VariantFiltering: filtering of coding and non-coding genetic variants

Dei M. Elurbe [1,2] and Robert Castelo [3,4]

June 16, 2014

[1]CIBER de Enfermedades Raras (CIBERER), Barcelona, Spain

[2]Present address: CMBI, Radboud University Medical Centre, Nijmegen, The Netherlands.

[2]Department of Experimental and Health Sciences, Universitat Pompeu Fabra, Barcelona, Spain.

[3]Research Program on Biomedical Informatics (GRIB), Hospital del Mar Medical Research Institute, Barcelona, Spain.

1 Overview

The aim of this software package is to facilitate the filtering and annotation of coding and non-coding genetic variants from a group of related or unrelated individuals among which at least one of them is affected by a genetic disorder. When working with related individuals, *VariantFiltering* can search for variants from the affected individuals that segregate according to a particular inheritance model acting on autosomes (dominant, recessive homozygous or recessive heterozygous -also known as compound heterozygous) or allosomes (X-linked), or that occur *de novo*. When working with unrelated individuals, no mode of inheritance is used for filtering but it can be used to search for variants shared among individuals affected by a common genetic disorder.

VariantFiltering exploits the R/Bioconductor infrastructure to annotate the sought variants with a number of functional annotations. Many of these annotations come from annotation packages, such as *MafDb.ALL.wgs.phase1.release.v3.20101123*, which stores and exposes to the user minimum allele frequency (MAF) values frozen from the latest realease of the 1000 Genomes project.

The main input are Variant Call Format (VCF) files which are parsed using the functionality from the *VariantAnnotation* package for that purpose.

This package contains a toy data set for illustrative purposes only, consisting of a multisample VCF file with variants from chromosomes 20, 21, 22 and allosomes from a trio of CEU individuals from the 1000 Genomes project. To further reduce the execution time of this vignette, only the code for the first analysis that searches for autosomal recessive homozygous variants is actually called and its results reported.

2 Using the package with parallel execution

Functions in *VariantFiltering* to annotate and filter variants leverage the functionality of the Bioconductor package *BiocParallel* to perform in parallel some of the tasks and calculations and reduce the overall execution time. These functions have an argument called `BPPARAM` that allows the user to control how this parallelism is exploited. In particular the user must give as value to this argument the result from a call to the function `bpparam()`, which actually is its default behavior. Here below we modify that behavior to force a call being executed without parallelism. The interested reader should consult the help page of `bpparam()` and the vignette of the *BiocParallel* for further information.

3 Setting up the analysis

To start using *VariantFiltering* the user should consider installing the packages listed in the `Suggests:` field from its `DESCRIPTION` file. After loading *VariantFiltering* the first step is to build a parameter object, of class *VariantFilteringParam* which requires at least a character vector of VCF filenames, as follows:

```
> library(VariantFiltering)
> CEUvcf <- file.path(system.file("extdata", package="VariantFiltering"), "CEUtrio.vcf.bgz")
> CEUped <- file.path(system.file("extdata", package="VariantFiltering"), "CEUtrio.ped")
> param <- VariantFilteringParam(vcfFilenames=CEUvcf, pedFilename=CEUped)
> param
```

```
VariantFiltering parameter object

  VCF file(s): CEUtrio.vcf.bgz
  PED file: CEUtrio.ped
  Gene-centric annotation package: org.Hs.eg.db
  Transcript-centric annotation package: TxDb.Hsapiens.UCSC.hg19.knownGene
  SNP-centric annotation package: SNPlocs.Hsapiens.dbSNP.20120608 (dbSNP Build 137)
  Radical/Conservative AA changes file: AA_chemical_properties_HanadaGojoboriLi2006.tsv
  Other annotation pkg/obj: MafDb.ESP6500SI.V2.SSA137.dbSNP138,
                            MafDb.ALL.wgs.phase1.release.v3.20101123,
                            PolyPhen.Hsapiens.dbSNP131,
                            SIFT.Hsapiens.dbSNP137,
                            phastCons100way.UCSC.hg19,
                            humanGenesPhylostrata
  All transcripts: FALSE
  Filter tag: NA
```

In this case, we are also providing a PED file because the analysis illustrated below works on a trio filtering variants with a particular inheritance model.

4 Autosomal recessive inheritance analysis: Homozygous

Homozygous variants responsible for a recessive trait in the affected individuals can be identified calling the `autosomalRecessiveHomozygous()` function. This function takes a *VariantFilteringParam* object as main argument and selects homozygous variants that are present in all the affected individuals and occur in heterozygosity in the unaffected ones. We change the default value of the `BPPARAM` argument to the object returned by a call to `bpparam("SerialParam")` to disable the parallel execution of this function when building this vignette. For this reason, we need to load the *BiocParallel* package first and perform the call to the function as follows:

```
> library(BiocParallel)
> reHo <- autosomalRecessiveHomozygous(param, BPPARAM=bpparam("SerialParam"))
> reHo
```

```
VariantFiltering results object

  Variants segregate according to a/an autosomal recessive homozygous inheritance model
  Functional annotation filters
    No filtering on presence in dbSNP Build 137
    Variant type: Any
    Amino acid change type: Any
    Populations used for MAF filtering: AFESP, EA_AFESP, AA_AFESP, AFKG, AMR_AFKG, ASN_AFKG, AFR_AFKG, EUR_
    Include MAF NA values: yes
    Maximum MAF: 1.00
    No filtering on nucleotide conservation
    No filtering on gene conservation
    No filtering on cryptic 5'ss
    No filtering on cryptic 3'ss

  Total number of variants: 127
     30 (23.6%) are coding non-synonymous
     49 (38.6%) are coding synonymous
      0 (0.0%) located in known splice sites
      9 (7.1%) located in promoter regions
      8 (6.3%) located in 5' UTR regions
     10 (7.9%) located in 3' UTR regions
     21 (16.5%) located in intronic regions
```

The resulting object belongs to the *VariantFilteringResults* class of objects which eases the task of applying further functional annotation filters to select and prioritize variants. The help page of the *VariantFilteringResults* class contains a list of available accessor methods. Here we illustrate the use of a few of them:

```
> maxMAF(reHo) <- 0.05
> MAFmask <- MAFpop(reHo)
> MAFmask
```

```
   AFESP EA_AFESP AA_AFESP     AFKG AMR_AFKG ASN_AFKG
    TRUE     TRUE     TRUE     TRUE     TRUE     TRUE
AFR_AFKG EUR_AFKG
    TRUE     TRUE
```

```
> MAFpop(reHo) <- !MAFmask
> MAFpop(reHo, "ASN_AFKG") <- TRUE
> MAFpop(reHo)
```

```
   AFESP EA_AFESP AA_AFESP     AFKG AMR_AFKG ASN_AFKG
   FALSE    FALSE    FALSE    FALSE    FALSE     TRUE
AFR_AFKG EUR_AFKG
   FALSE    FALSE
```

```
> minCRYP5ss(reHo) <- 0
> reHo
```

```
VariantFiltering results object

  Variants segregate according to a/an autosomal recessive homozygous inheritance model
  Functional annotation filters
    No filtering on presence in dbSNP Build 137
    Variant type: Any
    Amino acid change type: Any
    Populations used for MAF filtering: ASN_AFKG
    Include MAF NA values: yes
    Maximum MAF: 0.05
    No filtering on nucleotide conservation
    No filtering on gene conservation
    Minimum score for cryptic 5'ss: 0.00
    No filtering on cryptic 3'ss

  Total number of variants: 1
    0 (0.0%) are coding non-synonymous
    1 (100.0%) are coding synonymous
    0 (0.0%) located in known splice sites
    0 (0.0%) located in promoter regions
    0 (0.0%) located in 5' UTR regions
    0 (0.0%) located in 3' UTR regions
    0 (0.0%) located in intronic regions
```

```
> filteredVariants(reHo)
```

```
GRanges with 1 range and 37 metadata columns:
           seqnames               ranges strand | LOCATION
              <Rle>            <IRanges>  <Rle> | <factor>
  rs916425    chr22 [26388337, 26388337]      + |   coding
                      REF                 ALT      FILTER
           <DNAStringSet> <DNAStringSetList> <character>
  rs916425              C                  T           .
                 dbSNP      varAllele       CDSLOC
           <character> <DNAStringSet>    <IRanges>
  rs916425    rs916425              T [6165, 6165]
              PROTEINLOC        TXID     CDSID      GENEID
           <IntegerList> <character> <integer> <character>
  rs916425          2055       73793    214772       84700
           CONSEQUENCE       REFCODON       VARCODON
              <factor> <DNAStringSet> <DNAStringSet>
  rs916425  synonymous            TAC            TAT
                   REFAA          VARAA     TYPE CRYP5ssREF
           <AAStringSet> <AAStringSet> <factor>  <numeric>
  rs916425             Y              Y      SNV       1.45
           CRYP5ssALT CRYP5ssPOS CRYP3ssREF CRYP3ssALT
            <numeric>  <numeric>  <numeric>  <numeric>
  rs916425       1.52          4      -5.64      -5.32
           CRYP3ssPOS        GENE        OMIM      TXNAME
            <numeric> <character> <character> <character>
  rs916425          5      MYO18B      607295  uc003abz.1
                   CDS    AAchange AAchangeType   ASN_AFKG
           <character> <character>  <character> <numeric>
  rs916425       C6165T      Y2055Y Conservative     0.002
              PolyPhen2     PROVEAN phastCons
           <character> <character> <numeric>
  rs916425          NA     Neutral       0.1
           GenePhylostratumTaxID GenePhylostratumIndex
                     <character>             <integer>
  rs916425                  2759                     2
           GenePhylostratum    maxMAF
                <character> <numeric>
  rs916425        Eukaryota     0.002
  ---
  seqlengths:
       chr20     chr21     chr22 ...      chr18     chr19
    63025520  48129895  51304566 ...   78077248  59128983
```

5 Autosomal recessive inheritance analysis: Heterozygous

To filter by this mode of inheritance, also known as compound heterozygous, we need two unaffected parents/ancestors and at least one affected descendant. Variants are filtered in five steps: 1. select heterozygous variants in one of the parents and homozygous in the other; 2. discard previously selected variants that are common between the two parents; 3. group variants by gene; 4. select those genes, and the variants that occur within them, which have two or more variants and there is at least one from each parent; 5. from the previously selected variants, discard those that do not occur in the affected descendants. This is implemented in the function `autosomalRecessiveHeterozygous()`.

```
> compHet <- autosomalRecessiveHeterozygous(param)
```

6 Autosomal dominant inheritance analysis

The function `autosomalDominant()` identifies variants present in all the affected individual(s) discarding the ones that also occur in at least one of the unaffected subjects.

```
> dom <- autosomalDominant(param)
```

7 X-Linked inheritance analysis

The function `xLinked()` identifies variants that appear only in the X chromosome of the unaffected females as heterozygous, don't appear in the unaffected males analyzed and finally are present (as homozygous) in the affected male(s). This function is currently restricted to affected males, and therefore, it cannot search for X-linked segregating variants affecting daughters.

```
> xlid <- xLinked(param)
```

8 De Novo variants analysis

The function `deNovo()` searches for *de novo* variants which are present in one descendant and present in both parents/ancestors. It is currently restricted to a trio of individuals.

```
> deNovo <- deNovo(param)
```

9 Variants analysis for all possible inheritance models

Sometimes more than one mode of inheritance may be compatible with the phenotype of the individuals. To facilitate exploring the different inheritance models we provide a function called `allInheritanceModels()` which does not filter upfront for any mode of inheritance but it allows the user to switch among them by using the shiny app launched by the function `reportVariants()`. As a consequence, the object returned by `allInheritanceModels()` will be have a larger memory footprint. This analysis can be currently performed to a number of related individuals between one and four with at least one of them being affected.

```
> aim <- allInheritanceModels(param)
> aim <- reportVariants(aim)
```

10 Variants analysis on unrelated individuals

When the individuals to analyze are unrelated, and therefore, there is no mode of inheritance to use as a filter, we can filter variants using the function `unrelatedIndividuals()`. Thus, the goal here will be to identify variants that are responsible for a phenotype common to all the individuals.

Similarly to `allInheritanceModels()`, no variants are filtered out upfront, and therefore, the main memory requirements may be larger.

```
> uind <- unrelatedIndividuals(param)
```

11 Create a report from the filtered variants

The function `reportVariants()` allows us to easily create a report from the filtered variants into a CSV or a TSV file as follows:

```
> reportVariants(reHo, type="csv", file="reHo.csv")
```

However, the default value on the `type` argument (`"shiny"`) starts a shiny web app which allows one to interactively filter the variants, obtaining an udpated *VariantFilteringResults* object and downloading the filtered variants and the corresponding full reproducible R code, if necessary.

This `reportVariants()` function is designed to provide for different options according to the exact type of *VariantFilteringResults* object is dispatching. For example, for results objects resulting from the function `allInheritanceModels()`, it will show a tab named "Inheritance" where the user can allocate the individuals present in the VCF file according to different inheritance models.

```
> aim <- reportVariants(aim)
```

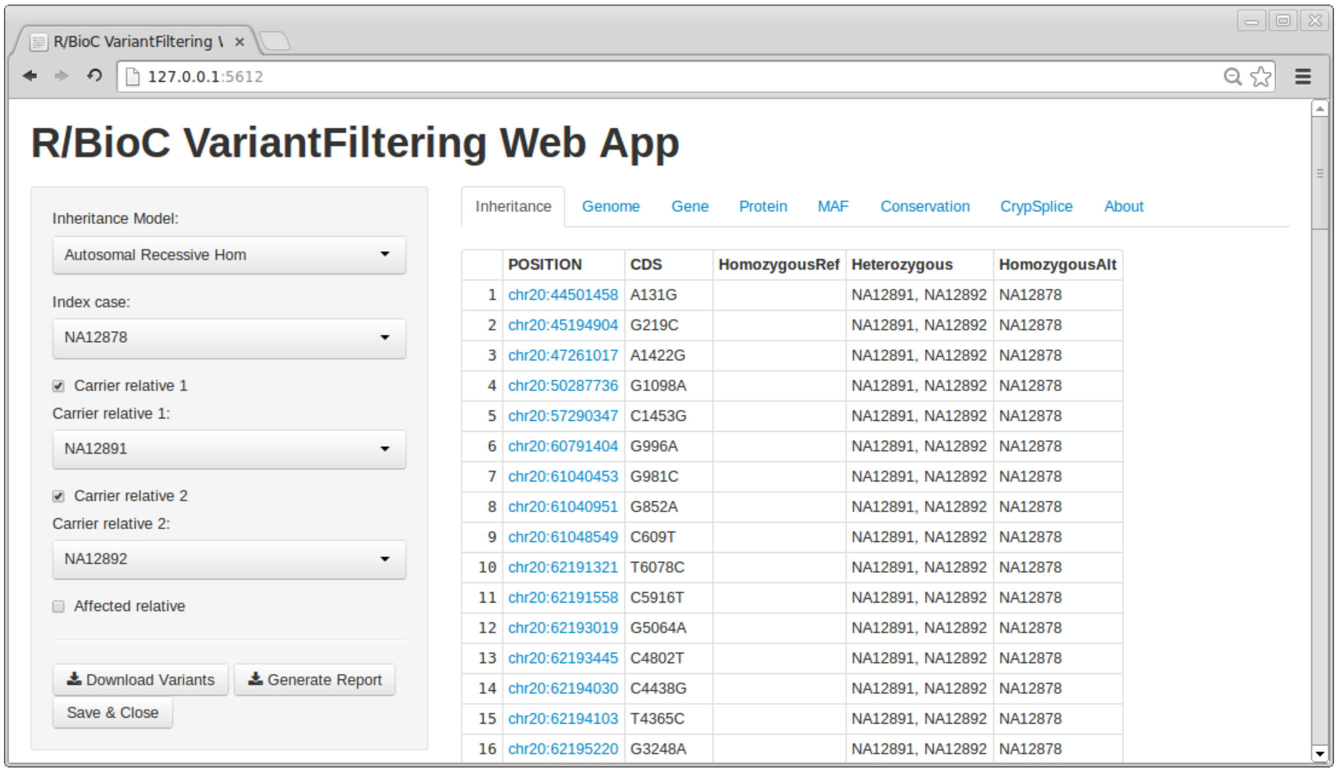
The previous call should open a browser window with a web app similar to the one in Figure 1. The button "Save & Close" will close the web app and return the results object updated according to the filters switched on or off at the web app.

12 Current limitations of the package

The package has a number of limitations and shortcomings which we will list here and update with feedback from users. We will make our best to try to address these issues in forthcoming releases of the package:

- Only SNVs are identified to a particular version of dbSNP using a Bioconductor SNPlocs.* package.
- Multi-allelic variants, i.e., variants with more than one alternate allele, are not yet supported and can prompt errors.
- Filtering by the autosomal recessive heterozygous mode of inheritance is currently restricted to coding variants.
- The X-linked mode of inheritance is currently restricted to affected sons, and therefore, it cannot search for X-linked segregating variants from affected daughters.
- The *de novo* mode of inheritance is currently restricted to a trio with two parents and one descendant.
- The function `allInheritanceModels` is currently restricted to a maximum of four individuals.
- The "Generate Report" button from the shiny app does not provide the full reproducible code when using `allInheritanceModels()` function for the analysis.
- Scoring of splice sites is currently restricted to human and the whole package is currently pretty much human-oriented.

13 Session information



R/BioC VariantFiltering Web App

Inheritance Model: Autosomal Recessive Hom

Index case: NA12878

[✓] Carrier relative 1

Carrier relative 1: NA12891

[✓] Carrier relative 2

Carrier relative 2: NA12892

[ ] Affected relative

	POSITION	CDS	HomozygousRef	Heterozygous	HomozygousAlt
1	chr20:44501458	A131G		NA12891, NA12892	NA12878
2	chr20:45194904	G219C		NA12891, NA12892	NA12878
3	chr20:47261017	A1422G		NA12891, NA12892	NA12878
4	chr20:50287736	G1098A		NA12891, NA12892	NA12878
5	chr20:57290347	C1453G		NA12891, NA12892	NA12878
6	chr20:60791404	G996A		NA12891, NA12892	NA12878
7	chr20:61040453	G981C		NA12891, NA12892	NA12878
8	chr20:61040951	G852A		NA12891, NA12892	NA12878
9	chr20:61048549	C609T		NA12891, NA12892	NA12878
10	chr20:62191321	T6078C		NA12891, NA12892	NA12878
11	chr20:62191558	C5916T		NA12891, NA12892	NA12878
12	chr20:62193019	G5064A		NA12891, NA12892	NA12878
13	chr20:62193445	C4802T		NA12891, NA12892	NA12878
14	chr20:62194030	C4438G		NA12891, NA12892	NA12878
15	chr20:62194103	T4365C		NA12891, NA12892	NA12878
16	chr20:62195220	G3248A		NA12891, NA12892	NA12878

Figure 1: Snapshot of the shiny web app run from VariantFiltering with the function `reportVariants()`. Some of the parameters has been filled for illustrative purposes.

```
> toLatex(sessionInfo())
```

- R version 3.1.0 (2014-04-10), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, utils
- Other packages: AnnotationDbi 1.26.0, BSgenome 1.32.0, Biobase 2.24.0, BiocGenerics 0.10.0, BiocParallel 0.6.1, Biostrings 2.32.0, DBI 0.2-7, GenomeInfoDb 1.0.2, GenomicFeatures 1.16.2, GenomicRanges 1.16.3, IRanges 1.22.9, MafDb.ALL.wgs.phase1.release.v3.20101123 1.0.0, MafDb.ESP6500SI.V2.SSA137.dbSNP138 1.0.0, PolyPhen.Hsapiens.dbSNP131 1.0.2, RSQLite 0.11.4, Rsamtools 1.16.1, SIFT.Hsapiens.dbSNP137 1.0.0, SNPlocs.Hsapiens.dbSNP.20120608 0.99.9, TxDb.Hsapiens.UCSC.hg19.knownGene 2.14.0, VariantAnnotation 1.10.2, VariantFiltering 1.0.4, XVector 0.4.0, org.Hs.eg.db 2.14.0, phastCons100way.UCSC.hg19 1.0.0
- Loaded via a namespace (and not attached): BBmisc 1.6, BSgenome.Hsapiens.UCSC.hg19 1.3.99, BatchJobs 1.2, BiocStyle 1.2.0, GenomicAlignments 1.0.1, RCurl 1.95-4.1, RJSONIO 1.2-0.2, Rcpp 0.11.2, XML 3.98-1.1, biomaRt 2.20.0, bitops 1.0-6, brew 1.0-6, caTools 1.17, codetools 0.2-8, digest 0.6.4, fail 1.2, foreach 1.4.2, htmltools 0.2.4, httpuv 1.3.0, iterators 1.0.7, plyr 1.8.1, rtracklayer 1.24.2, sendmailR 1.1-2, shiny 0.10.0, stats4 3.1.0, stringr 0.6.2, tools 3.1.0, xtable 1.7-3, zlibbioc 1.10.0