Jannovar Documentation

Release 0.11.0

Peter N Robinson, Marten Jaeger, Manuel Holtgrewe

July 03, 2015

Contents

Jannovar is a Java-based program and library for the functional annotation of VCF files.

CHAPTER 1

Quickstart

This short How-To guides you from downloading the Jannovar program to annotating a VCF file in 5 steps.

1. Download the current stable release from our GitHub project by clicking here.
2. Extract the ZIP archive.
   - you should find file called `jannovar-cli-0.14.jar` in the ZIP
   - you should also find a file `small.vcf` file in the folder `examples`
3. Download the RefSeq transcript database for the release *hg19/GRCh37*.

   Note: If you are behind a proxy then you have to pass its path to the `--proxy` option, e.g., `--proxy http://proxy.example.com:8080`. See the section *Proxy Settings* for more information.

```
# java -jar jannovar-cli-0.14.jar download hg19/refseq
```

   This will create the file `data/hg19_refseq.ser` which is a self-contained transcript database and can be used for functional annotation.

4. Annotate the file `small.vcf` from the `examples` directory.

```
# java -jar jannovar-cli-0.14.jar annotate data/hg19_refseq.ser examples/small.vcf
```

   Jannovar will now load the transcript database from `data/hg19_refseq.ser` and then read `examples/small.vcf` file. Each contained variant in this file will be annotated with an `EFFECT` and an `HGVS` field in the `VCF` info column. The `EFFECT` field contains an effect, e.g., `SYNONYMOUS` and the `HGVS` field contains a HGVS representation of the variant. The result will be written out to `small.jv.vcf`.

   The following excerpt shows the first three variants of the `small.vcf` file with their effect and HGVS annotation.

```
1 866511  rs60722469      C       CCCCT   258.62  PASS    EFFECT=INTRONIC;HGVS=SAMD11:NM_152486.
1 879317  rs7523549       C       T       150.77  PASS    EFFECT=MISSENSE;HGVS=SAMD11:XM_0052447
1 879482  .       G       C       484.52  PASS    EFFECT=MISSENSE;HGVS=SAMD11:XM_005244727.1:exc
```

1.1 Next Steps

Of course, you can follow the other manual chapters and get more extensive information on Jannovar. In addition, here are some external links that can help you in your understanding:

Current VCF Specification can be found in the **hts-specs** project on GitHub here.

HGVS Mutation Nomenclature. is mainainted by the Human Genome Variation Society and the nomenclature can be found here.

CHAPTER 2

Installation

There are two options of installing Jannovar. The recommended way for most users is to download a prebuilt binary and is well-described in the *Quickstart* section. This section describes how to build Jannovar from scratch.

2.1 Prequisites

For building Jannovar, you will need

1. Java JDK 6 or higher for compiling Jannovar,
2. Maven 3 for building Jannovar, and
3. Git for getting the sources.

2.2 Git Checkout

In this tutorial, we will download the Jannovar sources and build them in `~/Development/jannovar`.

```
~ # mkdir -p ~/Development
~ # cd ~/Development
Development # git clone https://github.com/charite/jannovar.git jannovar
Development # cd jannovar
```

2.3 Maven Proxy Settings

If you are behind a proxy, you will get problems with Maven downloading dependencies. If you run into problems, make sure to also delete `~/.m2/repository`. Then, execute the following commands to fill `~/.m2/settings.xml`.

```
jannovar # mkdir -p ~/.m2
jannovar # test -f ~/.m2/settings.xml || cat >~/.m2/settings.xml <<END
<settings>
  <proxies>
   <proxy>
      <active>true</active>
      <protocol>http</protocol>
      <host>proxy.example.com</host>
      <port>8080</port>
```

```
      <nonProxyHosts>*.example.com</nonProxyHosts>
    </proxy>
  </proxies>
</settings>
END
```

2.4 Building

You can build Jannovar using `mvn package`. This will automatically download all dependencies, build Jannovar, and run all tests.

```
jannovar # mvn package
```

In case that you have non-compiling test, you can use the *-DskipTests=true* parameter for skipping them.

```
jannovar # mvn install -DskipTests=true
```

2.5 Creating Eclipse Projects

Maven can be used to generate Eclipse projects that can be imported by the Eclipse IDE. This can be done calling `mvn eclipse:eclipse` command after calling `mvn install`:

```
jannovar # mvn install
jannovar # mvn eclipse:eclipse
```

CHAPTER 3

Downloading Transcript Databases

The first step after installing Jannovar is to obtain a **transcript database**. This database stores information about the transcripts, such as the location of a transcript and its exons, its CDS start and end position, and the transcript sequence. There are three major sources of annotation databases for the main model organisms: (1) the UCSC genome browser, (2) the Ensembl project, and (3) the RefSeq database at NCBI. Each database is linked to a certain release of a reference genome.

3.1 Displaying Available Database

Note: TODO: link to writing your own INI file

Jannovar has built-in support for the human and mouse genomes in releases `hg18`, `hg19`, `hg38`, `mm9`, and `mm10`. For each release, the database can originate from the sources `ucsc`, `ensembl`, and `refseq`. Further, the database can be limited to the curated transcripts only when using RefSeq: `refseq_curated`.

The genome release names and the source names are joint into database descriptors such as `hg19/ucsc` and `hg38/refseq`. You can view the built-in database names using the `db-list` Jannovar command:

```
# java -jar jannovar-cli-0.14.jar db-list
[...]
    hg18/refseq_curated
    hg19/ucsc
[...]
```

3.2 Database Download

A database can be downloaded using the `download` command. You can pass a list of database source names to this command. For each, Jannovar will download the database files over the network to the directory `data/${source}` This directory is created if necessary. When a to be downloaded file already exists, Jannovar will not attempt to overwrite this file.

Note: If you have problems with downloading files (e.g., because of proxy settings) and later on building the database fails then you should delete the directory `data/${source}` and retry downloading the file.

Finally, Jannovar will build a file with the extension `.ser` in the directory `data`, e.g. `data/hg19_ucsc.ser`.

Note: If you are behind a proxy then you have to pass the appropriate argument to Jannovar download. For most

users, adding `--proxy http://proxy.example.com:8080/` should suffice. Advanced proxy settings and details are explained in the section *Proxy Settings*

Let us now download the RefSeq and UCSC annotations for human release *hg19*:

```
# java -jar jannovar-cli-0.14.jar download hg19/refseq hg19/ucsc
```

CHAPTER 4

Proxy Settings

If you have to use a proxy for connecting to the internet then you can do so either using command line parameters to Jannovar `dowload` or using environment variables. If you do not have to use a proxy then you can ignore this section.

4.1 Proxy Command Line Arguments

You can specify one proxy URL for all protocols or give a different proxy for each protocol. Below is a list of the proxy-related command line arguments with an example value. The value of `--proxy` can be overridden by the protocol-specific options.

`--proxy http://proxy.example.com:8080/` Fallback proxy URL **most users only have to specify this**.

`--http-proxy http://proxy.example.com:8080/` Proxy URL for the HTTP protocol.

`--ftp-proxy http://proxy.example.com:8080/` Proxy URL for the FTP protocol.

`--https-proxy http://proxy.example.com:8080/` Proxy URL for the HTTPS protocol.

For most users, it is sufficient to use `--proxy` only:

```
# java -jar jannovar-cli-0.14.jar download --proxy http://proxy.example.com:8080/ hg19/ucsc
```

4.2 Proxy Environment Variables

It might be more convenient to use environment variables that can be configured globally. Jannovar interprets the following environment variables that are commonly used on Unix systems (and also interpreted by tools such as `curl`). For each protocol, Jannovar accepts both the upper and the lower case version and it is sufficient to specify one for each protocol.

`http_proxy`, `HTTP_PROXY` Proxy URL for the HTTP protocol.

`https_proxy`, `HTTPS_PROXY` Proxy URL for the HTTPS protocol.

`ftp_proxy`, `FTP_PROXY` Proxy URL for the FTP protocol.

If you are on Linux and have not already done so, you can add the following lines to your startup script (e.g., `~/.profile`):

```
export http_proxy=http://proxy.example.com:8080/
export https_proxy=http://proxy.example.com:8080/
export ftp_proxy=http://proxy.example.com:8080/
export no_proxy="localhost,127.0.0.1,localaddress,.localdomain.com,*.example.com"
```

```
export HTTP_PROXY=http://proxy.example.com:8080/
export HTTPS_PROXY=http://proxy.example.com:8080/
export FTP_PROXY=http://proxy.example.com:8080/
```

If you have write access to `/etc/environment`, you can add the following lines there:

```
http_proxy=http://proxy.example.com:8080/
https_proxy=http://proxy.example.com:8080/
ftp_proxy=http://proxy.example.com:8080/
no_proxy="localhost,127.0.0.1,localaddress,.localdomain.com,*.example.com"
HTTP_PROXY=http://proxy.example.com:8080/
HTTPS_PROXY=http://proxy.example.com:8080/
FTP_PROXY=http://proxy.example.com:8080/
```

CHAPTER 5

Annotating VCF Files

The main purpose of Jannovar is the annotation of all variants in a VCF file. That is, for each annotation, predict the results for all transcripts that can be afflicted by the change. Depending on the configuration, the one effect that is most pathogenic, or all, are written out.

This is done using the `annotate` command. You pass the path to an annotation database and one or more paths to VCF files that are to be annotated. For each file, the resulting annotated file is to the current directory, the file name is derived by replacing the file name suffix `.vcf` to `.jv.vcf`.

For example, for annotating the `small.vcf` file in the `examples` directory:

```
# java -jar jannovar-cli-0.14.jar annotate data/hg19_ucsc.ser examples/small.vcf
[...]
# ls examples/small.jv.vcf
small.jv.vcf
```

The first three variant lines of `examples/small.jv.vcf` will look as follows.

```
1   866511  rs60722469      C       CCCCT   258.62  PASS    ANN=CCCCT|5_prime_utr_variant|LOW|SAMD11
1   879317  rs7523549       C       T       150.77  PASS    ANN=T|missense_variant|MODERATE|SAMD11|EN
1   879482  .       G       C       484.52  PASS    ANN=C|missense_variant|MODERATE|SAMD11|ENTREZ148
```

5.1 Disabling 3' Shifting

The HGVS Nomenclature for the description fo sequence variants requires that variants are to be shifted towards the 3' end of transcripts in case of ambiguities. This is in partial conflict with the VCF standard which requires all variant calls to be shifted towards the 3' end of the genome. In the case that Jannovar shifted the variants towards the 3' end of the transcript, it will generate a `INFO_REALIGN_3_PRIME` information in the message field of the annotation (`ANN` field).

To comply with the VCF annotation standard, Jannovar also implements the `--no-3-prime-shifting` option. Using this switch suppresses this shifting and the variant will be kept as given in the VCF file. Here is an example of using this command line option:

```
# java -jar jannovar-cli-0.14.jar annotate --no-3-prime-shifting \
    data/hg19_refseq.ser examples/small.vcf
```

5.2 The Show-All Option

By default, Jannovar will only write out one most pathogenic variant as predicted. You can use the `--show-all/-a` option to write out all functional annotations:

```
# java -jar jannovar-cli-0.14.jar annotate --show-all \
    data/hg19_refseq.ser examples/small.vcf
```

For example, the first line of `small.jv.vcf` will look as follows and contain multiple effects and HGVS annotations.

```
1   866511  rs60722469      C       CCCCT   258.62  PASS    ANN=CCCCT|5_prime_utr_variant|LOW|SAMD11
```

CHAPTER 6

Annotating Positions

Sometimes, it is useful to annotate a single position only, for example for quick checks or for debugging purposes. You can do this using the `annotate-pos` command of Jannovar.

You have to pass a path to a annotation database file and one or more chromosomal change specifiers. Jannovar will then return the effect and the HGVS annotation for each chromosomal change.

```
# java -jar jannovar-cli-0.14.jar annotate-pos data/hg19_ucsc.ser 'chr1:12345C>A' 'chr1:12346C>A'
[...]
#change     effect  hgvs_annotation
chr1:12345C>A       CODING_TRANSCRIPT_INTRON_VARIANT        DDX11L1:uc010nxq.1:c.38+118C>A:p.=
chr1:12346C>A       CODING_TRANSCRIPT_INTRON_VARIANT        DDX11L1:uc010nxq.1:c.38+119C>A:p.=
```

The format for the chromsomal change is as follows:

```
{CHROMOSOME}:{POSITION}{REF}>{ALT}
```

CHROMOSOME name of the chromosome or contig

POSITION position of the first change base on the chromosome; in the case of insertions the first base after the insertion; the first base on the chromosome has position `1`

REF the reference bases

ALT the alternative bases

CHAPTER 7

JPed - Filter for Compatible Variants

The Jannovar package `de.charite.compbio.jannovar.filter` contains functionality to load PED files and check lists of genotype calls for compatibility with a given pedigree and a selected mode of inheritance. This functionality is exposed in the `jped-cli` program. You can get the command line help for `jped-cli` as follows:

```
# java -jar jped-cli-0.14.jar -h
```

A basic call looks as follows:

```
# java -jar jped-cli-0.14.jar -m MODE IN.ped IN.vcf OUT.vcf
```

This call of `jped-cli` will first read in the pedigree from `IN.ped`. Then, it will read the file `IN.vcf` and filter the variants therein for compatibility with the given `MODE` of inheritance and the pedigree. The resulting VCF file will be written to `OUT.vcf`.

7.1 Available Modes of Inheritance

You can select one of the following modes of inheritance:

`AUTOSOMAL_DOMINANT` Require compatibility with autosomal dominant mode of inheritance. This can also be used to filter for *de novo* mutations.

`AUTOSOMAL_RECESSIVE` Require compatibility with autosomal recessive mode of inheritance.

`X_RECESSIVE` Require compatibility with X-recessive mode of inheritance.

`X_DOMINANT` Require compatibility with X-dominant mode of inheritance.

7.2 Gene-Wise Processing

By default, `jped-cli` checks each record individually for compatibility. Of course, this does not account for composite recessive autosomal mode of inheritance. Here, all variants for a given gene have to be analyzed.

To enable gene-wise processing, you have to pass the `--gene-wise` flag and pass in a path to a Jannovar database (a `.ser` file, as previously downloaded with `jannovar download`). In this case, `jped-cli` will check all variants for compatibility with the selected mode of inheritance and will write out all variants in genes with possible compatibility.

Note: When doing gene-wise processing, all variants are written out for a gene for which a compatible mutation was found. This sometimes causes confusion for users.

CHAPTER 8

Library: Coordinates

TODO

CHAPTER 9

Variant Effects

This section describes the variant effect names that Jannovar uses for annotating variants. These descriptions are Sequence Ontology (SO) terms and meant to be compatible with the Variant annotations in VCF format standard.

9.1 Effect Names

The following table gives a list of the used SO terms, the putative impact, and the SO ID. The section *Classic Jannovar Effects* lists the effect annotations used in the previous Jannovar versions and the corresponding SO-based effect name. The putative impact is one of HIGH, MODERATE, LOW, and MODIFIER. The impact class MODIFIER is both used for terms with hard-to-predict effects and markers (e.g. non_coding_transcript_variant).

Putative Impact	SO ID	SO Term
HIGH	SO:1000182	chromosome_number_variation
HIGH	SO:0001624	transcript_ablation
HIGH	SO:0001572	exon_loss_variant
HIGH	SO:0001909	frameshift_elongation
HIGH	SO:0001910	frameshift_truncation
HIGH	SO:0001589	frameshift_variant
HIGH	SO:0001908	internal_feature_elongation
HIGH	SO:0001906	feature_truncation
HIGH	SO:0001583	mnv
HIGH	SO:1000005	complex_substitution
HIGH	SO:0002012	stop_gained
HIGH	SO:0002012	stop_lost
HIGH	SO:0002012	start_lost
HIGH	SO:0001619	splice_acceptor_variant
HIGH	SO:0001575	splice_donor_variant
HIGH	SO:0001619	rare_amino_acid_variant
MODERATE	SO:0001583	missense_variant
MODERATE	SO:0001821	inframe_insertion
MODERATE	SO:0001824	disruptive_inframe_insertion
MODERATE	SO:0001822	inframe_deletion
MODERATE	SO:0001826	disruptive_inframe_deletion
MODERATE	SO:0002013	5_prime_utr_truncation
MODERATE	SO:0001819	3_prime_utr_truncation
MODERATE	SO:0001630	splice_region_variant
LOW	SO:0001567	stop_retained_variant

Continued on next page

Table 9.1 – continued from previous page

Putative Impact	SO ID	SO Term
LOW	SO:0001582	initiator_codon_variant
LOW	SO:0001819	synonymous_variant
LOW	SO:0001969	coding_transcript_intron_variant
LOW	SO:0001583	non_coding_transcript_exon_variant
LOW	SO:0001970	non_coding_transcript_intron_variant
LOW	SO:0001983	5_prime_UTR_premature_start_codon_gain_variant
LOW	SO:0001623	5_prime_utr_variant
LOW	SO:0001624	3_prime_utr_variant
MODIFIER	SO:1000039	direct_tandem_duplication
MODIFIER	<custom>	<custom>
MODIFIER	SO:0001624	upstream_gene_variant
MODIFIER	SO:0001632	downstream_gene_variant
MODIFIER	SO:0001628	intergenic_variant
MODIFIER	SO:0001819	tf_binding_site_variant
MODIFIER	SO:0001619	regulatory_region_variant
MODIFIER	SO:0002018	conserved_intron_variant
MODIFIER	SO:0001908	intragenic_variant
MODIFIER	SO:0002017	conserved_intergenic_variant
MODIFIER	SO:0001537	structural_variant
MODIFIER	SO:0001580	coding_sequence_variant
MODIFIER	SO:0001908	intron_variant
MODIFIER	SO:0001791	exon_variant
MODIFIER	SO:0001568	splicing_variant
MODIFIER	SO:0001908	miRNA
MODIFIER	SO:0001564	gene_variant
MODIFIER	SO:0001968	coding_transcript_variant
MODIFIER	SO:0001619	non_coding_transcript_variant
MODIFIER	SO:0001624	transcript_variant
MODIFIER	SO:0000605	intergenic_region
MODIFIER	SO:0000340	chromosome
MODIFIER	SO:0001060	sequence_variant

9.2 Classic Jannovar Effects

The original Jannovar used the following terms together with priority levels.

Priority	Classic Term	Description
1	FS_DELETION	frameshift truncation
1	FS_DUPLICATION	frameshift duplication
1	FS_INSERTION	frameshift elongation
1	FS_SUBSTITUTION	frameshift substitution
1	MISSENSE	missense
1	NON_FS_DELETION	inframe deletion
1	NON_FS_DUPLICATION	inframe duplication
1	NON_FS_INSERTION	inframe insertion
1	NON_FS_SUBSTITUTION	inframe substitution
1	SPLICING	splicing
1	START_LOSS	startloss
1	STOPGAIN	stopgain
1	STOPLOSS	stoploss
1	SV_DELETION	1k+ deletion
1	SV_INSERTION	1k+ insertion
1	SV_INVERSION	1k+ inversion
1	SV_SUBSTITUTION	1k+ substitution
2	ncRNA_EXONIC	ncRNA exonic
2	ncRNA_SPLICING	ncRNA splicing
3	UTR3	UTR3
4	UTR5	UTR5
5	SYNONYMOUS	synonymous
6	INTRONIC	intronic
7	ncRNA_INTRONIC	ncRNA intronic
8	DOWNSTREAM	downstream
8	UPSTREAM	upstream
9	INTERGENIC	intergenic
10	ERROR	error

The following table gives a mapping between classic Jannovar terms to SO-based terms. In some cases, two SO attributes are combined to achieve the same annotation.

Priority	Classic Term
1	MISSENSE
1	FS_DELETION
1	FS_INSERTION
1	NON_FS_DELETION
1	NON_FS_INSERTION
1	SPLICING
1	STOPGAIN
1	STOPLOSS
1	FS_DUPLICATION
1	NON_FS_DUPLICATION
1	FS_SUBSTITUTION
1	NON_FS_SUBSTITUTION
1	STARTLOSS
2	ncRNA_EXONIC
2	ncRNA_SPLICING
3	UTR3
4	UTR5
5	SYNONYMOUS
6	INTRONIC
7	ncRNA_INTRONIC
8	UPSTREAM
8	DOWNSTREAM
9	INTERGENIC
10	ERROR

CHAPTER 10

Mode Of Inheritance Filters

Jannovar includes functionality to filter variants for being compatible with a given pedigree and a mode of inheritance. These filters work well for single individuals and the common case of two parents and a number of children. However, there are limitations when using them for larger pedigrees. For such larger families, the filters lose *specificity* but not *sensitivity*. That is, they can fail to filter out less than theoretically possible, but they should not lose any data.

This section describes in detail the checks performed on the variant and pedigrees to give the user a clear understanding on the algorithms and limitations.

The filters are passed a pedigree and a list of genotype calls. The mode of inheritance is selected by the filter choice. The whole list of genotype calls (usually the genotypes of variants falling onto one gene or transcripts) is then checked for compatibility with the given mode of inheritance and pedigree.

The genotype calls of the variants can either be `NOCALL` (no genotype was determined for the person), `REF` (homozygous wild-type), `HET` (heterozygous alternative), or `HOM` (homozygous alternative). In general a caller calls a hemizygous mutations as homozygous. Therefore we use `HOM` and for sensitivity `HET` on known males as hemizygous. The persons can either be affected, unaffected, or their affection state is unknown.

10.1 Autosomal Dominant Filter

This filter can be used to filter for *de novo* mutations as well.

- If the pedigree only contains one person then the variant call list must contain one `HET` call.
- If there is more than one person in the pedigree then there must be at least one compatible call, meaning:
  - at least one affected person has a `HET` call for this variant,
  - no affected person has a `REF` or `HOM` call, and
  - no unaffected person has a `HET` or `HOM` call.

10.2 Autosomal Recessive Filter

The filter first checks for compatibility with autosomal recessive (AR) homozygous and then AR compound heterozygous mode of inheritance.

For AR homozygous, the following checks are performed.

- If the pedigree only contains one person then the variant call list must contain one `HOM` call.
- If there is more than one person in the pedigree then there must be at least one compatible variant call in the list. For this, the following must be true for one variant in the list:

  - at least one affected person has a HOM call for this variant and
  - no affected person has a REF or HET call.
  - The unaffected parents of affected persons must not be REF or HOM.
  - There is no unaffected person that has a HOM call.

For AR compound heterozygous, the following checks are performed.

- If the pedigree only contains one person then there must be at least two HET entries in the variant list.
- If there is more than one person in the pedigree then the algorithm first enumerates *candidate pairs* of variants. The pairs are enumerated for all affected persons that have a father, a mother, or both in the pedigree.
  - The first entry in the pair is compatible with inheritance from the maternal side and the second entry in the pair is compatible from the paternal side.
  - A variant is compatible regarding the paternal side if:
    * the person has calls HET or NOCALL,
    * the person has no father or the father has calls HET or NOCALL,
    * the person has no mother or the mother has calls REF or NOCALL.
  - A variant is compatible regarding the maternal side if:
    * the person has calls HET or NOCALL,
    * the person has no mother or the mother has calls HET or NOCALL,
    * the person has no father or the father has calls REF or NOCALL.
  - Further, no candidate pair may contain the same call for both the maternal and the paternal side, and
  - there must be at least one call for the person, mother, or father that is not NOCALL.
- Each candidate pair is then check for compatibility with affected persons. The following is performed as described below and also with a role swap of the paternal and maternal variant call list.
  - For each affected person, the maternal and paternal variant call list is performed for compatibility. For this, each of the following must be checked:
    * If the maternal list is not empty then the genotype of the person in the paternal list must not be REF or HOM.
    * If the paternal list is not empty then the genotype of the person in the maternal list must not be REF or HOM.
    * If the paternal list is not empty and the person has a father then the father's genotype in the paternal list must not be REF or HOM.
    * If the maternal list is not empty and the person has a mother then the mother's genotype in the maternal list must not be REF or HOM.
    * None of the affected person's unaffected siblings must be both HET in the paternal or maternal list.
    * Every affected siblings of an afffected person must have HET in the paternal or maternal list.
- Finally, we check every unaffected person in the pedigree.
  - For each unaffected person in the pedigree, neither the maternal nor the paternal call list from the candidate can contain a HOM call for the unaffected person.
  - If the call for the unaffected persons is HET in both the paternal and the maternal call list. Then, the father's and mother's genotype are checked in the maternal call list of the candidate their genotypes in the paternal call list are considered.

    * Let the first two genotypes be `pp` and `mp` and the second two genotypes be `pm` and `mm`.
    * In the case of `pp == HET and mp == REF and pm == REF and mm == HET` and the case of `pp == REF and mp == HET and pm == HET and mm == REF`, the candidate pairs incompatible and compatible otherwise.

10.3 Autosomal X-Dominant Filter

- First of all variants must be X-Chromosomal.
- If the pedigree only contains one person then we decide if * the person is female then the variant call list must contain one `HET` call. * else the variant call list must contain a `HET` or a `HOM` call.
- If there is more than one person in the pedigree then there must be at least one compatible call, meaning: * at least one affected male has a `HET` or `HOM` call or a affected female a `HET` call for this variant, * no affected person has a `REF` call, * no a affected female has a `HOM` call, and * no unaffected person has a `HET` or `HOM` call.

10.4 Autosomal X-Recessive Filter

The filter first checks for compatibility with X-chromosomal recessive (XR) homozygous and then XR compound heterozygous mode of inheritance. XR is different to the AR filter, because affected males are always hemizygous (homozygous for the callers). So males do not have compund heterozygous variants.

For XR homozygous, the following checks are performed.

- First of all variants must be X-Chromosomal.
- **If the pedigree only contains one person then we decide if**
    - the person is female then the variant call list must contain one `HOM` call,
    - else the variant call list must contain a `HET` or a `HOM` call.
- If there is more than one person in the pedigree then there must be at least one compatible variant call in the list. For this, the following must be true for one variant in the list:
    - at least one affected male has a `HET` or `HOM` call or a affected female a `HOM` call for this variant,
    - no affected person has a `REF` or no affected female person has a `HET` call.
    - **For the parents of affected femals**
        * the father must be affected and
        * the mother cannot have it `REF` or `HOM`
    - For the parents of affected males * the unaffected father cannot have the variant `HET` or `HOM` * the mother cannot be `HOM`
    - There is no unaffected person that has a `HOM` call.
    - There is no unaffected male person that has a `HET` call.

For XR compound heterozygous, the following checks are performed.

- First of all variants must be X-Chromosomal.
- **If the pedigree only contains one person then we decide if**
    - the person male we do not allow any call. Please use the XR filter.
    - else we use the AR compound heterozygous filter.

- If there is more than one person in the pedigree then the algorithm first enumerates *candidate pairs* of variants. The pairs are enumerated for all affected persons that have a father, a mother, or both in the pedigree.
  - The first entry in the pair is compatible with inheritance from the maternal side and the second entry in the pair is compatible from the paternal side.
  - A variant is compatible regarding the paternal side if:
    * the person has calls `HET`, `NOCALL`, or if not female `HOM`,
    * the person has no father or the father has calls `HET`, `HOM`, or `NOCALL`,
    * the person has no mother or the mother has calls `REF` or `NOCALL`.
  - A variant is compatible regarding the maternal side if:
    * the person has calls `HET`, `NOCALL`, or if not female `HOM`,
    * the person has no mother or the mother has calls `HET` or `NOCALL`, and
    * no restrcition to the father because he must be affected. See ckecks later.
  - Further, no candidate pair may contain the same call for both the maternal and the paternal side, and
  - there must be at least one call for the person, mother, or father that is not `NOCALL`.
- Each candidate pair is then check for compatibility with affected persons. The following is performed as described below and also with a role swap of the paternal and maternal variant call list.
  - For each affected person, the maternal and paternal variant call list is performed for compatibility. For this, each of the following must be checked:
    * If the maternal list is not empty then the genotype of a female person in the paternal list must not be `REF` or `HOM`.
    * If the paternal list is not empty then the genotype of the person in the paternal list must not be `REF` or in case of a female `HOM`.
    * If the paternal list is not empty and the person has a father then the father's genotype in the paternal list must not be `REF`.
    * If the maternal list is not empty and the person has a mother then the mother's genotype in the maternal list must not be `REF` or `HOM`.
    * None of the affected person's unaffected siblings must be both `HET` in the paternal or maternal list.
    * Every affected siblings of an afffected person must have `HET` in the paternal or maternal list.
- Finally, we check every unaffected person in the pedigree.
  - For each unaffected person in the pedigree, neither the maternal nor the paternal call list from the candidate can contain a `HOM` or for males also a `HET` call for the unaffected person.
  - If the call for the unaffected persons is `HET` in both the paternal and the maternal call list. Then, the father's and mother's genotype are checked in the maternal call list of the candidate their genotypes in the paternal call list are considered.

CHAPTER 11

Custom Datasources

Jannovar ships with a number of predefined data sources (e.g., UCSC, Ensembl, and RefSeq for human releases hg18 to hg38, and mouse mm9 and mm10). However, it is quite easy to define your own data source by writing a datasource INI file. This section describes how to define your own data source.

Note: If you think that your new data source would be useful for others, please send them to us either using our issue tracker or by sending an email to Peter N Robinson <peter.robinson@charite.de>.

11.1 Datasource INI Files

The data sources are defined in INI files. For example, consider the following definition of human release `hg19` from UCSC:

```
[hg19/ucsc]
type=ucsc
alias=MT,M,chrM
chromInfo=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/chromInfo.txt.gz
chrToAccessions=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/Assemble
chrToAccessions.format=chr_accessions
knownCanonical=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownCanonical.txt.gz
knownGene=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownGene.txt.gz
knownGeneMrna=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownGeneMrna.txt.gz
kgXref=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/kgXref.txt.gz
knownToLocusLink=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownToLocusLink.txt.gz
```

The section name `hg19/ucsc` defines the data source name. When saving the above file contents as `my_ucsc.ini`, you can pass it to the Jannovar `download` command `--data-source-list`/`-s`.

```
java -Xms2G -Xmx2G -jar jannovar-cli-0.14.jar download -s my_ucsc.ini hg19/ucsc
```

Your INI file can either add new definitions or override the built-in ones. In fact, the definition from above is part of the INI file that is contained in the Jannovar JAR file and used by default.

The `type` setting of the data source section defines the type of the data source. Currently, Jannovar supports the types `ensembl`, `refseq`, and `ucsc`. The sections below explain the general settings and the data source types further.

11.2 Chromosome Aliasing

The `alias` setting defines an aliasing of the contigs and chromosomes. It can be used regardless of the used data source type.

The names of the contigs from the different data sources usually differ between UCSC and RefSeq (and Ensembl which uses the same names as RefSeq). Usually, the UCSC names can be derived from the RefSeq names by prepending `"chr"`. However, this is not true for the important case of the mitochondrial chromosome.

The `alias` line from above defines an alias between the chromosome names *MT*, *M*, and *chrM*. The first entry (*MT*) is implicitely added if it is not in the *chromInfo* file (see *Name Mapping and Lengths*). This is the case for older RefSeq releases.

11.3 Name Mapping and Lengths

The `chromInfo` setting defines the URL to the `chromInfo.txt.gz` file from UCSC. Usually, this URL is `http://hgdownload.soe.ucsc.edu/goldenPath/${RELEASE}/database/chromInfo.txt.gz`. This file contains the contig lengths for each chromosome with the UCSC name of the chromosome/contig (e.g., `chr19`).

The `chrToAccessions` setting defines the URL to the RefSeq file that contains the mapping from the RefSeq names to the RefSeq and GenBank contig sequence accessions. It is assumed that the UCSC contig names are derived from the RefSeq contig names by prepending `"chr"`, also see *Chromosome Aliasing*. This information is required as it is equally common to use the RefSeq names, UCSC names, or Genbank or RefSeq contig sequence accessions.

The two settings `chromInfo` and `chrToAccessions` have to be provided for all data source types.

The `chroToAccessions` file can have different formats, specified as `chrToAccessions.format`. The "modern" one is `chr_accessions` where the file is a TSV file with five columns, e.g.:

```
#Chromosome RefSeq Accession.version        RefSeq gi       GenBank Accession.version      GenBank g
1   NC_000001.10    224589800       CM000663.1      224384768
2   NC_000002.11    224589811       CM000664.1      224384767
3   NC_000003.11    224589815       CM000665.1      224384766
[...]
```

The first column gives the RefSeq name, the second the RefSeq sequence accession number, and the fourth one the GenBank accession number.

The `chr_NC_gi` file format has four columns and contains the mapping for the HuRef but also alternative assemblies, e.g.:

```
#Chr        Accession.ver   gi      Assembly
1   AC_000044.1     89161184        Celera
2   AC_000045.1     89161198        Celera
[...]
1   AC_000133.1     157704448       HuRef
2   AC_000134.1     157724517       HuRef
```

In this case, you have to specify a value that the last column should match to. The hg18 release uses the `chr_NC_gi` format, for example. Here, we filter the lines to those having `"HuRef"` in the last column:

```
[hg18/refseq]
type=refseq
alias=MT,M,chrM
chromInfo=http://hgdownload.soe.ucsc.edu/goldenPath/hg18/database/chromInfo.txt.gz
chrToAccessions=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/BUILD.36.3/Assembled_chromosomes
```

```
chrToAccessions.format=chr_NC_gi
chrToAccessions.matchLast=HuRef
gff=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/BUILD.36.3/GFF/ref_NCBI36_top_level.gff3.gz
rna=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/BUILD.36.3/RNA/rna.fa.gz
```

11.4 Ensembl Data Sources

When selecting the `ensembl` data source type then you have to pass the transcript definition GTF URL to `gtf` and the cDNA FASTA file to `cdna`. Below is an example for the Ensemble data source for human release hg19.

```
[hg19/ensembl]
type=ensembl
alias=MT,M,chrM
chromInfo=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/chromInfo.txt.gz
chrToAccessions=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/Assembled
chrToAccessions.format=chr_accessions
gtf=ftp://ftp.ensembl.org/pub/release-74/gtf/homo_sapiens/Homo_sapiens.GRCh37.74.gtf.gz
cdna=ftp://ftp.ensembl.org/pub/release-74/fasta/homo_sapiens/cdna/Homo_sapiens.GRCh37.74.cdna.all.fa
```

11.5 RefSeq Data Sources

When selecting the `ensembl` data source type then you have to pass the transcript definition GFF URL to `gff` and the RNA FASTA file to `rna`. Below is an example for the RefSeqe data source for human release hg19.

```
[hg19/refseq]
type=refseq
alias=MT,M,chrM
chromInfo=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/chromInfo.txt.gz
chrToAccessions=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/Assembled
chrToAccessions.format=chr_accessions
gff=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/GFF/ref_GRCh37.p13_to
rna=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/RNA/rna.fa.gz
```

For RefSeq, you can also limit building the database to those transcripts that are curated (e.g., that do not have a name starting with `"XM_"` or `"XR_"`. You can do this by setting `onlyCurated` to `true`:

```
[hg19/refseq_curated]
type=refseq
alias=MT,M,chrM
onlyCurated=true
chromInfo=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/chromInfo.txt.gz
chrToAccessions=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/Assembled
chrToAccessions.format=chr_accessions
gff=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/GFF/ref_GRCh37.p13_to
rna=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/RNA/rna.fa.gz
```

11.6 UCSC Data Sources

For UCSC data sources, you have specify the settings `knownCanonical`, `knownGene`, `knownGeneMrna`, `kgXref`, and `knownToLocusLink`. These can usually be derived from the example below by exchanging `hg19` by the release id (e.g., `mm10` for mouse release 10).

```
[hg19/ucsc]
type=ucsc
alias=MT,M,chrM
chromInfo=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/chromInfo.txt.gz
chrToAccessions=ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/ANNOTATION_RELEASE.105/Assembled
chrToAccessions.format=chr_accessions
knownCanonical=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownCanonical.txt.gz
knownGene=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownGene.txt.gz
knownGeneMrna=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownGeneMrna.txt.gz
kgXref=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/kgXref.txt.gz
knownToLocusLink=http://hgdownload.soe.ucsc.edu/goldenPath/hg19/database/knownToLocusLink.txt.gz
```

CHAPTER 12

Java Memory Settings

Jannovar is a Java program that runs using the Java Virtual Machine (JVM). The program does not allocate memory directly but through the JVM which uses a fixed maximal memory limit. If Jannovar terminates by throwing an exception `java.lang.OutOfMemoryError` then you have to increase the memory limit of the JVM.

One way of doing this by setting the environment variable `JAVA_TOOL_OPTIONS`. For example, the following line increases the available memory to 2 GB of RAM.

```
export JAVA_TOOL_OPTIONS="-Xms2G -Xmx2G"
```

If you prefer, then you can also pass these options to the invokation of JVM. The following Jannovar invocation allows to use up to 2 GB of RAM:

```
java -Xms2G -Xmx2G -jar jannovar-cli-0.14.jar [...]
```

CHAPTER 13

Jannovar License

Jannovar is licensed under the 3-Clause BSD License:

```
Copyright (c) 2013-2015, Charite Universitaetsmedizin Berlin
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this
list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this
list of conditions and the following disclaimer in the documentation and/or
other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER
CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
```

If you have any problems running Jannovar or find any bugs then please report them on our GitHub bug tracker or by sending an email to Peter N Robinson <peter.robinson@charite.de>.