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

PURPOSE

Variant Association Tools [VAT, Wang et al (2014)] [1] was developed to perform quality control and association 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 Association 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.

RESOURCES

A list of all commands that are used in this exercise can be found at
https://statgenetics.github.io/statgen-courses/notebooks/VAT.html

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

VAT 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 data sets (`CEU.exon.2010_03.genotypes.vcf.gz`, `YRI.exon.2010_03.genotypes.vcf.gz`) are 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.

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

1.1 Getting started

Check the available subcommands by typing:

```
vtools -h
```

Subcommand system is used for various data manipulation tasks (to check details of each subcommand use `vtools name of subcommand -h`). This tutorial is mission oriented and focuses on a subset of the commands that are relevant to variant-phenotype association analysis, rather than introducing them systematically. For additional functionality, please refer to documentation and tutorials online.

Initialize a project

```
vtools init VATDemo
```

```
OUTPUT
INFO: variant tools 3.0.9 : Copyright (c) 2011 - 2016 Bo Peng
INFO: Please visit https://github.com/vatlab/varianttools 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 from CEU.exon.2010_03.genotypes.vcf.gz (1/2)
CEU.exon.2010_03.genotypes.vcf.gz: 100% [==============================================] 4,306 3.1K/s in 00:00:01
INFO: 3,489 new variants (3,489 SNVs) from 3,500 lines are imported.
Importing genotypes: 100% [============================================================] 3,489 10.7K/s in 00:00:00
INFO: Importing variants from YRI.exon.2010_03.genotypes.vcf.gz (2/2)
YRI.exon.2010_03.genotypes.vcf.gz: 100% [==============================================] 5,967 10.8K/s in 00:00:00
INFO: 3,498 new variants (5,175 SNVs) from 5,186 lines are imported.
Importing genotypes: 100% [============================================================] 6,987 22.7K/s in 00:00:00
```

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 two vcf files sequentially 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 first vcf the project was empty so there was 0 site. The fourth `INFO` message tells that 5,175 variant sites are imported from the second vcf file, but only 3,498 of them are new (which are not seen in the existing 3,489) because prior to importing the second vcf there were already 3,489 existing variant sites from first vcf.

Thus, 5,175 - 3,498 = 1,677 variant sites are overlapped sites between first and second vcfs. 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 --flip
```

```
OUTPUT
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exporting variants: 100% [=========================================================================================] 6,987
333.2K/s in 00:00:00
INFO: Running UCSC liftOver tool
INFO: Flipping primary and alternative reference genome
Updating table variant: 100% [=============================================================================================] 6,987
46.1K/s in 00:00.00
```

Import phenotype data

The aim of the association test is to find variants that modulate the phenotype BMI. We simulated BMI values for each of the individuals. 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,BMI
NA06984,ILLUMINA,1,36.353
NA06985,NA,2,21.415
NA06986,ABI_SOLID+ILLUMINA,1,26.898
NA06989,ILLUMINA,2,25.015
NA06994,ABI_SOLID+ILLUMINA,1,23.858
NA07000,ABI_SOLID+ILLUMINA,2,36.226
NA07037,ILLUMINA,1,32.513
NA07048,ILLUMINA,2,17.57
NA07051,ILLUMINA,1,37.142
```

The phenotype file includes information for every individual, the sample name, sequencing panel, sex and BMI. 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 BMI of type FLOAT
INFO: 3 field (3 new, 0 existing) phenotypes of 202 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` 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 6987 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 is 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	13	6815.77028768	3434.28040091	4301	9143

In the test data, the maximum DP for variant sites is 25490, minimum DP 13, average DP about 6815, standard deviation of DP about 3434, lower quartile of DP 4301 and upper quartile of DP 9143.

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 several 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 6987 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_quartile(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
- 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
INFO: Reading genotype info for processing....
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% [============================================================] 6,987 42.5K/s in 00:00:00
```

```
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 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 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
INFO: Reading genotype info for processing....
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% [============================================================] 6,987 52.1K/s in 00:00:00
```

```
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 6987 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 (MAFs)

In previous steps, we calculated MAFs 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 20
```

OUTPUT

```
chr       pos           Maf                 mafGD10
1         1105366       0.0350877192982     0.0512820512821
1         1105411       0.00943396226415    0.0128205128205
1         1108138       0.192307692308      0.18023255814
1         1110240       0.00561797752809    0.0
1         1110294       0.228125            0.242307692308
1         3537996       0.12012987013       0.152173913043
1         3538692       0.0410256410256     0.0432098765432
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.

Next, we examine population specific MAFs. Our data is imported from two files, a CEU dataset (90 samples) and an YRI dataset (112 samples). To calculate allele frequency for each population, let us first assign an additional RACE phenotype (0 for YRI samples and 1 for CEU samples):

```
vtools phenotype --set "RACE=0"  --samples "filename like 'YRI%'"
vtools phenotype --set "RACE=1" --samples "filename like 'CEU%'"
vtools show samples --limit 10
```

OUTPUT

```
sample_name      filename                    panel                SEX       BMI         RACE
NA06984          CEU.exon...notypes.vcf.gz   ILLUMINA             1         36.353      1
NA06985          CEU.exon...notypes.vcf.gz   .                    2         21.415      1
NA06986          CEU.exon...notypes.vcf.gz   ABI_SOLID+ILLUMINA   1         26.898      1
NA06989          CEU.exon...notypes.vcf.gz   ILLUMINA             2         25.015      1
NA06994          CEU.exon...notypes.vcf.gz   ABI_SOLID+ILLUMINA   1         23.858      1
NA07000          CEU.exon...notypes.vcf.gz   ABI_SOLID+ILLUMINA   2         36.226      1
NA07037          CEU.exon...notypes.vcf.gz   ILLUMINA             1         32.513      1
NA07048          CEU.exon...notypes.vcf.gz   ILLUMINA             2         17.57       1
NA07051          CEU.exon...notypes.vcf.gz   ILLUMINA             1         37.142      1
NA07346 CEU.exon...notypes.vcf.gz . 2 30.978 1 (192 records omitted)
```

Population specific MAF calculations will be performed using those genotypes that passed the read depth filter (DP_geno>10)

```
vtools update variant --from_stat 'CEU_mafGD10=maf()' --genotypes 'DP_geno>10' --samples "RACE=1"
vtools update variant --from_stat 'YRI_mafGD10=maf()' --genotypes 'DP_geno>10' --samples "RACE=0"
vtools output variant chr pos mafGD10 CEU_mafGD10 YRI_mafGD10 --header --limit 10
```

OUTPUT

```
chr       Pos           mafGD10             CEU_mafGD10          YRI_mafGD10
1         1105366       0.0512820512821     0.0512820512821      0.0
1         1105411       0.0128205128205     0.0128205128205      0.0
1         1108138       0.18023255814       0.0212765957447      0.371794871795
1         1110240       0.0                 0.0                  0.0
1         1110294       0.242307692308      0.025                0.428571428571
1         3537996       0.152173913043      0.170454545455       0.135416666667
```

```
1       3538692     0.0432098765432     0.0833333333333     0.00595238095238
1       3541597     0.00617283950617    0.00617283950617    0.0
1       3541652     0.0533333333333     0.0533333333333     0.0
1       3545211     0.00581395348837    0.00581395348837    0.0
```

You will observe zero values because some variant sites are monomorphic or they are population specific.

Sample level genotype summaries

Similar operations could be performed on a sample level instead of on a variant level. More details about obtaining genotype level summary information using `vtools phenotype --from stat` can be found at http://varianttools.sourceforge.net/Vtools/Phenotype

```
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 sample_nameCEU_totalGD10CEU_numGD10YRI_totalGD10YRI_numGD10 --header
```

OUTPUT

```
sample_name     CEU_totalGD10   CEU_numGD10     YRI_totalGD10   YRI_numGD10
NA06984     2774        849     NA      NA
NA06985     1944        570     NA      NA
NA06986     3386        1029    NA      NA
NA06989     2659        819     NA      NA
NA06994     1730        486     NA      NA
...
NA19257     NA      NA      4969    1229
NA19259     NA      NA      4182    1005
NA19260     NA      NA      4404    1076
NA19262     NA      NA      4308    1044
NA19266     NA      NA      4878    1211
```

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
...
INFO: Fields mut_type, function of 6,920 variants are updated
```

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

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

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
...
```

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

Ti/Tv ratio evaluations

Before performing any data QC we examine the transition/transversion (Ti/Tv) ratio for all variant sites. Note that here we are obtaining Ti/Tv ratios for the entire sample, Ti/Tv ratios can also be obtained for each sample.

```
vtools_report trans_ratio variant -n num
```

num of transition	num of transversion	ratio
161,637	44,641	3.62082

The command above counts the number of transition and transversion variants and calculates its ratio. More details about `vtools report trans ratio` command can be found at http://varianttools.sourceforge.net/VtoolsReport/TransRatio

If only genotype calls having depth of coverage greater than 10 are considered:

```
vtools_report trans_ratio variant -n numGD10
```

num of transition	num of transversion	ratio
140,392	38,710	3.62676

We can see that Ti/Tv ratio has increase slightly if low depth of coverage calls are removed. There is only a small change in the Ti/Tv ratio since only a few variant sites become monomorphic and are no longer included in the calculation. In practice Ti/Tv ratios can be used to evaluate which threshold should be used in data QC.

Removal of low quality variant sites

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

```
remove those with a total read depth {$\(\le\)$} 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 one variant site has been removed from master variant table. 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 genotype read depth of coverage lower than 10X. The 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
```

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'`)

3367 functional variant sites are selected

2 Association Tests for Quantitative Traits - Part II

2.1 View phenotype data

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

OUTPUT

sample_name	filename	panel	SEX	BMI	...
NA06984	CEU.exon...notypes.vcf.gz	ILLUMINA	1	36.353	...
NA06985	CEU.exon...notypes.vcf.gz	.	2	21.415	...
NA06986	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	1	26.898	...
NA06989	CEU.exon...notypes.vcf.gz	ILLUMINA	2	25.015	...
NA06994	CEU.exon...notypes.vcf.gz	ABI_SOLID+ILLUMINA	1	23.858	...

2.2 Analysis plan

We want to carry out the association analysis for CEU and YRI separately. For starters we demonstrate analysis of CEU samples; and the same commands will be applicable for YRI samples. After completing the analysis of CEU samples please use the same commands to analyze the YRI data set. 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.

2.3 Subset data by MAFs

To carry out association tests we need 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_mafGD10>=0.05" -t common_ceu
```

```
vtools select v_funct "CEU_mafGD10<0.01" -t rare_ceu
```

Notice that for selection of rare variants we only keep those that are annotated as functional (chosen from `v funct` table). There are 1450 and 604 variant sites selected for MAF≥0.05 and MAF<0.01, respectively.

2.4 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 annoDB/refGene-hg19_20130904.ann
INFO: Downloading annotation database from annoDB/refGene-hg19_20130904.DB.gz refGene-hg19_20130904.DB.gz:
100%  [================================================================================  ==]  8,056,345.0
411.6K/s in 00:00:19
INFO: Using annotation DB refGene as refGene in project ceu.
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 'cmpl'
  cdsEndStat (char)  cds end stat, can be 'non', 'unk', 'incompl', and 'cmpl'
```

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

2.5 Association testing of 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 account for "SEX" as a covariate in the regression framework. More details about `vtools associate` command can be found at http://varianttools.sourceforge.net/Vtools/Associate

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
```

OUTPUT

```
INFO: 90 samples are found
INFO: 1450 groups are found
Loading genotypes: 100% [==================================================================] 90 56.7/s in 00:00:01
Testing for association: 100% [=================================================] 1,450/5 684.5/s in 00:00:02
INFO: Association tests on 1450 groups have completed. 5 failed.
INFO: Using annotation DB EA_CV as EA_CV in project ceu.
INFO: Annotation database used to record results of association tests. Created on Fri, 25 Mar 2016 17:45:52
INFO: 1450 out of 3484 variant.chr, variant.pos are annotated through annotation database EA_CV
```

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 1450 groups have completed. 5 failed.

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
```

OUTPUT

```
2016-03-25 12:45:57,373: DEBUG: An ERROR has occurred in process 0 while processing '6:30018583':
Sample size too small (2) to be analyzed for '6:30018583'.
2016-03-25 12:45:57,378: DEBUG: An ERROR has occurred in process 0 while processing '6:30018721':
Sample size too small (2) to be analyzed for '6:30018721'.
2016-03-25 12:45:57,574: DEBUG: An ERROR has occurred in process 0 while processing '7:148552665':
Sample size too small (2) to be analyzed for '7:148552665'.
2016-03-25 12:45:57,662: DEBUG: An ERROR has occurred in process 0 while processing '8:145718728':
Sample size too small (4) to be analyzed for '8:145718728'.
2016-03-25 12:45:57,669: DEBUG: An ERROR has occurred in process 0 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
```

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 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, for example:

```
vtools show fields
```

association analysis result columns

```
Field name Description
EA_CV.variant_chr
EA_CV.variant_pos
EA_CV.sample_size_LinRegBurden
EA_CV.beta_x_LinRegBurden
EA_CV.pvalue_LinRegBurden
EA_CV.wald_x_LinRegBurden
EA_CV.beta_2_LinRegBurden
EA_CV.beta_2_pvalue_LinRegBurden
EA_CV.wald_2_LinRegBurden
variant_chr
variant_pos
sample size
test statistic. In the context of regression, this is estimate of effect size for x p-value
Wald statistic for x (beta_x/SE(beta_x))
estimate of beta for covariate 2
p-value for covariate 2
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. More details about outputting annotation fields for significant findings can be found at http: //varianttools.sourceforge.net/Vtools/Output

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 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
```

OUTPUT

```
INFO: 90 samples are found
INFO: 254 groups are found
Loading genotypes: 100% [============================================================] 90 48.6/s in 00:00:01
Testing for association: 100% [=====================================================] 254/20 685.4/s in 00:00:00
INFO: Association tests on 254 groups have completed. 20 failed.
INFO: Using annotation DB EA_RV as EA_RV in project ceu.
INFO: Annotation database used to record results of association tests. Created on Fri, 25 Mar 2016 17:47:26
INFO: 254 out of 25360 refGene.refGene.name2 are annotated through annotation database EA_RV
```

Association tests on 254groups have completed. 20 failed. To view failed tests:

```
grep -i error *.log | tail -10
```

OUTPUT

```
2016-03-25 12:49:49,553: DEBUG: An ERROR has occurred in process 0 while processing 'ABCC1': No variant found in geno type data for 'ABCC1'.
2016-03-25 12:49:49,620: DEBUG: An ERROR has occurred in process 0 while processing 'ANO9': No variant found in genot ype data for 'ANO9'.
2016-03-25 12:49:49,781: DEBUG: An ERROR has occurred in process 0 while processing 'C10orf71': No variant found in g enotype data for 'C10orf71'.
2016-03-25 12:49:49,875: DEBUG: An ERROR has occurred in process 0 while processing 'CCDC127': No variant found in ge notype data for 'CCDC127'.
2016-03-25 12:49:50,313: DEBUG: An ERROR has occurred in process 0 while processing 'FBXL13': No variant found in genotype data for 'FBXL13'.
...
```

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

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

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

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

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_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 test that failed,

```
grep -i error *.log | tail -10
```

To view results,

```
less EA_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 EA_RV_VT.asso.res | head
```

Why do some tests fail?

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 ABCC1 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='ABCC1'" -o chr pos ref alt CEU_mafGD10 numGD10 mut_type --header
```

chr	Pos	ref	alt	CEU mafGD10	numGD10	mut type
16	16178858	T	C	0.0	243	nonsynonymous SNV

After applying our QC filters we are left with one variant within the ABCC1 gene to analyze. Because the MAF 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 and Manhattan plots for association results

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

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

QQ plots aid in evaluating if there is systematic inflation of test statistics. A common cause of inflation is population structure or batch effects. If you observe significant inflation of test you may consider including MDS components in the association test model.

```
vtools associate rare_ceu BMI --covariate SEX KING_MDS1 KING_MDS2 -m "LinRegBurden --name RVMDS2 --alternative 2" -\
g refGene.name2 -j1 --to_db EA_RV > EA_RV_MDS2.asso.res
vtools_report plot_association qq -o QQRV_MDS2 -b -- label_top 2 -f 6 < EA_RV_MDS2.asso.res
```

To visualize the plots copy them to the work directory by typing:

$ cp MHRV.pdf /home/jovyan/work

$ cp QQRV.pdf /home/jovyan/work

Now visualize from your computer's home directory

You should not arbitrarily include MDS (or PCA) components in the analysis. Instead put in each MDS component and examine the lambda value, i.e. include MDS component 1 them MDS components 1 and 2, etc. Visualization of the QQ plot is also useful to determine if population substructure/admixture is controlled

2.6 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

```
cd ..
vtools select variant --samples "RACE=0" -t YRI
mkdir -p yri; cd yri
vtools init yri --parent ../ --variants YRI --samples
"RACE=0" --build hg19 vtools select variant
"YRI_mafGD10>=0.05" -t common_yri vtools select v_funct
"YRI_mafGD10<0.01" -t rare_yri
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 cd ..
```

2.7 MDS analysis and PC adjustment

This pipeline needs PLINK 1.9 and KING.

```
vtools execute KING
```

$ cp KING.mds.pdf /home/jovyan/work

2.8 Meta-analysis

Here we demonstrate the application of meta-analysis to combine association results from the two populations via `vtools report meta analysis`. More details about `vtools report meta analysis` command can be found at http://varianttools.sourceforge.net/VtoolsReport/MetaAnalysis

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 > ME\ TA_RV_VT.asso.res
```

To view the results,

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

refgene_name2	pvalue_meta
CASP7	4.751E-01
POLR2J2	3.110E-01
GNAO1	6.875E-02
C18orf25	9.456E-01
GBP7	3.498E-01
MSH5	5.905E-01
OR51B5	5.521E-01
MAPK14	3.063E-01
BAZ2B	7.941E-01

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, will be ignored from meta-analysis.

2.8 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 laxer with certain criteria or force you to be more stringent with others.

Questions

Question 1 List the four lowest p-values and associated variants or gene regions for the EA CV.asso.res, EA RV.asso.res, and EA RV VT.asso.res test outputs, which are results from single variant Wald test, rare variant BRV and VT tests, respectively, using the European American (CEU) population. Also, list the results using Yoruba African (YRI) population from YA CV.asso.res, YA RV.asso.res and YA RV VT.asso.res

EA_CV.asso.res - single variant tests using CEU

1)_____; 2)_____

3)_____; 4)_____

EA_RV.asso.res - BRV tests using CEU

1) _____; 2) _____

3) _____; 4) _____

EA_RV_VT.asso.res - VT tests using CEU

1) _____; 2) _____

3) _____; 4) _____

YA_CV.asso.res - single variant tests using YRI

1)_____; 2)_____

3)_____; 4)_____

YA_RV.asso.res - BRV tests using YRI

1) _____; 2) _____

3) _____; 4) _____

YA_RV_VT.asso.res - VT tests using YRI

1) _____; 2) _____

3) _____; 4) _____

Question 2 List any gene regions that show up in the lowest eight p-values for both the BRV and the VT tests. Why might the p-values for the VT tests be higher than the p-values for the BRV tests? Are any of the top p-value hits significant? Why or why not?

_____

Answers

Question 1

EA CV.asso.res
107888886 0.000105185
1) 15869257 0.00038548
2) 56293401 0.000386273
3) 15869388 0.00279873

EA RV.asso.res
1) CIDEA 0.00504822
2) UGT1A10 0.00549521
3) UGT1A5 0.00549521
4) UGT1A6 0.00549521

EA_RV_VT.asso.res
1) UGT1A9 0.007996
2)CPED1 0.00999001
3) UGT1A10 0.00999001
4) UGT1A6 0.011988

YA_CV.asso.res

1) 107888886 0.00000974
2) 6003506 0.000211457
3) 25901623 0.001329
4) 3392651 0.00194995

YA_RV.asso.res

1) EMILIN2 0.00262487
2) ASIC2 0.0551664
3) MDN1 0.0593085
4) BAZ2B 0.0607625

YA_RV_VT.asso.res

1) EMILIN2 0.00533156
2) MDN1 0.013986
3) VLDLR 0.01998
4) LRRC9 0.025974

Question 2: The p-values do not achieve significance based on the corrected p 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. Also, because of the multiple testing, the p-values for the VT tests might be higher than the p-values for the BRV tests.

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 com-plex 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

[10] Manichaikul A, Mychaleckyj JC, Rich SS, Daly K, Sale M, Chen WM. Robust relationship inference in genome-wide association studies. Bioinformatics 2010 26(22):2867-2873

[11] Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MAR, Bender D, Maller J, Sklar P, de Bakker PIW, Daly MJ & Sham PC. PLINK: a toolset for whole-genome association and population-based linkage analysis. Am J Hum Genet, 2007 81:559-75