Association Analysis of Sequence Data using Variant Association Tools (VAT)

The Rockefeller University, New York, February 11, 2015

Purpose

Variant Association Tools [VAT, Wang *et al* (2014)][1] was developed to perform quality control and assocation analysis of sequence data. It can also be used to analyze genotype data, e.g. *exome chip* data and imputed data. The software incorporates many rare variant associations methods which include but is not limited to Combined Multivariate Collapsing (CMC)[2], Weighted Sum Statistic (WSS)[3], Kernel Based Adaptive Cluster (KBAC)[4], Variable Threshold (VT)[5] and Sequence Kernel Assocation Test (SKAT)[6].

Resource

Software documentation

http://varianttools.sourceforge.net/Main/Documentation

Genotype data

Exome genotype data was downloaded from the 1000 Genomes pilot data July 2010 release for both the CEU and YRI populations. Only the autosomes are contained in the datasets accompanying this exercise.

The dataset (`CEU.exon.2010_03.genotypes.vcf.gz`, `YRI.exon.2010_03.genotypes.vcf.gz`) is available from:

ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/release/2010_07/exon/snps

Phenotype data

To demonstrate the association analysis, we simulated a quantitative trait phenotype (BMI). Please note that these phenotypes are NOT from the 1000 genome project.

Computation resources

Due to the nature of next-generation sequencing data, a reasonably powerful machine with high speed internet connection is needed to use this tool for real-world applications. For this reason, in this tutorial we will use a small demo dataset to demonstrate association analysis. We will provide pre-annotated variants instead of going into details on variant annotation pipelines.

Data Cleaning and Variant/sample Selection

Getting started

Please navigate to the `VATData` directory and check the available subcommands by typing:

```
vtools -h
```

We use a subcommand system for various data manipulation tasks. This tutorial is mission oriented such that we will focus on a subset of the commands that are relevant to variant-phenotype association analysis, rather than introducing them systematically. For more functionality, please check out our documentation and tutorials online.

Initialize a project

```
vtools init myproj
```

Import variant data

Import all vcf files under the current directory:

```
vtools import *.vcf.gz --var_info DP filter --geno_info DP_geno --build hg18 -j1
```

Logging and Summary

There are four types of messages that will be displayed from the program. "INFO" shows various useful messages during runtime including progress and statistics; "WARNING" shows warning messages including reminders of pitfalls, suspicious pattern in data, etc; "ERROR" shows runtime error messages and forces the program to quit; "DEBUG" displays more detailed runtime information useful for diagnostic purposes. The verbosity level of command line output can be controlled by option `--verbosity`, which are:

- 0: suppress all output except for warnings and errors (no INFO or DEBUG)

- 1: display/log progress information, including progress bars on screen output (no DEBUG)
- 2: display/log progress and debugging information

Verbosity levels are specified by `-vXY` where X= {0,1,2} are verbosity for screen output and Y={0,1,2} are for verbosity levels in the log file. For example, for a production pipeline to avoid any debugging information written to the log file (which will result in a very large log file otherwise) or on screen, use option `-v00` which will suppress most output to the screen and log file. If -v10 is used, "INFO" will be printed to the screen but not the log file, and -v11 will record only INFO lines in the log file.

Summary information for the project can be viewed anytime using the command `vtools show`. Some useful data summary commands are:

```
vtools show project
vtools show tables
vtools show table variant
vtools show samples
vtools show genotypes
vtools show fields
```

Overview of variant and genotype data

Total number of variants

The number of imported variants may be greater than number of lines in the vcf file, because we treat cases where there are two alternative alleles (e.g. A->T/C) as two separate variants.

```
vtools select variant -c
```

There are 6987 variants in our test data.

Genotype Summary

```
vtools show genotypes -v2 > genotypesummary.txt
```

sample_name	filename	num_genotypes	sample_genotype_fields
NA06984	CEU.exon.2010_03.genotypes.vcf.gz	3162	GT,DP_geno
NA06985	CEU.exon.2010_03.genotypes.vcf.gz	3144	GT,DP_geno
NA06986	CEU.exon.2010_03.genotypes.vcf.gz	3437	GT,DP_geno
NA06989	CEU.exon.2010_03.genotypes.vcf.gz	3130	GT,DP_geno
NA06994	CEU.exon.2010_03.genotypes.vcf.gz	3002	GT,DP_geno
NA07000	CEU.exon.2010_03.genotypes.vcf.gz	3388	GT,DP_geno
NA07037	CEU.exon.2010_03.genotypes.vcf.gz	3374	GT,DP_geno
NA07048	CEU.exon.2010_03.genotypes.vcf.gz	3373	GT,DP_geno
NA07051	CEU.exon.2010_03.genotypes.vcf.gz	3451	GT,DP_geno
NA07346	CEU.exon.2010_03.genotypes.vcf.gz	3419	GT,DP_geno
...	...	...	...

Variant overall qualities

The following commands calculate the max/min/avg of total variant depth of coverage.

```
vtools select variant -o "max(DP)"
vtools select variant -o "min(DP)"
vtools select variant -o "avg(DP)"
```

To select only variants having passed all quality filters,

```
vtools select variant "filter='PASS'" -c
```

All 6987 variants have passed the quality filters.

Removal of low quality variants

We should not need to remove any variants based on the filter because all variants in our example dataset pass the quality filters. To demonstrate removal of variants, let us suppose that we demand that variants have a high read depth and those calls with a read depth $\leqslant 15$ will be considered low quality and will be removed.

```
vtools select variant "DP<15" -t to_remove
vtools show fields
vtools show tables
vtools remove variants to_remove -v0
```

Only one variant is removed. Using a combination of select/remove subcommands you can filter out low quality variants easily.

The `vtools show fields`, `vtools show tables`, and `vtools show table variant` commands will allow you to see the new fields, tables, and values, respectively, you have added to the project.

Data summaries

Variant level summaries

The command below will calculate:

- total: Total number of genotypes (GT) for a variant
- num: Total number of alternative alleles across all samples
- het: Total number of heterozygote genotypes 1/0
- hom: Total number of homozygote genotypes 1/1
- other: Total number of double-homozygotes 1/2
- min/max/meanDP_geno: Summaries for depth of coverage and genotype quality across samples

```
vtools update variant --from_stat 'total=#(GT)' 'num=#(alt)' 'het=#(het)' 'hom=#(hom)' \
'other=#(other)' 'minDP=min(DP_geno)' 'maxDP=max(DP_geno)' 'meanDP=avg(DP_geno)'
vtools show fields
vtools show table variant
```

Summaries for different genotype depth (GD) and genotype quality (GQ) filters

The `--genotypes CONDITION` option restricts calculation to genotypes satisfying a given condition. Later we will remove individual genotypes by `DP_geno` filters. The command below will calculate summary statistics on sample genotypes per variant site. It can assist us in determining filtering criteria for genotype call quality.

```
vtools update variant --from_stat 'totalGD10=#(GT)' 'numGD10=#(alt)' 'hetGD10=#(het)' 'homGD10=#(hom)' \
'otherGD10=#(other)' --genotypes "DP_geno > 10"
vtools show fields
vtools show table variant
```

You will notice the change in genotype counts when applying the filter on genotype depth of coverage greater than 10X. Data should now be updated to 6976 variants because only 6976 variants pass the filter of DP_geno>10.

Calculate alternative allele frequency

Calculate observed allele frequency. Notice that the resulting AF will not be accurate for sex chromosomes when there are males in the samples. In our data we do not have this problem.

```
vtools update variant --set "af=num/(total*2.0)"
vtools show fields
vtools show table variant
```

Calculate observed allele frequency if we remove genotypes having genotype depth not exceeding 10.

```
vtools update variant --set "afGD10=numGD10/(totalGD10*2.0)"
```

Compare these allele frequencies

```
vtools output variant chr pos af afGD10 --header --limit 20
```

OUTPUT

```
chr         pos                     af                              afGD10
1         1105366         0.03508771929824561             0.05128205128205128
1         1105411         0.009433962264150943            0.01282051282051282
1         1108138         0.19230769230769232             0.18023255813953487
1         1110240         0.0056179775280898875           0.0
1         1110294         0.228125                        0.2423076923076923
1         3537996         0.8798701298701299              0.8478260869565217
1         3538692         0.041025641025641026            0.043209876543209874
1         3541597         0.0056179775280898875           0.006172839506172839
1         3541652         0.044444444444444446            0.05333333333333334
1         3545211         0.0056179775280898875           0.005813953488372093
...
```

Adding "> `filename.txt`" at the end of the above command will write the output to a file.

Calculate allele frequency for different populations

Our data is imported from two files, a CEU dataset (90 samples) and an YRI dataset (112 samples). To calculate allele frequency for different populations, let us first assign them with an additional RACE phenotype:

```
vtools phenotype --set "RACE=0" --samples "filename like 'YRI%'"
vtools phenotype --set "RACE=1" --samples "filename like 'CEU%'"
vtools show samples
```

OUTPUT

```
sample_name        filename                          RACE
NA06984        CEU.exon.2010_03.genotypes.vcf.gz      1
NA06985        CEU.exon.2010_03.genotypes.vcf.gz      1
NA06986        CEU.exon.2010_03.genotypes.vcf.gz      1
NA06989        CEU.exon.2010_03.genotypes.vcf.gz      1
NA06994        CEU.exon.2010_03.genotypes.vcf.gz      1
NA07000        CEU.exon.2010_03.genotypes.vcf.gz      1
NA07037        CEU.exon.2010_03.genotypes.vcf.gz      1
NA07048        CEU.exon.2010_03.genotypes.vcf.gz      1
NA07051        CEU.exon.2010_03.genotypes.vcf.gz      1
NA07346        CEU.exon.2010_03.genotypes.vcf.gz      1
(192 records omitted, use parameter --limit to see more)
```

Notice that the phenotype "RACE" has been added from the first `vtools show samples` command to the second `vtools show samples` command.

Population specific MAF calculations will be done based on restricted samples (DP_geno>10)

```
vtools update variant --from_stat 'CEU_totalGD10=#(GT)' 'CEU_numGD10=#(alt)' \
--genotypes 'DP_geno>10' --samples "RACE=1"
vtools update variant --from_stat 'YRI_totalGD10=#(GT)' 'YRI_numGD10=#(alt)' \
--genotypes 'DP_geno>10' --samples "RACE=0"
vtools update variant --set "CEU_afGD10=CEU_numGD10/(CEU_totalGD10*2.0)"
vtools update variant --set "YRI_afGD10=YRI_numGD10/(YRI_totalGD10*2.0)"
vtools output variant chr pos afGD10 CEU_afGD10 YRI_afGD10 --header --limit 20
```

3483 variants should be updated for CEU samples and 5167 variants should be updated for the YRI samples.

OUTPUT

```
chr        pos              afGD10                 CEU_afGD10             YRI_afGD10
1        1105366        0.05128205128205128        0.05128205128205128        NA
1        1105411        0.01282051282051282        0.01282051282051282        NA
1        1108138        0.18023255813953487        0.02127659574468085        0.371794871
1        1110240        0.0                        0.0                        NA
1        1110294        0.2423076923076923         0.025                      0.428571428
1        3537996        0.8478260869565217         0.8295454545454546         0.864583333
1        3538692        0.043209876543209874       0.08333333333333333        0.005952380
1        3541597        0.006172839506172839       0.006172839506172839       NA
...
```

You will see some "NA" values within the output because variants found in samples from one population are not always observed in samples from other populations.

Genotype level summaries

Similar operations could be performed on a sample level instead of on a variant level. Please refer to our website wiki page for obtaining genotype level summary information using `vtools phenotype --from_stat`.

```
vtools phenotype --from_stat 'CEU_totalGD10=#(GT)' 'CEU_numGD10=#(alt)' --genotypes 'DP_geno>10' --samples "RACE=1"
vtools phenotype --from_stat 'YRI_totalGD10=#(GT)' 'YRI_numGD10=#(alt)' --genotypes 'DP_geno>10' --samples "RACE=0"
```

OUTPUT

```
180 values of 2 phenotypes (2 new, 0 existing) of 90 samples are updated.
224 values of 2 phenotypes (2 new, 0 existing) of 112 samples are updated.
```

```
vtools phenotype --output CEU_totalGD10 CEU_numGD10 YRI_totalGD10 YRI_numGD10
```

OUTPUT

CEU_totalGD10	CEU_numGD10	YRI_totalGD10	YRI_numGD10
2774	849	NA	NA
1944	570	NA	NA
3386	1029	NA	NA
2659	819	NA	NA
...			
NA	NA	4182	1005
NA	NA	4404	1076
NA	NA	4308	1044
NA	NA	4878	1211

Data Cleaning

Filter data by genotype depth 10

We have calculated various summary statistics using the command `--genotypes 'CONDITION'` but we have not yet removed genotypes having genotype depth of coverage smaller than 10X. The command below removes these genotypes.

```
vtools remove genotypes "DP_geno<10" -v0
```

Select variants by annotated functionality

For aggregated association tests for rare variants, we want to focus on aggregating only the variants having a potential functional contribution to a disease. Annotation is performed using `variant annotation tools` [7] which incorporates ANNOVAR annotation[8].

```
vtools output variant chr pos pos ref alt > allvariants.dat
```

In practice, you will need to run the ANNOVAR pipeline on `allvariants.dat` to obtain the annotations. In this tutorial, however, we have provided the annotated file, `allvariants.dat.exonic_variant_function`, since we are doing excerise on computers that are not suited to this task. If you wish

to run the ANNOVAR pipeline on your data please see http://varianttools.sourceforge.net/Pipeline/Annovar. The command below will allow you to use the variant file which has already been annotated.

```
vtools update variant --format ANNOVAR_exonic_variant_function --from_file allvariants.dat.exonic_variant_function \
--build hg18
```

The following command will output the annotated variants to the screen.

```
vtools output variant chr pos ref alt mut_type --limit 20 --header
vtools show fields
vtools show table variant
```

OUTPUT

```
chr        pos         ref      alt          mut_type
1        1105366        T        C        nonsynonymous SNV
1        1105411        G        A        nonsynonymous SNV
1        1108138        C        T        synonymous SNV
1        1110240        T        A        nonsynonymous SNV
1        1110294        G        A        nonsynonymous SNV
1        3537996        T        C        synonymous SNV
1        3538692        G        C        nonsynonymous SNV
1        3541597        C        T        nonsynonymous SNV
1        3541652        G        A        synonymous SNV
1        3545211        G        A        synonymous SNV
```

Alternatively you may check out the variety of annotation databases we provide:

http://vtools.houstonbioinformatics.org/annoDB/

Due to the database size and computational power constraints, we will not introduce the various annotation databases here. ANNOVAR is sufficient for now.

To select variants by annotation:

```
vtools select variant "mut_type like 'nonsynonymous%' OR mut_type like 'splicing%' OR mut_type like 'stoploss%' OR \
mut_type like 'stopgain%' OR mut_type like 'ncRNA%' " -t v_funct
vtools show tables
```

3653 variants are selected.

Association Tests for Quantitative Traits

Import phenotype data

The aim of the association test is to find variants that have contributions to the phenotype BMI. We simulated BMI values for each of the individuals. To import the phenotype data:

```
vtools phenotype --from_file phenotypes.txt
vtools show samples
```

In order to import phenotype data the file names and sample names in the phenotype file must match the file names and sample names in your project.

OUTPUT

```
sample_name        filename                          RACE        SEX        BMI
NA06984        CEU.exon.2010_03.genotypes.vcf.gz       1        1        36.353
NA06985        CEU.exon.2010_03.genotypes.vcf.gz       1        2        21.415
NA06986        CEU.exon.2010_03.genotypes.vcf.gz       1        1        26.898
NA06989        CEU.exon.2010_03.genotypes.vcf.gz       1        2        25.015
NA06994        CEU.exon.2010_03.genotypes.vcf.gz       1        1        23.858
NA07000        CEU.exon.2010_03.genotypes.vcf.gz       1        2        36.226
NA07037        CEU.exon.2010_03.genotypes.vcf.gz       1        1        32.513
NA07048        CEU.exon.2010_03.genotypes.vcf.gz       1        2        17.57
NA07051        CEU.exon.2010_03.genotypes.vcf.gz       1        1        37.142
NA07346        CEU.exon.2010_03.genotypes.vcf.gz       1        2        30.978
(192 records omitted, use parameter --limit to see more)
```

phenotypes.txt

```
filename                          sample_name        RACE        SEX        BMI
CEU.exon.2010_03.genotypes.vcf    NA06984             1           1         36.353
CEU.exon.2010_03.genotypes.vcf    NA06985             1           2         21.415
CEU.exon.2010_03.genotypes.vcf    NA06986             1           1         26.898
CEU.exon.2010_03.genotypes.vcf    NA06989             1           2         25.015
CEU.exon.2010_03.genotypes.vcf    NA06994             1           1         23.858
CEU.exon.2010_03.genotypes.vcf    NA07000             1           2         36.226
CEU.exon.2010_03.genotypes.vcf    NA07037             1           1         32.513
CEU.exon.2010_03.genotypes.vcf    NA07048             1           2         17.57
CEU.exon.2010_03.genotypes.vcf    NA07051             1           1         37.142
CEU.exon.2010_03.genotypes.vcf    NA07346             1           2         30.978
```

Create sub-projects for association analysis with CEU samples

We want to carry out the association analysis for CEU and YRI separately. It is recommended that we create two projects containing variants and samples for each population. This will greatly improve the computational efficiency. Note that we need to create empty folders to hold each of the projects:

```
vtools select variant --samples "RACE=1" -t CEU
mkdir -p ceu
cd ceu
vtools init ceu --parent ../ --variants CEU --samples "RACE=1"
vtools show project
```

From now on we will only demonstrate analysis of CEU samples (and all the following commands in this chapter will be executed for this project), although the same commands will be applicable for YRI samples. Please use the same commands to analyze the YRI data set. Though you should not analyze the data from different populations together, once you have the p-values from each analysis, you may perform a meta-analysis.

Subset data by AFs

In order to carry out association tests we have to treat common and rare variants separately. The dataset for our tutorial has very small sample size, but with large sample size it is reasonable to define rare variants as having observed MAF<0.01, and common variants as variants having observed MAF>0.05. First, we create variant tables based on calculated alternative allele frequencies for both populations

```
vtools select variant "CEU_afGD10>=0.05 AND CEU_afGD10<=0.99" -t common_ceu
vtools select variant "CEU_afGD10>=0.01 AND CEU_afGD10<=0.99" -t semi_common_ceu
vtools select v_funct "CEU_afGD10<0.01 OR CEU_afGD10>0.99" -t rare_ceu
```

Notice that for selection of rare variants we only keep those that are annotated as functional (chosen from `v_funct` table).

Annotate variants to genes

For gene based rare variant analysis we need annotations that tells us the boundary of genes. We provide the refGene annotation database for this purpose. Since our project has hg18 as the reference genome while we want to use the refGene hg19 database, we need to liftover our project using hg19 first before we are able to use the annotation with hg19 as the reference.

```
vtools liftover hg19
vtools use refGene.DB
vtools show annotation refGene
```

The names of genes are in the `refGene.name2` field.

Association testing on common/rare variants

The association test program `VAT` is currently under development and is temporarily implemented as the `vtools associate` subcommand. To list available association test options,

```
vtools associate -h
vtools show tests
vtools show test LinRegBurden
```

Note that we use the quantitative trait BMI as the phenotype, and we will allow for "SEX" as a covariate in the regression framework.

Analysis of common variants

By default the program will perform single variant tests using a simple linear model, and the Wald test statistic will be evaluated for p-values:

```
vtools associate common_ceu BMI --covariate SEX -m "LinRegBurden --alternative 2" \
-j1 --to_db EA_CV > EA_CV.asso.res
```

Note

Option `-j1` specifies that 1 CPU core be used for association testing. You may use larger number of jobs for real world data analysis, e.g., use `-j16` if your computational resources has 16 CPU cores available. Linux command `cat /proc/cpuinfo` shows the number of cores and other information related to the CPU on your computer.

Association tests on 1474 groups have completed. 5 failed.

```
grep -i error *.log
```

OUTPUT

```
2013-05-24 11:58:26,414: DEBUG: An ERROR has occurred in process 7 while processing '6:30018583': Sample size too sma
ll (2) to be analyzed for '6:30018583'.
2013-05-24 11:58:26,423: DEBUG: An ERROR has occurred in process 2 while processing '6:30018721': Sample size too sma
ll (2) to be analyzed for '6:30018721'.
2013-05-24 11:58:26,723: DEBUG: An ERROR has occurred in process 7 while processing '7:148552665': Sample size too sm
all (2) to be analyzed for '7:148552665'.
2013-05-24 11:58:26,859: DEBUG: An ERROR has occurred in process 2 while processing '8:145718728': Sample size too sm
all (4) to be analyzed for '8:145718728'.
2013-05-24 11:58:26,885: DEBUG: An ERROR has occurred in process 2 while processing '9:205057': Sample size too small
 (4) to be analyzed for '9:205057'.
```

A summary from the association test is written to the file `EA_CV.asso.res`. The first column indicates the variant chromosome and base pair position so that you may follow up on the top signals using various annotation sources that we will not introduce in this tutorial. The result will be automatically built into annotation database if `--to_db` option is specified.

You may view the summary using the `less` command

```
less EA_CV.asso.res
```

less EA_CV.asso.res

```
variant_chr variant_pos sample_size_LinRegBurden num_variants_LinRegBurden total_mac_LinRegBurden beta_x_LinRegBurden
 pvalue_LinRegBurden wald_x_LinRegBurden beta_2_LinRegBurden beta_2_pvalue_LinRegBurden wald_2_LinRegBurden
1       1105366 39      1       4       -3.79867        0.303847        -1.04312        1.81933 0.423273        0.809
982
1       3538692 78      1       13      1.29502        0.562724        0.581386        -0.753517 0.651351    -0.45370
6
...
```

To sort the results by p-value and output the first 10 lines of the file use the command:

```
sort -g -k7 EA_CV.asso.res | head
```

If you happen to get significant p-values be sure to also observe the accompanying sample size. Significant p-values from too small of a sample size may not be results you can trust.

Also, depending on your phenotype you may have to add additional covariates to your analysis. `VAT` allows you to test many different models for the various phenotypes and covariates you may have. P-values for other covariates are also reported.

Similar to using an annotation database, you can use the results from the association test to annotate the project and follow up variants of interest, for example:

```
vtools show fields
```

association analysis result columns

```
Field name   Description
EA_CV.chr                    chr
EA_CV.pos                    pos
EA_CV.sample_size_LNBT       Sample size
EA_CV.beta_x_LNBT            Test statistic. In the context of regression, this is
                             estimate of effect size for x
EA_CV.pvalue_LNBT            p-value
EA_CV.wald_x_LNBT            Wald statistic for x (beta_x/SE(beta_x))
EA_CV.beta_2_LNBT            estimate of beta for covariate 2
EA_CV.beta_2_pvalue_LNBT     p-value for covariate 2
EA_CV.wald_2_LNBT            Wald statistic for covariate 2
```

You see additional annotation fields starting with EA_CV, the name of the annotation database you just created from association test (if you used the --to_db option mentioned above). You can use them to easily select/output variants of interest.

Analysis of rare variants

We use the -g option and use the 'refGene.name2' field to define the boundaries of a gene. By default the test is a linear regression using aggregated counts of variants in a gene region as the regressor.

```
vtools associate rare_ceu BMI --covariate SEX -m "LinRegBurden --alternative 2" -g refGene.name2 -j1 --to_db EA_RV \
> EA_RV.asso.res
```

To view failed tests due to too small of a sample size, i.e. too few samples have a variant,

```
grep -i error *.log
```

OUTPUT

```
2013-05-24 11:58:26,414: DEBUG: An ERROR has occurred in process 7 while processing '6:30018583': Sample size too sma
ll (2) to be analyzed for '6:30018583'.
2013-05-24 11:58:26,423: DEBUG: An ERROR has occurred in process 2 while processing '6:30018721': Sample size too sma
ll (2) to be analyzed for '6:30018721'.
```

The output file is EA_RV.asso.res. The first column is the gene name, with corresponding p-values for the entire gene.

```
less EA_RV.asso.res
```

less EA_RV.asso.res

```
#refgene_name2   sample_size_LinRegBurden        num_variants_LinRegBurden       total_mac_LinRegBurden  beta_x_LinRe
gBurden     pvalue_LinRegBurden     wald_x_LinRegBurden     beta_2_LinRegBurden     beta_2_pvalue_LinRegBurden wald_2
_LinRegBurden
AATF    89      3       3       4.06571 0.371806        0.897786        0.819087        0.617609        0.501059
ABCB9   58      1       1       4.29374 0.561422        0.584278        0.0901042       0.962807        0.0468439
ABLIM3  90      2       2       -7.83832        0.158126        -1.42364        0.466136
        0.774715        0.287105
```

P-values from this test

You can also sort these results by p-value using the `sort -g -k6` command:

```
sort -g -k6 EA_RV.asso.res | head
```

Variable thresholds test

The variable thresholds method will carry out multiple testing in the same gene region using groups of variants based on observed allele frequencies. This test will maximize/minimize over the statistics thus obtained as the final test statistic, and calculate the empirical p-value so that multiple comparisons are adjusted for correctly. Since we will be testing for 432 genes in our dataset, we use an α level 0.05/432 = 0.00012, thus an "adaptive" cutoff 0.0005 for permutation tests. The command using variable thresholds method on our data is:

```
vtools associate rare_ceu BMI --covariate SEX -m "VariableThresholdsQt --alternative 2 -p 100000 --adaptive 0.0005"\
-g refGene.name2 -j1 --to_db EA_RV > EA_RV_VT.asso.res
```

To view failed tests,

```
grep -i error *.log
```

OUTPUT

```
2013-05-24 11:58:26,414: DEBUG: An ERROR has occurred in process 7 while processing '6:30018583': Sample size too sma
ll (2) to be analyzed for '6:30018583'.
2013-05-24 11:58:26,423: DEBUG: An ERROR has occurred in process 2 while processing '6:30018721': Sample size too sma
ll (2) to be analyzed for '6:30018721'.
```

```
less EA_RV_VT.asso.res
```

less EA_RV_VT.asso.res

```
refgene_name2   sample_size_VTQt        num_variants_VTQt       total_mac_VTQt  beta_x_VTQt
     pvalue_VTQt     std_error_VTQt  num_permutations_VTQt   MAF_threshold_VTQt
ABCB9   58      1       1       4.29374 0.663337        7.18148 1000    0.00862069
ACCN3   56      1       1       9.84035 0.151848        7.0135  1000    0.00892857
AATF    89      3       3       4.06571 0.405594        4.50659 1000    0.00561798
ABLIM3  90      2       2       -7.83832        0.157842        5.7447  1000    0.00555556
ACHE    76      1       1       1.51292 0.813187        7.4606  1000    0.00657895
```

Note

The p values you obtained for VT might be slightly different from the values above. This is due to the randomness in permutation tests. The β values should be the same.

Sort and output the lowest p-values using the command:

```
sort -g -k6 EA_RV_VT.asso.res | head
```

Why some tests failed?

Notice that `vtools associate` command will fail on some association test units. Instances of failure are printed to terminal in red and are recorded in the project log file. Most failures occur due to an association test unit having too few samples or number of variants (for gene based analysis). You should view these error messages after each association scan is complete, e.g., using the Linux command `grep -i error *.log` and make sure you are informed of why failures occur.

In the variable thresholds analysis above, gene *ABCC6* failed the association test. If we look at this gene more closely we can see which variants are being analyzed by our test:

```
vtools select rare_ceu "refGene.name2='ABCC6'" -o chr pos ref alt CEU_afGD10 mut_type
```

OUTPUT

```
chr     pos          ref     alt     CEU_afGD10     mut_type
16      16178858      T       C        1.0           nonsynonymous SNV
```

As you can see, after applying all of our QC filters we are left with one variant within the *ABCC6* gene to analyze. Because the allele frequency for this variant is 1.0 (minor allele frequency is 0.0) there is nothing in the gene to be analyzed and this gene is ignored.

Association analysis of YRI samples

Procedures for YRI sample association analysis is the same as for CEU samples as previously has been described, thus is left as an extra exercise for you to work on your own. Commands to perform analysis for YRI are found below:

BASH

```bash
cd ..
vtools select variant --samples "RACE=0" -t YRI
cp -a ceu yri; cd yri
vtools init yri --parent ../ --variants YRI --samples "RACE=1"
vtools select variant "YRI_afGD10>=0.05 AND YRI_afGD10<=0.99" -t common_yri
vtools select variant "YRI_afGD10>=0.01 AND YRI_afGD10<=0.99" -t semi_common_yri
vtools select v_funct "YRI_afGD10<0.01 OR YRI_afGD10>0.99" -t rare_yri
vtools liftover hg19
vtools use refGene
vtools associate common_yri BMI --covariate SEX -m "LinRegBurden --alternative 2" -j1 --to_db YA_CV > YA_CV.asso.res
vtools associate rare_yri BMI --covariate SEX -m "LinRegBurden --alternative 2" -g refGene.name2 -j1 --to_db YA_RV > YA_RV.asso.res
vtools associate rare_yri BMI --covariate SEX -m "VariableThresholdsQt --alternative 2 -p 100000 --adaptive 0.0005" -g refGene.name2\
 -j1 --to_db YA_RV > YA_RV_VT.asso.res
```

Meta analysis

Here we demonstrate the application of meta-analysis to combine association results from the two populations via `vtools_report meta_analysis`. The input to this command are the association results files generated from previous steps, for example:

```
vtools_report meta_analysis ceu/EA_RV_VT.asso.res yri/YA_RV_VT.asso.res --beta 5 --pval 6 --se 7 -n 2 --link 1> MET\
A_RV_VT.asso.res
```

To view the results,

```
cut -f1,3 META_RV_VT.asso.res
```

OUTPUT

```
refgene_name2        pvalue_meta
COL6A3        2.580E-01
F2RL1        6.031E-01
SETD2        4.192E-01
TMEM225        9.469E-01
PRRC2A        2.536E-01
CRELD1        2.539E-01
TIAM2        1.173E-01
SLC34A2        4.699E-01
RTEL1        1.049E-01
SLC22A14        7.479E-02
... ...
```

Note that for genes that only appears in one study but not the other, or only have a valid p-value in one study but not the other, then the gene will be ignored from meta analysis.

Summary

Analyzing variants with `variant tools/VAT` is much like any other analysis software with a general workflow of:

- Variant level cleaning
- Sample genotype cleaning
- Variant annotation and phenotype information processing
- Sample/variant selection
- Association analysis
- Interpreting the findings

The data cleaning and filtering conditions within this exercise should be considered as general guidelines. Your data may allow you to be more lax with certain criteria or force you to be more stringent with others.

Questions

Q1

List the four lowest p-values and associated variant or gene region for the EA_CV.asso.res, EA_RV.asso.res, and EA_RV_VT.asso.res aggregation test output files.

EA_CV.asso.res

1)_____; 2)_____

3)_____; 4)_____

EA_RV.asso.res

1) _____; 2) _____

3) _____; 4) _____

EA_RV_VT.asso.res

1) _____; 2) _____

3) _____; 4) _____

Q2

List any gene regions that show up in the lowest eight p-values for both the rare variant aggregation test and the variable thresholds test. Why might the p-values for the variable thresholds test not be as low as the p-values for the rare variant aggregation test? Are any of the top p-value hits significant? Why or why not?

_____

Q3

If there are 1472, 432, and 432 individual tests for the single variant test (EA_CV.asso.res), aggregated variant test (EA_RV.asso.res), and variable thresholds test, respectively, what α value of significance would a variant or gene region have to achieve to be considered significant?

Single variant test α = _____

Aggregated variant test α = _____

Variable thresholds test α = _____

Answers

A1

EA_CV.asso.res

1) 107888886 0.000105185

2) 15869257 0.00038548

3) 56293401 0.000386273

4) 15869388 0.00279873

EA_RV.asso.res

1) CIDEA 0.005048216866444212

2) SPP2 0.00549521414622887

3) WNT16 0.006833757978682209

4) CRTAP 0.007234338751298487

EA_RV_VT.asso.res

Note
Due to the use of permutation test, the p values may be slightly different from what you observe

1) CRTAP 0.00999000999000999

2) WNT1 0.00999000999000999

3) CIDEA 0.01198801198801198

4) SPP2 0.011988011988011988

A2

Note
Your "top gene" list might be slightly different for the permutation based VT method.

SPP2, CRTAP, WNT16, CYP24A1, CIDEA. The p-values do not achieve significance based on the corrected α values above (Bonferroni correction for multiple tests). Since the BMI values were randomly generated for each individual it is unlikely that any of the p-values for the single variant and aggregation tests would have achieved significance.

A3

Single variant test $\alpha = 0.05/1472 = 3.39 \times 10^{-5}$

Aggregated variant test $\alpha = 0.05/432 = 1.12 \times 10^{-4}$

Variable thresholds test $\alpha = 0.05/432 = 1.12 \times 10^{-4}$

References

[1] Wang, G.T., Peng, B., and Leal, S.M. (2014). Variant Association Tools for Quality Control and Analysis of Large-Scale Sequence and Genotyping Array Data. Am. J. Hum. Genet. 94, 770783.

[2] Li B, Leal SM. Methods for detecting associations with rare variants for common diseases: application to analysis of sequence data. Am J Hum Genet 2008 83:311-21

[3] Liu DJ, Leal SM. A novel adaptive method for the analysis of next-generation sequencing data to detect complex trait associations with rare variants due to gene main effects and interactions. PLoS Genet 2010 6:e1001156

[4] Madsen BE, Browning SR. A groupwise association test for rare mutations using a weighted sum statistic. PLoS Genet 2009 5:e1000384

[5] Price AL, Kryukov GV, de Bakker PI, Purcell SM, Staples J, Wei LJ, Sunyaev SR. Pooled association tests for rare variants in exon-resequencing studies. Am J Hum Genet 20010 86:832-8

[6] Wu MC, Lee S, Cai T, Li Y, Boehnke M, Lin X. Rare-variant association testing for sequencing data with the sequence kernel association test. Am J Hum Genet 2011 89:82-93

[7] Lucas FAS, Wang G, Scheet P, Peng B. Integrated annotation and analysis of genetic variants from next-generation sequencing studies with variant tools. Bioinformatics 2012 28:421-2

[8] Wang K, Li M, Hakonarson H. ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. Nucleic Acids Res 2010 38:e164