vikngs Documentation

Release 0.0.1

Scott Mastromatteo

Aug 13, 2019

Table of Contents

1 Quick Start **3**
1.1 User Interface . . . 3
1.2 Running Example on User Interface . . . 3
1.3 Running Command Line . . . 3

2 Source Code **5**
2.1 Compiling the VikNGS User Interface . . . 5
2.2 Compiling the VikNGS Command Line Tool . . . 6

3 Input Files **7**
3.1 Quick Summary . . . 7
3.2 Multi-sample VCF . . . 7
3.3 Sample Information File . . . 10
3.4 BED File (Optional) . . . 11

4 Choosing Parameters **13**
4.1 Command Line Parameters . . . 13
4.2 Parameter Explaination . . . 14

5 Association Tests **17**
5.1 Common Single Variant Association Test . . . 17
5.2 Rare Variant Association Test . . . 18

6 Output **21**
6.1 P-value File . . . 21
6.2 Filtered File . . . 21

7 Power Simulation Package **23**
7.1 Parameters . . . 23
7.2 Output . . . 24

For quick information on how to use VikNGS, see *Quick Start*.

The Variant Integration Kit for Next Generation Sequencing (VikNGS) was designed for association analysis of NGS data integrated across different studies. Compared to the usual score test, it uses a score test with a robust variance calculated from the expected genotype probability given the observed sequencing data, rather than the hard genotype call.

The software package includes tests for both rare and common variant association and can handle both case-control and quantitative studies.

Provide a multisample VCF file and a tab-separated sample information file. The variants will be extracted from the VCF and expected genotypes are computed for every individual. Variants are then filtered based on user-provided filtering parameters. A series of association tests are performed and the resulting p-values are provided as output.

The code is freely available on GitHub

CHAPTER 1

Quick Start

1.1 User Interface

To download VikNGS, go to the GitHub repository release page and find the latest release. The releases contains precompiled versions of VikNGS for Windows, Mac and Linux operating systems. Download the appropriate release ZIP folder for your opperating system.

Unzip the folder and running file VikNGS-X.X.X should start up the user interface. The available versions were compiled on the following systems, so please try to run it under a similar setting:

- Windows: Windows 10 64x (compiled with Microsoft Visual C++ 2017)
- Mac: macOS 10.13.4 High Sierra
- Linux: Ubuntu 18.04 64x

If there is an issue running the software, we recommend trying a different system or try *compiling the software from the source code*.

1.2 Running Example on User Interface

In the same directory where the VikNGS-X.X.X application is found, two example files are present. Within the VikNGS interface, provide *example.vcf* as input in the section labelled “VCF File” and *example_info.txt* as input in the section labelled “Sample Information File”. Clicking the “RUN” button should then trigger the association tests using these files as input.

1.3 Running Command Line

From the command line, run the following commands:

```
wget https://github.com/ScottMastro/VikNGS/archive/master.zip
unzip master.zip
cd VikNGS-master/bin
make
```

To test the binary executable file, try running the following command:

```
./VikNGS --vcf ../example/example.vcf --sample ../example/example_info.txt
```

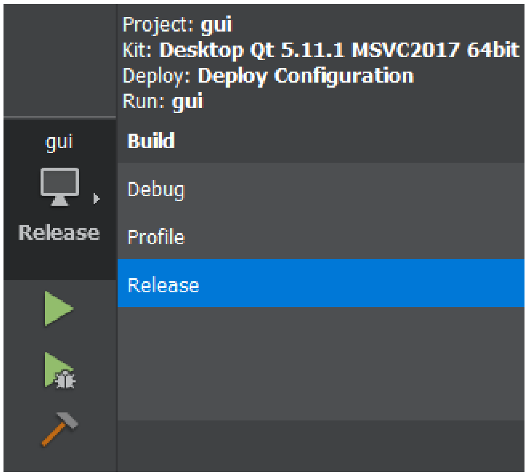
CHAPTER 2

Source Code

2.1 Compiling the VikNGS User Interface

The VikNGS source code is contained in `VikNGS/src/` and the files specific to the graphical user interface (GUI) are found in `VikNGS/src/gui`. To build the GUI version of the software, we recommend downloading and installing QT 5.11+ and QT Creator.

Open QT Creator after downloading and press the "Open Project" button to load the user interface QT project. Navigate to the directory where the VikNGS was downloaded and load the file `VikNGS/src/gui/gui.pro`. This should load the source code and prompt you to choose a compiler. After selecting a compiler, the program can be build by switching to "Release" mode and pressing the top green arrow as seen below:



Project: **gui**
Kit: **Desktop Qt 5.11.1 MSVC2017 64bit**
Deploy: **Deploy Configuration**
Run: **gui**

This should begin compiling the code (will take a few minutes) and will automatically open a window when it is complete.

2.2 Compiling the VikNGS Command Line Tool

The command line-specific files are contained in `VikNGS/src/cmd`. A Makefile is provided to compile the code for command line use and can be found in `VikNGS/bin`. Simply going into this `bin` directory and typing `make` from the command line will begin compiling the code.

Note: g++ and C++11 or later is required.

CHAPTER 3

Input Files

3.1 Quick Summary

VikNGS takes 3 different file types as input:

- **a multi=sample VCF that provides genotype information**
    - genotype information is extracted from **GL** (preferred), **PL**, and **GT** fields
- **a tab-separated sample information text file containing phenotype and covariate information**
    - must be created by user ref:*see column details code<sample_info>* for what this file should contain
- **a BED file to specify the variant collapsing strategy (optional)**
    - genes and exons specified in the file can be used to define regions to collapse upon
    - can be generated automatically from the UCSC Table Browser

Below, the different types of files are explained in further detail.-

3.2 Multi-sample VCF

A Variant Call Format (VCF) file is a standard way of storing variant information called from sequencing data. Each row of a VCF file corresponds to a genetic variant (insertion, deletion or substitution) and contains information such as the genomic position of the variant, the confidence in the variant call and many other additional annotations.

A multi-sample VCF is formatted identically to a single-sample VCF except it contains an extra set of columns corresponding to sample-specific data.

The first set of lines in a VCF file make up the header and are denoted by the characters *##*. The header includes information about the data source and how the VCF file was constructed. This information is ignored when by VikNGS. The last line of the header is denoted with a single # and includes the column names in addition to a unique identifier for every sample. The first nine columns must be (tab-delimited, in order) **CHROM**, **POS**, **ID**, **REF**, **ALT**, **QUAL**, **FILTER**, **INFO** and **FORMAT**. Every subsequent value is expected to be a unique sample identifier.

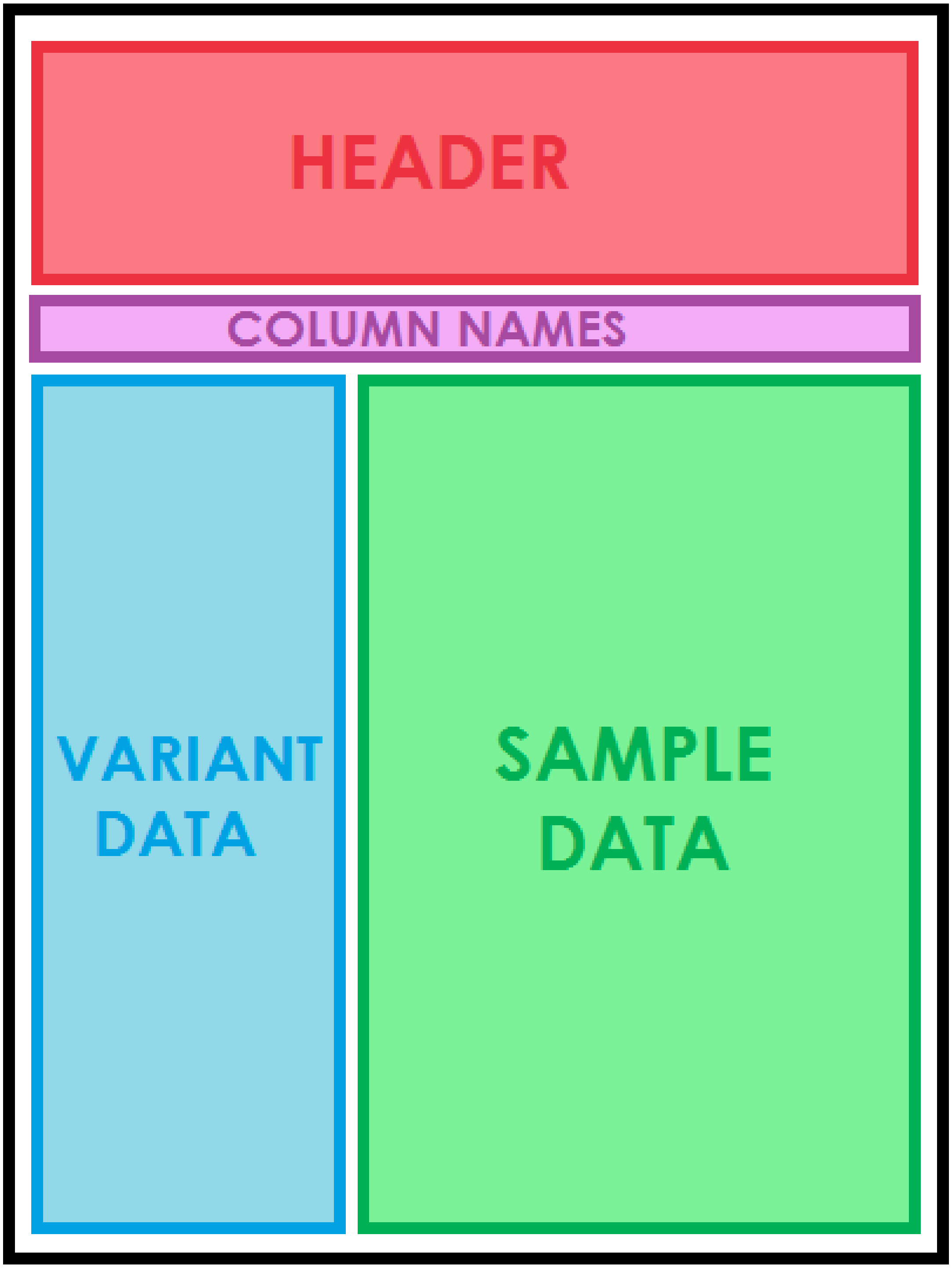


Fig. 1: The general layout of a multi-sample VCF file.

Listing 1: *Example of a multi-sample VCF*

```
##fileformat=VCFv4.1
##FILTER=<ID=PASS,Description="All filters passed">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GL,Number=.,Type=Integer,Description="Genotype Likelihood">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="Normalized, Phred-scaled␣
↪likelihoods">
#CHROM      POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  SAMPLE_1_
↪ID     SAMPLE_2_ID     ...
  22        160036  snp_1   A       C       .       PASS    .       GT:PL:GL       ␣
↪0/0:0,15,73:.   1/1:50,8,0:-5,-0.84,-0.07
  22        160408  snp_2   T       C       .       PASS    .       GT:PL   ./.:.  ␣
↪0/0:0,36,248
  22        160612  snp_2   C       G       .       PASS    .       GT:GL   0/1:-0.
↪48,-0.48,-0.48   1/1::-4.4,-0.27,-0.33
```

Note: While parsing the file only SNPs will be retained (a single A, T, C, or G in both the REF and ALT columns). Multiallelic sites are also currently ignored by VikNGS.

3.2.1 Variant Specific Columns

The first nine columns contain information relevant to the called variants.

The first two required columns (**CHROM**, **POS**) specify the genomic coordinates of the variant. The first base pair of a chromosome is denoted as position 1. **CHROM** is treated as a unique identifier for a chromosome. **POS** is expected to be a positive integer value.

The third column (**ID**) assigns an identifier to each variant.

The forth and fifth columns (**REF**, **ALT**) refer to reference alleles derived from a reference genome and alternative alleles observed in the sequencing data. These columns specify whether the variant is an indel or a SNP.

The sixth column (**QUAL**) provides a Phred-scaled probability that the called base is correct. This field is ignored and not used by vikNGS.

The seventh column (**FILTER**) Indicates whether or not the variant has passed all of the quality control checks present in the variant calling pipeline that produced the VCF file. The value of this field will be **PASS** if the variant passed all of the filters and will be "." if no checks were applied.

The eighth column (**INFO**) contains additional variant annotations, delimited by a semicolon. This field is ignored and not used by vikNGS.

The ninth column (**FORMAT**) specifies the ordering of sample-level annotations, delimited by a colon. All subsequent sample-specific columns will provide values for each annotation, placed in the same order and also delimited by a colon. In particular, the ordering of **GT**, **GL**, and **PL** is extracted from this column.

3.2.2 Sample Specific Columns

All columns following the **FORMAT** column should correspond to a single sample. The header of each column specifies a unique identifier for each sample. Values in subsequent columns are colon separated and are ordered with respect to the labels in the **FORMAT** column. The following values are extracted for each sample.

GT (Genotype): A pair of values specifying the predicted genotype for a specific sample (0/0 homozygous reference, 0/1 heterozygous, 1/1 homozygous alternative).

GL (Genotype likelihood): A set of three numbers indicating the log-scaled genotype probability of all three possible genotypes. Each number corresponds to the value for homozygous reference, heterozygous, and homozygous alternative, respectively.

PL (Phred-scaled likelihood): A set of three integers indicating the Phred-scaled probability of all three possible genotypes (homozygous reference, heterozygous, and homozygous alternative). These values are normalized since all the values are relative to the most likely genotype (and are therefore not probabilities).

Note: **GL** and **PL** are both ways of representing genotype probabilities. Both values can be used to derive genotype probabilities but because **PL** is a rounded integer, **GL** is more accurate.

$genotype\_probability = 10^{\mathbf{GL}} = 10^{-\mathbf{PL}/10}$

$-10\mathbf{GL} = \mathbf{PL}$

When running an association test in VikNGS, different genotype values can potentially be used. If the “Use VCF GT” option is checked, VikNGS will extract the **GT** value only and convert the value to a genotype $\{0, 1, 2\}$. If “Use Expected GT” is checked, the software will attempt to parse the **GL** values from each sample first. If **GL** values are missing or are formatted incorrectly, then the **PL** values will be extracted. If parsing of both **GL** and **PL** values fail, the **GT** values will be extracted. An expected genotype value [0,2] will be calculated from the extracted column. If “Use GT calls” is chosen, genotype probabilities will be extracted in the same way the expected gentype method but a hard genotype call $\{0, 1, 2\}$ will be made instead of calculating the expected value.

> If the relevant information is not present for a given variant, that variant will be skipped and not included in analysis.

Note: Note that if **GT** values are indicated as “missing” (ex. ./.) then the variant will be skipped even if values for **GL** and **PL** are present.

3.3 Sample Information File

To utilize phenotypic data and sample-specific information, vikNGS requires the user to provide this information in a separate file. This tab-separated file is defined specifically for use in vikNGS. This file should *not* have headers and is expected to contain one sample per line.

The columns are defined as follows:

3.3.1 Sample ID

Every line in the sample information file should begin with a *unique* sample ID. The only additional requirement is that every sample ID needs to identically match exactly one of the IDs that appear after the **FORMAT** column in the multisample VCF file. This column specifies the relationship between the sample-specific data and the data in the VCF file.

3.3.2 Phenotype

This column contains phenotypic data which will be used to identify association with genotype information.

Note: If looking to find association information between case-control groups, this column is used to specify case-control status. Please designate cases with a 1 and controls with a 0 in this column.

3.3.3 Group ID

Use this column to specify if samples are from different groups or studies. Any samples with the same value in the column will be put in the same group.

3.3.4 Read Depth

If using the expected genotype method, the score test calculates the variance for each group separately and needs to be aware of which groups are high read depth versus low read depth. The read depth of each sample must be specified in this column. The numerical read depth value (ex. 32) can be provided for each sample or simply a letter specifying whether a sample is from a high or low sequencing run (H = high, L = low). Note that all samples with a shared group ID must also share high/low read depth status. Therefore, the first read depth value encountered for a group will be applied to all members of that group.

3.3.5 Covariates

The remaining columns are used to specify covariates. Covariates can either be continuous or categorical. If every value in a covariate column is numeric, the column will be treated as a continuous covariate. If a single non-numeric value is identified, the covariate will be treated as categorical and a new dummy covariate will be made for EVERY unique value (high cardinality categorical variables can result in significantly longer computation time).

3.4 BED File (Optional)

Elucidation of associated rare variants can be challenging because the frequency of the associated allele can be extremely low. To improve the power of statistical tests that identify rare alleles, it is necessary to collapse a group of linked variants and perform the association test on a genetic region rather than individual SNPs. For rare variant association in vikiNGS, a collapsing strategy must be specified.

There are three types of collapsing strategies available: - Collapse every \(k\) - Collapse by gene - Collapse by exon

By default, the variants will be read and filtered from an input VCF file. After the filtering step, the first \(k\)th variants will be collapsed together, followed by the next set of \(k\) non-overlapping variants and so on (\(k=5\) by default).

To collapse variants in a more biologically relevant way, a BED file must be provided specifying the collapsible regions. A BED file is a tab-delimited table which describes genomic features intended to be used for visualization in a genome browser. The format is specified by UCSC Genome Bioinformatics, detailed information can be found on their web page web page. Every line describes a single region as follows

The first three columns specify the gene

1. **chrom** - The name of the chromosome matching the first column in the VCF file.
2. **chromStart** - The starting position of the gene on the chromosome (starting from 0)
3. **chromEnd** - The ending position of the gene on the chromosome. This base is not included in the gene.

For example, to specify the first 250 bases on chromosome 4: chr4 0 250

The next six column are specified by the BED format but are not used in variant collapsing:

4. **name** - Optional identifier for this region.
5. **score** - Not used.
6. **strand** - Not used.
7. **thickStart** - Not used.
8. **thickEnd** - Not used.
9. **itemRgb** - Not used.

The last three columns are potentially used if collapsing:

10. **blockCount** - The number of blocks (exons) in the gene.
11. **blockSizes** - The size of each exon, comma separated list the size of blockCount.
12. **blockStarts** - Positions where each exon should begin, relative to chromStart. Comma separated list the size of blockCount

To collapse variants by gene, the first three columns are required to indicate where each gene begins and ends.

To collapse variants by exon, all twelve columns must be present. The coding region is defined to be where the first block/exon starts to where the last one ends. Each exon is specified by a block and variants within that block will be collapsed.

Warning: Genes and exons can be very large and could contain thousands of variants to collapse in a single test. This can cause large computational burden, especially if a permutation test is used to calculate the p-value. Therefore, a maximum collapse size can be specified. Variants will be collapsed into a single test until the maximum size is reached and subsequent variants will be put into a new collapsed set.

CHAPTER 4

Choosing Parameters

4.1 Command Line Parameters

A command line version of VikNGS is available for users who wish to do association testing without running a user interface. The command line tool requires specification of a *multi-sample VCF file* and corresponding *sample information file*. By default, the command will run a common association tests on a single thread.

Run `vikNGS -h` for the list of relevant commands.

Parameter	Value/Default	Description
–vcf, -v	[DIRECTORY]	Directory of a multi-sample VCF file (required)
–sample, -i	[DIRECTORY]	Directory of a file containing sample information (required)
–bed,-b	[DIRECTORY]	directory of a BED file for collapsing variants
–out, -o	[DIRECTORY]= .	Directory for output (defaults to current directory)
–help, -h		Print a help message and exit
–common, -c		Perform a common variant association test (default)
–rare, -r	[TEST NAME]	Perform a rare variant association test
–boot, -n	[INT]=1000	Number of bootstrap iterations to calculate
–stop, -s		Stop bootstrapping if p-value looks to be > 0.05
–collapse, -k	[INT]=5	Collapse every k variants (rare only)
–gene		Collapse variants by gene if BED file specified (default)
–exon		Collapse variants by exon if BED file specified
–from	[INT]	Only include variants with **POS** larger than this value
–to	[INT]	Only include variants with **POS** smaller than this value
–chr	[CHR NAME]	Only include variants on this chromosome
–maf, -m	[FLOAT]=0.05	Minor allele frequency cut-off (common-rare threshold)
–depth, -d	[INT]=30	Read depth cut-off (low-high read depth threshold)
–missing, -x	[FLOAT]=0.1	Missing data cut-off (maximum tolerance for missing data)
–all, -a		Include variants which do not have *PASS* in the **FILTER** column
–threads, -t	[INT]=1	Number of threads
–batch, -h	[INT]=1000	Number of variants to read from VCF before beginning tests

Example 1. Running a common test on 16 threads for variants on chromosome 7 with minor allele frequency > 10% and ignoring what is in the **FILTER** column of the VCF:

```
./VikNGS --vcf [...] --sample [...] --chr chr7 -m 0.1 --all -t 16
```

Example 2. Running a rare test (CAST) on 4 threads, collapsing variants along genes and using one million bootstrap iterations with early stopping:

```
./VikNGS --vcf [...] --sample [...] --bed [...] -r cast --gene -n 1000000 --stop -t 4
```

4.2 Parameter Explaination

4.2.1 Minor Allele Frequency Cutoff

While reading the VCF file, VikNGS computes an allele frequency for each variant. The minor allele frequency (MAF) is estimated only using the samples included in the multisample VCF file. The MAF cutoff is used to define which variants are considered "rare" versus "common". When running a common association test, variants with estimated minor allele frequencies *less than* the MAF cutoff (ie. rare variants) will be excluded from testing. Likewise, when running a rare association test, variants with estimated minor allele frequencies *greater than* the MAF cutoff (ie. common variants) will be excluded from testing.

4.2.2 Missing Data Threshold

Variants may have ambiguous or missing genotype information (ex. **GT** = ./.) for some of the individuals in the multi-sample VCF file. If too much data is missing, association tests may produce misleading results. Any variant that is missing more data than this threshold will be excluded from testing. The default value is 0.1 which means if more than 10% of sample calls cannot be determined, the variant will be ignored.

Note: If running a quantitative association test, the proportion of missing data will be calculated from all samples. In a case-control test, two proportions will be calculated (one for all cases, one for all controls) if either cases *or* controls fail to satisfy the missing threshold, the variant will be excluded.

4.2.3 Filter By Genomic Coordinate

Enables filtering of variants based on the **CHR** and **POS** values in the VCF file. Variants outside a specific chromosome or range of positions can be excluded.

4.2.4 Must *PASS*

Variants which do not contain "*PASS*" in the **FILTER** column of the VCF are filtered out. By default this filtering step is on, turning it off will cause the contents of the **FILTER** column to be ignored.

4.2.5 Read Depth High/Low Cutoff

Samples with read depth above this threshold are considered high read depth samples (default=30). This is only used for the vRVS test if read depth values are provided in the sample infomation file.

4.2.6 Collapse Variants

See information on the *BED file section* on the Input page for details.

4.2.7 Testing Parameters

See information on the *Tests* page for details on the tests available.

Note: Use `-r cast` and `-r skat` for the CAST-like and SKAT-like tests, respectively.

4.2.8 Threads and Batch Size

Number of threads to perform association testing on. Batch size is the number of variants to process at one time on a given thread.

Warning: VikNGS will parse the VCF file line-by-line and store the data in memory. When using a large batch size, please keep in mind the memory limits of your device as these settings will determine how much memory is used.

4.2.9 Plot Results

Only available on the graphical user interface. A plotting interface will be displayed following the association testing in a new window if this setting is checked.

4.2.10 Explain Filter

Writes a file that explain why filtered variants were filtered if checked. See *Output <output>* for more details.

4.2.11 Retain Genotypes

This setting will store genotypes parsed from the VCF file in memory and will enable exploration of these values after p-values have been calculated.

Warning: Retaining all genotypes is extremely memory-intensive since a large amount of the data from the VCF file is being stored in memory simultaneously. Please only use this option for small datasets or on machines with very large amounts of memory.

CHAPTER 5

Association Tests

5.1 Common Single Variant Association Test

For both quantitative and binary trait analyses, a common variant test refers to a score test which has a Chi-squared distribution with 1 degree of freedom under no association hypothesis. The general form of the score test appears as follows:

$$T = \frac{S^2}{var(S)}$$

Where \(T\) is the test statistic following a Chi-squared distribution and \(S\) is the score. This test is used to perform a genetic association analysis between the phenotype \(Y\) and a single variant \(G_j\). For testing variant \(j\) given \(n\) individuals and phenotype vector \(Y\) and genotype matrix \(G\),

$$S_j = \sum_{i=1}^{n}(Y_i - E(Y_i))G_{ij}$$

$$E(Y_i) = Y_i - \hat{Y}$$

\(E(Y_i)\) is estimated from a vector of fitted values \(\hat{Y}\) which is dependent on the underlying distribution of \(Y\) (ex. case-control vs quantatitive). With no covariates, \(\hat{Y}=\bar{Y}\) which is the simple average of the observed phenotypes.

Under a case-control setting with no covariates, the score is an indication of how often the tested genotype appears in one group over the other. When coded as \(Y_i=1\) for cases and \(Y_i=0\) for controls, and genotypes coded as {0,1,2} corresponding to the number of alleles a particular individual possesses. Given this framework, cases with the allele of interest contribute positively to the overall score and controls contribute negatively. Therefore, the more a particular allele is associated with one group, the larger the magnitude of the score.

For genotypes coded strictly as {0,1,2}, the conventional variance formula is used to calculate \(var(S_j)\). To produce the test statistic \(T_j\), the square of the score \(S_j\) is normalized by the variance \(var(S_j)\) and a p-value is produced by evaluating \(T_j\) with respect to a Chi-squared distribution with 1 degree of freedom. In general, a large score and a small variance will result in a small (more significant) p-value.

In the vRVS methodology available in VikNGS, the genotype value \(G_{ij}\) is replaced with the expected genotype value calculated from the sequence read data \(E(G_{ikj}\mid D_{ikj})\). When integrating data from an arbitrary number of cohorts, the variance is calculated for each group separately and summed together to produce \(var(S_j)\).

The details of the derivation of \(var(S_j)\) are given in the Supplementary document of the VikNGS paper *VIKNGS: A C++ Variant Integration Kit for next generation sequencing association analysis*.

5.2 Rare Variant Association Test

For joint variant analysis, the score statistics for \(J\) variants, \(\boldsymbol{S}=[S_1,…,S_J]\). Please review the common variant section above to review the general structure of a score test. In VikNGS, multiple different genetic association tests are available which are described in the sections below.

For the CAST- and SKAT-like tests, we recommend the use of permutation to calculate p-values. This involves shuffling the phenotype vector \(Y\) and recalculating the p-value many times for every variant. After iteratively calculating a set of p-values, the final p-value is calculated based on the number of values that are less than or equal to the value that was calculated for the unshuffled data set divided by the number of iterations plus 1.

Note: Using permutation, the smallest p-value obtainable is 1/(# iterations + 1). Since this method can be very computationally expensive, an an early stopping procedure is available to terminate the calculation early if the p-value appears to be > 0.05. This uses the method designed by Jiang and Salzman (2012).

When using expected genotypes and the vRVS methodology, the fact that data could be combined from multiple different cohorts prevents the use of a simple permutation test. Instead, the bootstrap approach defined by Derkach *et al*. (2012) was adopted the for binary trait (case-control) analysis. Given a matrix of expected genotypes, the mean genotype is subtracted from each matrix element and rows are selected at random with replacement to form a shuffled matrix. This is done for every group separately. Covariates are also bootstrapped independently from the genotypes. For quantitative trait analysis using expected genotypes, we implement the permutation methodology defined by Lin and Tang (2011) within each combined group.

Warning: In VikNGS, these tests can be run by assuming the asymptotic distribution by setting the number of iterations to 1. Based on our limited testing, the results appear to behave as expected but we offer no statistical guarantees.

5.2.1 Linear Test (CAST-like)

This test related to the CAST method described by Morgenthaler and Thilly (2007). In this test, a score vector of size \(J\) is calculated, each element corresponding to a different variant. Each score in the vector is calculated using the method described in the common variant section above. A single score value is produced by summing the elements of the score vector.

Note: Since this test uses a sum of scores, it is very powerful when all variants have the same directional impact on disease risk. Combining protective and harmful variants in the same test will result in severely reduced statistical power.

5.2.2 Quadratic Test (SKAT-like)

This test related to the SKAT method described by Wu *et al*. (2011). Similar to the linear test, a score vector of size \(J\) is calculated, each element corresponding to a different variant. Variants are weighted based on minor allele frequency (MAF): \(w^{1/2}=1/[MAF(1-MAF)]^{1/2}\). The p-value is calculated using the C++ code underlying the CompQuadForm (Distribution Function of Quadratic Forms in Normal Variables) R library which is used in the R SKAT package.

Note: This method should be preferred over the linear test when both protective and harmful variants being collapsed together (of if it is unclear whether the variants are potentially protective or harmful).

5.2.3 Likelihood Method (Coming soon)

This method refers to the test described in *Association testing for next-generation sequencing data using score statistics* _ from Skotte and Albrechtsen (2012) Their method provides a score test where genotype calls are substituted by their expected values, \(E(G_{ikj}\mid D_{ikj})\). The variance of the score test is obtained from the second derivative of the joint likelihood of the observed \(Y_i\) and the observed sequencing data, \(D_{ij}\) individual \(i\) at locus \(j\). The p-values are calculated using the asymptotic distribution of the score test. For a joint rare analysis of \(J\) variants, the score test is distributed as a chi-square distribution with \(J\) degrees of freedom. This can also be used for common single variant association test which is distributed as chi-squared with one degree of freedom.

CHAPTER 6

Output

VikNGS will output the p-values calculated to a text file in the current directory (i.e. ".") by default. Output directory can be changed from the user interface or using `-o [DIRECTORY]` on the command line version.

In addition to p-values, VikNGS can also produce a file detailing which variants were filtered out prior to testing and an explanation why. In future versions of the software, we hope to provide more types of output in addition to these files.

6.1 P-value File

This file will be produced upon completion of a series of association tests. Each line corresponds to a tested variant and records the resulting p-value. The file is tab-separated with the following columns:

```
Chromosome Position Reference Alternative P-value TestName CollapseGroupID
```

`CollapseGroupID` is only present if variants were collapsed into groups (single p-value per collapsed group).

6.2 Filtered File

This file explains which variants were filtered prior to testing and the reasoning behind the filter. Each line corresponds to a single variant. This file will only be written if "Explain Filter" is checked on the user interface or if `--explain-filter` is added when running VikNGS from the command line. The file is tab-separated with the following columns:

```
Chromosome Position Reference Alternative Explanation
```

The `Explanation` values refer to the following:

- Invalid information: variant was not properly parsed from the VCF file
- Not SNP: variant was filtered for not being a single nucleotide polymorphism
- PASS fail: **FILTER** column of VCF file did not report "PASS"

- Missing: variant was missing (e.x. "./.") more data than tolerable by the missing threshold parameter
- No variation: no variation detected in this particular variant (all individuals have the same genotype)
- MAF: variant has a minor allele frequency > the MAF threshold (rare test) or < the MAF threshold (common test)

CHAPTER 7

Power Simulation Package

Power simulation package can be used for two purposes:

Type I error: The user can test the performance of the association tests with respect to the control of Type I error under different sequencing settings, e.g. different combinations of read depths with varying sample sizes, different base calling errors. In this setting odds ratio (\(OR\)) for a binary trait analysis is set to 1, and the proportion of variation explained by the genetic effect (\(R^2\)-coefficient of determination) for quantitative trait analysis is set to 0.

Power analysis: The user can calculate the minimum sample size required to detect a prespecified effect size, e.g. \(OR=1.2\) for a binary trait analysis and (\(R^2=0.1\) for quantitative trait analysis.

7.1 Parameters

Many parameters are unique to the simulation package. Please check the *relevant section* for information regarding parameters not specific to this component.

7.1.1 Phenotype and Cohort Parameters

Data can be simulated for either a case-control phenotype or a Normally distributed quantitative phenotype. The mean and standard deviation of the Normal phenotype can be altered but these parameters have little to no influence on the result. Groups of individuals are specified in a table. The size of each group can be a single value or a range (annotated as two numbers separated by a colon ":", e.g. 500:1500). If a range is given, the simulation will run multiple times with the sample size of the group increasing from the low end of the range to the high end. The "Cohort" column of the table indicates case/control status or will simply say "quantitative" depending on the distribution of the phenotype. The "Mean Depth" and "Depth SD" refer to the average read depth and standard deviation for each simulated group. The package will simulate a set of reads for each variant and the read depth will be sampled from a Normal distribution with these parameters. "Error Rate" is the base calling error rate. A value of 0.01 means that 1% of all the reads will report an incorrect base call for a given variant.

7.1.2 Variant Parameters

Effect size (odds ratio for case-control and \(R^2\) correlation for quantitative) determines the strength of the relationship between the genotype and the phenotype. An odds ratio of 1 or an \(R^2\) of 0 is what is used to simulate data under the null hypothesis (no association). A range of minor allele frequencies (MAFs) must also be provided. For each variant, the true MAF is selected uniformly at random for each variant. The minor allele is always simulated to be the causal variant.

7.1.3 Phenotype and Cohort Parameters

High/low cut-off defines the value which discriminates between high and low read depth groups. A high/low cut-off of 30 indicates that cohorts with a mean read depth less than 30x will be considered a low read depth cohort. This parameter is used by vRVS. Changing the number of steps alters the sample size increment on the cohort table (e.g. given a sample size of 500:1500 defined for controls and steps=3, three sets of simulations will run with the number of controls as 500, 1000 and 1500). The results will be saved and plotted on the sample size versus power graph.

7.2 Output

The program generates a Q-Q plot and histograms of the p-values when Type I error rate is of interest. For power analysis, the relationship between power and sample size can be studied by changing the step size and the sample size values in the cohort table.