on the likelihood ratio test. Note that for the additive, recessive, dominant and overdominant genetic models this is a χ^2 of 1 degree of freedom (df), whereas for the genotypic model this is a χ^2 of 2df. If the `--mmscore` option is used, the χ^2 values are calculated from the Wald statistic (1df)[k].

6 Preparing input files

After installing `ProbABEL` you can find the `prepare_data.R` file in the `scripts` directory. It is an R script that arranges phenotypic data in the right format. Please read this script for details.

[k]For the genotypic (2df) model the χ^2 values can't be simply calculated using the Wald statistic, consequently the χ^2 values are set to `nan`. A fix for this needs to be implemented still.

7 Memory use and performance

Maximum likelihood regression is implemented in `ProbABEL`. With 6,000 people and 2.5 million SNPs, a genome-wide scan is completed in less that an hour for a linear model with 1-2 covariates and overnight for logistic regression or the Cox proportional hazards model (figures for a PC bought back in 2007).

Memory may be an issue with `ProbABEL` if you use MaCH/`minimac` text dose/probability files, e.g. for large chromosomes, such as chromosome one consumed up to 5 GB of RAM with 6,000 people.

We suggest that the genotype dosage/probability file is to be supplied in filevector format in which case `ProbABEL` will operate about 2-3 times faster, and in low-RAM mode (approx. 128 MB). See the R libraries `GenABEL` and `DatABEL` on how to convert MaCH and IMPUTE files to filevector format (functions: `mach2databel()` and `impute2databel()`, respectively).

When the `--mmscore` option is used, the analysis takes more time.

8 Methodology

8.1 Analysis of population-based data

8.1.1 Linear regression assuming normal distribution

Standard linear regression theory is used to estimate coefficients of regression and their standard errors. We assume a linear model with expectation

$$E[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta} \tag{1}$$

and variance-covariance matrix

$$\mathbf{V} = \sigma^2\mathbf{I},$$

where $\mathbf{Y}$ is the vector of phenotypes of interest, $\mathbf{X}$ is the design matrix, $\boldsymbol{\beta}$ is the vector of regression parameters, σ^2 is the variance and $\mathbf{I}$ is the identity matrix.

The maximum likelihood estimates (MLEs) for the regression parameters are given by

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y} \tag{2}$$

and the MLE of the residual variance is

$$\hat{\sigma}^2 = \frac{(\mathbf{Y} - \mathbf{X}\hat{\beta})^T(\mathbf{Y} - \mathbf{X}\hat{\beta})}{N - r_X}, \tag{3}$$

where N is the number of observations and r_X is the rank of $\mathbf{X}$ (i.e. the number of columns of the design matrix).

The variance-covariance matrix for the parameter estimates under alternative hypothesis can be computed as

$$\mathbf{var}_{\hat{\beta}} = \hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}. \tag{4}$$

For the j-the element $\hat{\beta}(j)$ of the vector of estimates the standard error under the alternative hypothesis is given by the square root of the corresponding diagonal element of the above matrix, $\mathbf{var}_{\hat{\beta}}(jj)$, and the Wald test can be computed with

$$T^2(j) = \frac{\hat{\beta}(j)^2}{\mathbf{var}_{\hat{\beta}}(jj)},$$

which asymptotically follows the χ^2 distribution with one degree of freedom under the null hypothesis.

When testing significance for more than one parameter simultaneously, several alternatives are available. Let us first partition the vector of parameters into two components, $\beta = (\beta_g, \beta_x)$, and our interest is testing the parameters contained in β_g (SNP effects), while β_x (e.g. effects of sex, age, etc.) are considered nuisance parameters. Let us define the vector of the parameters of interest which are fixed to certain values under the null hypothesis as $\beta_{g,0}$.

Firstly, the likelihood ratio test can be obtained with

$$LRT = 2\left(\text{logLik}(\hat{\beta}_g, \hat{\beta}_x) - \text{logLik}(\beta_{g,0}, \hat{\beta}_x)\right),$$

which under the null hypothesis is asymptotically distributed as χ^2 with the number of degrees of freedom equal to the number of parameters specified by β_g. Assuming the normal distribution, the log-likelihood of a model specified by the vector of parameters β and residual variance σ^2 can be computed as

$$\text{logLik}(\beta, \sigma^2) = -\frac{1}{2}\left(N \cdot \log_e \sigma^2 + (\mathbf{Y} - \beta\mathbf{X})^T(\mathbf{I}/\sigma^2)(\mathbf{Y} - \beta\mathbf{X})\right).$$

Secondly, the Wald test can be used; for that the inverse variance-covariance matrix of $\hat{\beta}_g$ should be computed as

$$\mathbf{var}^{-1}_{\hat{\beta}_g} = \mathbf{var}^{-1}_{\hat{\beta}}(g,g) - \mathbf{var}^{-1}_{\hat{\beta}}(g,x)\left(\mathbf{var}^{-1}_{\hat{\beta}}(x,x)\right)^{-1}\mathbf{var}^{-1}_{\hat{\beta}}(x,g),$$

where $\mathbf{var}^{-1}_{\hat{\beta}}(a,b)$ correspond to sub-matrices of the inverse of the variance-covariance matrix of $\hat{\beta}$, involving either only parameters of interest (g,g), nuisance parameters (x,x) or combination of these (x,g), (g,x).

The Wald test statistics is then computed as

$$W^2 = (\hat{\beta}_g - \beta_{g,0})^T \ \mathbf{var}_{\hat{\beta}_g}^{-1} (\hat{\beta}_g - \beta_{g,0}),$$

which asymptotically follows the χ^2 distribution with the number of degrees of freedom equal to the number of parameters specified by β_g. The Wald test generally is computationally easier than the LRT, because it avoids estimation of the model specified by the parameter's vector $(\beta_{g,0}, \hat{\beta}_x)$.

Lastly, similar to the Wald test, the score test can be performed by use of $\mathbf{var}_{(\beta_{g,0},\hat{\beta}_x)}$ instead of $\mathbf{var}_{\hat{\beta}}$.

8.1.2 Logistic regression

For logistic regression, the procedure to obtain parameters estimates, their variance-covariance matrix, and tests are similar to these outlined above with several modifications.

The expectation of the binary trait is defined as the expected probability of the event as defined by the logistic function

$$E[\mathbf{Y}] = \pi = \frac{1}{1 + e^{-(\mathbf{X}\beta)}}.$$

The estimates of the parameters are obtained not in one step, as is the case of the linear model, but using an iterative procedure (iteratively re-weighted least squares). This procedure is not described here for the sake of brevity.

The log-likelihood of the data is computed using the binomial probability formula:

$$\text{logLik}(\beta) = \mathbf{Y}^T \log_e \pi + (\mathbf{1} - \mathbf{Y})^T \log_e (\mathbf{1} - \pi),$$

where $\log_e \pi$ is a vector obtained by taking the natural logarithm of every value contained in the vector π.

8.1.3 Robust variance-covariance matrix of parameter estimates

For a linear model, these are computed using the equation

$$\mathbf{var}_r = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{R}\mathbf{X})(\mathbf{X}^T\mathbf{X})^{-1},$$

where $\mathbf{R}$ is a diagonal matrix containing squares of residuals of $\mathbf{Y}$. The same equation may be used for "standard" analysis, in which case the elements of the $\mathbf{R}$ matrix are constant, namely the mean residual sum of squares (the estimate of σ^2).

Similar to that, the robust matrix is computed for logistic regression with

$$\mathbf{var}_r = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{R}\mathbf{X})(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1},$$

where $\mathbf{1}$ is the vector of ones and $\mathbf{W}$ is the diagonal matrix of "weights" used in logistic regression.

8.1.4 Cox proportional hazards model

The implementation of the Cox proportional hazard model used in `ProbABEL` is entirely based on the code of `R` library `survival` developed by Thomas Lumley (function `coxfit2`), and is therefore not described here.

Many thanks to Thomas for making his code available under GNU GPL!

8.2 Analysis of pedigree data

The framework for analysis of pedigree data follows the two-step logic developed in the works of Aulchenko *et al.* (2007) and Chen and Abecasis (2007). The general analysis model is a linear mixed model where the expectation of the trait is defined as

$$E[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta},$$

identical to that defined for a linear model (cf. Eq. 1). To account for correlations between the phenotypes of relatives which may be induced by family relations the variance-covariance matrix is defined to be proportional to the linear combination of the identity matrix $\mathbf{I}$ and the relationship matrix $\boldsymbol{\Phi}$:

$$\mathbf{V}_{\sigma^2,h^2} = \sigma^2 \left(2h^2\boldsymbol{\Phi} + (1-h^2)\mathbf{I}\right),$$

where h^2 is the heritability of the trait. The relationship matrix $\boldsymbol{\Phi}$ is twice the matrix containing the coefficients of kinship between all pairs of individuals under consideration; its estimation is discussed separately in section 8.2.2.

Estimation of a model defined in such a way is possible by numerical maximization of the likelihood function, however, the estimation of this model for large pedigrees is laborious, and is not computationally feasible for hundreds of thousands to millions of SNPs to be tested in the context of GWAS, as we have demonstrated previously (Aulchenko *et al.*, 2007).

8.2.1 Two-step score test for association

A two-step score test approach is therefore used to decrease the computational burden. Let us first re-define the expectation of the trait by splitting the design matrix in two parts, the "base" part $\mathbf{X}_x$, which includes all terms not changing across all SNP models fit in GWAS (e.g. effects of sex, age, etc.), and the part including SNP information, $\mathbf{X_g}$:

$$E[\mathbf{Y}] = \mathbf{X}_x\boldsymbol{\beta}_x + \mathbf{X}_g\boldsymbol{\beta}_g.$$

Note that the latter design matrix may include not only the main SNP effect, but e.g. SNP by environment interaction terms.

In the first step, a linear mixed model not including SNP effects

$$E[\mathbf{Y}] = \mathbf{X}_x \boldsymbol{\beta}_x$$

is fitted. The maximum likelihood estimates (MLEs) of the model parameters (regression coefficients for the fixed effects $\hat{\boldsymbol{\beta}}_x$, the residual variance $\hat{\sigma}_x^2$ and the heritability $\hat{h}_x^2$) can be obtained by numerical maximization of the likelihood function

$$\text{logLik}(\beta_x, h^2, \sigma^2) = -\frac{1}{2}\left(\log_e |\mathbf{V}_{\sigma^2,h^2}| + (\mathbf{Y} - \beta_x \mathbf{X}_x)^T \mathbf{V}_{\sigma^2,h^2}^{-1} (\mathbf{Y} - \beta_x \mathbf{X}_x)\right),$$

where $\mathbf{V}_{\sigma^2,h^2}^{-1}$ is the inverse and $|\mathbf{V}_{\sigma^2,h^2}|$ is the determinant of the variance-covariance matrix.

In the second step, the unbiased estimates of the fixed effects of the terms involving SNP are obtained with

$$\hat{\beta}_g = (\mathbf{X}_g^T \mathbf{V}_{\hat{\sigma}^2,\hat{h}^2}^{-1} \mathbf{X}_g)^{-1} \mathbf{X}_g^T \mathbf{V}_{\hat{\sigma}^2,\hat{h}^2}^{-1} \mathbf{R}_{\hat{\beta}_x}, \tag{5}$$

where $\mathbf{V}_{\hat{\sigma}^2,\hat{h}^2}^{-1}$ is the inverse variance-covariance matrix at the point of the MLE estimates of $\hat{h}_x^2$ and $\hat{\sigma}_x^2$, and $\mathbf{R}_{\hat{\beta}_x} = \mathbf{Y} - \hat{\beta}_x \mathbf{X}_x$ is the vector of residuals obtained from the base regression model. Under the null model, the inverse variance-covariance matrix of the parameter's estimates is defined as

$$\mathbf{var}_{\hat{\beta}_g} = \hat{\sigma}_x^2 (\mathbf{X}_g^T \mathbf{V}_{\hat{\sigma}^2,\hat{h}^2}^{-1} \mathbf{X}_g)^{-1}.$$

Thus the score test for joint significance of the terms involving SNP can be obtained with

$$T^2 = (\hat{\beta}_g - \beta_{g,0})^T \, \mathbf{var}_{\hat{\beta}_g}^{-1} (\hat{\beta}_g - \beta_{g,0}),$$

where $\beta_{g,0}$ are the values of parameters fixed under the null model. Under the null hypothesis this test statistic asymptotically follows the χ^2 distribution with the number of degrees of freedom equal to the number of parameters tested. The significance of an individual j-th element of the vector $\hat{\beta}_g$ can be tested with

$$T_j^2 = \hat{\beta}_g^2(j) \, \mathbf{var}_{\hat{\beta}_g}^{-1}(jj),$$

where $\hat{\beta}_g^2(j)$ is the square of the j-th element of the vector of estimates $\hat{\beta}_g$, and $\mathbf{var}_{\hat{\beta}_g}^{-1}(jj)$ corresponds to the j-th diagonal element of $\mathbf{var}_{\hat{\beta}_g}^{-1}$. The latter statistic asymptotically follows χ_1^2.

8.2.2 Estimation of the kinship matrix

The relationship matrix $\mathbf{\Phi}$ used in estimation of the linear mixed model for pedigree data is twice the matrix containing the coefficients of kinship between all pairs of individuals under consideration. This coefficient is defined as the probability that two gametes randomly sampled from each member of the pair are identical-by-descent (IBD), that is they are copies of exactly the same ancestral allele. The expectation of kinship can be estimated from pedigree data using standard methods, for example the kinship for two outbred sibs is 1/4, for grandchild-grandparent is 1/8, etc. For an outbred person, the kinship coefficient is 1/2 – that is two gametes sampled from this person at random are IBD only if the same gamete is sampled. However, if the person is inbred, there is a chance that a maternal and paternal chromosomes are also IBD. The probability of this is characterized by kinship between individual's parents, which is defined as the individual's inbreeding coefficient, F. In this case, the kinship coefficient for the individual is $F + 1/2$. Similar logic applies to computation of the kinship coefficient for other types of pairs in inbred pedigrees.

The kinship matrix can be computed using the pedigree data using standard methods. However, in many cases, pedigree information may be absent, incomplete, or not reliable. Moreover, the estimates obtained using pedigree data reflect the expectation of the kinship, while the true realization of kinship may vary around this expectation. In presence of genomic data it may therefore be desirable to estimate the kinship coefficient from these, and not from pedigree. It can be demonstrated that unbiased and positive semi-definite estimator of the kinship matrix can be obtained (Astle and Balding, 2010; Amin *et al.*, 2007) by computing the kinship coefficients between individuals i and j with

$$\hat{K}_{ij} = \frac{1}{L} \sum_{l=1}^{L} \frac{(g_{l,i} - p_l)(g_{l,j} - p_l)}{p_l(1 - p_l)}$$

where L is the number of loci, p_l is the allelic frequency at l-th locus and $g_{l,j}$ is the genotype of j-th person at the l-th locus, coded as 0, 1/2, and 1, corresponding to the homozygous, heterozygous, and other type of homozygous genotype. The frequency is computed for the allele which, when homozygous, corresponds to the genotype coded as "1".

9 How to cite

If you used ProbABEL for your analysis please give a link to the GenABEL project home page

> http://www.genabel.org/

and cite the ProbABEL paper to give us some credit:

> Aulchenko YS, Struchalin MV, van Duijn CM.
> *ProbABEL package for genome-wide association analysis of imputed data.*
> BMC Bioinformatics. 2010, 11:134.

A proper reference may look like

> For the analysis of imputed data, we used ProbABEL v.0.5.0 from the GenABEL suite of programs (Aulchenko *et al.*, 2010).

If you have used the Cox proportional hazard model, please mention the R package survival by Thomas Lumley. Additionally to the above citation, please include

> The Cox proportional hazards model implemented in ProbABEL makes use of the source code of the R package "survival" as implemented by T. Lumley.

10 Licence

ProbABEL is licenced under the GNU Public Licence (GPL) v2.0, which means it is free/libre open source software, basically allowing you to the freedoms to run, study, share, and modify the software. The full text of the licence can be found in the file doc/COPYING or on https://gnu.org/licenses/old-licenses/gpl-2.0.html.

The Eigen library that is included with ProbABEL is licenced under the Mozilla Public Licence (MPL) v2.0. The full text of this licence can be found in the file lib/eigen-3.2.1/COPYING.MPL2 or on https://www.mozilla.org/en-US/MPL/2.0/.

Index