Software Implementation of Methods to Prioritize Rare Risk Variants

This software package implements the backward elimination procedure and the hierarchical model described in the paper, in order to prioritize rare risk variants in a gene of interest. The software is applicable to case-control designs. The instructions on the input/output files and how to run the programs are given below. Included is also an example dataset containing sequencing data for one gene (VPS13B), part of a larger WES study for autism spectrum disorders.

In an exome-sequencing study, the backward elimination procedure can be applied separately to non-synonymous and synonymous variants. The result will be a list of selected variants that are enriched in causal variants.

For the hierarchical model, various bioinformatics predictors (such as PolyPhen-2, SIFT, GERP etc.) can be incorporated into the analyses. An estimate of the odds ratio is obtained for each variant, and variants can be ranked according to these effect size estimates. Ranking within the list of variants selected by the backward elimination procedure is expected to improve the ranking of truly causal variants.

The software package consists of two programs, one implementing the backward elimination procedure, and one implementing the hierarchical model.

SKATbe: Implementation of the Backwards elimination procedure

Programming language

The SKATbe package is written in R. It was developed and tested in the MAC OSX environment.

Installation

SKATbe can be installed in R by setting the working directory to the directory containing ‘SKATbe_1.0.tar.gz’ and using the command

```
install.packages(“SKATbe_1.0.tar.gz”,type=”source”)
```

SKATbe requires the SKAT package which is available from CRAN.

Input Files

The program accepts the following input files:

pedigreeFile - is a standard pedigree file. The first 6 columns are:

- Family ID
- Individual ID
- Father ID
- Mother ID
- Gender - 1 for male/ 2 for female
- Phenotype - 2 for case/1 for control

  The remaining columns encode the genotypes at individual SNV positions. Each genotype is coded in an additive fashion: 0, 1 or 2 (i.e. the number of minor alleles).

weightsFile - information about the weighs of individual SNVs, one weight per variant (therefore the total number of weights in the file is the number of SNVs in the pedigree file). In the simplest scenario, when all variants are tested each weight is 1. If only non-synonymous variants are tested then weights are 1 for non-synonymous variants and 0 for synonymous variants. The file should contain one weight per line.

The output files

saveBase – The base name for the output files. Further information is appended to the base, so an extension should not be included. For example, if

saveBase=my_results, R will generate files:
my_results_ BIG.ro1 - Gives the full results from the backwards elimination (when applied to the entire list of variants), with the variant eliminated in each step in the first column and the resulting pvalue (for the reduced set) in the second column.

my_results_ BIG.ro1.curr_set.ro1 - The final set of variants selected in the backwards elimination procedure (when applied to the entire list of variants).

my_results .ro1.big_set20.ro1 - The set of variants selected in the backwards elimination for each random subsample of the data (when random subsamples of 20 variants are selected). Each line corresponds to one subsample of 20 variants.

my_results .ro1.big_curr_set.ro1 - The final set of variants selected from each subsample. Each line corresponds to one subsample.

Other parameters:

Base – specifies a base directory to which the other file names are appended.

ro - specifies which test to run: ro=0 SKAT; ro=1 BURDEN.

N.SIMR – specifies the number of random subsamples to use (2000 is recommended, but larger numbers can be used especially for large genes).

Running SKATbe

SKATbe can be run either from an interactive R session or from the command line.

We provide an example of the wrapper shell ***RunSKATbe.R***. Example data from gene VPS13B is also provided and consists of a PED file along with weight files for selecting non-synonymous, synonymous, or all variants.

To run the shell open a command line terminal, navigate to the directory containing the shell, and enter:

```
R --slave --vanilla --args <file.ped> <file.weights> <dir.save>
<base> <ro> <N.SIMR> < RunSKATbe.R
```

For the example dataset included in this package, from the directory containing RunSKATbe.R enter:

```
R --slave --vanilla --args VPS13B.ped weights_VPS13B_nonsyn.txt
VPS13B_results ExampleData 1 2000 < RunSKATbe.R
```

Details on running from an interactive R session are in the documentation in the package and can be viewed with the command help(“skat.be”).

Non-parametric mixture for non-synonymous variants

Finally, to get the non-synonymous variants selected we fit a non-parametric mixture to the return counts calculated above. The R program nonparametric_mixture.R can be used for this purpose.

Next we describe the hierarchical model to estimate the odds ratios of the variants in a gene.

Implementation of the hierarchical modeling approach

Programming language

The hierarchical modeling approach is implemented using SAS.

Installation

The implementation of the hierarchical modeling approach requires the SAS %glimmix macro which is available for download at http://support.sas.com/kb/25/030.html . The main macro performing the analysis is %HM_approach.

Input Files

The %HM_approach macro accepts the following input files (note that the files **Ydata** and **Xdata** can be obtained from the **pedigree file** described in the BE procedure above: the 6th column of the pedigree file describes the phenotype while the remaining columns from 7th column on contain the genotype data; note also that for the hierarchical modeling approach the case control status is defined as 1/0 as opposed to 2/1 for the BE procedure):

dir_input - the directory where the input datasets are stored

dir_output – the directory where the results are to be saved

Ydata - Phenotype - 1 for case/0 for control

Xdata - the genotypes at individual SNV positions. Each genotype is coded in an additive fashion: 0, 1 or 2 (i.e. the number of minor alleles)

Zdata – the higher level covariates to be used in the second stage of the hierarchical model: those could be the different bioinformatic predictors and/or the nonsynonymous versus synonymous status.

cov – YES when individual level confounders are available (to be stored in the Wdata), NO if there are no individual level confounders to adjust for.

Wdata – individual level confounders such as age, gender, etc.

nvarW – number of individual level confounders; if "cov=NO" then nvarW=0.

nvarZ – number of higher level covariates; this is the number of variables in the Zdata file plus 1 to account for the intercept in the second stage model.

maf_rare – the MAF threshold under which the variants are called rare; the code restricts the analysis only to the variants with minor allele frequencies below maf_rare.

IGprior – contains the prior densities of the variance components used by the sampling algorithm for the Bayesian estimation of the variance component. Non-informative Inverse-Gamma distribution with shape and scale parameters both set at 0.01 are used as default.

The output files

out_results – contains summaries of the data (total number of variants, number of variants with MAF below maf_rare; it also prints the matrix of higher level covariates for verification purposes)

log_results – log file describing the different procedures invoked in the implementation; this file should be checked for Errors to ensure the implementation proceeds correctly.

HM_results – the main results file including the estimated z-values, the estimated log odds ratios of the variants, their standard errors, followed by the higher level covariates used in the model as well as the corresponding case control frequencies for each variant.

Running %HM_approach

To run the %HM_approach for all variants with MAF below 0.05 sequenced in the VPS13B gene and using Polyphen-2 and GERP as the higher level covariates (contained in the PolyGerp_Syn0.txt) one can use the following macro call:

```
%HM_approach(dir_input=,dir_output=,Ydata=vps13b_case_contr,
Xdata=vps13b_ped, Zdata=PolyGerp_Syn0, cov=NO,
IGprior=igparameters01,nvarW=0, nvarZ=3, maf_rare=0.05)
```

when no individual level confounders are included in the model or,

```
%HM_approach(dir_input=,dir_output=,Ydata=vps13b_case_contr,
Xdata=vps13b_ped, Wdata=vps13b_Wdata, Zdata=PolyGerp_Syn0, cov=YES,
IGprior=igparameters01,nvarW=0, nvarZ=3, maf_rare=0.05),
```

when individual level confounders are available in the dataset Wdata.