

---

# **GREEN-DB and GREEN-VARAN Documentation**

**Edoardo Giacobuzzi**

**Jan 24, 2023**



---

## Contents:

---

<b>1</b>	<b>1. The GREEN-DB collection</b>	<b>3</b>
<b>2</b>	<b>2. The GREEN-VARAN tool set</b>	<b>5</b>
<b>3</b>	<b>3. The prioritization workflow</b>	<b>7</b>
<b>4</b>	<b>How to cite</b>	<b>9</b>
4.1	The GREEN-DB . . . . .	9
4.2	GREEN-VARAN tool set . . . . .	16
4.3	Download resources . . . . .	29
4.4	How to cite . . . . .	32
<b>5</b>	<b>Indices and tables</b>	<b>35</b>



Welcome to the house of GREEN-DB and GREEN-VARAN! This documentation describes the resources part of the Genomic Regulatory Elements Encyclopedia

The GREEN project is made by 3 main components:



---

## 1. The GREEN-DB collection

---

The collection includes information useful for the annotation of non-coding variants in regulatory regions

- a database (GREEN-DB) containing ~2.4M regulatory regions in the human genome with information on controlled gene(s) and tissue(s) of activity
- pre-processed indexed BED files representing functional genomic signals (TFBS, DNase peaks, UCNE, TADs)
- pre-processed indexed tables for 12 non-coding variant impact prediction scores and PhyloP100 conservation

The GREEN-DB files can be downloaded from Zenodo: <https://zenodo.org/record/5636209> All the additional pre-processed datasets are also available from Zenodo, see the Download section.





---

### 2. The GREEN-VARAN tool set

---

This include tools and workflows that can be used to interact with information in the GREEN-DB and annotate VCF files

- annotate small variants or structural variants with regulatory impact information, including possibly controlled genes
- add additional annotations on functional elements and non-coding prediction scores
- prioritize small variants for possible regulatory impact
- given a list of variants or regions, query the GREEN-DB for detailed information

Available from GitHub: <https://github.com/edg1983/GREEN-VARAN>



---

### 3. The prioritization workflow

---

A Nextflow workflow is available to automate download of the GREEN-DB and supporting resources and run the prioritization workflow on a VCF file. This workflow can be run on one or multiple VCF file(s) and will automatically annotated the desired scores and regions and then perform GREEN-VARAN annotation.



If you find GREEN-DB and GREEN-VARAN useful for your research please cite our manuscript (<https://academic.oup.com/nar/article/50/5/2522/6541021>) See also the how to cite section

The Download section lists locations to download the GREEN-DB and other resource files for annotation

## 4.1 The GREEN-DB

GREEN-DB is a comprehensive collection of potential regulatory regions in the human genome including ~2.4M regions from 16 data sources and covering ~1.5Gb evenly distributed across chromosomes. The regulatory regions are grouped in 5 categories: enhancer, promoter, silencer, bivalent, insulator.

Each region is described by its genomic location, region type, method(s) of detection, data source and closest gene; ~35% of regions are annotated with controlled genes, ~40% with tissue(s) of activity, and ~14% have associated phenotype(s). GREEN-DB is available as an SQLite database and regions information with controlled genes are also provided as extended BED files for easy integration into existing analysis pipelines.

For details on how the database was compiled please refer to the original publication <https://doi.org/10.1101/2020.09.17.301960>

The GREEN-DB database is available for free for **academic use** and available for download in a [Zenodo repository](#). The full database is available as SQLite and a summary of region-based information is provided in BED files.

### 4.1.1 Database region content

## GRCh38

GREEN-DB	N	Bases covered
N Enhancer	1832830	1449153178
N Promoter	565323	234315553
N Silencer	4302	894792
N Bivalent	8409	11210309
N Insulator	23	17504
All regions	2410887	1502180018

## GRCh37

GREEN-DB	N	Bases covered
N Enhancer	1834183	1450755698
N Promoter	566102	234890654
N Silencer	4306	895868
N Bivalent	8413	11215000
N Insulator	23	17504
All regions	2413027	1504116499

### 4.1.2 Summary statistics on the database

### 4.1.3 SQLite database structure

The SQLite database contains 16 tables (expected columns are listed in the image):

- **GRCh37 / GRCh38 regions** GREEN-DB regions coordinate; region type; constraint percentile; closest gene symbol, Ensembl ID and distance; PhyloP100 statistics
- **Tissues** tissue(s) of activity for a region or a region-gene interaction
- **Genes** controlled gene(s)
- **Methods** method(s) supporting each region and region-gene interaction. This may correspond to the data source when no specific method information was available.
- **Phenotypes** potentially associated phenotypes
- **GRCh37 / GRCh38 TFBS** transcription factor binding sites
- **GRCh37 / GRCh38 DNase** DNase hypersensitivity peaks
- **GRCh37 / GRCh38 dbSuper** super-enhancers as defined by dbSuper
- **GRCh37 / GRCh38 LoF\_tolerance** the probability of LoF tolerance for enhancers
- **GRCh37 / GRCh38 UCNE** ultraconserved noncoding elements
- **GRCh37 / GRCh38 TAD** TAD domains from TADKB

Main tables (regions, tissues, genes and methods) are linked by the unique region ID. Additionally, a unique interaction ID identifies each gene-region pair in the gene table and it's linked to methods and tissues tables. Linking tables are included that map the overlap between GREEN-DB region IDs and each of TFBS, DNase, dbSuper and LoF\_tolerance region IDs, reporting also the fraction of overlap.

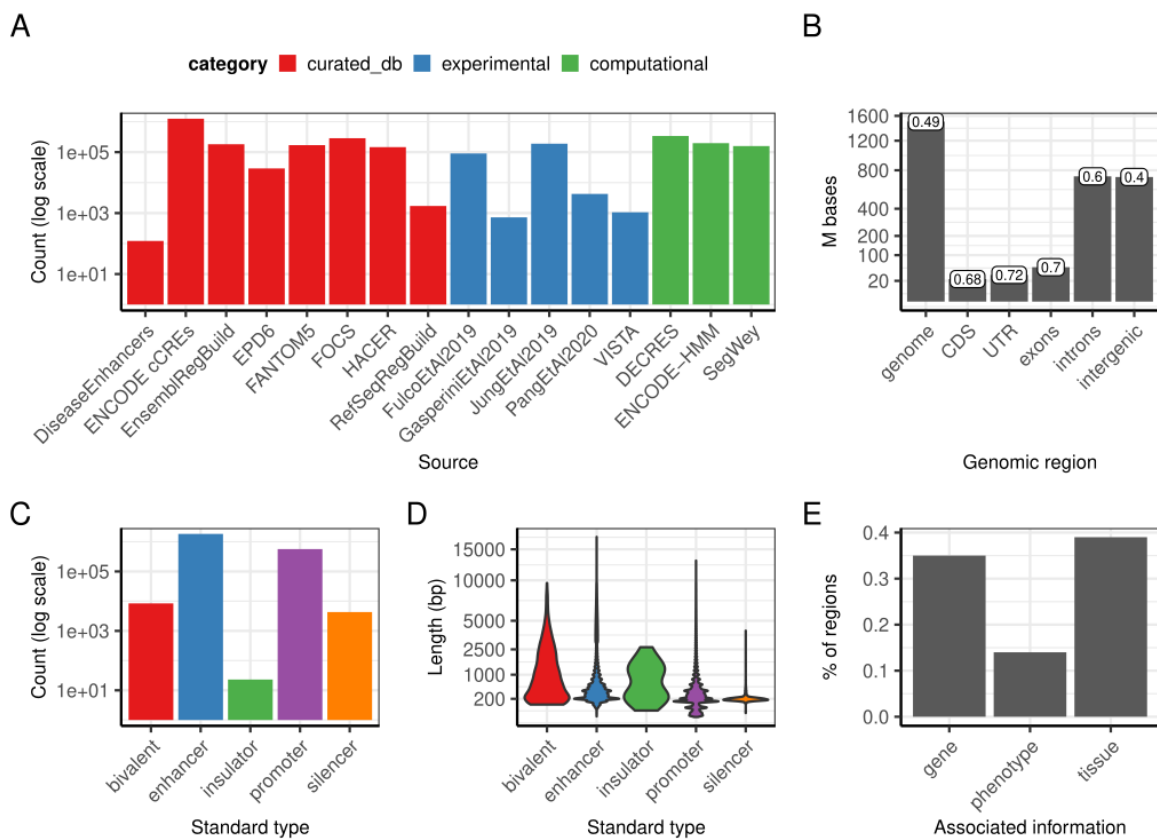


Fig. 1: Main information in the database

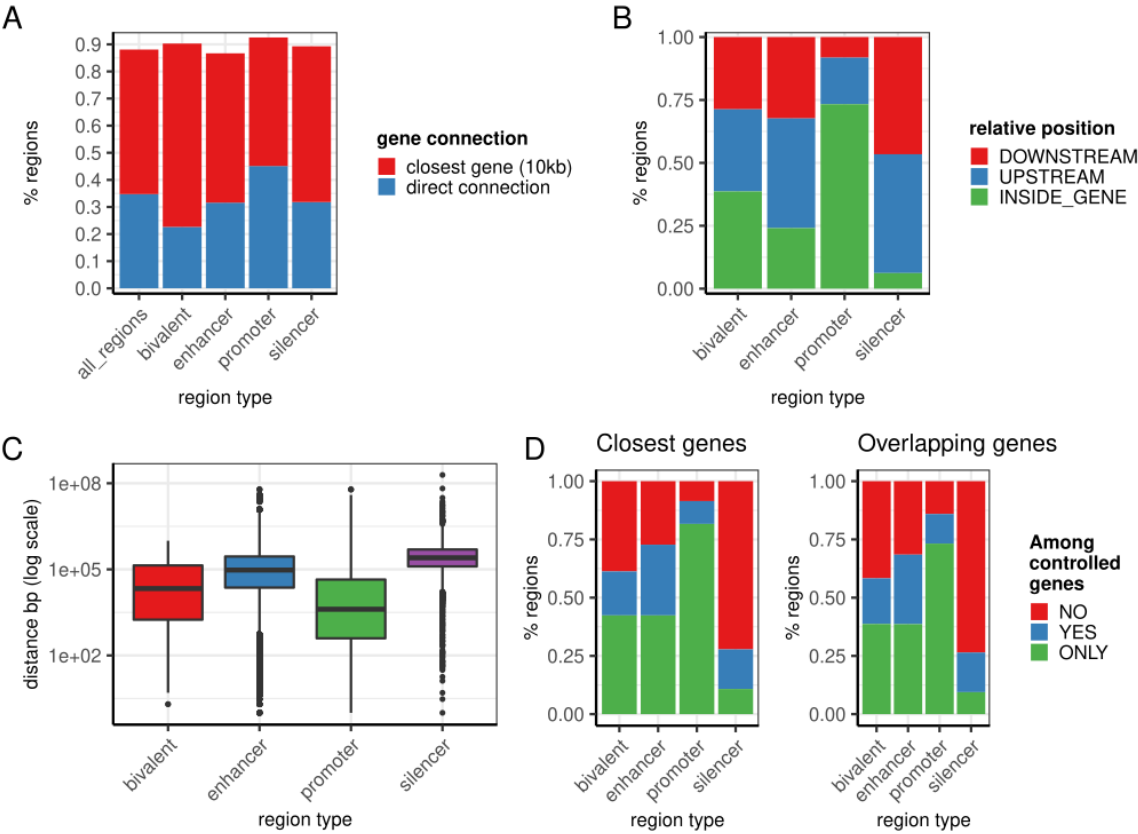


Fig. 2: Summary information on gene-region connections



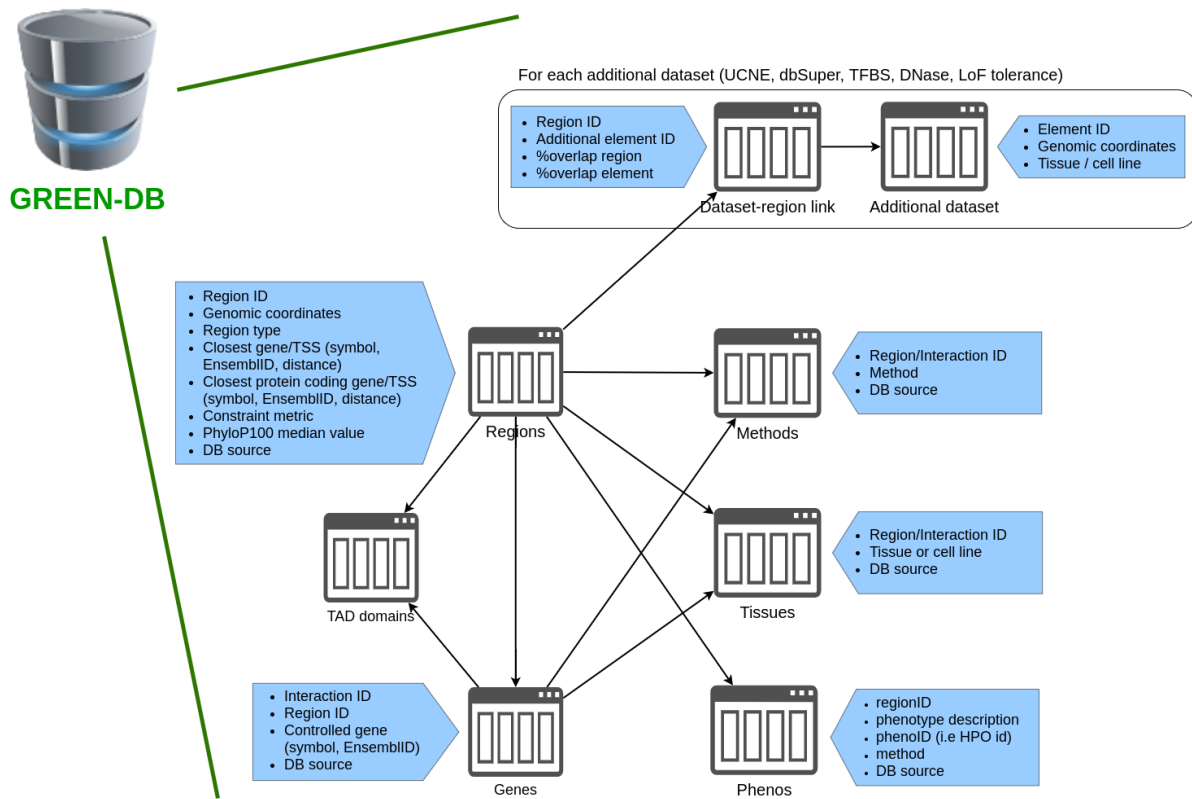


Fig. 3: A schematic representation of GREEN-DB.

#### 4.1.4 The constraint metric

For each region we calculated a constraint metric representing the tolerance to genetic variations. Constraint ranges 0-1 with higher values associated to higher level of variation constraint. Regions with high constraint values (especially > 0.9) are more likely to control essential genes and genes involved in human diseases. The constraint value is also higher for genes intolerant to LoF variants according to the gnomAD oe\_lof metric

#### 4.1.5 Summary of the building process

In GREEN-DB we collected and aggregated information from 17 different sources, including

- 8 previously published curated databases
- 6 experimental datasets from recently published articles
- predicted regulatory regions from 3 different algorithms

Four additional datasets were included to integrate region to gene / phenotype relationships. We also collected additional data useful in evaluating the regulatory role of genomic regions, including - TFBS and DNase peaks - ultraconserved non-coding elements (UCNE) - super-enhancer definitions - enhancer LoF tolerance

#### 4.1.6 Extract database tables

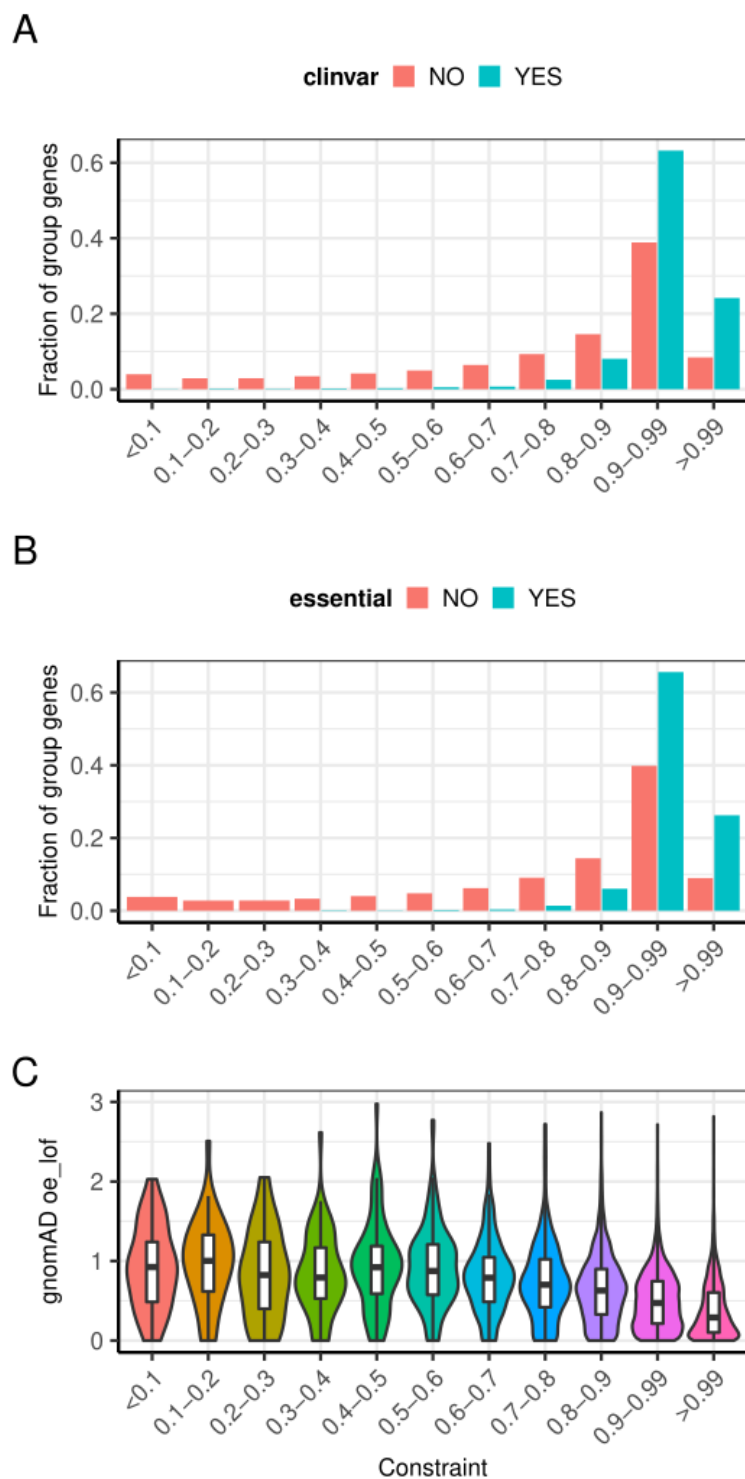


Fig. 4: Constraint values for regions associated to essential/pathogenic genes

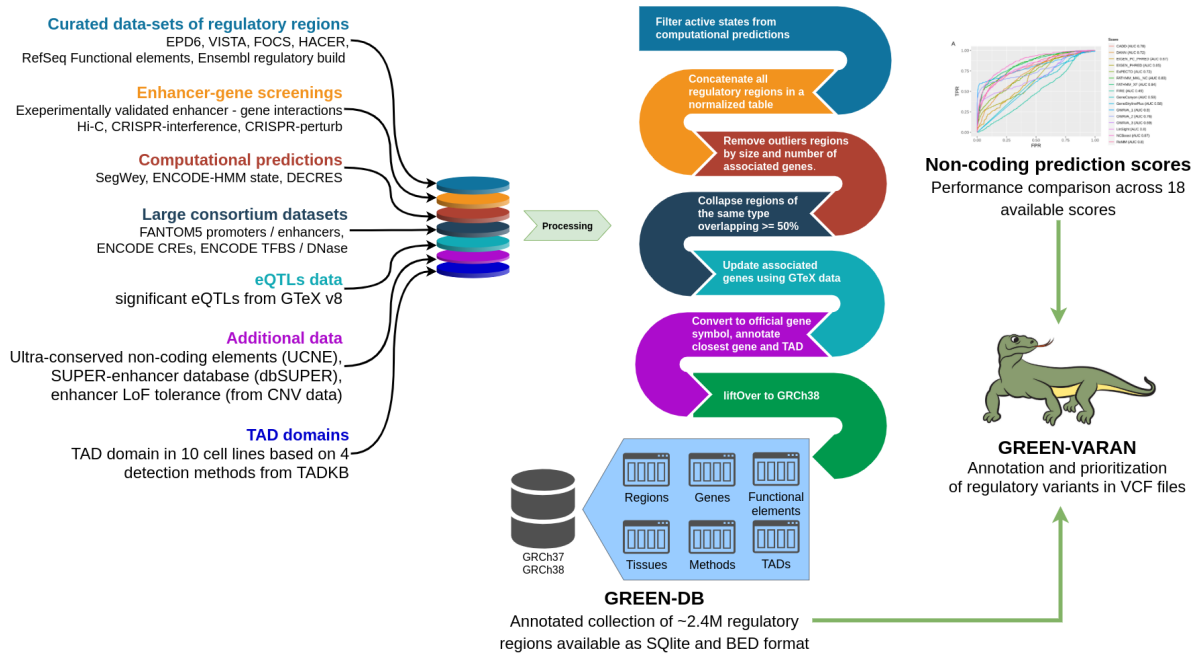


Fig. 5: Summary of the GREEN-DB building process

## Using bash

You can extract all tables of the database to tab-separated tables using a bash script. In the following example the db file is provided as argument and all tables are saved as .tsv files in the present folder

```
dbfile=$1

# obtains all data tables from database
TS=`sqlite3 $1 "SELECT tbl_name FROM sqlite_master WHERE type='table' and tbl_name_
↪not like 'sqlite_%';"`

# exports each table to tsv
for T in $TS; do
sqlite3 $1 <<!.
.headers on
.mode tabs
.output $T.tsv
select * from $T;
!
done
```

## Using R

You can extract tables from the database in R using the RSQLite package. In the example below we extract all tables to data frames in a named list (databases)

```
library("RSQLite")

## connect to the SQLite database
con <- dbConnect(drv=RSQLite::SQLite(), dbname="SQLite/RegulatoryRegions.db")
```

(continues on next page)

(continued from previous page)

```
## list all data tables
tables <- dbListTables(con)

## create a data.frame for each table
for (i in seq(along=tables)) {
  dbtables[[tables[i]]] <- dbGetQuery(conn=con, statement=paste("SELECT *
↳FROM '", tables[[i]], "'", sep=""))
}
```

## 4.2 GREEN-VARAN tool set

### Genomic Regulatory Elements ENcyclopedia VARIant ANnotation

Annotate variants in a VCF using GREEN-DB to provide information on non-coding regulatory variants and the controlled genes. Additionally perform prioritization summing up evidences of regulatory impact from GREENDB, population AF, functional regions and prediction scores

#### 4.2.1 Installation

##### 1. Get the tool binary from the repository

The easiest way to run GREEN-VARAN is to download the pre-compiled binaries from the latest release at <https://github.com/edg1983/GREEN-VARAN>

##### 2. Compile the tool

Alternatively, you can clone the repository `git clone https://github.com/edg1983/GREEN-VARAN.git`

And then compile the greenvaran using Nim compiler (<https://nim-lang.org/>). GREEN-VARAN requires - nim >= 0.10 - hts-nim >= 0.3.4 - argparse 0.10.1

If you have Singularity installed, you can use the script `nim_compile.sh` to create a static binary with no dependencies This uses musl-hts-nim as described in hts-nim repository (see <https://github.com/brentp/hts-nim#static-binary-with-singularity>)

#### 4.2.2 Get GREEN-DB files

To perform annotations with GREEN-VARAN you will need the GREEN-DB bed files for your genome build. You can download the GREEN-DB BED file for GRCh37 or GRCh38 from <https://zenodo.org/record/5636209>

The complete SQLite database is also available from the same repository

#### 4.2.3 GREEN-VARAN Nextflow workflow

We also provide a Nextflow workflow that can be used to automate VCF annotation and resource download. Given a small variants VCF annotated for gene consequences using snpEff or bcftools the workflow can be used to - automatically add functional regions annotations and non-coding prediction scores - perform greenvaran prioritization

Missing datasets will be downloaded automatically during the process. See the dedicated page for more usage information

## 4.2.4 Singularity

The tool binaries should work on most linux based system. In case you have any issue, we also provide GREEN-VARAN as Singularity image (tested on singularity >= 3.2). A Singularity recipe is included in the repository or you can pull the image from Singularity Library using

```
singularity pull library://edg1983/greenvaran/greenvaran:latest
```

See GREEN-VARAN usage for more details Usage #####

The image contains both greenvaran and greendb\_query tools. The general usage is:

```
singularity run \
greenvaran.sif \
tool_name [tool arguments]
```

### Bind specific folders for resources or data

The tool need access to input VCF file, required GREEN-DB bed file and config files so remember to bind the corresponding locations in the container

See the following example where we use the current working directory for input/output, while other files are located in the default config / resources folder within greenvaran folder. In the example we use GRCh38 genome build

```
singularity run \
--bind /greenvaran_path/resources/GRCh38:/db_files \
--bind /greenvaran_path/config:/config_files \
--bind ${PWD}:/data \
greenvaran.sif \
greenvaran -i /data/input.vcf.gz \
-o /data/output.vcf.gz \
--db /db_files/GRCh38_GREEN-DB.bed.gz \
--dbschema /config_files/greendb_schema_v2.5.json
--config /config_files/prioritize_smallvars.json
[additional tool arguments]
```

## 4.2.5 Single tools usage

The GREEN-VARAN tool set includes 2 main tools to annotate variants and interact with GREEN-DB.

1. **greenvaran** Perform annotation on small variants or structural variants VCF. Provides prioritization of regulatory variants summing up evidences of impact from GREENDB, population AF, functional regions and prediction scores. Variants can also be tagged based on a list of genes of interest. Finally, the tool can update standard gene consequence in ANN or BCQS fields to reflect regulated genes.
2. **greendb\_query** Assists in querying the GREEN-DB. Given a list of region IDs, a list of variants or a table of variants and relevant GREENDB regions the tool generates a set of tables containing detailed information on the regions of interest, region-gene connections, functional regions and tissues.

For detailed instruction on the single tools usage please refer to the corresponding page

## GREEN-VARAN tool usage

GREEN-VARAN performs annotation of small variants or structural variants VCF adding information on potential regulatory variants from GREEN-DB. Especially it can annotate possible controlled genes and a prioritization level (this latter need the presence of some additional annotations, see below) It provides also ability to tag variants linked to genes of interest and update existing gene-level annotations from SnpEff or bcftools.

### Basic usage

```
greenvaran [run mode] [options]
```

The running mode can be one of:

- **smallvars** In this mode the tool will perform annotation for a small variants VCF. It will annotate variants with information on the possible regulatory role based on GREENDB and eventually provide prioritization levels
- **sv** In this mode the tool will perform annotation for a structural variants VCF. Capability in this case is limited to annotation of overlapping GREENDB regions and controlled genes. No prioritization is provided
- **querytab** This mode is a convenient way to automatically prepare input table to be used with the query tool to extract detailed information from GREENDB database.
- **version** Print the tool version

**NB.** To perform prioritization of small variants some additional annotation fields are expected in the input VCF, see the prioritization section below. By default, when these information are not present the prioritization level will be set to zero for all annotated variants. We also provide pre-processed datasets and Nextflow workflow to automate the whole process (see #TODO nextflow workflow page).

### Command line options

#### smallvars and sv shared options

- i, --invcf INVCF** path to indexed input vcf.gz/bcf.
- o, --outvcf OUTVCF** output vcf / vcf.gz file
- d, --db DB** GREEN-DB bed.gz file for your build (see download section)
- s, --dbschema DBSCHEMA** json file containig greendb column mapping  
A default configuration for GREENDB v2.5 is available in config folder
- u, --noupdate** do not update ANN / BCSQ field in the input VCF
- f, --filter** filter instead of annotate. Only variants with greendb overlap will be written.  
If `--genes` is active, the output will contain only variants connected to the input genes of interest
- m, --impact IMPACT** Which impact to assign when updating snpEff field  
Possible values: [HIGH, MODERATE, LOW, MODIFIER] (default: MODIFIER)
- chrom CHROM** Annotate only for a specific chromosome  
Useful to parallelize across chromosomes
- g, --genes GENES** Gene symbols for genes of interest, variants connected to those will be flagged with greendb\_VOI tag

This can be a comma-separated list or a text file listing genes one per line

**--connection CONNECTION** Region-gene connections accepted for annotation  
Possible values: [all, closest, annotated] (default: all)

**--log LOG** Log file. Default is greenvaran\_[now].log

### sv specific options

**-p, --padding PADDING** Value to add on each side of BND/INS, this override the CIPOS when set

**--cipos CIPOS** INFO field listing the confidence interval around breakpoints  
It is expected to have 2 comma-separated values (default: CIPOS)

**-t, --minoverlap MINOVERLAP** Min fraction of GREENDB region to be overlapped by a SV  
(default: 0.000001)

**-b, --minbp MINBP** Min number of bases of GREENDB region to be overlapped by a SV (default: 1)

### smallvars specific options

**-c, --config CONFIG** json config file for prioritization  
A default configuration for the four level described in the paper is provided in config folder

**-p, --permissive** Perform prioritization even if one of the INFO fields required by prioritization config is missing  
By default, when one of the expected fields is not defined in the header, the prioritization is disabled and all variants will get level zero

## Annotations added by GREEN-VARAN

### INFO fields

Fields in the following table are added to INFO fields by GREEN-VARAN. greendb\_level will be added only for small variants

Annotation tag	Data type	Description
greendb_id	String	Comma-separated list of GREEN-DB IDs identifying the regions that overlap this variant
greendb_stdtype	String	Comma-separated list of standard region types as annotated in GREEN-DB for regions overlapping the variant
greendb_dbsource	String	Comma-separated list of data sources as annotated in GREEN-DB for regions overlapping the variant
greendb_level	Integer	Variant prioritization level computed by GREEN-VARAN. See Prioritization section below
greendb_constraint	Float	The maximum constraint value across GREEN-DB regions overlapping the variant
greendb_genes	String	Possibly controlled genes for regulatory regions overlapping this variant
greendb_VOI	Flag	When <code>--genes</code> option is active this flag is set when any of the input genes is among the possibly controlled genes for overlapping regulatory regions.

## Updated gene consequences

By default, GREEN-VARAN update gene consequences in the SnpEff ANN field or the bcftools BCSQ if one is present in the input VCF file. In this way the annotation can be processed by most downstream tools evaluating segregation. If none is found, GREEN-VARAN will create a new ANN field. To switch off gene consequence update use the `--noupdate` option.

Here the tool will add one a new consequence for each possibly controlled genes, limited by the `--connection` option. The new consequence will follow standard format according to SnpEff or bcftools and have MODIFIER impact by default. This can be adjusted using the `--impact` option. The gene effect will be set according to the GREEN-DB region type, adding 5 new terms: *bivalent*, *enhancer*, *insulator*, *promoter*, *silencer*.

Example ANN / BCSQ field added by GREEN-VARAN.

```
ANN=C|enhancer|MODIFIER|GeneA| | | | | | | | | |
BCSQ=enhancer|GeneA| |
```

## Prioritization of small variants

GREEN-VARAN will consider GREEN-DB annotations, additional functional regions and non-coding impact prediction scores to provide a prioritization level for each annotated variant. This level is annotated under `greenvara_level` tag in the INFO field. This field is an integer from 0 to N which summarize evidences supporting a regulatory impact for the variant. Higher values are associated to a higher probability of regulatory impact.

**NB.** You need the following INFO fields in your input VCF to run prioritization mode as described in the GREEN-DB manuscript using the default config provided.

1. `gnomAD_AF`, `gnomAD_AF_nfe` float values describing global and NFE population AF from gnomAD
2. `ncER`, `FATHMM-MKL` and `ReMM` float values providing scores predictions
3. `TFBS`, `DNase` and `UCNE` flags describing overlap with additional functional regions

This configuration resembles the four levels prioritization described in the GREEN-DB manuscript. Note that the exact names of these annotations and the score thresholds are defined in the json file passed to `--config` options.

The following table summarizes the four prioritization levels defined in the manuscript and this is the default behaviour you will obtain using the default config file and the default option `--prioritization_strategy levels`



Level	Description
1	Rare variant (population AF < 1%) overlapping one of GREEN-DB regions
2	Level 1 criteria and overlap at least one functional element among transcription factors binding sites (TFBS), DNase peaks, ultra conserved elements (UCNE)
3	Level 2 criteria and prediction score value above the suggested FDR50 threshold for at least one among ncER, FATHMM MKL, ReMM
4	Level 3 criteria and region constraint value greater or equal 0.7

Alternatively, you can chose a “pile-up” approach setting *-prioritization\_strategy pileup* which simply sum evidences across levels.

This means that the criteria described above are tested independently and the level reported is increased by one for each satisfied criteria.

### Personalize the prioritization schema

The prioritization schema is defined in a config json file. The default is provided in the config folder. An example of expected file structure is reported below

```
{
  "af": ["gnomAD_AF", "gnomAD_AF_nfe"],
  "maxaf": 0.01,
  "regions": ["TFBS", "DNase", "UCNE"],
  "scores": {
    "FATHMM_MKLNC": 0.908,
    "ncER": 98.6,
    "ReMM": 0.963
  },
  "constraint": 0.7,
  "more_regions": [],
  "more_values": {}
}
```

Sections definitions:

1. af: INFO fields containing AF annotations. The tool will consider the max value across all these
2. maxaf: if the max value across af fields is below this, the variant get +1 point
3. regions: INFO fields for overlapping regions. If any of these is set, the variant get +1 point
4. scores: series of key, value pairs. If any of key value is above the configured value, the variant get +1 point
5. constraint: if the max constraint value across overlapping GREEN-DB regions is above this value, the variant get +1 point
6. more\_regions: any additional INFO fields representing overlap with custom regions. The variant get +1 point for each positive overlap
7. more\_values: series of key, value pairs. The variant get +1 point fro each key value above the configured value

**NB.** more\_regions and more\_values must always been present. Leave them empty like in the example above if you don't want to configure any custom value.

**NB2.** INFO fields specified by af, scores and more\_values are expected to be float, while those specified by regions and more\_regions are expected as flags.

### structural variants annotations

The annotation of structural variants is based on overlap with the regulatory regions defined in GREEN-DB. This is treated differently according to the SV type:

- For **DEL, DUP, INV** an interval is constructed based on position field and the END info field from INFO. When END is missing, the tool will try to use SVLEN instead. If none is not found the variant is not annotated. The user can then set a minimum level of overlap as either overlap fraction (`--minoverlap`) or N bp overlap (`--minbp`). A GREEN-DB region is added to annotation only if its overlapping porting is larger or equal to both threshold
- For **INS and BND**, an interval is constructed using the position and the coordinates in the CIPOS field (an alternative field can be set using `--cipos`). This is done since INS and BND are often represented as single positions in structural variants VCF. Alternatively, the user can provide a padding values using `--padding` and this value will be added around position. For these kind of variants any overlapping GREEN-DB region will be reported, disregarding the overlap thresholds

### Singularity

The tool binaries should work on most linux based system. In case you have any issue, we also provide GREEN-VARAN as Singularity image (tested on singularity  $\geq 3.2$ ). A Singularity recipe is included in the repository or you can pull the image from Singularity Library using

```
singularity pull library://edg1983/greenvaran/greenvaran:latest
```

### Usage

The image contains both greenvaran and greendb\_query tools. The general usage is:

```
singularity exec \  
greenvaran.sif \  
tool_name [tool arguments]
```

### Bind specific folders for resources or data

The tool needs access to input VCF file, required GREEN-DB bed file and config files so remember to bind the corresponding locations in the container

See the following example where we use the current working directory for input/output, while other files are located in the default config / resources folder within greenvaran folder. In the example we use GRCh38 genome build

```
singularity exec \  
--bind /greenvaran_path/resources/GRCh38:/db_files \  
--bind /greenvaran_path/config:/config_files \  
--bind ${PWD}:/data \  
greenvaran.sif \  
greenvaran -i /data/input.vcf.gz \  
-o /data/output.vcf.gz \  
--db /db_files/GRCh38_GREEN-DB.bed.gz \  
--dbschema /config_files/greendb_schema_v2.5.json \  
--config /config_files/prioritize_smallvars.json \  
[additional tool arguments]
```

## Example usage

### small variants test

```
greenvaran smallvars \
--invvcf test/VCF/GRCh38.test.smallvars.vcf.gz \
--outvcf test/out/smallvars.annotated.vcf.gz \
--config config/prioritize_smallvars.json \
--dbschema config/greendb_schema_v2.5.json \
--db resources/GRCh38/GRCh38_GREEN-DB.bed.gz \
--genes test/VCF/genes_list_example.txt
```

### structural variants test

```
greenvaran sv \
--invvcf test/VCF/GRCh38.test.SV.vcf.gz \
--outvcf test/out/SV.annotated.vcf.gz \
--dbschema config/greendb_schema_v2.5.json \
--db resources/GRCh38/GRCh38_GREEN-DB.bed.gz \
--minbp 10
```

### greendb\_query tool usage

greendb\_query assists in querying the GREEN-DB database. Given a list of region IDs, variant IDs or a table or variant and relevant regions, the tool generates a set of tables containing detailed information on the regions of interest, overlap with additional supporting regions (TFBS, DNase HS peaks, UCNE, dbSuper), gene-region connections, tissue of activity and associated phenotypes.

```
greendb_query [-h] (-v VARIDS | -r REGIDS | -t TABLE) -o OUTPREFIX -g
                {GRCh37,GRCh38} --db GREENDB [--logfile LOGFILE]
```

## Possible inputs

The tools allows to query GREEN-DB using 3 different type of inputs. Only one type of input can be specified.

### 1. List of regions (-r)

If you are simply interested in detailed information on a list of regions, you can use the -r input. This argument accepts a comma-separated list of regions (like ID1, ID2) or a text file with one region ID per line.

### 2. VCF file (-v)

If you have a small list of variants for which you want to extract overlapping regulatory regions, you can input a them as a comma-separated list of variant IDs (like var1, var2) or a text file with one variant ID per line A variant ID has the format chrom\_pos\_ref\_alt

### 3. Variant-regions table (-t)

If you have a list of variants of interest for which you know the relevant GREEN-DB region IDs you can query the DB directly providing a tab separated text file with **no header** and 2 columns:

- column 1: variant ID in the format chrom\_pos\_ref\_alt
- column 2: comma-separated list of region IDs overlapping the variant

This table can be generated automatically from a VCF annotated with greenvaran by using `greenvaran querytab`

### Output tables

The tool will generate 6 tables with the provided prefix. Some table may be empty if the corresponding information is missing. Output tables structure is described below

#### regions

Details on the regions of interest

1. **regionID**: GREEN-DB region ID
- 2-4. **chrom, start, stop**: genomic location of the region
5. **type**: region type as extracted from the source dataset
6. **std\_type**: one of the 5 main region types (enhancer, promoter, silencer, bivalent, insulator)
7. **DB\_source**: comma-separated list of sources supporting the region
8. **PhyloP100\_median**: median PhyloP100 conservation value across the region
9. **constraint\_pct**: constraint metric. range 0-1 with higher values equals more intolerant to variants
10. **controlled\_gene**: comma-separated list of gene symbols for controlled genes with experimental support
- 11-13. **closestGene\_symbol, \_ensg, \_dist**: symbol, ensembl IDs and distance for the closest gene
14. **cell\_or\_tissues**: comma-separated list of cell types and tissues where the region is active
15. **detection\_method**: comma-separated list of methods supporting this regions
16. **phenotype**: comma-separated list of phenotypes eventually associated to this region

#### gene\_details

Details on the controlled genes, reporting the tissue where the gene-region interaction is detected

1. **regionID**: GREEN-DB region ID
- 2-4. **chrom, start, stop**: genomic location of the region
5. **std\_type**: one of the 5 main region types (enhancer, promoter, silencer, bivalent, insulator)
6. **controlled\_gene**: gene symbol for controlled gene
7. **detection\_method**: method supporting this interaction
8. **tissue\_of\_interaction**: comma-separated list of cell types and tissues where this region-gene interaction is detected
9. **same\_TAD**: 0/1 value indicating if the reported interaction occurs in the same TAD according to TADKB

## pheno\_details

Details on the phenotypes potentially associated with the regions of interest

1. **regionID**: GREEN-DB region ID
- 2-4. **chrom, start, stop**: genomic location of the region
5. **std\_type**: one of the 5 main region types (enhancer, promoter, silencer, bivalent, insulator)
6. **phenotype**: phenotype eventually associated to this region
7. **detection method**: method supporting this association. Note that when the method is GENE2HPO this means that the phenotype is inferred from HPOs associated to the controlled gene(s)
8. **DB source**: source supporting this association

## DNase, dbSuper, TFBS, UCNE

For each of the 4 functional elements a table is generated with details on each element overlapping the region(s) / variant(s) of interest.

1. **regionID**: GREEN-DB region ID
- 2-4. **dataset\_chrom, \_start, \_stop**: genomic location of the functional element
5. **dataset\_ID**: database ID of the functional element
6. **dataset\_cell\_or\_tissue**: comma-separated list of cell types and tissues where the element is detected  
cell and tissue information is not available for UCNE

## Variant(s) of interest

When the input contains variants of interest (-t, -v), an additional column is added to all tables. A region or element is reported in the output only if it overlaps with one of the variants.

**var\_id**: comma-separated list of variant ID(s) (chrom\_pos\_ref\_alt) of the variant(s) overlapping this feature

## Arguments list

- v VARID, --vcf VARID** Comma separated list of variant IDs or file with a list of variant IDs
- r REGIDS, --regIDs REGIDS** Comma separated list of region IDs or file with a list of region IDs
- t TABLE, --table TABLE** Tab-separated file with  
col1 (chr\_pos\_ref\_alt)  
col2 comma-separated list of region IDs
- o OUTPREFIX, --outprefix OUTPREFIX** Prefix for output files
- g BUILD, --genome BUILD** Possible values: {GRCh37, GRCh38}  
Genome build for the query
- db GREENDB** Location of the GREEN-DB SQLite database file (.db)
- logfile LOGFILE** Custom location for the log file

## GREEN-VARAN workflow

To perform small variants prioritization as described in the GREEN-DB manuscript, GREEN-VARAN need some annotations to be already present in your input VCF (see [Prioritization of small variants](#))

This Nextflow workflow automate the whole process annotating additional information and then performing greenevaran annotation. The workflow is tested on Nextflow >=v20.10

## Usage

The typical usage scenario start with a VCF file already containing gene consequences annotations from SnpEff or bcftools. Then from the GREEN-VARAN tool main folder you can perform all annotations using the following command. This will add a minimum set of information to you VCF including:

- population allele AF from gnomAD genomes v3.1.1 (GRCh38) or v2.1.1 (GRCh37)
- functional regions overlaps for TFBS, DNase peaks and UCNE
- prediction score values for ncER, FATHMM, ReMM
- GREEN-DB information on regulatory variants with prioritization levels

```
nextflow workflow/main.nf \  
  -profile local \  
  --input input_file.vcf.gz \  
  --build GRCh38 \  
  --out results \  
  --scores best \  
  --regions best \  
  --AF \  
  --greenvaran_config config/prioritize_smallvars.json \  
  --greenvaran_dbschema config/greendb_schema_v2.5.json
```

If requested annotation files are missing, they will be automatically downloaded in the default location (resources folder within the main GREEN-VARAN folder)

Note that --input can accept multiple vcf.gz files using a pattern like inputdir/\*.vcf.gz

## Add additional custom annotations

If you have additional custom annotation you want to add to your VCF before greenvaran processing they can be configured in a .toml and then you can pass this file to the workflow using --anno\_toml.

A toml file is a annotation configuration file used by the vcfanno tool and is described in ‘[vcfanno repository<https://github.com/brentp/vcfanno>](https://github.com/brentp/vcfanno)’\_

A minimal example is reported below

```
[[annotation]]  
file="ExAC.vcf" #source file  
fields = ["AF", "AF_nfe"] #INFO fields to be extracted from source  
ops=["self", "max"] #How to treat source values  
names=["exac_af", "exac_af_nfe_max"] #names used in the annotated file  
  
[[annotation]]  
file="regions_score.bed.gz"  
columns = [4, 5] #When using a BED or TSV files you can refer to values by col index
```

(continues on next page)

(continued from previous page)

```
names=["regions_ids", "score_max"]
ops=["uniq", "max"]
```

## Resources

To perform annotations GREEN-VARAN Nextflow workflow requires a series of supporting files. By default, various resources are expected in the `resources` folder within the main tool folder. You pass an alternative resource folder using `--resource_folder` option, but the same structure is expected in this folder

The expected folder structure is as follows

```
.
|-- SQLite
|   `-- GREEN-DB_v2.5.db
|-- GRCh37
|   `-- BED / TSV files used for GRCh37 genome build
`-- GRCh38
    `-- BED / TSV files used for GRCh38 genome build
```

Use the `--list_data` option to see the full list of available resources and the expected path for each one.

## Automated download

A supporting workflow is provided to automate data download for all resources included in the GREEN-DB collection. You can list the available resources and their resulting download location using

```
nextflow workflow/download.nf --list_data
```

The recommended set of annotations can be downloaded to the default location using the following command or you can set an alternative resource folder using `--resource_folder` option

```
nextflow workflow/download.nf \
-profile local \
--scores best \
--regions best \
--AF \
--db
```

Otherwise, single files are available for download from Zenodo repository and all file locations are listed in the `GREENDB_collection.txt` file under `resources` folder.

## Workflow configuration

The workflow has pre-configured profiles for most popular schedulers (sge, lsf, slurm) and also a local profile (local). These profiles determine how many download jobs can be submitted concurrently and the number of threads used for annotation.

You can activate the desired profile using `-profile` argument when launching the workflow

**NB.** You need to update the queue name parameter to reflect your local settings, see how to edit the config below

The default settings for each profile are reported below:

### Editing the profile configuration

To adjust the configuration you need to edit the `nextflow.config` file in the workflow folder

The main parameters you may need to adjust are - `ncpus`: this controls the number of threads request for annotation - `max_local_jobs`: this controls the max number of concurrent jobs submitted in local profile (when not submitting job to a scheduler) - `queue`: this is the name of the queue to be used when submitting jobs

### Editing the annotation file schema

The annotation file schema contain the expected files names, repositories and annotation sources. In case you need to adjust this you can modify the `resources.conf` file located in `workflow/config` in the GREEN-VARAN folder.

### Available parameters for main workflow

- input INPUT\_VCF** Input VCF file(s), compressed and indexed  
You can input multiple files from a folder using quotes like `--input mypath/*.vcf.gz`
- build GENOME\_BUILD** Genome build  
Accepted values: [GRCh37, GRCh38]
- out output\_dir** Output directory
- scores SCORE\_NAME** Annotate prediction scores  
Accepted values: [best, all, name]  
best: annotate ncER, FATHMM-MKL, ReMM  
all: annotate all scores  
name: annotate only the specified score(s) (can be comma-separated list)
- regions REGIONS\_NAME** Annotate functional regions  
Accepted values: [best, all, name]  
best: annotate TFBS, DNase, UCNE  
all: annotate all regions  
name: annotate only the specified region(s) (can be comma-separated list)
- AF** Annotate global AF from gnomAD genomes
- greenvaran\_config JSON\_FILE** A json config file for GREEN-VARAN tool
- greenvaran\_dbschema JSON\_FILE** A json db schema file for GREEN-VARAN tool
- nochr** Chromosome names in the input file do not have chr prefix
- prioritization\_strategy** Set prioritization strategy [levels, pileup]
- resource\_folder** Specify a custom folder for the annotation files  
Default is the resources folder in GREEN-VARAN main folder
- anno\_tom TOML\_FILE** A custom toml annotation config file.  
This file is a toml file as specified by `vcfanno` tool  
This will be added to other annotations defined with scores, regions and AF.
- list\_data** Output the list of available scores / regions and the expected paths



## 4.3 Download resources

### 4.3.1 greenvaran tool

The greenvaran annotation tool only need the GREEN-DB BED file and index for your genome build available from <https://zenodo.org/record/5636209>

### 4.3.2 greendb\_query tool

The greendb query tool only need the GREEN-DB SQLite file (.db.gz) available from <https://zenodo.org/record/5636209> Remember to decompress this before use

### 4.3.3 GREEN-VARAN workflow

To perform annotations GREEN-VARAN Nextflow workflow requires a series of supporting files. By default, various resources are expected in the `resources` folder within the main tool folder. If you pass an alternative resource folder using `--resource_folder` option, the same structure is expected in this folder The expected folder structure is as follows and the expected file names are those listed in the Zenodo repository table below

```
.
|-- SQLite
|   `-- GREEN-DB_v2.5.db
|-- GRCh37
|   `-- BED / TSV files used for GRCh37 genome build
`-- GRCh38
    `-- BED / TSV files used for GRCh38 genome build
```

When you clone the GREEN-VARAN repository you can use the Nextflow workflow `workflow/download.nf` to download files and prepare the resource folder. Use the `--list_data` option to see the full list of available resource and the expected path for each one.

Otherwise, single files are available for download from Zenodo repository

Annotation	Category	File
GRCh37_CADD	scores	<a href="https://zenodo.org/record/3956385/files/GRCh37_CADD.tsv.gz">https://zenodo.org/record/3956385/files/GRCh37_CADD.tsv.gz</a>
GRCh37_CADD	scores	<a href="https://zenodo.org/record/3956385/files/GRCh37_CADD.tsv.gz.csi">https://zenodo.org/record/3956385/files/GRCh37_CADD.tsv.gz.csi</a>
GRCh37_DANN	scores	<a href="https://zenodo.org/record/3957486/files/GRCh37_DANN.tsv.gz">https://zenodo.org/record/3957486/files/GRCh37_DANN.tsv.gz</a>
GRCh37_DANN	scores	<a href="https://zenodo.org/record/3957486/files/GRCh37_DANN.tsv.gz.csi">https://zenodo.org/record/3957486/files/GRCh37_DANN.tsv.gz.csi</a>
GRCh37_ExPECTO	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_ExPECTO.tsv.gz">https://zenodo.org/record/3956168/files/GRCh37_ExPECTO.tsv.gz</a>
GRCh37_ExPECTO	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_ExPECTO.tsv.gz.csi">https://zenodo.org/record/3956168/files/GRCh37_ExPECTO.tsv.gz.csi</a>
GRCh37_FIRE	scores	<a href="https://zenodo.org/record/3957356/files/GRCh37_FIRE.tsv.gz">https://zenodo.org/record/3957356/files/GRCh37_FIRE.tsv.gz</a>
GRCh37_FIRE	scores	<a href="https://zenodo.org/record/3957356/files/GRCh37_FIRE.tsv.gz.csi">https://zenodo.org/record/3957356/files/GRCh37_FIRE.tsv.gz.csi</a>
GRCh37_LinSight	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_LinSight.bed.gz">https://zenodo.org/record/3956168/files/GRCh37_LinSight.bed.gz</a>
GRCh37_LinSight	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_LinSight.bed.gz.csi">https://zenodo.org/record/3956168/files/GRCh37_LinSight.bed.gz.csi</a>
GRCh37_NCBoost	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_NCBoost.tsv.gz">https://zenodo.org/record/3956168/files/GRCh37_NCBoost.tsv.gz</a>
GRCh37_NCBoost	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_NCBoost.tsv.gz.csi">https://zenodo.org/record/3956168/files/GRCh37_NCBoost.tsv.gz.csi</a>
GRCh37_ReMM	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_ReMM.tsv.gz">https://zenodo.org/record/3956168/files/GRCh37_ReMM.tsv.gz</a>

Continued on next page

Table 1 – continued from previous page

Annotation	Category	File
GRCh37_ReMM	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_ReMM.tsv.gz.csi">https://zenodo.org/record/3956168/files/GRCh37_ReMM.tsv.gz.csi</a>
GRCh37_PhyloP100	scores	<a href="https://zenodo.org/record/3973181/files/GRCh37_PhyloP100.bed.gz">https://zenodo.org/record/3973181/files/GRCh37_PhyloP100.bed.gz</a>
GRCh37_PhyloP100	scores	<a href="https://zenodo.org/record/3973181/files/GRCh37_PhyloP100.bed.gz.csi">https://zenodo.org/record/3973181/files/GRCh37_PhyloP100.bed.gz.csi</a>
GRCh37_Eigen	scores	<a href="https://zenodo.org/record/3982095/files/GRCh37_Eigen.tsv.gz">https://zenodo.org/record/3982095/files/GRCh37_Eigen.tsv.gz</a>
GRCh37_Eigen	scores	<a href="https://zenodo.org/record/3982095/files/GRCh37_Eigen.tsv.gz.csi">https://zenodo.org/record/3982095/files/GRCh37_Eigen.tsv.gz.csi</a>
GRCh37_FATHMM-XF	scores	<a href="https://zenodo.org/record/3982392/files/GRCh37_FATHMM-XF_NC.tsv.gz">https://zenodo.org/record/3982392/files/GRCh37_FATHMM-XF_NC.tsv.gz</a>
GRCh37_FATHMM-XF	scores	<a href="https://zenodo.org/record/3982392/files/GRCh37_FATHMM-XF_NC.tsv.gz.csi">https://zenodo.org/record/3982392/files/GRCh37_FATHMM-XF_NC.tsv.gz.csi</a>
GRCh37_FATHMM-MKL	scores	<a href="https://zenodo.org/record/3981113/files/GRCh37_FATHMM-MKL_NC.tsv.gz">https://zenodo.org/record/3981113/files/GRCh37_FATHMM-MKL_NC.tsv.gz</a>
GRCh37_FATHMM-MKL	scores	<a href="https://zenodo.org/record/3981113/files/GRCh37_FATHMM-MKL_NC.tsv.gz.csi">https://zenodo.org/record/3981113/files/GRCh37_FATHMM-MKL_NC.tsv.gz.csi</a>
GRCh37_GWAVA	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_gwava.bed.gz">https://zenodo.org/record/3956168/files/GRCh37_gwava.bed.gz</a>
GRCh37_GWAVA	scores	<a href="https://zenodo.org/record/3956168/files/GRCh37_gwava.bed.gz.csi">https://zenodo.org/record/3956168/files/GRCh37_gwava.bed.gz.csi</a>
GRCh37_gnomAD	AF	<a href="https://zenodo.org/record/3957637/files/GRCh37_gnomad.genomes.vcf.gz">https://zenodo.org/record/3957637/files/GRCh37_gnomad.genomes.vcf.gz</a>
GRCh37_gnomAD	AF	<a href="https://zenodo.org/record/3957637/files/GRCh37_gnomad.genomes.vcf.gz.csi">https://zenodo.org/record/3957637/files/GRCh37_gnomad.genomes.vcf.gz.csi</a>
GRCh37_ncER	scores	<a href="https://zenodo.org/record/5636163/files/GRCh37_ncER_perc.bed.gz">https://zenodo.org/record/5636163/files/GRCh37_ncER_perc.bed.gz</a>
GRCh37_ncER	scores	<a href="https://zenodo.org/record/5636163/files/GRCh37_ncER_perc.bed.gz.csi">https://zenodo.org/record/5636163/files/GRCh37_ncER_perc.bed.gz.csi</a>
GRCh38_CADD	scores	<a href="https://zenodo.org/record/3956227/files/GRCh38_CADD.tsv.gz">https://zenodo.org/record/3956227/files/GRCh38_CADD.tsv.gz</a>
GRCh38_CADD	scores	<a href="https://zenodo.org/record/3956227/files/GRCh38_CADD.tsv.gz.csi">https://zenodo.org/record/3956227/files/GRCh38_CADD.tsv.gz.csi</a>
GRCh38_DANN	scores	<a href="https://zenodo.org/record/3957428/files/GRCh38_DANN.tsv.gz">https://zenodo.org/record/3957428/files/GRCh38_DANN.tsv.gz</a>
GRCh38_DANN	scores	<a href="https://zenodo.org/record/3957428/files/GRCh38_DANN.tsv.gz.csi">https://zenodo.org/record/3957428/files/GRCh38_DANN.tsv.gz.csi</a>
GRCh38_ExPECTO	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_ExPECTO.tsv.gz">https://zenodo.org/record/3955933/files/GRCh38_ExPECTO.tsv.gz</a>
GRCh38_ExPECTO	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_ExPECTO.tsv.gz.csi">https://zenodo.org/record/3955933/files/GRCh38_ExPECTO.tsv.gz.csi</a>
GRCh38_FIRE	scores	<a href="https://zenodo.org/record/3957216/files/GRCh38_FIRE.tsv.gz">https://zenodo.org/record/3957216/files/GRCh38_FIRE.tsv.gz</a>
GRCh38_FIRE	scores	<a href="https://zenodo.org/record/3957216/files/GRCh38_FIRE.tsv.gz.csi">https://zenodo.org/record/3957216/files/GRCh38_FIRE.tsv.gz.csi</a>
GRCh38_LinSight	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_LinSight.bed.gz">https://zenodo.org/record/3955933/files/GRCh38_LinSight.bed.gz</a>
GRCh38_LinSight	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_LinSight.bed.gz.csi">https://zenodo.org/record/3955933/files/GRCh38_LinSight.bed.gz.csi</a>
GRCh38_NCBoost	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_NCBoost.tsv.gz">https://zenodo.org/record/3955933/files/GRCh38_NCBoost.tsv.gz</a>
GRCh38_NCBoost	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_NCBoost.tsv.gz.csi">https://zenodo.org/record/3955933/files/GRCh38_NCBoost.tsv.gz.csi</a>
GRCh38_ReMM	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_ReMM.tsv.gz">https://zenodo.org/record/3955933/files/GRCh38_ReMM.tsv.gz</a>
GRCh38_ReMM	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_ReMM.tsv.gz.csi">https://zenodo.org/record/3955933/files/GRCh38_ReMM.tsv.gz.csi</a>
GRCh38_PhyloP100	scores	<a href="https://zenodo.org/record/3973181/files/GRCh38_PhyloP100.bed.gz">https://zenodo.org/record/3973181/files/GRCh38_PhyloP100.bed.gz</a>
GRCh38_PhyloP100	scores	<a href="https://zenodo.org/record/3973181/files/GRCh38_PhyloP100.bed.gz.csi">https://zenodo.org/record/3973181/files/GRCh38_PhyloP100.bed.gz.csi</a>
GRCh38_Eigen	scores	<a href="https://zenodo.org/record/3982182/files/GRCh38_Eigen.tsv.gz">https://zenodo.org/record/3982182/files/GRCh38_Eigen.tsv.gz</a>
GRCh38_Eigen	scores	<a href="https://zenodo.org/record/3982182/files/GRCh38_Eigen.tsv.gz.csi">https://zenodo.org/record/3982182/files/GRCh38_Eigen.tsv.gz.csi</a>

Continued on next page

Table 1 – continued from previous page

Annotation	Category	File
GRCh38_FATHMM_XF	scores	<a href="https://zenodo.org/record/3982484/files/GRCh38_FATHMM-XF_NC.tsv.gz">https://zenodo.org/record/3982484/files/GRCh38_FATHMM-XF_NC.tsv.gz</a>
GRCh38_FATHMM_XF	scores	<a href="https://zenodo.org/record/3982484/files/GRCh38_FATHMM-XF_NC.tsv.gz.csi">https://zenodo.org/record/3982484/files/GRCh38_FATHMM-XF_NC.tsv.gz.csi</a>
GRCh38_FATHMM_MKL	scores	<a href="https://zenodo.org/record/3981121/files/GRCh38_FATHMM-MKL_NC.tsv.gz">https://zenodo.org/record/3981121/files/GRCh38_FATHMM-MKL_NC.tsv.gz</a>
GRCh38_FATHMM_MKL	scores	<a href="https://zenodo.org/record/3981121/files/GRCh38_FATHMM-MKL_NC.tsv.gz.csi">https://zenodo.org/record/3981121/files/GRCh38_FATHMM-MKL_NC.tsv.gz.csi</a>
GRCh38_GWAVA	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_gwava.bed.gz">https://zenodo.org/record/3955933/files/GRCh38_gwava.bed.gz</a>
GRCh38_GWAVA	scores	<a href="https://zenodo.org/record/3955933/files/GRCh38_gwava.bed.gz.csi">https://zenodo.org/record/3955933/files/GRCh38_gwava.bed.gz.csi</a>
GRCh38_ncER	scores	<a href="https://zenodo.org/record/5636163/files/GRCh38_ncER_perc.bed.gz">https://zenodo.org/record/5636163/files/GRCh38_ncER_perc.bed.gz</a>
GRCh38_ncER	scores	<a href="https://zenodo.org/record/5636163/files/GRCh38_ncER_perc.bed.gz.csi">https://zenodo.org/record/5636163/files/GRCh38_ncER_perc.bed.gz.csi</a>
GRCh37_TFBS	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_TFBS.merged.bed.gz">https://zenodo.org/record/5705936/files/GRCh37_TFBS.merged.bed.gz</a>
GRCh37_TFBS	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_TFBS.merged.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh37_TFBS.merged.bed.gz.csi</a>
GRCh37_DNase	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_DNase.merged.bed.gz">https://zenodo.org/record/5705936/files/GRCh37_DNase.merged.bed.gz</a>
GRCh37_DNase	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_DNase.merged.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh37_DNase.merged.bed.gz.csi</a>
GRCh37_UCNE	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_UCNE.bed.gz">https://zenodo.org/record/5705936/files/GRCh37_UCNE.bed.gz</a>
GRCh37_UCNE	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_UCNE.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh37_UCNE.bed.gz.csi</a>
GRCh37_dbSuper	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_dbSuper.bed.gz">https://zenodo.org/record/5705936/files/GRCh37_dbSuper.bed.gz</a>
GRCh37_dbSuper	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_dbSuper.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh37_dbSuper.bed.gz.csi</a>
GRCh37_TAD	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_TAD.bed.gz">https://zenodo.org/record/5705936/files/GRCh37_TAD.bed.gz</a>
GRCh37_TAD	regions	<a href="https://zenodo.org/record/5705936/files/GRCh37_TAD.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh37_TAD.bed.gz.csi</a>
GRCh38_TFBS	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_TFBS.merged.bed.gz">https://zenodo.org/record/5705936/files/GRCh38_TFBS.merged.bed.gz</a>
GRCh38_TFBS	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_TFBS.merged.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh38_TFBS.merged.bed.gz.csi</a>
GRCh38_DNase	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_DNase.merged.bed.gz">https://zenodo.org/record/5705936/files/GRCh38_DNase.merged.bed.gz</a>
GRCh38_DNase	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_DNase.merged.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh38_DNase.merged.bed.gz.csi</a>
GRCh38_UCNE	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_UCNE.bed.gz">https://zenodo.org/record/5705936/files/GRCh38_UCNE.bed.gz</a>
GRCh38_UCNE	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_UCNE.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh38_UCNE.bed.gz.csi</a>
GRCh38_dbSuper	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_dbSuper.bed.gz">https://zenodo.org/record/5705936/files/GRCh38_dbSuper.bed.gz</a>
GRCh38_dbSuper	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_dbSuper.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh38_dbSuper.bed.gz.csi</a>
GRCh38_TAD	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_TAD.bed.gz">https://zenodo.org/record/5705936/files/GRCh38_TAD.bed.gz</a>
GRCh38_TAD	regions	<a href="https://zenodo.org/record/5705936/files/GRCh38_TAD.bed.gz.csi">https://zenodo.org/record/5705936/files/GRCh38_TAD.bed.gz.csi</a>
GRCh38_gnomAD	AF	<a href="https://zenodo.org/record/3957637/files/GRCh38_gnomad.genomes.vcf.gz">https://zenodo.org/record/3957637/files/GRCh38_gnomad.genomes.vcf.gz</a>

Continued on next page

Table 1 – continued from previous page

Annotation	Category	File
GRCh38_gnomAD	AF	<a href="https://zenodo.org/record/3957637/files/GRCh38_gnomad.genomes.vcf.gz.csi">https://zenodo.org/record/3957637/files/GRCh38_gnomad.genomes.vcf.gz.csi</a>
SV_annotations	SV_annotations	<a href="https://zenodo.org/record/3970785/files/SV_annotations.tar.gz">https://zenodo.org/record/3970785/files/SV_annotations.tar.gz</a>
GRCh37_GREENDB	GREENDB_bed	<a href="https://zenodo.org/record/5636209/files/GRCh37_GREEN-DB.bed.gz">https://zenodo.org/record/5636209/files/GRCh37_GREEN-DB.bed.gz</a>
GRCh37_GREENDB	GREENDB_bed	<a href="https://zenodo.org/record/5636209/files/GRCh37_GREEN-DB.bed.gz.csi">https://zenodo.org/record/5636209/files/GRCh37_GREEN-DB.bed.gz.csi</a>
GRCh38_GREENDB	GREENDB_bed	<a href="https://zenodo.org/record/5636209/files/GRCh38_GREEN-DB.bed.gz">https://zenodo.org/record/5636209/files/GRCh38_GREEN-DB.bed.gz</a>
GRCh38_GREENDB	GREENDB_bed	<a href="https://zenodo.org/record/5636209/files/GRCh38_GREEN-DB.bed.gz.csi">https://zenodo.org/record/5636209/files/GRCh38_GREEN-DB.bed.gz.csi</a>
GREENDB_sqlite	GREENDB_sqlite	<a href="https://zenodo.org/record/5636209/files/GREEN-DB_v2.5.db.gz">https://zenodo.org/record/5636209/files/GREEN-DB_v2.5.db.gz</a>

## 4.4 How to cite

### 4.4.1 GREEN-DB

If you use any information from GREEN-DB please cite: [GREEN-DB: a framework for the annotation and prioritization of non-coding regulatory variants in whole-genome sequencing](#) Giacomuzzi E., Popitsch N., Taylor JC. BiorXiv (2020)

### 4.4.2 GREEN-VARAN

When you use greenvaran for annotation please cite

[GREEN-DB: a framework for the annotation and prioritization of non-coding regulatory variants in whole-genome sequencing](#)

Giacomuzzi E., Popitsch N., Taylor JC. BiorXiv (2021)

If you use the GREEN-VARAN Nextflow workflow for additional annotations also cite

[Vcfanno: fast, flexible annotation of genetic variants](#)

Brent S. Pedersen, Ryan M. Layer & Aaron R. Quinlan. Genome Biology volume 17, Article number: 118 (2016)

If you include annotation with a prediction score please also cite the corresponding paper

Score	PUBMED ID
CADD	30371827
DANN	25338716
EIGEN	26727659
ExPecto	30013180
FATHMM-MKL	25583119
FATHMM-XF	28968714
FINSURF	doi.org/10.1101/2021.05.03.442347
FIRE	28961785
GenoCanyon	26015273
GenoSkyline-plus	27058395
GWAVA	24487584
LinSight	28288115
NCBoost	30744685
ncER	31748530
ReMM	27569544

#### 4.4.3 Population AF

If you use population AF annotation with GREEN-VARAN workflow also cite the gnomAD paper: [The mutational constraint spectrum quantified from variation in 141,456 humans](#)



## CHAPTER 5

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`