Association Analysis of Sequence Data using Variant Association Tools (VAT) for Mendelian Traits

0.1 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 association methods which include but not limited to Combined Multivariate Collapsing (CMC)[2], Burden of Rare Variants (BRV)[3], Weighted Sum Statistic (WSS)[4], Kernel Based Adaptive Cluster (KBAC)[5], Variable Threshold (VT)[6] and Sequence Kernel Assocation Test (SKAT)[7].

VAT inherits the intuitive command-line interface of Variant Tools (VTools)[8] with re-design and implementation of its infrastructure to accommodate the scale of dataset generated from current sequencing efforts on large populations. Features of VAT are implemented into VTools subcommand system.

0.2 Resource

- **List of commands**

A list of all commands that are used in this exercise can be found at

`http://statgen.us/CourseExerciseScripts#VAT_exercise`

- **Concepts**

Basic concepts to handle sequence data using `vtools` can be found at `http://varianttools.sourceforge.net/Main/Concepts`

- **Software documentation**

VAT documentation can be found at `http://varianttools.sourceforge.net/Main/Documentation`

- **Genotype data**

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

The data set (`CEU.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 assigned half of the CEU individuals to be cases and the other half to be controls to generate data under the null for Mendelian disease (MD).

- **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.

Important

Please reset the user process resource limits by typing `ulimit -s 8000` on the command window. The default allocation of resource on the MDC cluster node is not sufficient for the program to run properly.

```
ulimit -s 8000
```

1 Data Quality Control, Annotation and Variant/sample Selection - Part I

1.1 Getting started

Please navigate to the exercise data directory and check the available subcommands by typing:

```
vtools -h
```

We use a subcommand system for various data manipulation tasks (to check details of each subcommand use `vtools name_of_subcommand -h`). 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 VATDemo
```

OUTPUT

```
INFO: variant tools 2.6.1 : Copyright (c) 2011 - 2014 Bo Peng
INFO: San Lucas FA, Wang G, Scheet P, Peng B (2012) Bioinformatics 28(3):421-422
INFO: Please visit http://varianttools.sourceforge.net for more information.
INFO: Creating a new project VATDemo
```

Command `vtools init` creates a new project in the current directory. A directory can only have one project. After a project is created, subsequent `vtools` calls will automatically load the project in the current directory. Working from outside of a project directory is not allowed.

▪ Import variant and genotype data

Import all vcf files under the current directory:

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

OUTPUT

```
INFO: Importing variants and genotypes from CEU.exon.2010_03.genotypes.vcf.gz (1/1)
CEU.exon.2010_03.genotypes.vcf.gz: 100% [============] 4,306 3.0K/s in 00:00:01
INFO: 3,489 variants (3,489 new, 3,489 SNVs) from 3,500 lines are imported, with a total of 288,291 genotypes from 90
 samples.
```

Command `vtools import` imports variants, sample genotypes and related information fields. The imported variants are saved to the master variant table for the project, along with their information fields.

The command above imports one vcf file into an empty `vtools` project. The second `INFO` message in the screen output shows that 3,489 variant sites are imported from the first vcf file, where *3,489 new* means that all of them are new because prior to importing the vcf file the project was empty so there was 0 site.

More details about `vtools import` command can be found at http://varianttools.sourceforge.net/Vtools/Import

Since the input VCF file uses hg18 as the reference genome while most modern annotation data sources are hg19-based, we need to *liftover* our project using hg19 in order to use various annotation sources in the analysis. Vtools provides a command which is based on the tool of USCS liftOver to map the variants from existing reference genome to an alternative build. More details about `vtools liftover` command can be found at http://varianttools.sourceforge.net/Vtools/Liftover

```
vtools liftover hg19
```

OUTPUT

```
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exporting variants: 100% [=========================] 3,489 420.1K/s in 00:00:00
INFO: Running UCSC liftOver tool
Updating table variant: 100% [======================] 3,489 295.2/s in 00:00:11
```

▪ Import phenotype data

The phenotype file must be in plain text format with sample names matching the sample IDs in the vcf file(s):

```
head phenotypes.csv
```

phenotypes.csv

```
sample_name,panel,SEX,MD
NA06984,ILLUMINA,1,0
NA06985,,2,0
NA06986,ABI_SOLID+ILLUMINA,1,0
NA06989,ILLUMINA,2,0
NA06994,ABI_SOLID+ILLUMINA,1,0
NA07000,ABI_SOLID+ILLUMINA,2,0
NA07037,ILLUMINA,1,0
NA07048,ILLUMINA,2,0
NA07051,ILLUMINA,1,0
...
```

The phenotype file includes information for each individual, the sample name, sequencing panel, sex and MD.

To import the phenotype data:

```
vtools phenotype --from_file phenotypes.csv --delimiter ","
```

OUTPUT

```
INFO: Adding phenotype panel of type VARCHAR(24)
INFO: Adding phenotype SEX of type INT
INFO: Adding phenotype MD of type INT
INFO: 3 field (3 new, 0 existing) phenotypes of 90 samples are updated.
```

Unlike `vtools import`, this command imports/adds properties to *samples* rather than to *variants*. More details about `vtools phenotype` command can be found at http://varianttools.sourceforge.net/Vtools/Phenotype

▪ View imported data

Summary information for the project can be viewed anytime using the command `vtools show`, which displays various project and system information. More details about `vtools show` properties can be found at http://varianttools.sourceforge.net/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
```

1.2 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 when a variant has two alternative alleles (e.g. A->T/C) it is treated as two separate variants.

```
vtools select variant --count
```

There are 3489 variants in our test data.

`vtools select table condition action` selects from a variant table `table` a subset of variants satisfying a specified `condition`, and perform an `action` of

- creating a new variant table if `--to_table` is specified.
- counting the number of variants if `--count` is specified.
- outputting selected variants if `--output` is specified.

The `condition` should be a SQL expression using one or more fields in a project (displayed in `vtools show` fields). If the condition argument is unspecified, then all variants in the `table` will be selected. An optional condition `--samples [condition]` can also be used to limit selected variants to specific samples. More details about `vtools select` command can be found at http://varianttools.sourceforge.net/Vtools/Select

▪ Genotype Summary

The command `vtools show genotypes` displays the number of genotypes for each sample and names of the available genotype information fields for each sample, e.g. GT - genotype; DP_geno - genotype read depth. Such information are useful for the calculation of summary statistics of genotypes (e.g. depth of coverage).

```
vtools show genotypes > GenotypeSummary.txt
head 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

▪ Variant Quality Overview

The following command calculates summary statistics on the variant site depth of coverage (DP). Below is the command to calculate depth of coverage information for all variant sites.

```
vtools output variant "max(DP)" "min(DP)" "avg(DP)" "stdev(DP)" "lower_quartile(DP)" "upper_quartile(DP)" --header
```

max_DP_	min_DP_	avg_DP_	stdev_DP_	lower_quartile_DP_	upper_quartile_DP_
25490	48	6992.37116652	3382.79101288	4620	9293

In the test data, the maximum DP for variant sites is 25490, minimum DP 48, average DP about 6992, standard deviation of DP about 3383, lower quartile of DP 4620 and upper quartile of DP 9293.

The same syntax can be applied to other variant information or annotation information fields. The command `vtools output name_of_variant_table` outputs properties of variants in a specified variant table. The properties include fields from annotation databases and variant tables, basically fields outputted from command `vtools show fields`, and SQL-supported functions and expressions. There are a number of freely available SQL resources on the web to learn more about SQL functions and expressions.

It is also possible to view *variant level* summary statistic for variants satisfying certain filtering criteria using `vtools select name_of_variant_table` command, for example to count only variants having passed all quality filters:

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

All 3489 variants have passed the quality filters. To combine variant filtering and summary statistics:

```
vtools select variant "filter='PASS'" -o "max(DP)" "min(DP)" "avg(DP)" "stdev(DP)" "lower_quartile(DP)" "upper_quar\
tile(DP)" --header
```

The output information of command above will be the same as the previous `vtools output` command, since all variants have passed quality filter.

1.3 Data exploration

▪ 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`: Summaries for depth of coverage and genotype quality across samples
- `maf`: Minor allele frequency

This command will also add calculated variant level statistics to `fields`, which can be shown by commands `vtools show fields` and `vtools show table variant`

```
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)' 'maf=maf()'
```

OUTPUT

```
Counting variants: 100% [================================================================================================
] 90 147.8/s in 00:00:00
INFO: Adding variant info field num with type INT
```

```
INFO: Adding variant info field hom with type INT
INFO: Adding variant info field het with type INT
INFO: Adding variant info field other with type INT
INFO: Adding variant info field total with type INT
INFO: Adding variant info field maf with type FLOAT
INFO: Adding variant info field minDP with type INT
INFO: Adding variant info field maxDP with type INT
INFO: Adding variant info field meanDP with type FLOAT
Updating variant: 100% [============================================================================================] 3
,489 91.4K/s in 00:00:00
INFO: 3489 records are updated
```

```
vtools show fields
vtools show table variant
```

Command `vtools update` updates *variant info fields* (and to a lesser extend *genotype info fields*) by adding more fields or updating values at existing fields. It does not add any new variants or genotypes, and does not change existing variants, samples, or genotypes. Using three parameters `--from_file`, `--from_stat`, and `--set`, variant information fields could be updated from external file, sample genotypes, and existing fields. More details about `vtools update` command can be found at http://varianttools.sourceforge.net/Vtools/Update

▪ Summaries for genotype depth and genotype quality 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 genotypes of all samples 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)' 'mafGD10=maf()' --genotypes "DP_geno > 10"
```

OUTPUT

```
Counting variants: 100% [============================================================================================
] 90 513.8/s in 00:00:00
INFO: Adding variant info field numGD10 with type INT
INFO: Adding variant info field homGD10 with type INT
INFO: Adding variant info field hetGD10 with type INT
INFO: Adding variant info field otherGD10 with type INT
INFO: Adding variant info field totalGD10 with type INT
INFO: Adding variant info field mafGD10 with type FLOAT
Updating variant: 100% [============================================================================================] 3,
483 117.9K/s in 00:00:00
INFO: 3483 records are updated
```

```
vtools show fields
vtools show table variant
```

You will notice the change in genotype counts when applying the filter on genotype depth of coverage and only retaining those genotypes with a read depth greater than 10X. There are now 3483 variant sites after filtering on `DP_geno`>10. Note that some variant sites will become monomorphic after removing genotypes due to low read depth.

- **Minor allele frequencies**

In previous steps, we calculated minor allele frequencies for each variant site before and after filtering on genotype read depth. Below is a summary of the results:

```
vtools output variant chr pos maf mafGD10 --header --limit 10
```

OUTPUT

```
chr         pos           maf                  mafGD10
1           1105366       0.0350877192982      0.0512820512821
1           1105411       0.00943396226415     0.0128205128205
1           1108138       0.0245901639344      0.0212765957447
1           1110240       0.00561797752809     0.0
1           1110294       0.038961038961       0.025
1           3537996       0.123287671233       0.170454545455
1           3538692       0.0730337078652      0.0833333333333
1           3541597       0.00561797752809     0.00617283950617
1           3541652       0.0444444444444      0.0533333333333
1           3545211       0.00561797752809     0.00581395348837
...
```

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

1.4 Variant Annotation

For rare variant aggregated association tests, we want to focus on analyzing aggregating variants having potential functional contribution to a phenotype. Thus, each variant site needs to be annotated for its functionality. Annotation is performed using `variant annotation tools` [7] which implements an ANNOVAR pipeline for variant function annotation[9]. More details about the ANNOVAR pipeline can be found at http://varianttools.sourceforge.net/Pipeline/Annovar

```
vtools execute ANNOVAR geneanno
```

OUTPUT

```
INFO: Running vtools update variant --from_file cache/annovar_input.variant_function --format ANNOVAR_variant_functio
n --var_info region_type, region_name
...
Running vtools update variant --from_file cache/annovar_input.exonic_variant_function --format ANNOVAR_exonic_variant
_function --var_info mut_type, function
...
```

The following command will output the annotated variant sites to the screen.

```
vtools output variant chr pos ref alt mut_type --limit 10 --header
```

OUTPUT

```
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
...
```

Many more annotation sources are available which are not covered in this tutorial. Please read http://varianttools.sourceforge.net/Annotation for annotation databases, and http://varianttools.sourceforge.net/Pipeline for annotation pipelines.

1.5 Data Quality Control (QC) and Variant Selection

▪ Removal of low quality variant sites

We should not need to remove any variant sites based on read depth because all variants passed the quality filter. To demonstrate removal of variant sites, let us remove those with a total variant site read depth < 15.

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

We can see that none has been removed from master variant table, which means all variant sites have read depth $\geqslant 15$. The `vtools remove` command can remove various items from the current project. More details about `vtools remove` command can be found at http://varianttools.sourceforge.net/Vtools/Remove. Using a combination of select/remove subcommands low quality variant sites can be easily filtered out. The `vtools show fields`, `vtools show tables`, and `vtools show table variant` commands will allow you to see the new/updated fields and tables you have added/changed to the project.

▪ Filter genotype calls by quality

We have calculated various summary statistics using the command `--genotypes 'CONDITION'` but we have not yet removed genotypes having a genotype read depth $<$ 10X. The below command below removes these genotypes.

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

▪ Select variants by annotated functionality

To select potentially functional variants for association mapping:

```
vtools select variant "mut_type like 'non%' or mut_type like 'stop%' or region_type='splicing'" -t v_funct
vtools show tables
```

OUTPUT

```
Running: 5 2.1K/s in 00:00:00
INFO: 1749 variants selected.
```

The command above selects variant sites that are either nonsynonymous (by condition `"mut_type like 'non%'`) or stop-gain/stop-loss (by condition `mut_type like 'stop%'`) or alternative splicing (by condition `region_type='splicing'`)

1749 functional variant sites are selected.

2 Association Tests - Part II

2.1 View phenotype data

```
vtools show samples --limit 5
```

OUTPUT

```
sample_name        filename                    panel                 SEX        MD
NA06984            CEU.exon...notypes.vcf.gz   ILLUMINA              1          0
NA06985            CEU.exon...notypes.vcf.gz                         2          0
NA06986            CEU.exon...notypes.vcf.gz   ABI_SOLID+ILLUMINA    1          0
NA06989            CEU.exon...notypes.vcf.gz   ILLUMINA              2          0
NA06994            CEU.exon...notypes.vcf.gz   ABI_SOLID+ILLUMINA    1          0
(85 records omitted)
```

2.2 Subset data by MAFs

In order to carry out association tests we will only analyze rare variants with a frequency of $< 1\%$.

```
vtools use ExAC.DB
vtools show annotation ExAC
vtools select v_funct "NFE_MAF<0.01" -t rare
```

Note

- This allele frequency cut-off is based upon frequency data bases, such as ExAC, and *not* the frequency within the data. Here we use `ExAC.NFE_MAF` field to select rare variant sites based on frequency.
- The `vtools use` command, attaches an annotation database to the project, effectively incorporating one or more attributes available to variants in the project. More details about `vtools use` command can be found at http://varianttools.sourceforge.net/Vtools/Use
- For selection of rare variants we only keep those that are annotated as functional (chosen from `v_funct` table). There are 677 variant sites selected.

2.3 Annotate variants to genes

For gene based rare variant analysis we need annotations that tell us the boundaries of genes. We use the `refGene` annotation database for this purpose.

```
vtools use refGene
```

OUTPUT

```
INFO: Downloading annotation database from annoDB/refGene-hg19_20130904.DB.gz
INFO: Using annotation DB refGene as refGene in project VATDemo.
INFO: Known human protein-coding and non-protein-coding genes taken from the NCBI RNA reference sequences collection
(RefSeq).
```

```
vtools show annotation refGene
```

OUTPUT

```
Annotation database refGene (version hg19_20130904)
Description:            Known human protein-coding and non-protein-coding genes taken from the NCBI RNA reference seq
uences collection (RefSeq).
Database type:          range
Reference genome hg19:  chr, txStart, txEnd
  name (char)           Gene name
  chr (char)
  strand (char)         which DNA strand contains the observed alleles
  txStart (int)         Transcription start position (1-based)
  txEnd (int)           Transcription end position
  cdsStart (int)        Coding region start (1-based)
  cdsEnd (int)          Coding region end
  exonCount (int)       Number of exons
  exonStarts (char)     Starting point of exons (adjusted to 1-based positions)
  exonEnds (char)       Ending point of exons
  score (int)           Score
  name2 (char)          Alternative name
  cdsStartStat (char)   cds start stat, can be 'non', 'unk', 'incompl', and 'cmp1'
  cdsEndStat (char)     cds end stat, can be 'non', 'unk', 'incompl', and 'cmp1'
```

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

2.4 Association testing for rare variants

The association test program `VAT` is implemented as the `vtools associate` subcommand. To list available association test options,

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

More details about `vtools associate` command can be found at http://varianttools.sourceforge.net/Vtools/Associate

▪ Burden test for rare variants (BRV)

BRV method uses the count of rare variants in given genetic region for association analysis, regardless of the region length.

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

```
vtools associate rare MD --covariate SEX -m "LogitRegBurden --alternative 2" -g refGene.name2 -j1 --to_db RV > RV_B\
RV.asso.res
```

OUTPUT

```
INFO: 90 samples are found
INFO: 438 groups are found
Loading genotypes: 100% [============================================] 90 51.0/s in 00:00:01
Testing for association: 100% [=================================] 438/4 161.0/s in 00:00:02
INFO: Association tests on 438 groups have completed. 4 failed.
INFO: Using annotation DB RV as RV in project VATDemo.
INFO: Annotation database used to record results of association tests. Created on Wed, 26 Aug 2015 15:58:00
INFO: 438 out of 25360 refGene.refGene.name2 are annotated through annotation database RV
```

Association tests on 438 groups have completed. 4 failed.

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.

The following command displays error messages about the failed tests. In each case, the sample size was too small to perform the regression analysis.

```
grep -i error *.log | tail -3
```

OUTPUT

```
2015-08-26 10:58:04,518: DEBUG: An ERROR has occurred in process 0 while processing 'NOM1': No variant found in genot
ype data for 'NOM1'.
2015-08-26 10:58:04,805: DEBUG: An ERROR has occurred in process 0 while processing 'PCDHGC5': No variant found in ge
notype data for 'PCDHGC5'.
2015-08-26 10:58:05,227: DEBUG: An ERROR has occurred in process 0 while processing 'SF3A2': No variant found in geno
type data for 'SF3A2'.
```

The output file is `RV_BRV.asso.res`. The first column is the gene name, with corresponding p-values in the sixth column for the entire gene. The result will be automatically built into annotation database if `--to_db` option is specified.

```
less RV_BRV.asso.res
```

You can also sort these results by p-value using command:

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

If you obtain 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. P-values for 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.

```
vtools show fields
```

You see additional annotation fields starting with `RV`, 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. More details about outputting annotation fields for significant findings can be found at http://varianttools.sourceforge.net/Vtools/Output

▪ Variable thresholds test for rare variants (VT)

The variable thresholds (VT) method will carry out multiple testing in the same gene region using groups of variants based on observed variant allele frequencies. This test will maximize over statistics thus obtain a final test statistic, and calculate the empirical p-value so that multiple comparisons are adjusted for correctly.

We will use adaptive permutation to obtain empirical p-values. Therefore, to avoid performing too large number of permutations we use a cutoff to limit the number of permutations when the p-value is greater than 0.0005, e.g. not all 100,000 permutations are performed. Generally, even more permutations are used but we limit it to 100,000 to save time for this exercise.

The command using variable thresholds method on our data is:

```
vtools associate rare MD --covariate SEX -m "VariableThresholdsBt --alternative 2 -p 100000 \
--adaptive 0.0005" -g refGene.name2 -j1 --to_db RV > RV_VT.asso.res
```

OUTPUT

```
INFO: 90 samples are found
INFO: 438 groups are found
Loading genotypes: 100% [============================================] 90 114.5/s in 00:00:00
Testing for association: 100% [=====================================] 438/27 0.7/s in 00:10:47
INFO: Association tests on 438 groups have completed. 27 failed.
INFO: Using annotation DB RV as RV in project VATDemo.
INFO: Annotation database used to record results of association tests. Created on Wed, 26 Aug 2015 15:58:00
INFO: 438 out of 25360 refGene.refGene.name2 are annotated through annotation database RV
```

To view failed tests,

```
grep -i error *.log | tail -5
```

To view results,

```
less RV_VT.asso.res
```

Note

The *p* values you obtained for VT might be slightly different for each run. This is due to the randomness in permutation tests.

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

```
sort -g -k6 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.

For example in the variable thresholds analysis above, gene *SF3A2* 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 "refGene.name2='SF3A2'" -o chr pos ref alt mafGD10 numGD10 mut_type --header
```

chr	pos	ref	alt	mafGD10	numGD10	mut_type
19	2199303	G	A	0.0	0	nonsynonymous SNV

As you can see, after applying all of our QC filters we are left with one variant within the *SF3A2* gene to analyze. Because the minor allele frequency for this variant is 0.0 there are no variants in the gene to analyze so that this gene is ignored. Note that all individuals are homozygous for the alternative allele for this variant site.

- **QQ/Manhattan plots for association results**

```
vtools_report plot_association qq -o QQRV -b --label_top 2 -f 6 < RV_BRV.asso.res
vtools_report plot_association manhattan -o MHRV -b --label_top 5 --color Dark2 --chrom_prefix None -f 6 < RV_BRV.a\
sso.res
```

The `vtools_report plot_association` command generates QQ and Manhattan plots from output of `vtools associate` command. More details about `vtools_report plot_association` can be found at http://varianttools.sourceforge.net/VtoolsReport/PlotAssociation

QQ plot aids in evaluating if there is systematic inflation of test statistics. A common cause of inflation is population structure. If you observe significant inflation of test you may consider including MDS (multi dimensional scaling) components in the association test model.

2.5 Summary

Analyzing variants with `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.

2.6 Questions

List the four lowest p-values and associated gene regions for BRV and VT tests.

RV_BRV.asso.res - BRV tests

1) _____; 2) _____

3) _____; 4) _____

RV_VT.asso.res - VT tests

1) _____; 2) _____

3) _____; 4) _____

2.7 Answers

RV_BRV.asso.res

1) RBPJ 0.0512186

2) CDHR2 0.156218

3) ANP32C 0.186753

4) MARCH1 0.186753

RV_VT.asso.res

1) ZNF493 0.025974

2) AKAP13 0.033966

3) MORC1 0.0739261

4) CRIPAK 0.0819181

Note

The p-values do not achieve significance based on the corrected α values above (Bonferroni correction for multiple tests). Since the data was generated under the null it is unlikely that a significant result will be observed.

3 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] Auer PL, Wang G, Leal SM. Testing for rare variant associations in the presence of missing data. Genet Epidemiol 2013 37:529-38.

[4] 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

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

[6] 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

[7] 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

[8] 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

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