Integrated Annotation and Analysis of Genetic Variants from Next-generation Sequencing Studies with *Variant Tools*

Bo Peng, Ph.D.

Department of Bioinformatics and Computational Biology
The University of Texas MD Anderson Cancer Center

Oct 3rd and 10th, 2013

Outline

Background

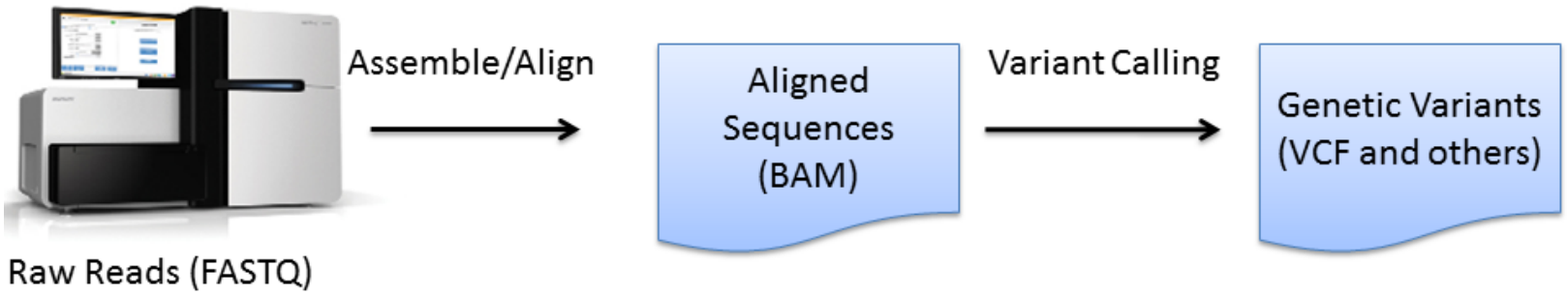
Introduction to variant tools
- Overview
- Basic concepts

A real-world example
- Import data
- Phenotype and sample statistics
- Annotation
- Select and filter variants
- Output variants and their summary statistics

More advanced features
- Definition and execution of pipelines
- Association Analysis Framework

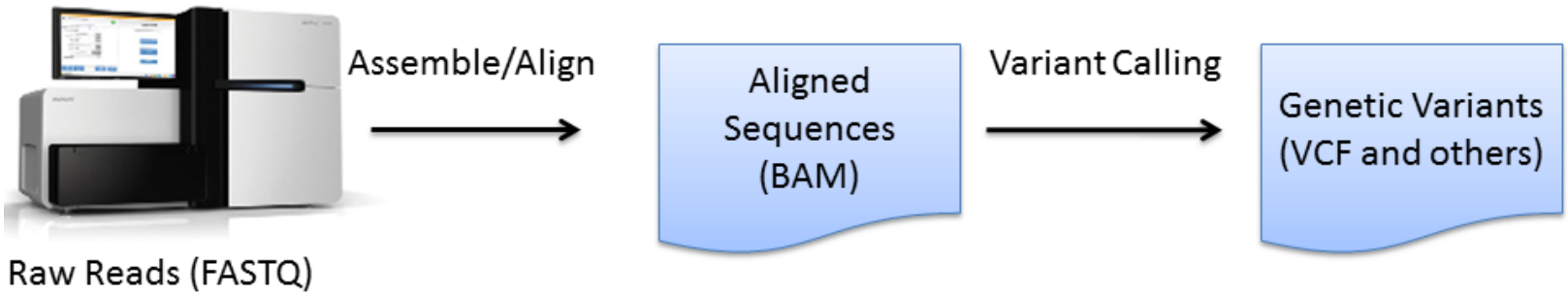
Conclusion

Sequencing Analysis: variant calling



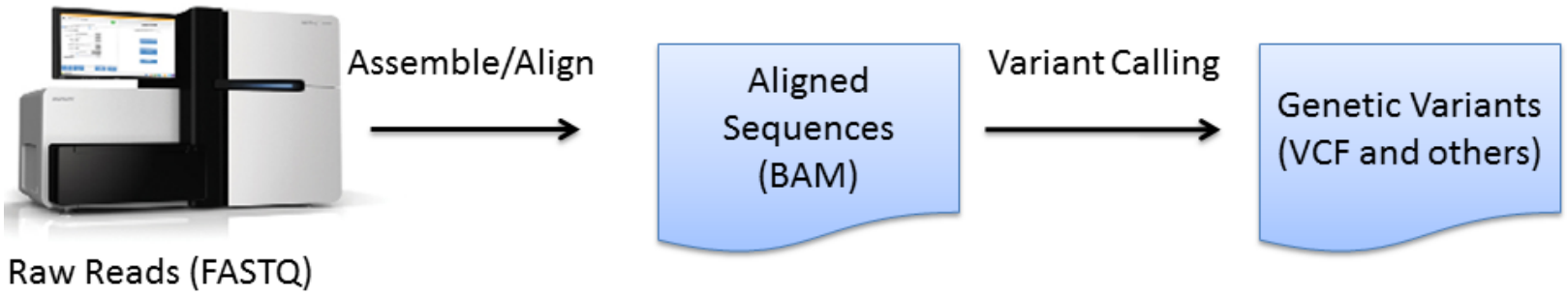
- Align raw reads from different platforms (Sanger Capillary, Roche 454, Illumina, Applied Biosystems SOLID, Complete Genomcs, Ion Torrent, ...) to a reference genome, using different aligners such as SNAP, iSAAC, NovoAlign, Razers3, bwa, bowtie, STAR, TopHat.

Sequencing Analysis: variant calling



- Align raw reads from different platforms (Sanger Capillary, Roche 454, Illumina, Applied Biosystems SOLID, Complete Genomcs, Ion Torrent, ...) to a reference genome, using different aligners such as SNAP, iSAAC, NovoAlign, Razers3, bwa, bowtie, STAR, TopHat.
- Call small (SNVs, insertions and deletions) and structural variants (difference in the copy number, orientation or location of genomic segments > 100bp) from aligned reads, using variant calling and SV discovery tools such as GATK, CASAVA, BreakDancer, CLEVER, VNCer, PEMer, SLOPE.

Sequencing Analysis: variant calling



- Align raw lary, Roche 454, Il omcs, Ion Torrent, igners such as SN vtie, STAR, Top

Variants called from different variant-calling pipelines usually do not quite agree with each other.

- Call small (SNVs, insertions and deletions) and structural variants (difference in the copy number, orientation or location of genomic segments > 100bp) from aligned reads, using variant calling and SV discovery tools such as GATK, CASAVA, BreakDancer, CLEVER, VNCer, PEMer, SLOPE.

Sequencing Analysis: Annotation and Prioritization

- Region: Is a variant in a gene (ref seq gene, known gene, CCDS gene), in exome regions of a gene, in a genomic duplication region?
- Database membership: Is the variant in dbSNP, 1000 genomes, dbNSFP, COSMIC (Catalogue of Somatic Mutations in Cancer), ESP (Exome Sequencing Project), gwas gatalog? Does it belong to any known cancer gene, pathway?
- Functional prediction: Is it predicted to be damaging (SIFT, Polyphen2, LRT, MutationTaster, FATHMM, GERP, PhyloP scores) or in an evolutionarily conserved region (PhastCons)?
- Population statistics: What are the population or sample frequency of the variant?

Sequencing Analysis: Association and Other Analyses

In addition to numerous applications in functional genomics, NGS data have been used to

- Identify De Novo mutations: Identify alterations that are present for the first time in one family member as a result of mutations in a germ cell (egg or sperm) of one of the parents or in the fertilized egg itself.
- Associate genotype to phenotype: Associate variants (for highly penetrant variants for Mendelian diseases) or genes (for complex traits) to qualitative or quantitative traits, using case control or family based study designs.

Challenges

- Many different pipelines for read alignment and variant calling

Challenges

- Many different pipelines for read alignment and variant calling
- Wide array of formats for sample variants and annotations
  - Text-based formats from different calling algorithms
  - Variant Call Format (VCF)
  - BED6, BED12, GFF and other annotation formats

Challenges

- Many different pipelines for read alignment and variant calling
- Wide array of formats for sample variants and annotations
  - Text-based formats from different calling algorithms
  - Variant Call Format (VCF)
  - BED6, BED12, GFF and other annotation formats
- Continually added and updated annotation sources
  - Updated data from 1000 genomes and other projects
  - Annotations might use newer reference genomes

Challenges

- Many different pipelines for read alignment and variant calling
- Wide array of formats for sample variants and annotations
  - Text-based formats from different calling algorithms
  - Variant Call Format (VCF)
  - BED6, BED12, GFF and other annotation formats
- Continually added and updated annotation sources
  - Updated data from 1000 genomes and other projects
  - Annotations might use newer reference genomes
- Availability of a number of evolving tools with different input/output formats
  - ANNOVAR for functional annotation
  - BEDTools for comparing genomic features
  - PLINK/SEQ and GoldenHelix SVS

Outline

Introduction to variant tools
- Overview
- Basic concepts

Design of *Variant Tools*

variant tools is a toolkit for the integrated annotation and analysis of genetic variants from next-gen sequencing studies.

DESIGN OF *Variant Tools*

variant tools is a toolkit for the integrated annotation and analysis of genetic variants from next-gen sequencing studies.

- Project-based organization to reduce intermediate results and files, with a flexible command line interface and extensive documentation

Design of *Variant Tools*

variant tools is a toolkit for the integrated annotation and analysis of genetic variants from next-gen sequencing studies.

- Project-based organization to reduce intermediate results and files, with a flexible command line interface and extensive documentation
- File format specification system, standardized annotation databases, and support for an alternative reference genome to free users from details about file formats and reference genomes

DESIGN OF *Variant Tools*

variant tools is a toolkit for the integrated annotation and analysis of genetic variants from next-gen sequencing studies.

- Project-based organization to reduce intermediate results and files, with a flexible command line interface and extensive documentation
- File format specification system, standardized annotation databases, and support for an alternative reference genome to free users from details about file formats and reference genomes
- Unified handling of variant info, annotation and track fields allows easy annotation, selection and reporting of variants according to multiple annotation sources

DESIGN OF *Variant Tools*

variant tools is a toolkit for the integrated annotation and analysis of genetic variants from next-gen sequencing studies.

- Project-based organization to reduce intermediate results and files, with a flexible command line interface and extensive documentation
- File format specification system, standardized annotation databases, and support for an alternative reference genome to free users from details about file formats and reference genomes
- Unified handling of variant info, annotation and track fields allows easy annotation, selection and reporting of variants according to multiple annotation sources
- An association analysis framework allows flexible and extensible association analysis

DESIGN OF *Variant Tools*

variant tools is a toolkit for the integrated annotation and analysis of genetic variants from next-gen sequencing studies.

- Project-based organization to reduce intermediate results and files, with a flexible command line interface and extensive documentation
- File format specification system, standardized annotation databases, and support for an alternative reference genome to free users from details about file formats and reference genomes
- Unified handling of variant info, annotation and track fields allows easy annotation, selection and reporting of variants according to multiple annotation sources
- An association analysis framework allows flexible and extensible association analysis
- Online resource repository of annotation databases, file formats, snapshots etc.

STATUS OF VARIANT TOOLS

- 16 `vtools` commands
- 8 `vtools_report` commands

- Bo Peng
  Gao Wang
  Anthony San Lucas

- GPL3
- Version 2.0.1 as of today
- Application note published in bioinformatics

- More than 200 registered users

A SVN-LIKE SUBCOMMAND INTERFACE

```
$ vtools -h
usage: vtools [-h] [--version]

              {init,import,phenotype,show,liftover,use,update,select,exclude,compare,output,export
                    ,remove,associate,admin,execute}
              ...

A variant calling, processing, annotation and analysis tool for next-
generation sequencing studies.

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

subcommands:
  {init,import,phenotype,show,liftover,use,update,select,exclude,compare,output,export,remove,
       associate,admin,execute}
    init                Create a new project, or a subproject from an existing
                        parent project, or merge several existing projects
                        into one
    import              Import variants and related sample genotype from files
                        in specified formats
    phenotype           Manage sample phenotypes
    show                Display content of a project
    liftover            Set alternative reference genome and update
                        alternative coordinates of all variant tables
    use                 Prepare (download or import if necessary) and use an
                        annotation database
    update              Add or update fields of existing variants and genotype
                        using information from specified existing fields,
                        sample genotype, or external files
    select              Output or save select variants that match specified
                        conditions
    exclude             Output or save variants after excluding variants that
```

GETTING HELP

```
$ vtools init -h
usage: vtools init [-h] [-f] [--parent DIR] [--variants [TABLE]]
                   [--samples [COND [COND ...]]]
                   [--genotypes [COND [COND ...]]] [--children DIR [DIR ...]]
                   [-v {0,1,2}]
                   project

Create a new project in the current directory. This command will fail if
another project already exists in this directory, unless option '--force' is
used to remove the existing project.

positional arguments:
  project               Name of a new project. This will create a new .proj
                        file under the current directory. Only one project is
                        allowed in a directory.

optional arguments:
  -h, --help            show this help message and exit
  -f, --force           Remove a project if it already exists.
  -v {0,1,2}, --verbosity {0,1,2}
                        Output error and warning (0), info (1) and debug (2)
                        information to standard output (default to 1).

Derive from a parent project:
  --parent DIR          Directory of a parent project (e.g. --parent ../main)
                        from which all or part of variants (--variants),
                        samples (--samples) and genotypes (--genotypes) will
                        be copied to the newly created project.
  --variants [TABLE]    A variant table of the parental project whose variants
                        will be copied to the new project. Default to variant
                        (all variants).
  --samples [COND [COND ...]]
                        Copy only samples of the parental project that match
                        specified conditions.
```

GETTING HELP

```
$ vtools init -h
usage: vtools init [-h] [-f] [--parent DIR] [-
                   [--samples [COND [COND ...]
                   [--genotypes [COND [COND ..
                   [-v {0,1,2}]
                   project

Create a new project in the current directory. This command will fail if
another project already exists in this directory, unless option '--force' is
used to remove the existing project.

positional arguments:
  project               Name of a new project. This will create a new .proj
                        file under the current directory. Only one project is
                        allowed in a directory.

optional arguments:
  -h, --help            show this help message and exit
  -f, --force           Remove a project if it already exists.
  -v {0,1,2}, --verbosity {0,1,2}
                        Output error and warning (0), info (1) and debug (2)
                        information to standard output (default to 1).

Derive from a parent project:
  --parent DIR          Directory of a parent project (e.g. --parent ../main)
                        from which all or part of variants (--variants),
                        samples (--samples) and genotypes (--genotypes) will
                        be copied to the newly created project.
  --variants [TABLE]    A variant table of the parental project whose variants
                        will be copied to the new project. Default to variant
                        (all variants).
  --samples [COND [COND ...]]
                        Copy only samples of the parental project that match
                        specified conditions.
```

The variant tools website has detailed documentation, sample projects, examples, and tutorials for all commands.

Outline

Introduction to variant tools
- Overview
- Basic concepts

PROJECT

A *project* contains one or more databases and some runtime data. A directory can have one and only one project that will be opened by subsequent variant tools commands.

```
$ vtools init concept
INFO: variant tools 2.0.1 : Copyright (c) 2011 - 2012 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 concept

$ wget ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/pilot_data/release/2010_07/exon/snps/CEU.exon
      .2010_03.sites.vcf.gz
$ wget ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/pilot_data/release/2010_07/exon/snps/JPT.exon
      .2010_03.sites.vcf.gz
$ vtools import CEU.exon.2010_03.sites.vcf.gz --sample_name CEU --var_info AA DP --build hg18
$ vtools import JPT.exon.2010_03.sites.vcf.gz --sample_name JPT --var_info AA DP

$ vtools show
Project name:                 concept
Created on:                   Wed Oct  7 11:13:16 2013
Primary reference genome:     hg18
Secondary reference genome:   None
Runtime options:              verbosity=1
Variant tables:               variant
Annotation databases:
```

VARIANT AND VARIANT TABLE

A variant refers to a mutation from `ref` to `alt` at `pos` of `chr`. A variant in *variant tools* can be SNV, small indel, or MNPs (Multiple-nucleotide polymorphism). All variants are assumed to be on the forward (+) strand.

```
$ vtools show tables
table      #variants     date message
variant        4,858    Oct02 Master variant table

$ vtools output variant chr pos ref alt --limit 5
1  1105366  T  C
1  1105411  G  A
1  1108138  C  T
1  1110240  T  A
1  1110294  G  A

$ vtools select variant 'ref="T"' --to_table refT 'variants with reference allele T'
Running: 2 846.4/s in 00:00:00
INFO: 787 variants selected.

$ vtools show tables
table      #variants     date message
refT             787    Oct02 variants with reference allele T
variant        4,858    Oct02 Master variant table

$ vtools output refT chr pos ref alt -l 5
1  1105366  T  C
1  1110240  T  A
1  3537996  T  C
1  6447088  T  C
1  6447275  T  C
```

VARIANT AND VARIANT TABLE

A variant refers to a mutation from `ref` to `alt` at `pos` of `chr`. A variant in *variant tools* can be SNV, small indel, or MNPs (Multiple-nucleotide polymorphism). All variants are assumed to be on the forward (+) strand.

```
$ vtools show tables
table       #variants     date message
variant         4,858    Oct02 Master variant table

$ vtools output variant
1   1105366   T   C
1   1105411   G   A
1   1108138   C   T
1   1110240   T   A
1   1110294   G   A

$ vtools select variant
Running: 2 846.4/s in
INFO: 787 variants sel

$ vtools show tables
table       #variants     date message
refT              787    Oct02 variants with reference allele T
variant         4,858    Oct02 Master variant table

$ vtools output refT chr pos ref alt -l 5
1   1105366   T   C
1   1110240   T   A
1   3537996   T   C
1   6447088   T   C
1   6447275   T   C
```

Variant Tools does not yet support large indels and structural variants such as inversions.

VARIANT INFO FIELD

Variant info fields provide annotation information for each variant. They are maintained inside the project.

```
$ vtools show fields
variant.chr
variant.pos
variant.ref
variant.alt
variant.AA
variant.DP

$ vtools output refT chr pos ref alt AA DP -l 5
1  1105366  T  C  T  3251
1  1110240  T  A  T  7275
1  3537996  T  C  C  1753
1  6447088  T  C  T  4691
1  6447275  T  C  T  6871

$ vtools update variant --from_file CEU.exon.2010_03.sites.vcf.gz --var_info id
INFO: Using primary reference genome hg18 of the project.
Getting existing variants: 100% [=======================] 3,188 231.4K/s in 00:00:00
INFO: Updating variants from CEU.exon.2010_03.sites.vcf.gz (1/1)
CEU.exon.2010_03.sites.vcf.gz: 100% [====================] 3,500 8.4K/s in 00:00:00
INFO: Field id of 1,531 variants are updated

$ vtools output refT chr pos ref alt id AA DP -l 5
1  1105366  T  C  .           T  3251
1  1110240  T  A  .           T  7275
1  3537996  T  C  rs2760321   C  1753
1  6447088  T  C  rs11800462  T  4691
1  6447275  T  C  rs3170675   T  6871
```

REFERENCE GENOME

A variant can have different chromosomal coordinates in different reference genomes. It is extremely important to know the reference genome used for your project.

```
$ vtools output variant chr pos ref alt 'ref_sequence(chr, pos, pos+5)' -l 5
1  1105366  T  C  TGTGGG
1  1105411  G  A  GGACCC
1  1108138  C  T  CAAGCC
1  1110240  T  A  TGCTGC
1  1110294  G  A  GTGACA

$ vtools liftover hg19
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exporting variants: 100% [==============] 4,858 129.0K/s in 00:00:00
INFO: Running UCSC liftOver tool
Updating table variant: 100% [===========] 4,858 28.4K/s in 00:00:00

$ vtools output variant chr pos ref alt 'ref_sequence(chr, pos, pos+5)' -l 5 --build hg19
1  1115503  T  C  TGTGGG
1  1115548  G  A  GGACCC
1  1118275  C  T  CAAGCC
1  1120377  T  A  TGCTGC
1  1120431  G  A  GTGACA
```

REFERENCE GENOME

A variant can have different chromosomal coordinates in different reference genomes. It is extremely important to k
project.

Functio `ref_sequence` returns the reference sequence at or around variant location, which is unrelated to your data. If `ref_sequence(chr, pos)` returns a different reference allele from the `ref` of your variant, you might have specified a wrong reference genome for your data.

```
$ vtools output variant chr pos ref alt 'ref_s
1  1105366  T  C  TGTGGG
1  1105411  G  A  GGACCC
1  1108138  C  T  CAAGCC
1  1110240  T  A  TGCTGC
1  1110294  G  A  GTGACA

$ vtools liftover hg19
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exporting variants: 100% [==============] 4,858 129.0K/s in 00:00:00
INFO: Running UCSC liftOver tool
Updating table variant: 100% [===========] 4,858 28.4K/s in 00:00:00

$ vtools output variant chr pos ref alt 'ref_sequence(chr, pos, pos+5)' -l 5 --build hg19
1  1115503  T  C  TGTGGG
1  1115548  G  A  GGACCC
1  1118275  C  T  CAAGCC
1  1120377  T  A  TGCTGC
1  1120431  G  A  GTGACA
```

Annotation Database

Variant tools supports four types of annotation databases:

- Variant: Annotate specific variant (`chr`, `pos`, `ref`, `alt`)
  dbNSFP, dbSNP, 1000 genomes
- Position: Annotate chromosomal position (`chr`, `pos`)
  gwasCatalog
- Range: Annotate regions (`chr`, `start`, `end`)
  refGene, knownGene, ccdsGene
  refGene_exon, knownGene_exon, ccdsGene_exon
- Attribute: Annotate attribute of variants (e.g. gene)
  keggPathway, Cancer Gene Census

Annotation databases are defined by `.ann` files. Database files (`.DB.gz`) are automatically downloaded from `http://vtools.houstonbioinformatics.org`.

Annotation Database

```
$ vtools show annotations -v0
CancerGeneCensus-20111215
CancerGeneCensus-20120315
CancerGeneCensus-20130711
CancerGeneCensus
CosmicCodingMuts-v61_260912
CosmicCodingMuts
CosmicMutantExport-v61_260912
CosmicMutantExport
CosmicNonCodingVariants-v61_260912
CosmicNonCodingVariants
ESP-6500SI-V2-SSA137
ESP
ccdsGene-hg19_20110909
ccdsGene-hg19_20111206
ccdsGene
ccdsGene_exon-hg19_20110909
ccdsGene_exon-hg19_20111206
ccdsGene_exon
ccdsGene_exon_hg19-20111206
ccdsGene_hg19-20111206
cytoBand-hg18_20111216
cytoBand-hg19_20111216
cytoBand
dbNSFP-hg18_hg19_1.1_2
dbNSFP-hg18_hg19_1_3
dbNSFP-hg18_hg19_2_0
dbNSFP
dbNSFP_gene-2_0
dbNSFP_gene
dbNSFP_light-hg18_hg19_1.0_0
dbNSFP_light-hg18_hg19_1_3
dbNSFP_light
dbSNP-hg18_129
```

Annotation Database

```
$ vtools show annotations -v0
CancerGeneCensus-20111215
CancerGeneCensus-20120315
CancerGeneCensus-20130711
CancerGeneCensus
CosmicCodingMuts-v61_260912
CosmicCodingMuts
CosmicMutantExport-v61_260912
CosmicMutantExport
CosmicNonCodingVariants-v61_260912
CosmicNonCodingVariants
ESP-6500SI-V2-SSA137
ESP
ccdsGene-hg19_20110909
ccdsGene-hg19_20111206
ccdsGene
ccdsGene_exon-hg19_20110909
ccdsGene_exon-hg19_20111206
ccdsGene_exon
ccdsGene_exon_hg19-20111206
ccdsGene_hg19-20111206
cytoBand-hg18_20111216
cytoBand-hg19_20111216
cytoBand
dbNSFP-hg18_hg19_1.1_2
dbNSFP-hg18_hg19_1_3
dbNSFP-hg18_hg19_2_0
dbNSFP
dbNSFP_gene-2_0
dbNSFP_gene
dbNSFP_light-hg18_hg19_1.0_0
dbNSFP_light-hg18_hg19_1_3
dbNSFP_light
dbSNP-hg18_129
```

Option `--verbosity 0/1/2` controls the verbosity of output. `-v1` of this command will output descriptions of annotation databases.

ANNOTATION DATABASE

```
$ vtools use dbNSFP
INFO: Downloading annotation database from annoDB/dbNSFP.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/dbNSFP-
      hg18_hg19_2_0.DB.gz
INFO: Using annotation DB dbNSFP in project concept.
INFO: dbNSFP version 2.0, maintained by Xiaoming Liu from UTSPH. Please cite
"Liu X, Jian X, and Boerwinkle E. 2011. dbNSFP: a lightweight database of human
non-synonymous SNPs and their functional predictions. Human Mutation. 32:894-899" and
"Liu X, Jian X, and Boerwinkle E. 2013. dbNSFP v2.0: A Database of Human Nonsynonymous
SNVs and Their Functional Predictions and Annotations. Human Mutation. 34:E2393-E2402."
if you find this database useful.
```

Under the hook, vtools will

- Check for a local database `dbNSFP.DB` and use it if possible
- If unavailable, download `dbNSFP.ann` from web
- If available, download the latest version of `dbNSFP-$version.DB.gz` from web and use it
- If failed, download source of dbNSFP from a URL specified in `dbNSFP.ann`
- If succeed, create a database from source

ANNOTATION DATABASE

```
$ vtools show annotation dbNSFP
Annotation database dbNSFP (version hg18_hg19_2_0)
Description:            dbNSFP version 2.0, maintained by Xiaoming Liu from
  UTSPH. Please cite "Liu X, Jian X, and Boerwinkle E. 2011. dbNSFP: a
  lightweight database of human non-synonymous SNPs and their functional
  predictions. Human Mutation. 32:894-899" and "Liu X, Jian X, and Boerwinkle
  E. 2013. dbNSFP v2.0: A Database of Human Nonsynonymous SNVs and Their
  Functional Predictions and Annotations. Human Mutation. 34:E2393-E2402." if
  you find this database useful.
Database type:          variant
Reference genome hg18:  chr, hg18_pos, ref, alt
Reference genome hg19:  chr, pos, ref, alt
  chr                   Chromosome number
  pos                   physical position on the chromosome as to hg19
                        (1-based coordinate)
  ref                   Reference nucleotide allele (as on the + strand)
  alt                   Alternative nucleotide allele (as on the + strand)
  aaref                 reference amino acid
  aaalt                 alternative amino acid
  hg18_pos              physical position on the chromosome as to hg19
                        (1-based coordinate)
  genename              common gene name
  Uniprot_acc           Uniprot accession number. Multiple entries separated
                        by ";".
  Uniprot_id            Uniprot ID number. Multiple entries separated by ";".
  Uniprot_aapos         amino acid position as to Uniprot. Multiple entries
                        separated by ";".
  Interpro_domain       Interpro_domain: domain or conserved site on which the
                        variant locates. Domain annotations come from Interpro
                        database. The number in the brackets following a
                        specific domain is the count of times Interpro assigns
                        the variant position to that domain, typically coming
                        from different predicting databases. Multiple entries
```

ANNOTATION DATABASE

```
$ vtools show fields
variant.chr
variant.pos
variant.ref
variant.alt
variant.AA
variant.AC
variant.AN
variant.DP
variant.id
dbNSFP.chr                  Chromosome number
dbNSFP.pos                  physical position on the chromosome as to hg19
                            (1-based coordinate)
dbNSFP.ref                  Reference nucleotide allele (as on the + strand)
dbNSFP.alt                  Alternative nucleotide allele (as on the + strand)
dbNSFP.aaref                reference amino acid
dbNSFP.aaalt                alternative amino acid
dbNSFP.hg18_pos             physical position on the chromosome as to hg19 (1-based
                            coordinate)
dbNSFP.genename             common gene name
dbNSFP.Uniprot_acc          Uniprot accession number. Multiple entries separated by ";".
dbNSFP.Uniprot_id           Uniprot ID number. Multiple entries separated by ";".
dbNSFP.Uniprot_aapos        amino acid position as to Uniprot. Multiple entries separated
                            by ";".
dbNSFP.Interpro_domain      Interpro_domain: domain or conserved site on which the variant
                            locates. Domain annotations come from
                            Interpro database. The number in the
                            brackets following a specific domain is
                            the count of times Interpro assigns the
                            variant position to that domain,
                            typically coming from different
                            predicting databases. Multiple entries
                            separated by ";".
```

ANNOTATION DATABASE

```
$ vtools output refT chr pos ref alt genename SIFT_score KGp1_AFR_AF -l5
1  1105366  T  C  TTLL10    0.07  0.00406504065041
1  1110240  T  A  TTLL10    0.92  0.0
1  3537996  T  C  .         .     .
1  6447088  T  C  TNFRSF25  0.29  0.211382113821
1  6447275  T  C  .         .     .

$ vtools select variant 'SIFT_score < 0.05' -o chr pos ref alt SIFT_score Polyphen2_HDIV_score
     Polyphen2_HDIV_pred -l 10
1  3541597   C  T  0.0   1.0          D
1  18022097  G  T  0.0   0.004        B
1  18022200  C  A  0.0   0.999        D
1  18022253  A  G  0.0   0.649        P
1  25442668  T  C  0.04  0.087        B
1  25445571  T  G  0.0   0.999        D
1  25445572  C  T  0.0   0.99         D
1  25445603  A  G  0.0   0.999        D
1  35999342  C  G  0.01  0.99;1.0     D;D
1  36002845  T  G  0.01  0.649;0.825  P;P
```

Annotation Database

```
$ vtools output refT chr pos ref alt genename SIFT_score KGp1_AFR_AF -l5
1  1105366  T  C  TTLL10    0.07  0.00406504065041
1  1110240  T  A  TTLL10    0.92  0.0
1  3537996  T  C  .         .     .
1  6447088  T  C  TNFRSF25  0.29  0.211382113821
1  6447275  T  C  .         .     .

$ vtools select variant 'SIFT_score < 0.05' -o chr pos ref alt SIFT_score Polyphen2_HDIV_score
      Polyphen2_HDIV_pred -l 10
1  3541597   C  T  0.0   1.0          D
1  18022097  G  T  0.0   0.004        B
1  18022200  C  A  0.0
1  18022253  A  G  0.0
1  25442668  T  C  0.0
1  25445571  T  G  0.0
1  25445572  C  T  0.0
1  25445603  A  G  0.0
1  35999342  C  G  0.0
1  36002845  T  G  0.0
```

Please pay close attention to the description of fields before using them. For example, a variant is predicted to be damaging with smaller SIFT score but higher Polyphen2 scores.

Track

Track files provide additional annotation information to variants (e.g. info fields in vcf files) or positions (e.g. alignment information at positions).

```
$ tabix -p vcf CEU.exon.2010_03.sites.vcf.gz
$ vtools show track CEU.exon.2010_03.sites.vcf.gz
Version                 VCF v4.0
Number of fields:       8

Header: (excluding INFO and FORMAT lines)
                        ##reference=human_b36_both.fasta

Available fields (with type VARCHAR if unspecified or all=1):
0 (INTEGER)             1 if matched
chr (1, chrom)          chromosome
pos (2, INTEGER)        position (1-based)
name (3)                name of variant
ref (4)                 reference allele
alt (5)                 alternative alleles
qual (6)                qual
filter (7)              filter
info (8, default)       variant info fields
info.DP (INTEGER)       Total Depth
info.HM2 (INTEGER, flag) HapMap2 membership
info.HM3 (INTEGER, flag) HapMap3 membership
info.AA                 Ancestral Allele, ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/
     technical/reference/ancestral_alignments/README
info.AC (INTEGER)       total number of alternate alleles in called genotypes
info.AN (INTEGER)       total number of alleles in called genotypes
format (9)              genotype format
```

TRACK

```
$ vtools output refT chr pos ref alt "track('CEU.exon.2010_03.sites.vcf.gz', 'info.AA')" -l5
1  1105366  T  C  T
1  1110240  T  A  T
1  3537996  T  C  C
1  6447088  T  C  T
1  6447275  T  C  T

$ vtools select variant "track('CEU.exon.2010_03.sites.vcf.gz', 'info.DP') > 1000" --output chr
    pos ref alt DP -l5
1  1105366  T  C  3251
1  1105411  G  A  2676
1  1108138  C  T  2253
1  1110240  T  A  7275
1  1110294  G  A  7639

$ vtools liftover hg19
$ vtools output variant chr pos "track('http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release
    /20110521/ALL.chr1.phase1_release_v3.20101123.snps_indels_svs.genotypes.vcf.gz', 'info')" --
    build hg19 -l 5
[get_local_version] downloading the index file...
1  1115503  LDAF=0.0133;AC=28;SNPSOURCE=LOWCOV,EXOME;AA=T;AN=2184;VT=SNP;THETA=0.0012;ERATE
    =0.0003;RSQ=0.9950;AVGPOST=0.9999;AF=0.01;AMR_AF=0.01;AFR_AF=0.0041;EUR_AF=0.03
1  1115548  AVGPOST=0.9983;THETA=0.0004;SNPSOURCE=LOWCOV,EXOME;AA=G;AN=2184;RSQ=0.9326;LDAF
    =0.0106;VT=SNP;AC=22;ERATE=0.0006;AF=0.01;AMR_AF=0.02;EUR_AF=0.02
1  1118275  AC=300;AA=C;THETA=0.0004;SNPSOURCE=LOWCOV,EXOME;AN=2184;AVGPOST=0.9981;LDAF=0.1372;VT=
    SNP;ERATE=0.0008;RSQ=0.9950;AF=0.14;ASN_AF=0.05;AMR_AF=0.14;AFR_AF=0.38;EUR_AF=0.04
1  1120377  THETA=0.0009;SNPSOURCE=LOWCOV,EXOME;AA=T;AN=2184;RSQ=0.9796;AC=16;AVGPOST=0.9996;VT=
    SNP;LDAF=0.0072;ERATE=0.0003;AF=0.01;AMR_AF=0.01;EUR_AF=0.02
1  1120431  AC=347;THETA=0.0096;ERATE=0.0063;AVGPOST=0.9977;RSQ=0.9945;SNPSOURCE=LOWCOV,EXOME;AN
    =2184;VT=SNP;LDAF=0.1592;AA=A;AF=0.16;ASN_AF=0.16;AMR_AF=0.06;AFR_AF=0.40;EUR_AF=0.05
```

TRACK

```
$ vtools output refT chr pos ref alt "track('CEU.exon.2010_03.sites.vcf.gz', 'info.AA')" -l5
1  1105366  T  C  T
1  1110240  T  A  T
1  3537996  T  C  C
1  6447088  T  C  T
1  6447275  T  C  T

$ vtools select variant "track('CEU.exon.2010_03.sites.vcf.gz', 'info.DP') > 1000" --output chr
      pos ref alt DP -l5
1  1105366  T  C  3251
1  1105411  G  A  2676
1  1108138  C  T  2253
1  1110240  T  A  7275
1  1110294  G  A  7639

$ vtools liftover hg19
$ vtools output variant chr pos "track('http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release
      /20110521/ALL.chr1.phase1_release_v3.20101123.snps_indels_svs.genotypes.vcf.gz', 'info')" --
      build hg19 -l 5
[get_local_version] downloading the index file...
1  1115503  LDAF=0.0133;AC=28;SNPSOURCE=LOWCOV,EXOME;AA=T;AN=2184;VT=SNP;THETA=0.0012;ERATE
      =0.0003;RSQ=0.9950;AVGPOST=0.9999;AF=0.01;AMR_AF=0.01;AFR_AF=0.0041;EUR_AF=0.03
1  1115548  AVGPOST=0.9983;THETA=0.0004;SNPSOURCE=LOWCOV,EXOME;AA=G;AN=2184;RSQ=0.9326;LDAF
      =0.0106;VT=SNP;AC=22;ERATE=0.0006;AF=0.01;AMR_AF=0.02;EUR_AF=0.02
1  1118275  AC=300;AA=C;THETA=0.0004;SNPSOURCE=LOWCOV,EXOME;AN=2184;AVGPOST=0.9981;LDAF=0.1372;VT=
      SNP;ERATE=0.0008;RSQ=0.9950;AF=0.14;ASN_AF=0.05;AMR_AF=0.14;AFR_AF=0.38;EUR_AF=0.04
1  1120377  THETA=0.0009;SNPSOURCE=LOWCOV,EXOME;AA=T;AN=2184;RSQ=0.9796;AC=16;AVGPOST=0.9996;VT=
      SNP;LDAF=0.0072;ERATE=0.0003;AF=0.01;AMR_AF=0.01;EUR_AF=0.02
1  1120431  AC=347;THETA=0.0096;ERATE=0.0063;AVGPOST=0.9977;RSQ=0.9945;SNPSOURCE=LOWCOV,EXOME;AN
      =2184;VT=SNP;LDAF=0.1592;AA=A;AF=0.16;ASN_AF=0.16;AMR_AF=0.06;AFR_AF=0.40;EUR_AF=0.05
```

`BigWig`, `BigBed`, and local and online indexed `VCF` and `BAM` files are supported.

SNAPSHOT

A *snapshot* contains a copy of all databases of a project. Local snapshots are used to save, restore, and transfer projects. Online snapshots are used extensively in documentation.

```
$ vtools admin --save_snapshot con1 'first snapshot for project concept'
INFO: Snapshot con1 has been saved

$ vtools show snapshots
con1                    first snapshot for project concept  (358.0KB, created:
                        Oct03 01:01:07)
vt_qc                   snapshot for QC tutorial, exome data of 1000 genomes
                        project with simulated GD and GQ scores (2.0GB, online
                        snapshot)
vt_ExomeAssociation     Data with ~26k variants from chr1 and 2, ~3k samples,
                        3 phenotypes, ready for association testing. (446.0MB,
                        online snapshot)
vt_quickStartGuide      A simple project with variants from the CEU and JPT
                        pilot data of the 1000 genome project (148.0KB, online
                        snapshot)
vt_illuminaTestData     Test data with 1M paired reads (49.0MB, online
                        snapshot)
vt_simple               A simple project with variants imported from three vcf
                        files (41.0KB, online snapshot)
vt_testData             An empty project with some test datasets (68.0KB,
                        online snapshot)

$ vtools admin --load_snapshot vt_testData
Downloading snapshot vt_testData.tar.gz from online
INFO: Snapshot vt_testData has been loaded
```

SAMPLE, GENOTYPE AND GENOTYPE INFO FIELDS

A *sample* contains a list of variants, their number (0 for homozygote reference, 1 for heterozygote and 2 for homozygote alternative), and additional info (e.g. depth of coverage) detected from a physical sample.

```
$ vtools import CEU.vcf.gz --build hg18 --var_info DP --geno_info DP_geno
INFO: Importing variants from CEU.vcf.gz (1/1)
CEU.vcf.gz: 100% [============================] 300 12.5K/s in 00:00:00
INFO: 0 new variants 288 SNVs from 300 lines are imported.
Importing genotypes: 100% [=================] 18,000 9.0K/s in 00:00:02
Copying samples: 100% [=======================] 65 64.9/s in 00:00:01

$ vtools show genotypes -l 5
sample_name  filename    num_genotypes  sample_genotype_fields
NA06985      CEU.vcf.gz  287            GT,DP_geno
NA06986      CEU.vcf.gz  287            GT,DP_geno
NA06994      CEU.vcf.gz  287            GT,DP_geno
NA07000      CEU.vcf.gz  287            GT,DP_geno
NA07037      CEU.vcf.gz  287            GT,DP_geno
(55 records omitted)
```

PHENOTYPE

Phenotypes are arbitrary properties of samples.

```
$ head -8 phenotype.txt
sample_name     aff     sex     BMI
NA06985 2       F       19.64
NA06986 1       M       None
NA06994 1       F       19.49
NA07000 2       F       21.52
NA07037 2       F       23.05
NA07051 1       F       21.01
NA07346 1       F       18.93

$ vtools phenotype --from_file phenotype.txt
INFO: Adding phenotype aff
INFO: Adding phenotype sex
INFO: Adding phenotype BMI
INFO: 3 field (3 new, 0 existing) phenotypes of 60 samples are updated.

$ vtools show phenotypes -l 8
sample_name  aff  sex  BMI
NA06985      2    F    19.64
NA06986      1    M    None
NA06994      1    F    19.49
NA07000      2    F    21.52
NA07037      2    F    23.05
NA07051      1    F    21.01
NA07346      1    F    18.93
NA07347      2    M    19.2
(50 records omitted)
```

Outline

A real-world example
- Import data
- Phenotype and sample statistics
- Annotation
- Select and filter variants
- Output variants and their summary statistics

DATA

Whole genome sequence of 44 independent probands from NARAC (North American Rheumatoid Arthritis Consortium) and MADGC (Multiple Autoimmune Disease Genetics Consortium) families. The data are prepared by BGI using hg18 reference genome, and are provided as

- 44 VCF files (one file for each sample) with on average 3.7M single nucleotide variants (SNVs).
- 44 text files with on average of 0.82M indels.

In order to compare variants of these patients with variants from healthy individuals, we (tentatively) include

- 200 VCF files with on average 0.07M variants (SNVs and INDELs), from exome sequencing of 2000 individuals of Danish nationality. The variants are called using hg19 reference genome.

DATA

Whole genome sequence of 44 independent probands from NARAC (North American Rheumatoid Arthritis Consortium) and MADGC (Multiple Autoimmune Disease Genetics Consortium) families. The data are prepared by BGI using hg18 reference genome, and are provided as

- 44 VCF files (one file for each sample) with average 3.7M si
- 44 text

5 cases and 5 controls are used for this presentation

In order to compare variants of these patients with variants from healthy individuals, we (tentatively) include

- 200 VCF files with on average 0.07M variants (SNVs and INDELs), from exome sequencing of 2000 individuals of Danish nationality. The variants are called using hg19 reference genome.

IMPORT SNV DATA

```
$ vtools init RA
INFO: variant tools 2.0.1 : Copyright (c) 2011 - 2012 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 RA

$ vtools import MG*.vcf --build hg18
INFO: Importing variants from MG3037-121.snp.txt.vcf (1/5)
MG3037-121.snp.txt.vcf: 100% [==============================] 4,075,605 24.4K/s in 00:02:46
INFO: 3,696,791 new variants (3,696,791 SNVs) from 3,694,582 lines are imported.
INFO: Importing variants from MG3046-303.snp.txt.vcf (2/5)
MG3046-303.snp.txt.vcf: 100% [==============================] 3,922,895 35.9K/s in 00:01:49
INFO: 1,274,983 new variants (1,274,983 SNVs) from 3,433,052 lines are imported.
INFO: Importing variants from MG3087-200.snp.txt.vcf (3/5)
MG3087-200.snp.txt.vcf: 100% [==============================] 3,819,332 28.8K/s in 00:02:12
INFO: 809,942 new variants (809,942 SNVs) from 3,444,085 lines are imported.
INFO: Importing variants from MG3140-300.snp.txt.vcf (4/5)
MG3140-300.snp.txt.vcf: 100% [==============================] 4,084,525 25.0K/s in 00:02:43
INFO: 616,326 new variants (616,326 SNVs) from 3,669,294 lines are imported.
INFO: Importing variants from MG3184-301.snp.txt.vcf (5/5)
MG3184-301.snp.txt.vcf: 100% [==============================] 4,127,946 26.1K/s in 00:02:38
INFO: 491,056 new variants (491,056 SNVs) from 3,726,488 lines are imported.
Importing genotypes: 100% [==============================] 28,913,578 127.8K/s in 00:03:46
Copying samples: 100% [============================================] 9 0.6/s in 00:00:15
INFO: 6,889,098 new variants (6,889,098 SNVs) from 17,967,501 lines (5 samples) are imported.
```

FORMAT OF INDEL DATA

```
$ head -30 MG3037-121.pileup.indel
chr10   51372   D1      A       *       hete    25      9       33
chr10   57161   D2      AG      *       hete    33      3       21
chr10   57414   I1      G       *       hete    21      2       20
chr10   62170   I1      T       *       hete    36      10      30
chr10   62899   I3      AAA     *       hete    38      9       30
chr10   66586   D1      A       *       hete    22      5       31
chr10   85429   I1      A       *       hete    53      10      26
chr10   86294   I4      CAGC    *       hete    46      4       35
chr10   87126   I24     TGCATTTACGTGATCTTGGCTCAC        *       hete    55      8       53
chr10   88705   I1      A       *       hete    53      10      55
chr10   89448   I3      AGG     *       hete    29      5       39
chr10   93591   D1      G       *       hete    40      6       33
chr10   93753   D1      T       *       hete    29      19      79
chr10   94117   I3      CAA     *       hete    27      38      106
chr10   97572   D1      T       *       hete    40      8       51
chr10   97938   D1      T       *       hete    32      29      65
chr10   98719   I1      T       *       hete    47      10      38
chr10   100799  I1      G       *       hete    47      10      36
chr10   101382  D1      G       *       hete    53      13      36
chr10   102510  D1      C       *       hete    52      8       38
chr10   103093  D1      T       *       hete    53      23      41
chr10   106216  D4      TTTT    *       hete    53      15      35
chr10   106509  I13     TGGCCAGGCACAG   *       hete    49      3       29
chr10   107368  D1      T       *       hete    51      5       27
chr10   108915  I1      G       *       hete    54      12      31
chr10   110337  D2      GG      *       hete    55      2       18
chr10   110565  D1      A       *       hete    45      4       15
```

FORMAT OF INDEL DATA

```
$ head -30 MG3037-121.pileup.indel
chr10   51372   D1      A       *       hete
chr10   57161   D2      AG      *       hete
chr10   57414   I1      G       *       hete
chr10   62170   I1      T       *       hete
chr10   62899   I3      AAA     *       hete
chr10   66586   D1      A       *       hete
chr10   85429   I1      A       *       hete    53      10      26
chr10   86294   I4      CAGC    *       hete    46      4       35
chr10   87126   I24     TGCATTTACGTGATCTTGGCTCAC        *       hete    55      8       53
chr10   88705   I1      A       *       hete    53      10      55
chr10   89448   I3      AGG     *       hete    29      5       39
chr10   93591   D1      G       *       hete    40      6       33
chr10   93753   D1      T       *       hete    29      19      79
chr10   94117   I3      CAA     *       hete    27      38      106
chr10   97572   D1      T       *       hete    40      8       51
chr10   97938   D1      T       *       hete    32      29      65
chr10   98719   I1      T       *       hete    47      10      38
chr10   100799  I1      G       *       hete    47      10      36
chr10   101382  D1      G       *       hete    53      13      36
chr10   102510  D1      C       *       hete    52      8       38
chr10   103093  D1      T       *       hete    53      23      41
chr10   106216  D4      TTTT    *       hete    53      15      35
chr10   106509  I13     TGGCCAGGCACAG   *       hete    49      3       29
chr10   107368  D1      T       *       hete    51      5       27
chr10   108915  I1      G       *       hete    54      12      31
chr10   110337  D2      GG      *       hete    55      2       18
chr10   110565  D1      A       *       hete    45      4       15
```

Variant tools provides an input format specification system that allows processing data in arbitrary delimiter separated formats.

INPUT FORMAT SPECIFICATION

```
$ vtools show formats -v0
CASAVA18_snps
CASAVA18_indels
plink
rsname
ANNOVAR
pileup_indel
ANNOVAR_exonic_variant_function
ANNOVAR_variant_function
twoalleles
map
polyphen2
basic
vcf
CGA
csv
tped

$ vtools show format pileup_indel
Input format for samtools pileup indel caller. This format imports chr, pos,
ref, alt and genotype.

Columns:
  None defined, cannot export to this format

variant:
  chr                   Chromosome name
  pos                   Start position of the indel event.
  ref                   reference allele, '-' for insertion
  alt                   alternative allele, '-' for deletion

Genotype:
  GT                    type of indel (homozygote or heterozygote)
```

IMPORT INDEL DATA

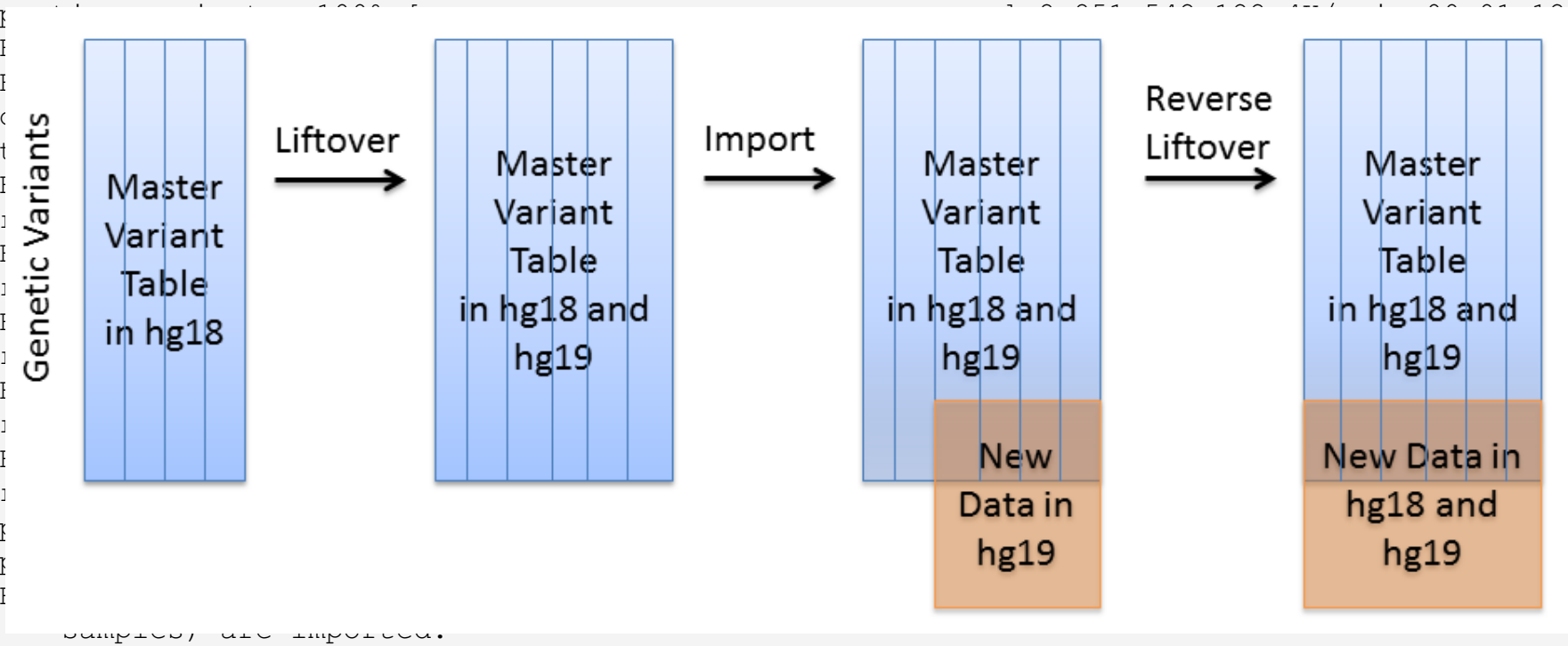
```
$ vtools import --format pileup_indel MG*.indel
INFO: Opening project RA.proj
INFO: Using primary reference genome hg18 of the project.
Getting existing variants: 100.0% [============================>] 6,901,157 162.2K/s in 00:00:42
INFO: Additional genotype fields: genotype
INFO: Importing genotype from ../data/indel/MG1000-240.pileup.indel (1/5)
MG1000-240.pileup.indel: 100.0% [=================================>] 712,688 9.2K/s in 00:01:17
INFO: 847,949 new variants from 847,949 records are imported, with 0 SNVs, 348,266 insertions,
      499,683 deletions, and 0 complex variants.
INFO: Importing genotype from ../data/indel/MG1004-200.pileup.indel (2/5)
MG1004-200.pileup.indel: 100.0% [=================================>] 706,906 10.8K/s in 00:01:05
INFO: 416,517 new variants from 836,944 records are imported, with 0 SNVs, 161,927 insertions,
      254,590 deletions, and 0 complex variants.
INFO: Importing genotype from ../data/indel/MG1022-121.pileup.indel (3/5)
MG1022-121.pileup.indel: 100.0% [=================================>] 758,880 11.8K/s in 00:01:04
INFO: 314,641 new variants from 857,899 records are imported, with 0 SNVs, 117,506 insertions,
      197,135 deletions, and 0 complex variants.
INFO: Importing genotype from ../data/indel/MG1057-203.pileup.indel (4/5)
MG1057-203.pileup.indel: 100.0% [=================================>] 676,350 11.2K/s in 00:01:00
INFO: 207,950 new variants from 798,406 records are imported, with 0 SNVs, 79,766 insertions,
      128,184 deletions, and 0 complex variants.
INFO: Importing genotype from ../data/indel/MG1078-200.pileup.indel (5/5)
MG1078-200.pileup.indel: 100.0% [=================================>] 709,018 11.7K/s in 00:01:00
INFO: 191,135 new variants from 842,633 records are imported, with 0 SNVs, 72,772 insertions,
      118,363 deletions, and 0 complex variants.
INFO: 1,978,192 new variants from 4,183,831 records in 5 files are imported, with 0 SNVs, 780,237
      insertions, 1,197,955 deletions, and 0 complex variants.
INFO: Creating index on master variant table. This might take quite a while.
```

IMPORTING DATA IN ANOTHER REFERENCE GENOME

```
$ vtools import varSRR02896*.vcf --build hg19
WARNING: The new files uses a different reference genome (hg19) from the primary reference genome
     (hg18) of the project.
INFO: Adding an alternative reference genome (hg19) to the project.
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exporting variants: 100% [==============================] 8,851,542 122.4K/s in 00:01:12
INFO: Running UCSC liftOver tool
INFO: 11023 records failed to map.
Updating table variant: 100% [============================] 8,858,166 29.7K/s in 00:04:58
Getting existing variants: 100% [=========================] 8,851,542 144.5K/s in 00:01:01
INFO: Importing variants from varSRR028961.filtered.vcf (1/5)
varSRR028961.filtered.vcf: 100% [============================] 58,294 14.1K/s in 00:00:04
INFO: Importing variants from varSRR028962.filtered.vcf (2/5)
varSRR028962.filtered.vcf: 100% [============================] 58,767 17.6K/s in 00:00:03
INFO: Importing variants from varSRR028963.filtered.vcf (3/5)
varSRR028963.filtered.vcf: 100% [============================] 48,211 8.4K/s in 00:00:05
INFO: Importing variants from varSRR028964.filtered.vcf (4/5)
varSRR028964.filtered.vcf: 100% [============================] 52,495 6.8K/s in 00:00:07
INFO: Importing variants from varSRR028965.filtered.vcf (5/5)
varSRR028965.filtered.vcf: 100% [============================] 58,753 9.7K/s in 00:00:06
Importing genotypes: 100% [================================] 427,136 42.6K/s in 00:00:10
Copying samples: 100% [=============================================] 9 9.0/s in 00:00:01
INFO: 54,327 new variants (45,903 SNVs, 3,938 insertions, 4,486 deletions) from 268,566 lines (5
     samples) are imported.
INFO: Analyzing project
INFO: Mapping new variants at 54327 loci from hg19 to hg18 reference genome
INFO: Downloading liftOver chain file from UCSC
INFO: Running UCSC liftOver tool
Updating coordinates: 100% [================================] 54,327 27.8K/s in 00:00:01
INFO: Coordinates of 54145 (54327 total, 182 failed to map) new variants are updated.
```

IMPORTING DATA IN ANOTHER REFERENCE GENOME

```
$ vtools import varSRR02896*.vcf --build hg19
WARNING: The new files uses a different reference genome (hg19) from the primary reference genome
      (hg18) of the project.
INFO: Adding an alternative reference genome (hg19) to the project.
INFO: Downloading liftOver chain file from UCSC
INFO: Exporting variants in BED format
Exp
INF
INF
Upd
Get
INF
var
INF
var
INF
var
INF
var
INF
var
Imp
Cop
INF                                                                                    es (5
      samples) are imported.
INFO: Analyzing project
INFO: Mapping new variants at 54327 loci from hg19 to hg18 reference genome
INFO: Downloading liftOver chain file from UCSC
INFO: Running UCSC liftOver tool
Updating coordinates: 100% [==================================] 54,327 27.8K/s in 00:00:01
INFO: Coordinates of 54145 (54327 total, 182 failed to map) new variants are updated.
```



Outline

A real-world example
- Import data
- Phenotype and sample statistics
- Annotation
- Select and filter variants
- Output variants and their summary statistics

Have a look at the project

```
$ vtools show project
Project name:                 RA
Created on:                   Wed Oct  8 12:20:24 2013
Primary reference genome:     hg18
Secondary reference genome:   hg19
Runtime options:              verbosity=1
Variant tables:               variant
Annotation databases:

$ vtools show tables
table      #variants     date message
variant    8,905,869    Oct03 Master variant table

$ vtools show genotypes
sample_name                filename                    num_genotypes  sample_genotype_fields
SAMP1                      MG3037-121.snp.txt.vcf      3696791        GT
SAMP1                      MG3046-303.snp.txt.vcf      3434868        GT
SAMP1                      MG3087-200.snp.txt.vcf      3446189        GT
SAMP1                      MG3140-300.snp.txt.vcf      3671426        GT
SAMP1                      MG3184-301.snp.txt.vcf      3728739        GT
                           MG3037-121.pileup.indel     843853         GT
                           MG3046-303.pileup.indel     835560         GT
                           MG3087-200.pileup.indel     834798         GT
                           MG3140-300.pileup.indel     818325         GT
                           MG3184-301.pileup.indel     833162         GT
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf   55818          GT
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf   56655          GT
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf   47887          GT
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf   51753          GT
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf   56449          GT
```

RENAME SAMPLES

```
$ vtools admin --rename_samples "filename like 'MG3037%'" MG3037
INFO: 2 samples with names , SAMP1 are renamed to MG3037
$ vtools admin --rename_samples "filename like 'MG3046%'" MG3046
INFO: 2 samples with names , SAMP1 are renamed to MG3046
$ vtools admin --rename_samples "filename like 'MG3087%'" MG3087
INFO: 2 samples with names , SAMP1 are renamed to MG3087
$ vtools admin --rename_samples "filename like 'MG3140%'" MG3140
INFO: 2 samples with names , SAMP1 are renamed to MG3140
$ vtools admin --rename_samples "filename like 'MG3184%'" MG3184
INFO: 2 samples with names , SAMP1 are renamed to MG3184

$ vtools show samples
sample_name                filename
MG3037                     MG3037-121.snp.txt.vcf
MG3037                     MG3037-121.pileup.indel
MG3046                     MG3046-303.snp.txt.vcf
MG3046                     MG3046-303.pileup.indel
MG3087                     MG3087-200.snp.txt.vcf
MG3087                     MG3087-200.pileup.indel
MG3140                     MG3140-300.snp.txt.vcf
MG3140                     MG3140-300.pileup.indel
MG3184                     MG3184-301.snp.txt.vcf
MG3184                     MG3184-301.pileup.indel
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf
```

MERGE SAMPLES

```
$ vtools admin --merge_samples
INFO: 10 samples that share identical names will be merged to 5 samples
Merging samples: 100% [============================================] 10 0.5/s in 00:00:21
Removing obsolete tables: 100% [=======================================] 10 8.6/s in 00:00:01

$ vtools show samples
sample_name               filename
MG3037                    MG3037-1...21.snp.txt.vcf
MG3046                    MG3046-3...03.snp.txt.vcf
MG3087                    MG3087-2...00.snp.txt.vcf
MG3140                    MG3140-3...00.snp.txt.vcf
MG3184                    MG3184-3...01.snp.txt.vcf
SRR028961.aln.sorted.bam  varSRR028961.filtered.vcf
SRR028962.aln.sorted.bam  varSRR028962.filtered.vcf
SRR028963.aln.sorted.bam  varSRR028963.filtered.vcf
SRR028964.aln.sorted.bam  varSRR028964.filtered.vcf
SRR028965.aln.sorted.bam  varSRR028965.filtered.vcf

$ vtools admin --save_snapshot imported_data 'Imported data, SNVs and INDELs from samples are
     merged'
INFO: Snapshot imported_data has been saved
```

MERGE SAMPLES

```
$ vtools admin --merge_samples
INFO: 10 samples that share identical names wi
Merging samples: 100% [=======================
Removing obsolete tables: 100% [==============

$ vtools show samples
sample_name                filename
MG3037                     MG3037-1...21.snp.txt.vcf
MG3046                     MG3046-3...03.snp.txt.vcf
MG3087                     MG3087-2...00.snp.txt.vcf
MG3140                     MG3140-3...00.snp.txt.vcf
MG3184                     MG3184-3...01.snp.txt.vcf
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf

$ vtools admin --save_snapshot imported_data 'Imported data, SNVs and INDELs from samples are
     merged'
INFO: Snapshot imported_data has been saved
```

It is a good practice to save snapshots of your project after the completion of major tasks, or before experimental processing steps.

COUNTING NUMBER OF VARIANTS IN SAMPLES

Command `vtools update` adds or updates variant info fields. This example uses special functions `#(alt)`, `#(hom)` and `#(het)` to count the number of variants, homozygotes and heterozygotes for each variant in the sample.

```
$ vtools update variant --from_stat 'num=#(alt)' 'hom=#(hom)' 'het=#(het)'
Counting variants: 100% [====================================] 15 0.2/s in 00:01:100
INFO: Adding variant info field num
INFO: Adding variant info field hom
INFO: Adding variant info field het
Updating variant: 100% [================================] 8,904,873 45.5K/s in 00:03:15
INFO: 8904873 records are updated

$ vtools output variant chr pos ref alt num hom het -l 10
1  583   G  A  5  0  5
1  4770  A  G  5  0  5
1  5931  T  C  4  1  2
1  5966  T  G  6  1  4
1  6120  G  C  2  0  2
1  6241  T  C  4  1  2
1  6360  A  G  2  0  2
1  7401  C  A  1  0  1
1  9131  C  T  2  0  2
1  9992  C  T  3  0  3
```

COUNT GENOTYPES IN CASES

```
$ vtools show samples
sample_name                filename
MG3037                     MG3037-1...21.snp.txt.vcf
MG3046                     MG3046-3...03.snp.txt.vcf
MG3087                     MG3087-2...00.snp.txt.vcf
MG3140                     MG3140-3...00.snp.txt.vcf
MG3184                     MG3184-3...01.snp.txt.vcf
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf

$ vtools update variant --from_stat 'case_num=#(alt)' --samples 'sample_name like "%MG%"'
INFO: 5 samples are selected
Counting variants: 100% [======================] 10 0.1/s in 00:01:24
INFO: Adding variant info field case_num
Updating variant: 100% [===============] 8,851,542 48.7K/s in 00:03:01
INFO: 8851542 records are updated

$ vtools output variant chr pos ref alt num case_num  -l 5
1  583   G  A  5  5
1  4770  A  G  5  5
1  5931  T  C  4  4
1  5966  T  G  6  6
1  6120  G  C  2  2
```

COUNT GENOTYPES IN CASES

```
$ vtools show samples
sample_name                filename
MG3037                     MG3037-1...21.snp.tx
MG3046                     MG3046-3...03.snp.tx
MG3087                     MG3087-2...00.snp.tx
MG3140                     MG3140-3...00.snp.txt.vcf
MG3184                     MG3184-3...01.snp.txt.vcf
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf

$ vtools update variant --from_stat 'case_num=#(alt)' --samples 'sample_name like "%MG%"'
INFO: 5 samples are selected
Counting variants: 100% [=======================] 10 0.1/s in 00:01:24
INFO: Adding variant info field case_num
Updating variant: 100% [================] 8,851,542 48.7K/s in 00:03:01
INFO: 8851542 records are updated

$ vtools output variant chr pos ref alt num case_num  -l 5
1  583   G  A  5  5
1  4770  A  G  5  5
1  5931  T  C  4  4
1  5966  T  G  6  6
1  6120  G  C  2  2
```

Samples can be selected by sample names, file names, and arbitrary phenotypes.

ADD PHENOTYPE

```
$ vtools show samples
sample_name                filename
MG3037                     MG3037-1...21.snp.txt.vcf
MG3046                     MG3046-3...03.snp.txt.vcf
MG3087                     MG3087-2...00.snp.txt.vcf
MG3140                     MG3140-3...00.snp.txt.vcf
MG3184                     MG3184-3...01.snp.txt.vcf
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf

$ vtools phenotype --set aff=2 --samples "sample_name like '%MG%'"
INFO: Adding phenotype aff
INFO: 10 values of 1 phenotypes (1 new, 0 existing) of 10 samples are updated.

$ vtools phenotype --set aff=1 --samples 'aff is NULL'
INFO: 5 values of 1 phenotypes (0 new, 1 existing) of 5 samples are updated.
```

Allele count by affection status

```
$ vtools show samples
sample_name                filename                    aff
MG3037                     MG3037-1...21.snp.txt.vcf   2
MG3046                     MG3046-3...03.snp.txt.vcf   2
MG3087                     MG3087-2...00.snp.txt.vcf   2
MG3140                     MG3140-3...00.snp.txt.vcf   2
MG3184                     MG3184-3...01.snp.txt.vcf   2
SRR028961.aln.sorted.bam   varSRR028961.filtered.vcf   1
SRR028962.aln.sorted.bam   varSRR028962.filtered.vcf   1
SRR028963.aln.sorted.bam   varSRR028963.filtered.vcf   1
SRR028964.aln.sorted.bam   varSRR028964.filtered.vcf   1
SRR028965.aln.sorted.bam   varSRR028965.filtered.vcf   1

$ vtools update variant --from_stat 'ctrl_num=#(alt)' --samples 'aff=1'
INFO: 5 samples are selected
Counting variants: 100% [=========================] 5 4.6/s in 00:00:01
INFO: Adding variant info field ctrl_num
Updating variant: 100% [==================] 171,861 22.5K/s in 00:00:07
INFO: 171861 records are updated

$ vtools output variant chr pos ref alt num case_num ctrl_num  -l 5
1  583   G  A  5  5  0
1  4770  A  G  5  5  0
1  5931  T  C  4  4  0
1  5966  T  G  6  6  0
1  6120  G  C  2  2  0
```

Outline

A real-world example
- Import data
- Phenotype and sample statistics
- Annotation
- Select and filter variants
- Output variants and their summary statistics

dbSNP

Use command `vtools use` to link to annotation databases. Databases without version name always refer to the latest version. If you need to use a particular version of database, use databases such as `dbSNP-hg18_130`.

```
$ vtools use dbSNP
INFO: Downloading annotation database from annoDB/dbSNP.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/dbSNP-
      hg19_138.DB.gz
INFO: Using annotation DB dbSNP in project RA.
INFO: dbSNP version 138, created using vcf file downloaded from NCBI

$ vtools output variant chr pos ref alt dbSNP.name -l10
1  583   G  A  rs58108140
1  4770  A  G  rs79585140
1  5931  T  C  rs372319358
1  5966  T  G  rs200358166
1  6120  G  C  rs78588380
1  6241  T  C  rs148220436
1  6360  A  G  rs150723783
1  7401  C  A  rs200046632
1  9131  C  T  .
1  9992  C  T  rs202081272
```

refGene and refGene_exon

Several gene databases are available based on different prediction criteria.

```
$ vtools use refGene
INFO: Downloading annotation database from annoDB/refGene.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/refGene-
      hg19_20130904.DB.gz
INFO: Using annotation DB refGene in project RA.
INFO: Known human protein-coding and non-protein-coding genes taken from the NCBI RNA reference
      sequences collection (RefSeq).

$ vtools use refGene_exon
INFO: Downloading annotation database from annoDB/refGene_exon.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/
      refGene_exon-hg19_20130904.DB.gz
INFO: Using annotation DB refGene_exon in project RA.
INFO: RefGene specifies known human protein-coding and non-protein-coding genes taken from the
      NCBI RNA reference sequences collection (RefSeq). This database contains all exome regions
      of the refSeq genes.

$ vtools output variant chr pos ref alt refGene.name refGene.name2 refGene_exon.name2 -l 10
1  583   G  A  .          .       .
1  4770  A  G  NR_024540  WASH7P  .
1  5931  T  C  NR_024540  WASH7P  .
1  5966  T  G  NR_024540  WASH7P  .
1  6120  G  C  NR_024540  WASH7P  .
1  6241  T  C  NR_024540  WASH7P  .
1  6360  A  G  NR_024540  WASH7P  .
1  7401  C  A  NR_024540  WASH7P  .
1  9131  C  T  NR_024540  WASH7P  .
1  9992  C  T  NR_024540  WASH7P  .
```

dbNSFP

dbNSFP provides a comprehensive set of annotations, most notably function-prediction scores, for non-symnonymous SNPs in CCDS genes.

```
$ vtools use dbNSFP
INFO: dbNSFP version 2.1, maintained by Xiaoming Liu from UTSPH. Please cite
"Liu X, Jian X, and Boerwinkle E. 2011. dbNSFP: a lightweight database of human
non-synonymous SNPs and their functional predictions. Human Mutation. 32:894-899" and
"Liu X, Jian X, and Boerwinkle E. 2013. dbNSFP v2.0: A Database of Human Nonsynonymous
SNVs and Their Functional Predictions and Annotations. Human Mutation. 34:E2393-E2402."
if you find this database useful.

$ vtools output variant chr pos ref alt SIFT_score PolyPhen2_HDIV_score -l 10
1  583   G  A  .  .
1  4770  A  G  .  .
1  5931  T  C  .  .
1  5966  T  G  .  .
1  6120  G  C  .  .
1  6241  T  C  .  .
1  6360  A  G  .  .
1  7401  C  A  .  .
1  9131  C  T  .  .
1  9992  C  T  .  .
```

Outline

A real-world example

- Import data
- Phenotype and sample statistics
- Annotation
- Select and filter variants
- Output variants and their summary statistics

IDENTIFY VARIANTS IN DBNSFP

Variants that are not covered by a database will conceptually have `NULL` values for all fields. Condition "`dbNSFP.chr IS NOT NULL`" can therefore be used to select all variants that are in dbNSFP.

```
$ vtools select variant 'dbNSFP.chr IS NOT NULL' -t NS 'Non-symnonymous SNPs'
Running: 20,519 234.4/s in 00:01:27
INFO: 26963 variants selected.

$ vtools output NS chr pos ref alt SIFT_score Polyphen2_HDIV_score -l 10
1  878522   T  C  1.0   0.0
1  879101   G  A  0.07  0.999;0.999;0.99
1  901458   A  G  0.0   0.518
1  904196   C  G  0.46  0.0
1  904715   G  C  1.0   0.0
1  904739   T  C  0.43  0.001
1  906412   A  G  0.37  .
1  939471   G  A  0.0   0.01
1  1148494  A  G  .     .
1  1548655  T  C  0.31  0.013;0.0;0.0

$ vtools show tables
table      #variants     date message
NS            26,963    Oct03 Non-symnonymous SNPs
variant    8,905,869    Oct03 Master variant table
```

SELECT VARIANTS

```
$ vtools select NS 'SIFT_score < 0.05' -t NS_damaging 'Non-symnonymous SNPs with SIFT score <
      0.05'
Running: 93 177.9/s in 00:00:00
INFO: 5619 variants selected.

$ vtools select NS 'SIFT_score < 0.05 OR Polyphen2_HDIV_score_max > 0.95' -t NS_or
Running: 105 195.5/s in 00:00:00
INFO: 7800 variants selected.

$ vtools compare NS_or NS_damaging --difference NS_pp2 'Variants in table NS_or but not in
      NS_damaging'
INFO: Reading 7,800 variants in NS_or...
INFO: Reading 5,619 variants in NS_damaging...
Writing to NS_pp2: 100% [============================] 2,181 78.2K/s in 00:00:00
2181

$ vtools output NS_pp2 chr pos ref alt SIFT_score PolyPhen2_HDIV_score LRT_pred -l 8
1  879101    G  A  0.07  0.999;0.999;0.99                                        N
1  1640705   G  A  0.08  0.097;1.0;0.243;1.0;1.0;0.998;1.0;1.0;1.0;0.999;1.0  U
1  4672577   G  A  0.32  0.999                                                   N
1  6447088   T  C  0.29  1.0;1.0;1.0;1.0                                        N
1  6553693   C  T  .     .                                                       .
1  8932038   G  C  .     1.0                                                     N
1  8939791   A  G  0.13  0.984;0.971                                            N
1  11778965  G  A  0.05  0.998;0.999                                            D
```

SELECT VARIANTS

```
$ vtools select NS 'SIFT_score < 0.05' -t NS_d
     0.05'
Running: 93 177.9/s in 00:00:00
INFO: 5619 variants selected.

$ vtools select NS 'SIFT_score < 0.05 OR Polyphen2_HDIV_score_max > 0.95' -t NS_or
Running: 105 195.5/s in 00:00:00
INFO: 7800 variants selected.

$ vtools compare NS_or NS_damaging --difference NS_pp2 'Variants in table NS_or but not in
     NS_damaging'
INFO: Reading 7,800 variants in NS_or...
INFO: Reading 5,619 variants in NS_damaging...
Writing to NS_pp2: 100% [==========================] 2,181 78.2K/s in 00:00:00
2181

$ vtools output NS_pp2 chr pos ref alt SIFT_score PolyPhen2_HDIV_score LRT_pred -l 8
1  879101    G  A  0.07  0.999;0.999;0.99                                          N
1  1640705   G  A  0.08  0.097;1.0;0.243;1.0;1.0;0.998;1.0;1.0;1.0;0.999;1.0  U
1  4672577   G  A  0.32  0.999                                                     N
1  6447088   T  C  0.29  1.0;1.0;1.0;1.0                                          N
1  6553693   C  T  .     .                                                         .
1  8932038   G  C  .     1.0                                                       N
1  8939791   A  G  0.13  0.984;0.971                                              N
1  11778965  G  A  0.05  0.998;0.999                                              D
```

Descriptions to variant tables are optional, but highly recommended.

How do tables compare?

```
$ vtools compare NS_damaging NS_or
INFO: Reading approximately 5,619 variants in NS_damaging...
INFO: Reading approximately 7,800 variants in NS_or...
INFO: Number of variants in A but not B, B but not A, A and B, and A or B
0       2181    5619    7800

$ vtools compare variant NS_or --difference 'Not Damaging' 'Variants that are not in NS_or table'
INFO: Reading 8,905,869 variants in variant...
INFO: Reading 7,800 variants in NS_or...
Writing to Not Damaging: 100% [================] 8,898,069 174.6K/s in 00:00:500
8898069

$ vtools show tables
table                          #variants     date message
NS                                26,963    Oct09 Non-symnonymous SNPs
NS_damaging                        5,619    Oct09 Non-symnonymous SNPs with SIFT score < 0.05
NS_or                              7,800    Oct09
NS_pp2                             2,181    Oct09 Variants in table NS_or but n ot in NS_damaging
Not Damaging                   8,898,069    Oct09 Variants that are not in NS_or table
variant                        8,905,869    Oct09 Master variant table
```

VARIANT SELECTING USING OTHER FIELDS

In addition to annotation fields, variant info fields, built-in function, and extended functions such as `track` can also be used for variant selection.

```
$ vtools select NS 'case_num=5' 'ctrl_num=0' -t case_only 'NS SNPs exist only in cases'
Running: 29 1.0/s in 00:00:28
INFO: 1060 variants selected.

$ vtools select NS "ref_sequence(chr, pos-1) = 'C'" "ref_sequence(chr, pos+1) = 'G'" -t CpG 'SNPs
      in CpG sites'
Running: 52 291.1/s in 00:00:00
INFO: 3144 variants selected.

$ vtools output CpG chr pos ref alt 'ref_sequence(chr, pos-2, pos+2)' -l 5
1  904739   T  C  GCTGG
1  1877105  G  A  GCGGC
1  1878053  C  A  GCCGA
1  2134648  A  G  ACAGC
1  2423760  C  T  CCCGC

$ vtools update variant --set "hwe=HWE_exact(num, het, hom)"
INFO: Adding variant info field hwe

$ vtools select NS 'hwe < 0.05' --output chr pos ref alt num het hom hwe -l 5
1  878522   T  C  17  1  8  0.000243679501334
1  904739   T  C  10  0  5  0.00136396111628
1  906412   A  G  6   0  3  0.021645021645
1  1148494  A  G  8   0  4  0.00543900543901
1  1876879  A  G  9   1  4  0.0364459070341
```

VARIANT SELECTING USING OTHER FIELDS

In addition to annotation fields, variant info fields, built-in function, and extended functions such as `track` can also be us

Genotype counts in subgroups are frequently used to detect variants that, for example, exist only in offspring (De Novo), exist only in probands (case only), or exist only as homozygotes in probands (recessive).

```
$ vtools select NS 'case_num=5' 'ctrl_num=0' -
Running: 29 1.0/s in 00:00:28
INFO: 1060 variants selected.

$ vtools select NS "ref_sequence(chr, pos-1) =
      in CpG sites'
Running: 52 291.1/s in 00:00:00
INFO: 3144 variants selected.

$ vtools output CpG chr pos ref alt 'ref_sequence(chr, pos-2, pos+2)' -l 5
1  904739   T  C  GCTGG
1  1877105  G  A  GCGGC
1  1878053  C  A  GCCGA
1  2134648  A  G  ACAGC
1  2423760  C  T  CCCGC

$ vtools update variant --set "hwe=HWE_exact(num, het, hom)"
INFO: Adding variant info field hwe

$ vtools select NS 'hwe < 0.05' --output chr pos ref alt num het hom hwe -l 5
1  878522   T  C  17  1  8  0.000243679501334
1  904739   T  C  10  0  5  0.00136396111628
1  906412   A  G  6   0  3  0.021645021645
1  1148494  A  G  8   0  4  0.00543900543901
1  1876879  A  G  9   1  4  0.0364459070341
```

WHAT PATHWAYS THESE VARIANTS BELONG?

```
$ vtools use ccdsGene
INFO: Downloading annotation database from annoDB/ccdsGene.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/ccdsGene
      -hg19_20130904.DB.gz
INFO: Using annotation DB ccdsGene in project RA.
INFO: High-confidence human gene annotations from the Consensus Coding Sequence (CCDS) project.

$ vtools use keggPathway --linked_by ccdsGene.name
INFO: Downloading annotation database from annoDB/keggPathway.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/
      keggPathway-20110823.DB.gz
INFO: Using annotation DB keggPathway in project RA.
INFO: kegg pathway for CCDS genes
INFO: 6821 out of 27731 ccdsgene.name are annotated through annotation database keggPathway
WARNING: 128 out of 6949 values in annotation database keggPathway are not linked to the project.

$ vtools output NS chr pos ccdsGene.name KgID KgDesc -l 10
1  878522   CCDS3.1      .         .
1  879101   CCDS3.1      .         .
1  901458   .            .         .
1  904196   .            .         .
1  904715   .            .         .
1  904739   .            .         .
1  906412   .            .         .
1  939471   CCDS6.1      hsa04622  RIG-I-like receptor signaling pathway
1  1148494  CCDS12.1     .         .
1  1548655  CCDS41224.2  .         .
```

WHAT PATHWAYS THESE VARIANTS BELONG?

```
$ vtools use ccdsGene
INFO: Downloading annotation database from ann
INFO: Downloading annotation database from htt                                ene
      -hg19_20130904.DB.gz
INFO: Using annotation DB ccdsGene in project
INFO: High-confidence human gene annotations f

$ vtools use keggPathway --linked_by ccdsGene.name
INFO: Downloading annotation database from annoDB/keggPathway.ann
INFO: Downloading annotation database from http://vtools.houstonbioinformatics.org/annoDB/
      keggPathway-20110823.DB.gz
INFO: Using annotation DB keggPathway in project RA.
INFO: kegg pathway for CCDS genes
INFO: 6821 out of 27731 ccdsgene.name are annotated through annotation database keggPathway
WARNING: 128 out of 6949 values in annotation database keggPathway are not linked to the project.

$ vtools output NS chr pos ccdsGene.name KgID KgDesc -l 10
1  878522   CCDS3.1      .         .
1  879101   CCDS3.1      .         .
1  901458   .            .         .
1  904196   .            .         .
1  904715   .            .         .
1  904739   .            .         .
1  906412   .            .         .
1  939471   CCDS6.1      hsa04622  RIG-I-like receptor signaling pathway
1  1148494  CCDS12.1     .         .
1  1548655  CCDS41224.2  .         .
```

The `keggPathway` database annotates genes through their CCDS gene ID, which are available in `ccdsGene` and `dbNSFP`. `ccdsGene` is preferred though.

FIND VARIANTS THAT BELONG TO A PATHWAY

```
$ vtools select NS 'kgID="hsa00760"' --output chr pos ref alt ccdsGene.name kgID kgDesc -l 20
1   1675900    G  T  CCDS30565.1  hsa01100  Metabolic pathways
11  70847195   G  C  CCDS8201.1   hsa01100  Metabolic pathways
11  70862326   A  C  CCDS8201.1   hsa01100  Metabolic pathways
14  20010446   G  A  CCDS9552.1   hsa01100  Metabolic pathways
16  29615851   A  G  CCDS10651.1  hsa01100  Metabolic pathways
4   15318290   G  A  CCDS3416.1   hsa04020  Calcium signaling pathway
5   43691831   C  T  CCDS3949.1   hsa01100  Metabolic pathways
5   102922572  T  C  CCDS4096.1   hsa04146  Peroxisome
6   86255952   A  G  CCDS5002.1   hsa01100  Metabolic pathways
6   132214061  A  C  CCDS5150.2   hsa01100  Metabolic pathways
1   1675941    G  A  CCDS30565.1  hsa01100  Metabolic pathways
6   132072584  T  G  CCDS47475.1  .         .
6   132071774  G  A  CCDS47475.1  .         .
6   132072589  T  C  CCDS47475.1  .         .
10  104924699  T  C  CCDS7544.1   hsa01100  Metabolic pathways
6   132071745  G  T  CCDS47475.1  .         .
2   201234575  A  G  CCDS33360.1  hsa01100  Metabolic pathways
6   132103113  G  A  CCDS5148.1   hsa01100  Metabolic pathways
11  70869465   C  T  CCDS8201.1   hsa01100  Metabolic pathways
16  29613945   C  T  CCDS10651.1  hsa01100  Metabolic pathways
```

FIND VARIANTS THAT BELONG TO A PATHWAY

```
$ vtools select NS 'kgID="hsa00760"' --output chr pos ref alt ccdsGene.name kgID kgDesc -l 20
1   1675900    G  T  CCDS30565.1  hsa01100  Metabolic pathways
11  70847195   G  C  CCDS8201.1   hsa01100  Metabolic pathways
11  70862326   A  C  CCDS8201.1   hsa01100  Metabolic pathways
14  20010446   G  A  CCDS9552.1   hsa01100  Metabolic pathways
16  29615851   A  G  CCDS10651.1  hsa01100  Metabolic pathways
4   15318290   G  A  CCDS3416.1   hsa04020  Calcium signaling pathway
5   43691831   C  T  CCDS3949.1   hsa01100  Metabolic pathways
5   102922572  T  C  CCDS4096.1   hsa04146  Peroxisome
6   86255952   A  G  CCDS5002.1   hsa01100  Metabolic pathways
6   132214061  A  C  C
1   1675941    G  A  C
6   132072584  T  G  C
6   132071774  G  A  C
6   132072589  T  C  C
10  104924699  T  C  C
6   132071745  G  T  CCDS47475.1  .         .
2   201234575  A  G  CCDS33360.1  hsa01100  Metabolic pathways
6   132103113  G  A  CCDS5148.1   hsa01100  Metabolic pathways
11  70869465   C  T  CCDS8201.1   hsa01100  Metabolic pathways
16  29613945   C  T  CCDS10651.1  hsa01100  Metabolic pathways
```

Notice any problem with the output?

THE –ALL OPTION

When there are multiple records for a variant in an annotation database, variant tools by default output one of them randomly. The `--all` options tells *variant tools* to output all matching records.

```
$ vtools select NS 'kgID="hsa00760"' --output chr pos ref alt ccdsGene.name kgID kgDesc --all -l
      20
1   1675900    G  T  CCDS55562.1  .          .
1   1675900    G  T  CCDS55561.1  .          .
1   1675900    G  T  CCDS30565.1  hsa00760   Nicotinate and nicotinamide metabolism
1   1675900    G  T  CCDS30565.1  hsa01100   Metabolic pathways
11  70847195   G  C  CCDS8201.1   hsa00760   Nicotinate and nicotinamide metabolism
11  70847195   G  C  CCDS8201.1   hsa01100   Metabolic pathways
11  70862326   A  C  CCDS8201.1   hsa00760   Nicotinate and nicotinamide metabolism
11  70862326   A  C  CCDS8201.1   hsa01100   Metabolic pathways
14  20010446   G  A  CCDS9552.1   hsa00230   Purine metabolism
14  20010446   G  A  CCDS9552.1   hsa00240   Pyrimidine metabolism
14  20010446   G  A  CCDS9552.1   hsa00760   Nicotinate and nicotinamide metabolism
14  20010446   G  A  CCDS9552.1   hsa01100   Metabolic pathways
16  29615851   A  G  CCDS10651.1  hsa00760   Nicotinate and nicotinamide metabolism
16  29615851   A  G  CCDS10651.1  hsa01100   Metabolic pathways
4   15318290   G  A  CCDS3416.1   hsa00760   Nicotinate and nicotinamide metabolism
4   15318290   G  A  CCDS3416.1   hsa01100   Metabolic pathways
4   15318290   G  A  CCDS3416.1   hsa04020   Calcium signaling pathway
5   43691831   C  T  CCDS3949.1   hsa00760   Nicotinate and nicotinamide metabolism
5   43691831   C  T  CCDS3949.1   hsa01100   Metabolic pathways
5   102922572  T  C  CCDS4096.1   hsa00760   Nicotinate and nicotinamide metabolism
```

USING ANNOVAR TO ANNOTATE VARIANTS

Formats such as `ANNOVAR` and `ANNOVAR_exonic_variant_function` are provided to export variants to be analyzed by other programs, and import results from output of these programs.

```
$ vtools export NS --format ANNOVAR > annovar.input
INFO: Using primary reference genome hg18 of the project.
Writing: 100% [==================================] 26,963 45.1K/s in 00:00:000
INFO: 26963 lines are exported from variant table NS

$ ~/bin/annovar/annotate_variation.pl annovar.input ~/bin/annovar/humandb/
NOTICE: The --geneanno operation is set to ON by default
NOTICE: The --buildver is set as 'hg18' by default
NOTICE: Reading gene annotation from /Users/bpeng/bin/annovar/humandb/hg18_refGene.txt ... Done
     with 42259 transcripts (including 7526 without coding sequence annotation) for 23769 unique
     genes
NOTICE: Reading FASTA sequences from /Users/bpeng/bin/annovar/humandb/hg18_refGeneMrna.fa ... Done
      with 16660 sequences
WARNING: A total of 329 sequences will be ignored due to lack of correct ORF annotation
NOTICE: Finished gene-based annotation on 26963 genetic variants in annovar.input
NOTICE: Output files were written to annovar.input.variant_function, annovar.input.
     exonic_variant_function

$ vtools update NS --format ANNOVAR_exonic_variant_function --from_file annovar.input.
     exonic_variant_function --var_info mut_type function
INFO: Using primary reference genome hg18 of the project.
Getting existing variants: 100% [================] 26,963 121.9K/s in 00:00:000
INFO: Updating variants from annovar.input.exonic_variant_function (1/1)
annovar.input.exonic_variant_function: 100% [=======] 23,683 8.1K/s in 00:00:020
INFO: Fields mut_type, function of 23,683 variants are updated
```

IDENTIFYING STOPGAIN MUTATIONS

```
$ vtools output NS mut_type | sort | uniq
.
nonsynonymous SNV
stopgain SNV
stoploss SNV
synonymous SNV
unknown

$ vtools select NS 'mut_type = "stopgain SNV"' --output chr pos ref alt mut_type -l 20
1   12776677   T  A  stopgain SNV
1   20374169   G  A  stopgain SNV
1   48480815   G  T  stopgain SNV
1   143787040  C  T  stopgain SNV
1   143984723  C  T  stopgain SNV
1   159742828  C  T  stopgain SNV
1   159779491  G  A  stopgain SNV
1   221351823  G  A  stopgain SNV
1   236115192  G  A  stopgain SNV
1   246179649  T  A  stopgain SNV
10  4879403    C  T  stopgain SNV
11  5400712    C  T  stopgain SNV
11  48242807   T  A  stopgain SNV
11  48303590   G  A  stopgain SNV
11  55127957   G  A  stopgain SNV
11  56066932   A  T  stopgain SNV
11  56187792   C  T  stopgain SNV
11  60021578   C  T  stopgain SNV
11  62605063   A  C  stopgain SNV
11  62814501   G  A  stopgain SNV
```

Outline

A real-world example

- Import data
- Phenotype and sample statistics
- Annotation
- Select and filter variants
- Output variants and their summary statistics

OUTPUT SUMMARY STATISTICS

```
$ vtools select variant 'ref="-"' --count
Counting variants: 3,059 734.6/s in 00:00:04
775833

$ vtools output variant refGene.name2 'count(*)' --group_by refGene.name2 -l 5
.          5358110
A1BG       17
A1BG-AS1   10
A1CF       144
A2M        145

$ vtools select variant "(ref='A' AND alt='G') OR (ref='G' AND alt='A') OR (ref='C' AND alt='T')
      OR (ref='T' AND alt='C')" --output 'sum(num)'
17120173

$ vtools select variant 'genename is not NULL' --output genename 'sum(case_num)' 'sum(ctrl_num)'
      --group_by genename -l 10
A1BG     10       6
A2ML1    37       0
A4GALT   2        2
A4GNT    9        0
AAAS     1        1
AADAC    9        4
AADACL2  5        8
AADACL3  32       0
AAGAB    7        0
AARS     0        1
```

VTOOLS_REPORT

`vtools_report` is built on top of `vtools` to perform tasks that would require the use of multiple `vtools` commands.

```
$ vtools_report -h
usage: vtools_report [-h] [--version]

                     {trans_ratio,avg_depth,variant_stat,discordance_rate,sequence,plot_fields,
                          plot_geno_fields,plot_association,meta_analysis}
                     ...

A collection of functions that analyze data using vtools and generate various
reports

optional arguments:
  -h, --help            show this help message and exit
  --version             show program's version number and exit

Available reports:
  {trans_ratio,avg_depth,variant_stat,discordance_rate,sequence,plot_fields,plot_geno_fields,
       plot_association,meta_analysis}
    trans_ratio         Transition count, transversion count and
                        transition/transversion ratio
    avg_depth           Average depth for each variant, can be divided by
                        sample variant count
    variant_stat        Reports number of snps, insertions, deletions and
                        substitutions for groups of samples with some size
                        metrics to characterize the indels
    discordance_rate    Calculate discordance rate between pairs of samples
    sequence            Obtain DNA sequence in specified chromosomal region.
                        This command by default outputs nucleotide sequence at
                        the reference genome.
    plot_fields         Dump values of specified variant info field(s) and/or
```

TRANSITION/TRANSVERSION RATIO

Command `trans_ratio` calculates transition - transversion ratio of all mutations in the samples, using an existing field that records the number of variants in the samples.

```
$ vtools_report trans_ratio variant -n num
num_of_transition       num_of_transversion     ratio
16,534,168              8,213,424               2.01307

$ vtools_report trans_ratio variant -n num --group_by num
num     num_of_transition       num_of_transversion     ratio
0       0                       0                       0.00000
1       1,471,898               789,039                 1.86543
10      2,176,350               1,062,220               2.04887
11      51,282                  20,757                  2.47059
12      74,784                  30,504                  2.45161
13      29,458                  12,064                  2.44181
14      43,596                  18,032                  2.41770
15      20,490                  8,055                   2.54376
16      34,896                  14,288                  2.44233
17      11,067                  4,624                   2.39338
18      25,560                  10,332                  2.47387
19      4,294                   1,634                   2.62791
2       1,490,186               763,804                 1.95101
20      11,580                  4,640                   2.49569
3       1,552,902               785,208                 1.97770
4       1,686,952               853,176                 1.97726
5       1,798,620               917,430                 1.96050
6       1,574,268               764,286                 2.05979
7       1,514,898               726,768                 2.08443
8       1,718,088               824,472                 2.08386
9       1,242,999               602,091                 2.06447
```

EXPORT VARIANTS AND GENOTYPES

```
$ vtools export NS -o ns.vcf
INFO: Using primary reference genome hg18 of the project.
ns.vcf: 100% [====================================] 26,963 40.0K/s in 00:00:000
INFO: 26952 lines are exported from variant table NS

$ head -5 ns.vcf
1       878522  .       T       C       .       PASS    .
1       879101  .       G       A       .       PASS    .
1       901458  .       A       G       .       PASS    .
1       904196  .       C       G       .       PASS    .
1       904715  .       G       C       .       PASS    .

$ vtools export NS --format csv --fields chr pos ref alt refGene.name2 SIFT_score -o ns.csv
INFO: Using primary reference genome hg18 of the project.
ns.csv: 100% [====================================] 26,963 14.7K/s in 00:00:018
INFO: 26963 lines are exported from variant table NS

$ head -5 ns.csv
1,878522,T,C,NOC2L,1.0
1,879101,G,A,NOC2L,0.07
1,901458,A,G,C1orf170,0.0
1,904196,C,G,C1orf170,0.46
1,904715,G,C,C1orf170,1.0

$ vtools export NS --samples 1 --format csv --fields chr pos ref alt dbSNP.name refGene.name2
     refGene.name meanQT dbNSFP.SIFT_score dbNSFP.Polyphen2_HDIV_score Polyphen2_HDIV_pred
     Polyphen2_HVAR_score Polyphen2_HVAR_pred dbSNP.func kgDesc  --order_by chr pos --header chr
     pos ref alt rsname gene 'refgene name' 'Quality score' 'SIFT score' 'Polyphen2 HDIV score'
     'Polyphen2 HDIV pred'  'Polyphen2 HVAR score'  'Polyphen2 HVAR pred' 'dbSNP func code' '
     pathway' '%(sample_names)s' --output NS.csv
```

Outline

More advanced features
- Definition and execution of pipelines
- Association Analysis Framework

AVAILABLE PIPELINES

Variant tools pipelines are defined by pipeline description files that are available online, and are executed by command `vtools execute`. Features of the execution process include logging, output-locking, and validation and skipping of executed steps.

```
$ vtools show pipelines
illumina                A pipeline to handle illumina data prepared by CASAVA
                        1.8+. It imports variants from SNPs.vcf and Indel.vcf
                        of multiple samples, separate maxgt and poly into
                        different projects, calculate a few standard
                        statistics and apply a few filters. All results are
                        saved as variant tools snapshots. This pipeline uses
                        command vtools so multi-processing is not supported.
anno_utils              This file defines a number of pipelines to manipulate
                        variant tools annotation databases.
bwa_gatk_hg19           A pipeline to align raw reads from fastq or BAW/SAM
                        files using BWA and GATK best practice. It uses hg19
                        of human reference genome and assumes paired-end reads
                        in plain text and compressed formats.
mosaik_gatk23_align     A pipeline to align raw reads from fastq or BAM/SAM
                        files using Mosaik-aligner. It uses hg19 of human
                        reference genome and assumes paired-end reads in plain
                        text and compressed formats.
```

DECRIPTION OF PIPELINE

A pipeline description file defines one or more pipelines. Additional command line arguments can be passed to customize pipelines.

```
$ vtools show pipeline bwa_gatk_hg19
A pipeline to align raw reads from fastq or BAW/SAM files using BWA and GATK
best practice. It uses hg19 of human reference genome and assumes paired-end
reads in plain text and compressed formats.

Available pipelines: align, call

Pipeline "align":  Align raw reads from input files using bwa, gatk, and
picard. This pipeline accepts raw input files in plain text format, SAM/BAM
format, and their compressed versions (.zip, .tar.gz, .tgz, .bz2, .tbz2 etc).
All input files are assumed to be raw reads from the same sample. This
pipeline generates a calibrated bam file (--output), and its reduced version
if an additional output file is specified.
  align_0:            Download required resources to resource directory
  align_10:           Check existence of commands bwa, samtools and java
  align_11:           Check the version of bwa. Version is 0.7.4 is
                      recommended
  align_12:           Check the version of picard. Version is 1.82 is
                      recommended.
  align_13:           Check the version of GATK. Version 2.4 is recommended.
  align_20:           Check existence of class files for Picard and GATK
  align_30:           Build bwa index for build hg19 of reference genome
  align_40:           Build samtools index for build hg19 of reference genome
  align_100:          Convert bam files to paired fastq files if the input is
                      in bam/sam format. Other input files are returned
                      untouched.
  align_101:          Decompress all input files (.tgz2, .tar, .tar.gz, .gz,
                      .tgz, .zip etc) to a cache directory. Uncompressed files
                      are hard-linked to the cache directory.
```

EXECUTE PIPELINES

```
$ vtools admin --load_snapshot vt_illuminaTestData
Downloading snapshot vt_illuminaTestData.tar.gz from online
INFO: Snapshot vt_illuminaTestData has been loaded

$ vtools execute bwa_gatk_hg19 align --input illumina_test_seq.tar.gz --output test.bam --
      gatk_path $HOME/bin/GATK --picard_path $HOME/bin/Picard
INFO: Executing step align_0 of pipeline bwa_gatk_hg19: Download required resources to resource
      directory
INFO: Using 38 existing resource files under /Users/bpeng/.variant_tools/pipeline_resource/
      gatk23_hg19.
Validating md5 signature: 100% [================================] 2,515,007,932 487.2M/s in
      00:00:05
INFO: Executing step align_10 of pipeline bwa_gatk_hg19: Check existence of commands bwa, samtools
       and java
INFO: Command bwa is located.
INFO: Command samtools is located.
INFO: Command java is located.
INFO: Executing step align_11 of pipeline bwa_gatk_hg19: Check the version of bwa. Version is
      0.7.4 is recommended
INFO: Executing step align_12 of pipeline bwa_gatk_hg19: Check the version of picard. Version is
      1.82 is recommended.
INFO: Executing step align_13 of pipeline bwa_gatk_hg19: Check the version of GATK. Version 2.4 is
       recommended.
INFO: Executing step align_20 of pipeline bwa_gatk_hg19: Check existence of class files for Picard
       and GATK
INFO: /Users/bpeng/bin/Picard/SortSam.jar is located.
INFO: /Users/bpeng/bin/GATK/GenomeAnalysisTK.jar is located.
INFO: Executing step align_30 of pipeline bwa_gatk_hg19: Build bwa index for build hg19 of
      reference genome
INFO: Reuse existing files /Users/bpeng/.variant_tools/pipeline_resource/gatk23_hg19/ucsc.hg19.
      fasta.amb
INFO: Executing step align_40 of pipeline bwa_gatk_hg19: Build samtools index for build hg19 of
      reference genome
```

A PEEK INTO BWA_GATK_HG19.PIPELINE

A pipeline consists of a series of numbered steps. Each step defines input files, how input files are organized and sent for processing, and action(s) to take for each group of input file.

```
[align_301]
# cannot use output of step align200, because we need a list of fastq files
input=${OUTPUT101}
action=RunCommand(cmd='%(bwa)s aln
        ${INPUT: open(INPUT[0] + ".aln_param").read().strip()}
        %(opt_bwa_aln)s -t 4 ${RESOURCE_DIR}/ucsc.hg19.fasta
        ${INPUT} > ${INPUT: INPUT[0] + '.sai'}',
    output="${INPUT: INPUT[0] + '.sai'}",
    max_jobs=10)
# remove all non-fastq files that might have been inputted
input_emitter=EmitInput('single', select='fastq', pass_unselected=False)
comment=Call bwa aln to produce .sai files
```

A PEEK INTO BWA_GATK_HG19.PIPELINE

A pipeline consists of a series of numbered steps. Each step defines input files, how input files are organized and sent to, and how to handle each group of input file.

Pipeline variables keep runtime information of pipelines (for example `${CMD_INPUT}` for command line input of option `--input`). Lambda functions can be used to change the value of pipeline variables.

```
[align_301]
# cannot use output of step align200, because
input=${OUTPUT101}
action=RunCommand(cmd='%(bwa)s aln
        ${INPUT: open(INPUT[0] + ".aln_param").read().strip()}
        %(opt_bwa_aln)s -t 4 ${RESOURCE_DIR}/ucsc.hg19.fasta
        ${INPUT} > ${INPUT: INPUT[0] + '.sai'}',
    output="${INPUT: INPUT[0] + '.sai'}",
    max_jobs=10)
# remove all non-fastq files that might have been inputted
input_emitter=EmitInput('single', select='fastq', pass_unselected=False)
comment=Call bwa aln to produce .sai files
```

A PEEK INTO BWA_GATK_HG19.PIPELINE

A pipeline consists of a series of numbered steps. Each step defines input files, how input files are organized and sent to the command, and how to handle each group of input file.

User parameters keep additional command line parameters (e.g. `%(gatk_path)s`).

```
[align_301]
# cannot use output of step align200, because we need a list of fastq files
input=${OUTPUT101}
action=RunCommand(cmd='%(bwa)s aln
        ${INPUT: open(INPUT[0] + ".aln_param").read().strip()}
        %(opt_bwa_aln)s -t 4 ${RESOURCE_DIR}/ucsc.hg19.fasta
        ${INPUT} > ${INPUT: INPUT[0] + '.sai'}',
    output="${INPUT: INPUT[0] + '.sai'}",
    max_jobs=10)
# remove all non-fastq files that might have been inputted
input_emitter=EmitInput('single', select='fastq', pass_unselected=False)
comment=Call bwa aln to produce .sai files
```

A PEEK INTO BWA_GATK_HG19.PIPELINE

A pipeline consists of a series of numbered steps. Each step defines input files, how input files are organized and sent t… group of input file.

`input_emitter` controls what input files to process, and if they should be passed individually, altogether, or in pairs.

```
[align_301]
# cannot use output of step align200, because we need a list of fastq files
input=${OUTPUT101}
action=RunCommand(cmd='%(bwa)s aln
        ${INPUT: open(INPUT[0] + ".aln_param").read().strip()}
        %(opt_bwa_aln)s -t 4 ${RESOURCE_DIR}/ucsc.hg19.fasta
        ${INPUT} > ${INPUT: INPUT[0] + '.sai'}',
    output="${INPUT: INPUT[0] + '.sai'}",
    max_jobs=10)
# remove all non-fastq files that might have been inputted
input_emitter=EmitInput('single', select='fastq', pass_unselected=False)
comment=Call bwa aln to produce .sai files
```

A PEEK INTO BWA_GATK_HG19.PIPELINE

A pipeline consists of a series of numbered steps. Each step defines input files, how input files are organized and sent to action, and how to group output files of the group of input file.

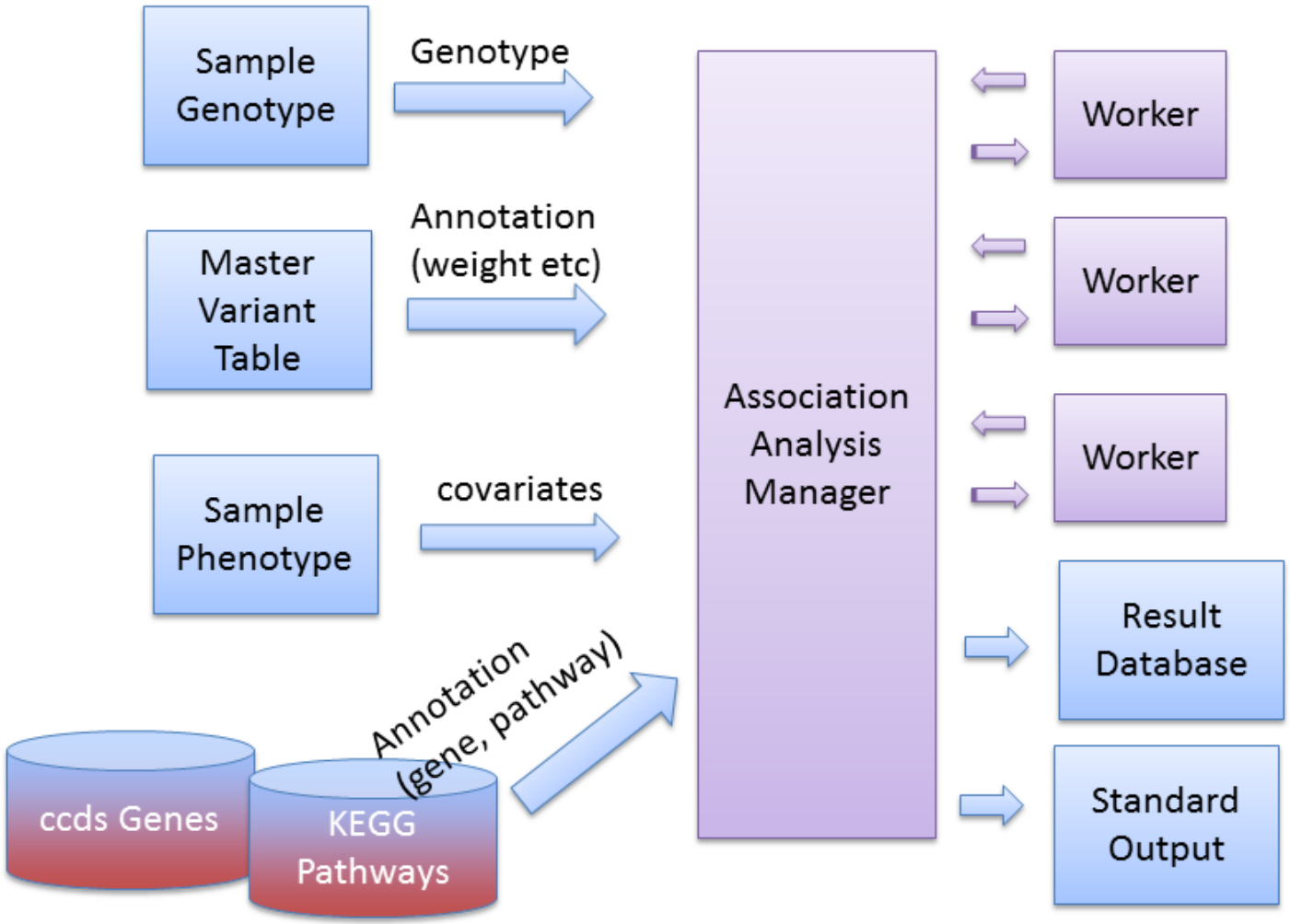
`action` controls the action(s) applies to input files. Output files of the action constitute output of the step.

```
[align_301]
# cannot use output of step align200, because we need a list of fastq files
input=${OUTPUT101}
action=RunCommand(cmd='%(bwa)s aln
        ${INPUT: open(INPUT[0] + ".aln_param").read().strip()}
        %(opt_bwa_aln)s -t 4 ${RESOURCE_DIR}/ucsc.hg19.fasta
        ${INPUT} > ${INPUT: INPUT[0] + '.sai'}',
    output="${INPUT: INPUT[0] + '.sai'}",
    max_jobs=10)
# remove all non-fastq files that might have been inputted
input_emitter=EmitInput('single', select='fastq', pass_unselected=False)
comment=Call bwa aln to produce .sai files
```

Outline

- More advanced features
  - Definition and execution of pipelines
  - Association Analysis Framework

Association analysis framework



SUPPORTED ASSOCIATION TESTS

```
$ vtools show tests
BurdenBt                Burden test for disease traits, Morris & Zeggini 2009
BurdenQt                Burden test for quantitative traits, Morris & Zeggini
                        2009
CFisher                 Fisher's exact test on collapsed variant loci, Li &
                        Leal 2008
Calpha                  c-alpha test for unusual distribution of variants
                        between cases and controls, Neale et al 2011
CollapseBt              Collapsing method for disease traits, Li & Leal 2008
CollapseQt              Collapsing method for quantitative traits, Li & Leal
                        2008
GroupStat               Calculates basic statistics for each testing group
GroupWrite              Write data to disk for each testing group
KBAC                    Kernel Based Adaptive Clustering method, Liu & Leal
                        2010
LinRegBurden            A versatile framework of association tests for
                        quantitative traits
LogitRegBurden          A versatile framework of association tests for disease
                        traits
RBT                     Replication Based Test for protective and deleterious
                        variants, Ionita-Laza et al 2011
RTest                   A general framework for association analysis using R
                        programs
RareCover               A "covering" method for detecting rare variants
                        association, Bhatia et al 2010.
SKAT                    SKAT (Wu et al 2011) and SKAT-O (Lee et al 2012)
SSeq_common             Score statistic / SCORE-Seq software (Tang & Lin
                        2011), for common variants analysis
SSeq_rare               Score statistic / SCORE-Seq software (Tang & Lin
                        2011), for rare variants analysis
VTtest                  VT statistic for disease traits, Price et al 2010
VariableThresholdsBt    Variable thresholds method for disease traits, in the
                        spirit of Price et al 2010
VariableThresholdsQt    Variable thresholds method for quantitative traits, in
```

DETAILS OF AN ASSOCIATION TEST

```
$ vtools show test WeightedBurdenBt
Name:           WeightedBurdenBt
Description:    Weighted genotype burden tests for disease traits, using one or many
                arbitrary external weights as well as one of 4 internal
                weighting themes
usage: vtools associate --method WeightedBurdenBt [-h] [--name NAME]
                                                   [--mafupper MAFUPPER]
                                                   [--alternative TAILED]
                                                   [-p N] [--permute_by XY]
                                                   [--adaptive C]
                                                   [--extern_weight [EXTERN_WEIGHT [EXTERN_WEIGHT
                                                        ...]]]
                                                   [--weight {Browning_all,Browning,KBAC,RBT}]
                                                   [--NA_adjust]
                                                   [--moi {additive,dominant,recessive}]

Weighted genotype burden tests for disease traits, using one or many arbitrary
external weights as well as one of 4 internal weighting themes. External
weights (variant/genotype annotation field) are passed into the test by
--var_info and --geno_info options. Internal weighting themes are one of
"Browning_all", "Browning", "KBAC" or "RBT". p-value is based on logistic
regression analysis and permutation procedure has to be used for "Browning",
"KBAC" or "RBT" weights.

optional arguments:
  -h, --help            show this help message and exit
  --name NAME           Name of the test that will be appended to names of
                        output fields, usually used to differentiate output of
                        different tests, or the same test with different
                        parameters.
  --mafupper MAFUPPER   Minor allele frequency upper limit. All variants
                        having sample MAF<=m1 will be included in analysis.
                        Default set to 0.01
  --alternative TAILED  Alternative hypothesis is one-sided ("1") or two-sided
```

NUMBER OF VARIANTS IN EACH GENE

```
$ vtools admin --load_snapshot vt_ExomeAssociation
$ vtools show phenotypes -l 5
sample_name  gender  age  bmi          status  exposure
SAMP10       1       44   27.93818994  0       0
SAMP100      1       47   33.47268746  0       0
SAMP1000     1       50   26.4845      0       0
SAMP1001     2       59   24.02405     0       1
SAMP1002     2       61   26.32636     0       0
(3175 records omitted)

$ vtools use refGene
$ vtools associate variant BMI -m GroupStat --name stat --stat num_variants sample_size --group_by
      refGene.name2 > groups
INFO: 3180 samples are found
INFO: 2701 groups are found
Loading genotypes: 100% [==========================================] 3,180 4.0/s in 00:13:14
Testing for association: 100% [==================================] 2,701/62 2.8/s in 00:15:58
INFO: Association tests on 2701 groups have completed. 62 failed.

$ head -n 10 groups
refgene_name2   num_variants_stat       sample_size_stat
AADACL4         6                       3180
AAMP            4                       3180
ABCA12          44                      3180
ABCA4           58                      3180
ABCB10          7                       3180
ABCB6           7                       3180
ABCD3           4                       3180
ABCG5           7                       3180
ABCG8           20                      3180
```

NUMBER OF VARIANTS IN EACH GENE

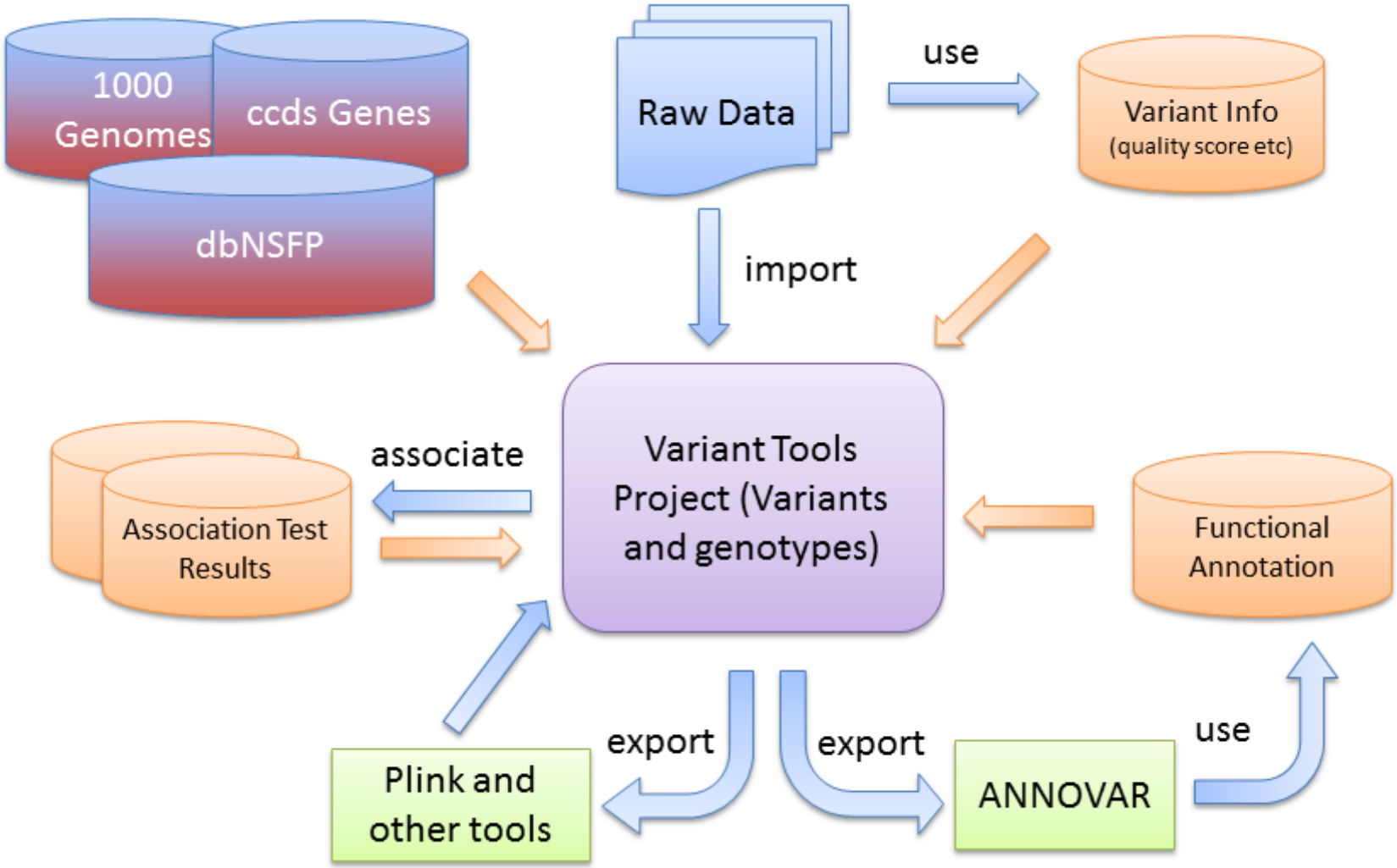
```
$ vtools admin --load_snapshot vt_ExomeAssociation
$ vtools show phenotypes -l 5
sample_name  gender  age  bmi           status
SAMP10       1       44   27.93818994   0
SAMP100      1       47   33.47268746   0
SAMP1000     1       50   26.4845       0       0
SAMP1001     2       59   24.02405      0       1
SAMP1002     2       61   26.32636      0       0
(3175 records omitted)

$ vtools use refGene
$ vtools associate variant BMI -m GroupStat --name stat --stat num_variants sample_size --group_by
      refGene.name2 > groups
INFO: 3180 samples are found
INFO: 2701 groups are found
Loading genotypes: 100% [==========================================] 3,180 4.0/s in 00:13:14
Testing for association: 100% [==================================] 2,701/62 2.8/s in 00:15:58
INFO: Association tests on 2701 groups have completed. 62 failed.

$ head -n 10 groups
refgene_name2   num_variants_stat       sample_size_stat
AADACL4         6                       3180
AAMP            4                       3180
ABCA12          44                      3180
ABCA4           58                      3180
ABCB10          7                       3180
ABCB6           7                       3180
ABCD3           4                       3180
ABCG5           7                       3180
ABCG8           20                      3180
```

Some tests fail because no qualifying variant can be found for a gene.

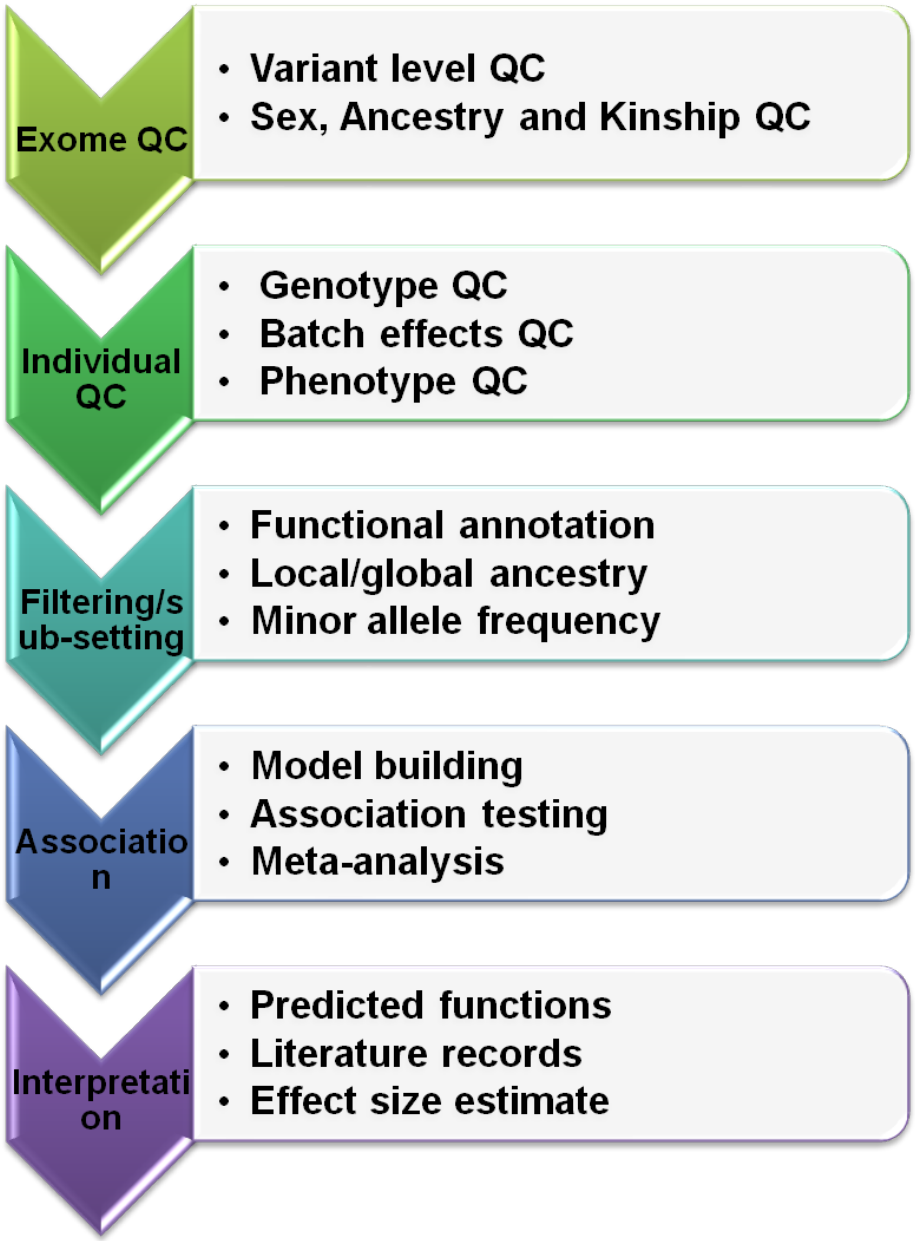
Association analysis

```
$ vtools associate variant BMI --covariate gender -m 'BurdenQt --alternative 2' -g refGene.name2 -
      j8 --to_db bqt > bqt.dat
INFO: 3180 samples are found
INFO: 2701 groups are found
INFO: Starting 8 processes to load genotypes
Loading genotypes: 100% [==========================================] 3,180 30.3/s in 00:01:44
Testing for association: 100% [==================================] 2,701/62 21.7/s in 00:02:04
INFO: Association tests on 2701 groups have completed. 62 failed.
INFO: Using annotation DB bqt in project RA.
INFO: Annotation database used to record results of association tests. Created on Fri, 11 Oct 2013
       23:06:57
INFO: 2701 out of 23953 refgene.name2 are annotated through annotation database bqt

$ head -5 bqt.dat | cut -f1-6
refgene_name2   sample_size_BQt num_variants_BQt       total_mac_BQt   beta_x_BQt      pvalue_BQt
AADACL4         3180            5                     138             -0.486274       0.285215
AAMP            3180            3                     35              1.81079         0.0524737
ABCB10          3180            6                     122             0.0180437       0.971633
ABCB6           3180            7                     151             -0.45799        0.310898

$ vtools output variant chr pos ref alt refGene.name2 bqt.pvalue_BQt bqt.wald_2_BQt -l 10
1       861292  C       G       SAMD11  0.560709073527  0.561785575814
1       866422  C       T       SAMD11  0.560709073527  0.561785575814
1       866517  C       G       SAMD11  0.560709073527  0.561785575814
1       871215  C       G       SAMD11  0.560709073527  0.561785575814
1       871239  C       T       SAMD11  0.560709073527  0.561785575814
1       878709  C       T       SAMD11  0.560709073527  0.561785575814
1       880483  A       G       NOC2L   0.11682257852   0.571101614294
1       880502  C       T       NOC2L   0.11682257852   0.571101614294
1       880943  G       A       NOC2L   0.11682257852   0.571101614294
1       881070  G       A       NOC2L   0.11682257852   0.571101614294
```

Analysis diagram



QC Pipeline



Association study

- > 5000 exome samples with multiple phenotypes
- Variants are removed using a SVM filter based on location, depth, missing calls etc
- Individual QC based on kinship, population structure and sex
- Phenotype QC based on inferred ethnicity, clinical and project-specific information.
- Association tests based on transformed phenotypes (outliers are removed)

FILTERING VARIANTS AND GENOTYPES

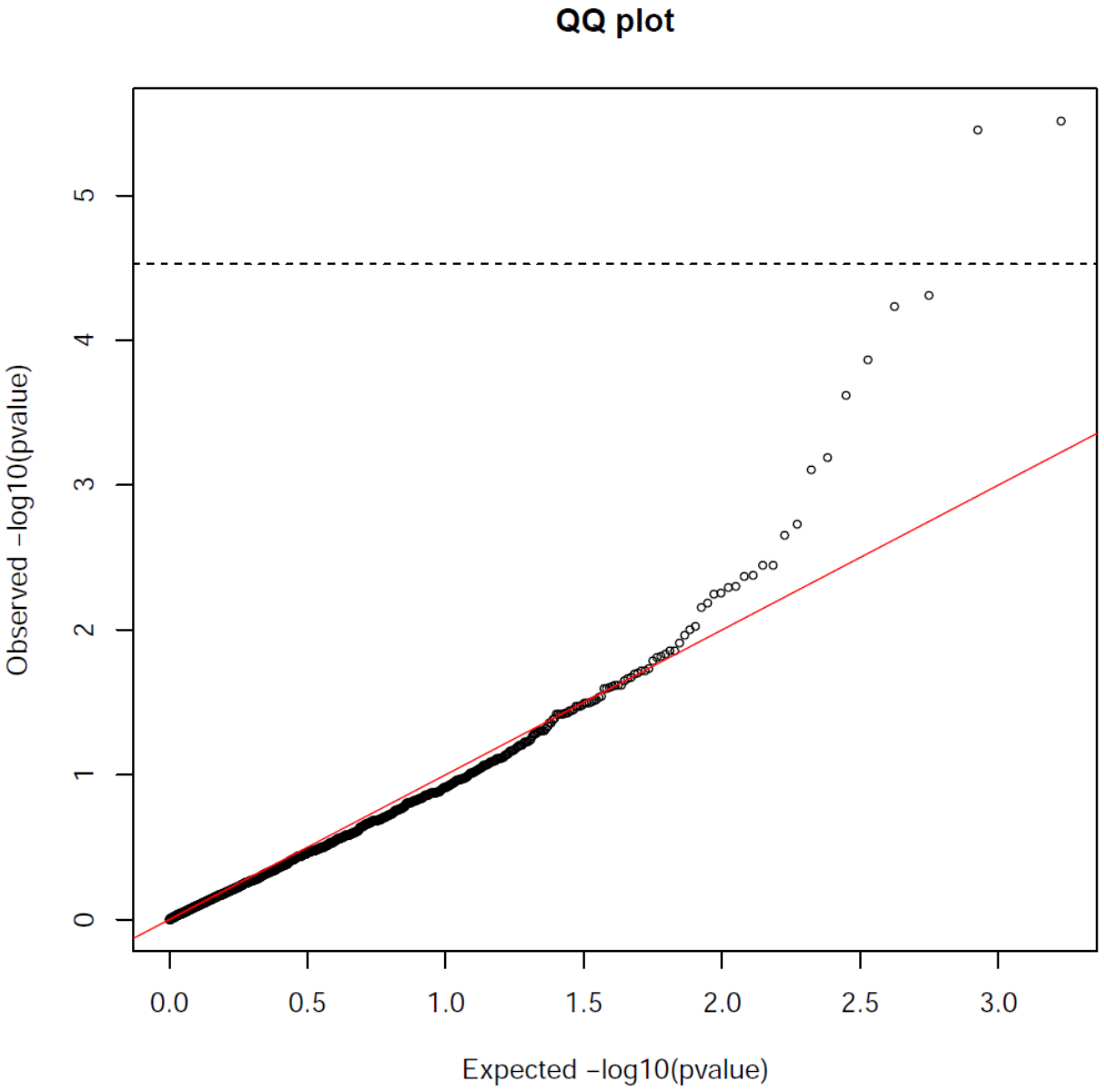
```
# remove SVM fail
vtools exclude variant "esp6800.filter='PASS'" -t variant_to_be_removed
vtools remove varaints variant_to_be_removed

# sample statistics
vtools update variant --from_stat 'totalGD10=#(GT)' 'numGD10=#(alt)' 'hetGD10=#(het)' 'homGD10=#(
     hom)' 'otherGD10=#(other)' --genotypes 'GD>10'
vtools update variant --from_stat 'target_broad_totalGD10=#(GT)' --samples 'Target="broad"' --
     genotypes 'GD>10' -j13
vtools update variant --from_stat 'target_uwrefseq_totalGD10=#(GT)' --samples 'Target="uwrefseq"'
     --genotypes 'GD>10' -j13
vtools update variant --from_stat 'target_V2refseq2010_totalGD10=#(GT)' --samples 'Target="
     V2refseq2010"' --genotypes 'GD>10' -j13
vtools update variant --from_stat 'target_ccds_totalGD10=#(GT)' --samples 'Target="ccds"' --
     genotypes 'GD>10' -j13
vtools update variant --from_stat 'AA_totalGD10=#(GT)' 'AA_numGD10=#(alt)' --genotypes 'GD>10' --
     samples "MDS_RACE=0" -j7

# calculate MAF
vtools update variant --set 'mafGD10=numGD10/(totalGD10*2.0)'

# Remove GD < 10
vtools remove genotypes 'GD < 10'

# export to TPED for plink analysis
vtools export chr5 --format tped --samples 1 -j7 > esp6000_chr5.tped
```

Association analysis

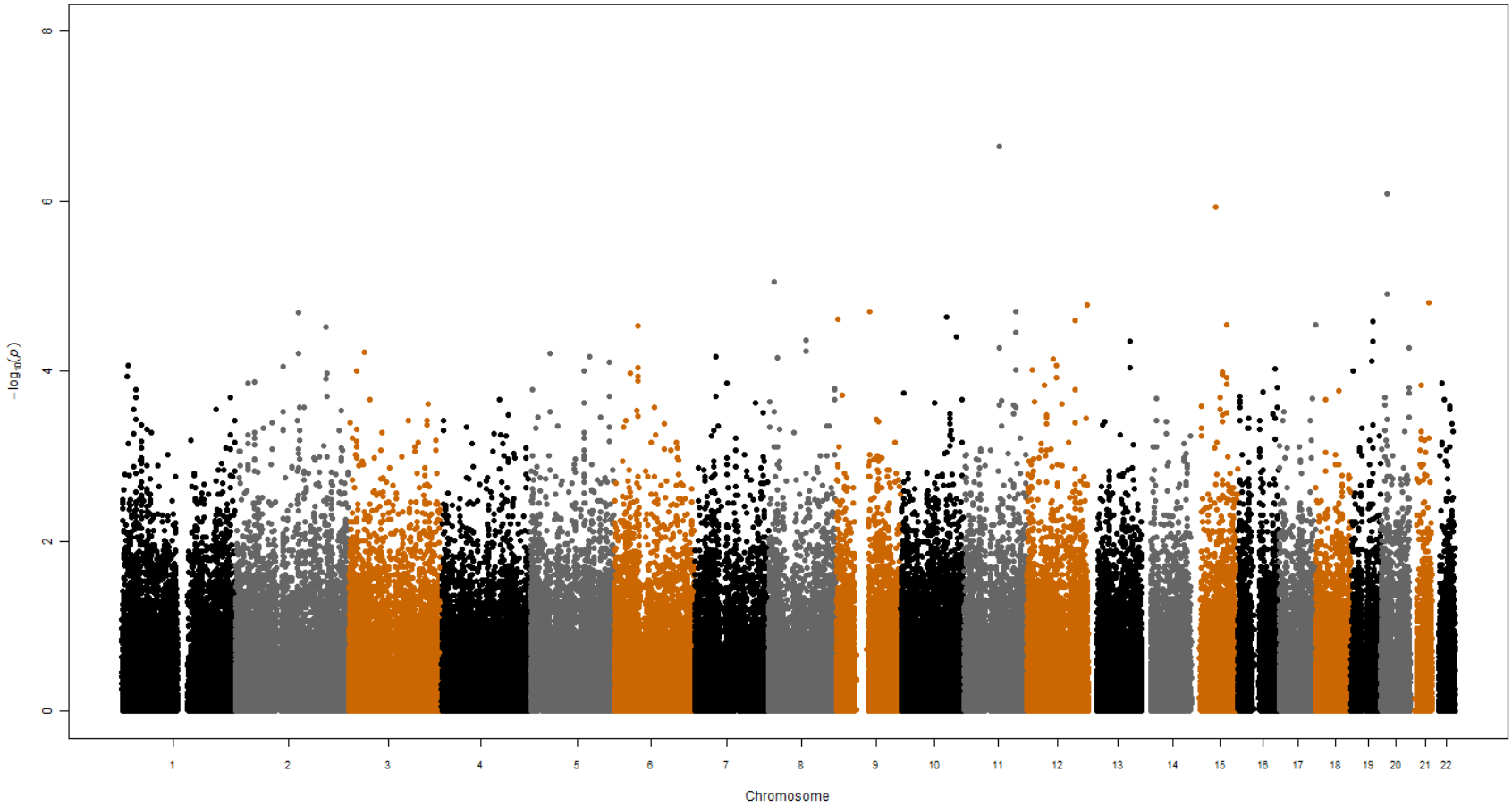
Association analyses were performed for a variety of combinations of methods, samples, parameters etc

```
vtools associate EA_WHR_female_variant_common001 ESP_WHR_BASELINE  \
    --covariates TARGET_1 TARGET_2 COHORT_1 COHORT_2 COHORT_3 COHORT_4 \
          ESP_AGE_BASELINE ESP_BMI_BASELINE ESP_CURRENT_SMOKER_BASELINE PC1_RACE PC2_RACE \
        -m "LinRegBurden --alternative 2" -j7 \
        > ./cache/EA_asso/EA_WHR_female_SNV_30May2012.asso

vtools associate AA_WHR_female_variant_common001 ESP_WHR_BASELINE \
    --covariates TARGET_1 TARGET_2 COHORT_1 COHORT_2 COHORT_3 COHORT_4 ESP_AGE_BASELINE \
          ESP_BMI_BASELINE ESP_CURRENT_SMOKER_BASELINE PC1_RACE PC2_RACE \
        -m "LinRegBurden --alternative 2" -j7 \
        > ./cache/AA_asso/AA_WHR_female_SNV_30May2012.asso

vtools associate EA_WHR_female_variant_rare001 ESP_WHR_BASELINE \
    --covariates TARGET_1 TARGET_2 COHORT_1 COHORT_2 COHORT_3 COHORT_4 \
          ESP_AGE_BASELINE ESP_BMI_BASELINE ESP_CURRENT_SMOKER_BASELINE PC1_RACE PC2_RACE \
        -m "VariableThresholdsQt --alternative 2 -p 10000000 --permute_by X --adaptive 0.000005" \
        -g refGene_exon.name2 -j13 --to_db esp69hEA_WHR_female_rare001_VT \
        > EA_WHR_female_rare_VT_1June2012.asso
```

QQ Plot for one of the tests



Manhattan Plot



Conclusions

- Variant tools greatly simplifies the annotation and analysis of next-gen sequencing data
- It provides a platform on which novel association testing methods could be easily implemented and tested
- It helps the creation but does not eliminate the need of project-specific pipelines
- It does not solve problems with the sequencing analysis itself, such as accuracy of variant calling, coverage and quality of annotation, and statistical power of association tests

Acknowledgment

Genetics

- Dr. Christopher Amos
- Long Ma
- Qiao Min

Epidemiology

- Dr. Paul Scheet
- F. Anthony San Lucas
- Richard Fowler

Baylor College of Medicine

- Dr. Suzanne Leal
- Gao Wang

National Institute of Health

- R01AR44422
- U01GM 92666
- 5R03CA143982
- 1R01HG005859

Schissler Foundattion

Lyda Hill Foundation

NHBLI Exome Sequencing Project

Duncan Family Institute

Prevent Cancer Foundation