cgap-annotations

Release v1.0.0

Michele Berselli, Soo Lee, CGAP team, HMS DBMI

Jan 31, 2023

CONTENTS

1 Gene Annotations **3**
- 1.1 Data Sources . . . 3

2 Variant Annotations **5**
- 2.1 Data Sources . . . 5

3 Other References **11**
- 3.1 hg38/GRCh38 Genome Build . . . 11
- 3.2 HaplotypeCaller Exome Region File . . . 11
- 3.3 BICseq2 Mappability File . . . 12
- 3.4 Unrelated Files and Panel of Normal . . . 13
- 3.5 ASCAT Resources . . . 15
- 3.6 Lift-over Chain Files . . . 15

This documentation covers the sources of annotation used in CGAP for genes and variants. It also provides information on the creation of custom reference files used in the CGAP Pipelines.

Copies of the files referred to in these docs are stored in the S3 bucket:

s3://cgap-annotations

The bucket is private and not meant for public files sharing. It is intended for internal back-up only and most of the files are stored in deeper archive tiers.

CHAPTER

ONE

GENE ANNOTATIONS

1.1 Data Sources

Data sources available for genes annotation.

1.1.1 RefSeq

Files

ncbi refseq RefSeqGene:

- LRG_RefSeqGene
- refseqgene.<n>.genomic.gbff.gz

ncbi refseq mRNA_Prot:

- human.<n>.rna.gbff.gz

ncbi gene:

- gene2ensembl.gz
- gene2refseq.gz

Description

LRG_RefSeqGene is a tab-delimited file reporting, for each gene, the *accession.version* of the genomic RefSeq (RSG) that is the standard reference. Additionally reports the *accession.version* of the associated RNA and protein RefSeqs.

```
#tax_id   GeneID   Symbol   RSG   LRG   RNA   t   Protein   p   Category
```

refseqgene.<n>.genomic.gbff report annotations for each RSG in GenBank format.

human.<n>.rna.gbff report annotations for each RNA and protein RefSeq in GenBank format.

gene2ensembl is a tab-delimited file matching NCBI to Ensembl annotations.

```
#tax_id   GeneID   Ensembl_gene_identifier   RNA_nucleotide_accession.version   Ensembl_
↪rna_identifier   protein_accession.version   Ensembl_protein_identifier
```

gene2refseq is a tab-delimited file reporting genomic/RNA/protein sets of matching RefSeqs.

```
#tax_id  GeneID  status  RNA_nucleotide_accession.version  RNA_nucleotide_gi  ␣
↪protein_accession.version  protein_gi  genomic_nucleotide_accession.version  ␣
↪genomic_nucleotide_gi  start_position_on_the_genomic_accession  end_position_on_the_
↪genomic_accession  orientation  assembly  mature_peptide_accession.version  mature_
↪peptide_gi  Symbol
```

Version

Current version accessed 2020-10-22.

- LRG_RefSeqGene: v20201020
- refseqgene.<n>.genomic.gbff.gz: v20201020
- human.<n>.rna.gbff.gz: v20201020
- gene2ensembl.gz: v20201022
- gene2refseq.gz: v20201022

CHAPTER

TWO

VARIANT ANNOTATIONS

2.1 Data Sources

Software and data sources for variants annotation.

2.1.1 VEP

Current software version is 101.

Annotation uses Variant Effect Predictor (VEP) software.

Source files for Software and Plugins.

Annotation Sources

VEP

This is the main annotation source for VEP.

Source file v101 for homo_sapiens on **hg38/GRCh38**.

```
$ wget ftp://ftp.ensembl.org/pub/release-101/variation/vep/homo_sapiens_vep_101_GRCh38.
↪tar.gz
```

MaxEnt

Current version v20040421.

This is the data source used by MaxEntScan plugin.

Source file fordownload.

ClinVar

Current version is v20201101. ClinVar is updated weekly.

This is the data source for ClinVar to be used with `--custom`.

```
# Compressed VCF file
$ curl -O ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/clinvar.vcf.gz
# Index file
$ curl -O ftp://ftp.ncbi.nlm.nih.gov/pub/clinvar/vcf_GRCh38/clinvar.vcf.gz.tbi
```

SpliceAI

Current version is v1.3.

This is the data source used by SpliceAI plugin.

Download requires a log in on illumina platform and BaseSpace sequence CLI.

```
# Authenticate
$ bs auth
# Get id for dataset genome_scores
$ bs list dataset
# Download
$ bs dataset download --id <datasetid> -o .
```

For annotation we are using the raw **hg38/GRCh38** files and their index:

- `spliceai_scores.raw.snv.hg38.vcf.gz`
- `spliceai_scores.raw.snv.hg38.vcf.gz.tbi`
- `spliceai_scores.raw.indel.hg38.vcf.gz`
- `spliceai_scores.raw.indel.hg38.vcf.gz.tbi`

dbNSFP

Current version 4.1a.

This is the data source used by dbNSFP plugin.

A small modification was made to the source code for the dbNSFP plugin to allow for annotation of non-missense variants. The change is shown below with the original code commented out.

```
#my %INCLUDE_SO = map {$_ => 1} qw(missense_variant stop_lost stop_gained start_lost);
my %INCLUDE_SO = map {$_ => 1} qw(missense_variant stop_lost stop_gained start_lost␣
↪splice_donor_variant splice_acceptor_variant splice_region_variant frameshift inframe_
↪insertion inframe_deletion);
```

Source file dbNSFP.

To create the data source:

```
# Download and unpack
$ wget ftp://dbnsfp:dbnsfp@dbnsfp.softgenetics.com/dbNSFP4.1a.zip
$ unzip dbNSFP4.1a.zip
# Get header
$ zcat dbNSFP4.1a_variant.chr1.gz | head -n1 > h
# Extract information and compress to bgzip
$ zgrep -h -v ^#chr dbNSFP4.1a_variant.chr* | sort -T /path/to/tmp_folder -k1,1 -k2,2n -␣
↪| cat h - | bgzip -c > dbNSFP4.1a.gz
# Create tabix index
$ tabix -s 1 -b 2 -e 2 dbNSFP4.1a.gz
```

gnomAD Genomes

Current genome version 3.1.

Files are available for download at https://gnomad.broadinstitute.org/downloads.

Files have been preprocessed to reduce the number of annotations using `filter_gnomAD.py` script inside scripts folder. The annotations that are used and maintained are listed in `gnomAD_3.1_fields.tsv` file inside variants folder.

gnomAD files have been filtered while splitting by chromosomes. The filtered `vcf` files have been concatenated, compressed with `bgzip` and indexed using `tabix`.

gnomAD Exomes

Current exome version 2.1.1 (hg38/GRCh38 lift-over).

The all chromosomes `vcf` was downloaded from https://gnomad.broadinstitute.org/downloads.

This file was preprocessed to reduce the number of annotations using the `gnomAD_exome_v2_filter.py` scripts inside the scripts folder. The annotations that are used and maintained are listed in the `gnomAD_2.1_fields.tsv` file inside the variants folder.

The filtered `vcf` was compressed with `bgzip` and indexed using `tabix`.

gnomAD Structural Variants

Current SV version is nstd166 (hg38/GRCh38 lift-over).

File was originally downloaded here: https://ftp.ncbi.nlm.nih.gov/pub/dbVar/data/Homo_sapiens/by_study/vcf/nstd166.GRCh38.variant_call.vcf.gz, but that same link now takes to a newer and incorrect file.

See `nstd166_GRCh38_readme.txt` in the `s3://cgap-annotations/gnomAD/SV/` for in-depth explanation. We have copies of both the original (currently used) and the newer file in the bucket.

CADD

Current version is v1.6

CADD SNV and INDEL files were downloaded from https://cadd-staging.kircherlab.bihealth.org/download

```
$ wget https://krishna.gs.washington.edu/download/CADD/v1.6/GRCh38/whole_genome_SNVs.tsv.
↪gz
$ wget https://krishna.gs.washington.edu/download/CADD/v1.6/GRCh38/gnomad.genomes.r3.0.
↪indel.tsv.gz
```

This is the data source used by CADD plugin.

Conservation Scores

Current version is UCSC hg38/GRCh38 for phyloP30way, phyloP100way, and phastCons100way

```
$ wget http://hgdownload.cse.ucsc.edu/goldenpath/hg38/phyloP30way/hg38.phyloP30way.bw
$ wget http://hgdownload.cse.ucsc.edu/goldenpath/hg38/phyloP100way/hg38.phyloP100way.bw
$ wget http://hgdownload.cse.ucsc.edu/goldenpath/hg38/phastCons100way/hg38.
↪phastCons100way.bw
```

These files were supplied to customs within VEP.

Run VEP

```
# Base command
vep \
-i input.vcf \
-o output.vep.vcf \
--hgvs \
--fasta <PATH/reference.fa> \
--assembly GRCh38 \
--use_given_ref \
--offline \
--cache_version 101 \
--dir_cache . \
--everything \
--force_overwrite \
--vcf \
--dir_plugins <PATH/VEP_plugins>

# Additional plugins
--plugin SpliceRegion,Extended
--plugin MaxEntScan,<PATH/fordownload>
--plugin TSSDistance
--plugin dbNSFP,<PATH/dbNSFP.gz>,phyloP100way_vertebrate_rankscore,GERP++_RS,GERP++_RS_
↪rankscore,SiPhy_29way_logOdds,SiPhy_29way_pi,PrimateAI_score,PrimateAI_pred,PrimateAI_
↪rankscore,CADD_raw_rankscore,Polyphen2_HVAR_pred,Polyphen2_HVAR_rankscore,Polyphen2_
↪HVAR_score,SIFT_pred,SIFT_converted_rankscore,SIFT_score,REVEL_rankscore,REVEL_score,
↪Ensembl_geneid,Ensembl_proteinid,Ensembl_transcriptid
--plugin SpliceAI,snv=<PATH/spliceai_scores.raw.snv.hg38.vcf.gz>,indel=<PATH/spliceai_
```

(continues on next page)

(continued from previous page)

```
↪scores.raw.indel.hg38.vcf.gz>
--plugin CADD,<PATH/whole_genome_SNVs.tsv.gz>,<PATH/gnomad.genomes.r3.0.indel.tsv.gz>

# Custom annotations
--custom <PATH/clinvar.vcf.gz>,ClinVar,vcf,exact,0,ALLELEID,CLNSIG,CLNREVSTAT,CLNDN,
↪CLNDISDB,CLNDNINCL,CLNDISDBINCL,CLNHGVS,CLNSIGCONF,CLNSIGINCL,CLNVC,CLNVCSO,CLNVI,
↪DBVARID,GENEINFO,MC,ORIGIN,RS,SSR
--custom <PATH/gnomAD.vcf.gz>,gnomADg,vcf,exact,0,AC,AC-XX,AC-XY,AC-afr,AC-ami,AC-amr,AC-
↪asj,AC-eas,AC-fin,AC-mid,AC-nfe,AC-oth,AC-sas,AF,AF-XX,AF-XY,AF-afr,AF-ami,AF-amr,AF-
↪asj,AF-eas,AF-fin,AF-mid,AF-nfe,AF-oth,AF-sas,AF_popmax,AN,AN-XX,AN-XY,AN-afr,AN-ami,
↪AN-amr,AN-asj,AN-eas,AN-fin,AN-mid,AN-nfe,AN-oth,AN-sas,nhomalt,nhomalt-XX,nhomalt-XY,
↪nhomalt-afr,nhomalt-ami,nhomalt-amr,nhomalt-asj,nhomalt-eas,nhomalt-fin,nhomalt-mid,
↪nhomalt-nfe,nhomalt-oth,nhomalt-sas
--custom <PATH/trimmed_gnomad.exomes.r2.1.1.sites.liftover_grch38.vcf.gz>,gnomADe2,vcf,
↪exact,0,AC,AN,AF,nhomalt,AC_oth,AN_oth,AF_oth,nhomalt_oth,AC_sas,AN_sas,AF_sas,nhomalt_
↪sas,AC_fin,AN_fin,AF_fin,nhomalt_fin,AC_eas,AN_eas,AF_eas,nhomalt_eas,AC_amr,AN_amr,AF_
↪amr,nhomalt_amr,AC_afr,AN_afr,AF_afr,nhomalt_afr,AC_asj,AN_asj,AF_asj,nhomalt_asj,AC_
↪nfe,AN_nfe,AF_nfe,nhomalt_nfe,AC_female,AN_female,AF_female,nhomalt_female,AC_male,AN_
↪male,AF_male,nhomalt_male,AF_popmax
--custom <PATH/hg38.phyloP100way.bw>,phylop100verts,bigwig,exact,0
--custom <PATH/hg38.phyloP30way.bw>,phylop30mams,bigwig,exact,0
--custom <PATH/hg38.phastCons100way.bw>,phastcons100verts,bigwig,exact,0
```

2.1.2 dbSNP

Current database version is v151.

```
# Download all variants file from the GATK folder
$ wget https://ftp.ncbi.nlm.nih.gov/snp/pre_build152/organisms/human_9606_b151_GRCh38p7/
↪VCF/GATK/00-All.vcf.gz
# Parse to reduce size
$ python vcf_parse_keep5.py 00-All.vcf.gz 00-All_keep5.vcf
# Compress and index
$ bgzip 00-All_keep5.vcf
$ bcftools index 00-All_keep5.vcf.gz
$ tabix 00-All_keep5.vcf.gz
```

2.1.3 Cytoband

The **hg38/GRCh38** Cytoband reference file from UCSC: http://hgdownload.cse.ucsc.edu/goldenpath/hg38/database/cytoBand.txt.gz.

2.1.4 HGVSg

Current version 20.05

The Human Genome Variation Society has strict guidelines and best practices for describing human genomic variants based on the reference genome, chromosomal position, and variant type. HGVSg can be used to describe all genomic variants, not just those within coding regions. The script used to generate HGVSg infomation in our pipeline implements the recommendations found here for DNA variants (http://varnomen.hgvs.org/recommendations/DNA/). We describe substitions, deletions, insertions, and deletion-insertions for all variants on the 23 nuclear chromosomes and the mitochondrial genome within this field.

2.1.5 Version

Current version accessed 2021-04-20.

- VEP: v101
- MaxEnt: v20040421
- ClinVar: v20201101
- SpliceAI: v1.3
- dbNSFP: v4.1a
- gnomAD: v3.1
- gnomAD_exomes: v2.1.1
- CADD: v1.6
- phyloP30way: hg38/GRCh38
- phyloP100way: hg38/GRCh38
- phastCons100way: hg38/GRCh38
- dbSNP: v151
- HGVSg: 20.05
- Cytoband: hg38/GRCh38

CHAPTER

THREE

OTHER REFERENCES

3.1 hg38/GRCh38 Genome Build

The reference files were downloaded here. This build include an additional index file that is required to flag alternate contigs as described here.

FASTA

- Homo_sapiens_assembly38.fasta
- Homo_sapiens_assembly38.fasta.fai
- Homo_sapiens_assembly38.dict

Burrows-Wheeler transformed

- Homo_sapiens_assembly38.fasta.64.bwt
- Homo_sapiens_assembly38.fasta.64.ann
- Homo_sapiens_assembly38.fasta.64.amb
- Homo_sapiens_assembly38.fasta.64.pac
- Homo_sapiens_assembly38.fasta.64.sa

Alternate contigs

- Homo_sapiens_assembly38.fasta.64.alt

3.2 HaplotypeCaller Exome Region File

Data sources and code used to generate the exome region file used by GATK `HaplotypeCaller` in WES runs.

3.2.1 VEP v101

Accessed 2021-10-21.

VEP v101 archive website.

VEP v101 gtf file:

- Homo_sapiens.GRCh38.101.gtf.gz

A copy of this file is also stored within the `exome_regions` folder of the cgap-annotations s3 bucket.

3.2.2 Reference File Creation

To transform this VEP `gtf` file into a comprehensive `bed` file of all possible transcripts and UTR regions, one python script and two BEDTools (v2.30.0) commands were used.

```
bgzip -d Homo_sapiens.GRCh38.101.gtf.gz
python exome_hg38_region_of_interest.py Homo_sapiens.GRCh38.101.gtf regions_bed_final.bed
bedtools sort -i regions_bed_final.bed > sort_regions_bed_final.bed
bedtools merge -i sort_regions_bed_final.bed > merge_sort_regions_bed_final.bed
```

`exome_hg38_region_of_interest.py` is available in this repository in `/genes/exome_regions/`.

3.3 BICseq2 Mappability File

`BICseq2-norm` makes use of a unique mappability file to aid in the process of normalizing the raw coverage data presented in the `seq` files. This mappability file must be generated for each library size (e.g., 150 bp) given that unique mappability will vary with read length. A 100 bp read might not map uniquely at a given position, but a 150 bp read starting from the same position might map uniquely given 50 additional bases at the end.

The current mappability file was generated for 150 bp reads using a custom workflow, as follows:

1. The file `chromosomes.txt` was created with only the 23 chromosomes from **hg38/GRCh38** (e.g., chr1, chr2 … chr22, chrX, chrY; each on their own line). These regions were extracted from our **hg38/GRCh38** reference genome `GAPFIXRDPDK5.fa` to generate `hg38_main_chrs.fa` and a `fasta` index file was generated for this output.

```
for file in $(cat chromosomes.txt); do samtools faidx GAPFIXRDPDK5.fa $file >> hg38_main_
↪chrs.fa; done
samtools faidx hg38_main_chrs.fa
```

2. Using an archived version of GEMTools (v 1.7.1-i3) distributed in the github repo below, the initial mappability file was generated and converted to `wig` format:

```
git clone https://github.com/LinjieWu/GenerateMappability
cd GenerateMappability
python setup.py
cd ..

SoftwareDir="<path_to_folder>/GenerateMappability"
export PATH=${SoftwareDir}/gemtools-1.7.1-i3/bin/:$PATH

gem-indexer -T 16 -c dna -i hg38_main_chrs.fa -o hg38_main_chr_index
gem-mappability -T 16 -I hg38_main_chr_index.gem -l 150 -o hg38_full_mappability_150
gem-2-wig -I hg38_main_chr_index.gem -i hg38_full_mappability_150.mappability -o hg38_
↪full_mappability_150
```

3. This `wig` mappability file must next be converted to a `bed` file through a series of conversion steps using tools available from UCSC:

```
./wigToBigWig hg38_full_mappability_150.wig hg38_full_mappability_150.sizes hg38_full_
↪mappability_150.bw
./bigWigToBedGraph hg38_full_mappability_150.bw  hg38_full_mappability_150.bedGraph
./bedGraphTobed hg38_full_mappability_150.bedGraph hg38_full_mappability_150.bed 1
```

4. After testing this mappability file, we determined that repetitive regions at the centromeres were causing large numbers of artefactual CNVs. BICseq2 had been optimized previously for **hg19/GRCh37** with mappability files that excluded the centromeres, so we decided to also exclude the centromeric regions from our **hg38/GRCh38** mappability file. The centromeres for **hg38/GRCh38** were pulled from UCSC as follows:

1. Navigate to http://genome.ucsc.edu/cgi-bin/hgTables
2. Under `assembly`, select `Dec. 2013 (GRCh38/hg38)`
3. Under `group`, select `Mapping and Sequencing`
4. Under `track`, select `Chromosome Band (Ideogram)`
5. Under `filter`, select `create`
6. Under `gieStain`, select `does` match, and type `acen` in the text box, then select `submit`
7. Under `output format`, select `BED - browser extensible data`
8. Select `get output`
9. Select `get BED`
5. This `bed` file was saved as `centromeres.bed` and subtracted from the existing mappability file:

```
bedtools subtract -a hg38_full_mappability_150.bed -b centromeres.bed > hg38_full_
↪mappability_150_no_centromeres.bed
```

6. Finally, the `bed` file was parsed to generate a single mappability file for each chromosome in the format required by `BICseq2-norm`:

```
for file in $(cat chromosomes.txt); do echo $file; grep -P ${file}'\t' hg38_full_
↪mappability_150_no_centromeres.bed | awk -v OFS='\t' '{print $2, $3}' > full_
↪mappability_hg38_150_no_centromeres/${file}_mappability; done
```

3.4 Unrelated Files and Panel of Normal

For many of the CGAP Pipelines, a collection of 20 de-identified UGRP samples are used to aid in filtering common variants. This documentation page outlines how they were created.

3.4.1 SNV Pipeline - Unrelated RCK files

Sentieon

1. 20 unrelated `fastq` files from UGRP dataset were run through the Upstream Sentieon module (v1.0.0) to generate analysis-ready `bam` files.
2. The `bam` files were then processed using a custom module (SNV Unrelated, v1.0.0) that executes granite `mpileupCounts` and `rckTar` commands.
3. The final file was uploaded to the CGAP Portal as: `196ef586-be28-40c5-a244-d739fd173984/GAPFIMO8Y4K1.rck.tar`

GATK

1. 20 unrelated `fastq` files from UGRP dataset were run through the Upstream GATK module (v1.0.0) to generate analysis-ready `bam` files.
2. The `bam` files were then processed using a custom module (SNV Unrelated, v1.0.0) that executes granite `mpileupCounts` and `rckTar` commands.
3. The final file was uploaded to the CGAP Portal as: `eac862c0-8c87-4838-83cb-9a77412bff6f/GAPFIMO8Y4PZ.rck.tar`

3.4.2 Somatic Sentieon - Panel of Normal (PON)

1. 20 unrelated `fastq` files from UGRP dataset were run through the Upstream Sentieon module (v1.0.0) to generate analysis-ready `bam` files.
2. Following this protocol from Sentieon each resulting `bam` file was run individually through the Somatic Sentieon Tumor Only module (v1.0.0), using `GAPFI4LJRN98.vcf.gz` dbSNP file for known SNPs.
3. The 20 resulting `vcf` output files were merged using BCFtools (1.10.2).
4. This file was uploaded to the CGAP Portal as: `833c91e9-a8cd-470e-8100-32b49ed14159/GAPFIV1QKYU9.vcf.gz`

3.4.3 SV Pipeline - Manta

1. 20 unrelated `fastq` files from UGRP dataset were uploaded to the (now decommissioned) cgap-wolf environment.
2. Each of the 20 samples was run through the Upstream GATK module (v24), ending with a final `bam` file following `workflow_gatk-ApplyBQSR`.
3. Each of the resulting final `bam` files was run through a proband-only Manta workflow (v2) to produce `vcf` files.
4. The resulting `vcf` files were downloaded to a folder named `unrelated`, which was compressed:

```
tar -cvf unrelated.tar unrelated
```

5. This file was uploaded to the CGAP Portal as: `cd647c0c-ac11-46db-9c51-bfe238e9ac13/GAPFIH794KXC.vcf.tar`

3.4.4 CNV Pipeline - BICseq2

1. 20 unrelated `fastq` files from UGRP dataset were retrieved from Glacier Deep Archive and uploaded to the current cgap-wolf environment.
2. Each of the 20 samples was run through the Upstream GATK module (v27), ending with a final `bam` file following `workflow_gatk-ApplyBQSR`.
3. Each of the resulting final `bam` files was run through the development version of the CNV module (v1), which included only 2 steps (`workflow_BICseq2_map_norm_seg` and `workflow_BICseq2_vcf_convert_vcf-check`). This development version still included chromosomes X and Y as well, which have since been removed from the production version.
4. The resulting `vcf` files were downloaded to a folder named `unrelated`, which was compressed:

```
tar -cvf unrelated.tar unrelated
```

5. This file was uploaded to the CGAP Portal as: `318788cd-661f-4327-b571-d58a9b7c301e/GAPFICPW2884.vcf.tar`

3.5 ASCAT Resources

Current software version is 3.0.0.

Source files for Software .

ASCAT requires a set of external reference data that are provided as additional data sources in the main repository of the software, here. For convenience, we collected and packaged these resources into a single `tar` archive that contains the following set of files.

3.5.1 Loci Files

ASCAT repository commit 7fc8c9d, files version 20092021

Loci files contain SNP positions derived from the 1000Genomes prepared for **hg38/GRCh38**, available here. We operate on chr-based `bam` files, so the original loci files were modified and the `chr-` prefix was added by running the command:

```
for i in {1..22} X; do sed -i 's/^/chr/' G1000_loci_hg38_chr${i}.txt; done
```

3.5.2 Allele Files

ASCAT repository commit 7fc8c9d, files version 20092021

Allele files contain SNP positions with their reference and alternative nucleotide bases based on the 1000Genomes prepared for **hg38/GRCh38**, available here.

3.5.3 GC Correction File

ASCAT repository commit 7fc8c9d, files version 20092021

The GC correction file contains the GC content around every SNP for increasing window sizes, available here.

3.6 Lift-over Chain Files

3.6.1 hg19/GRCh37 to hg38/GRCh38

Chain file that translate coordinates from **hg19/GRCh37** to **hg38/GRCh38** genome build.

The chain file was downloaded from UCSC, available here.

3.6.2 hg38/GRCh38 to hg19/GRCh37

Chain file that translate coordinates from **hg38/GRCh38** to **hg19/GRCh37** genome build.

The chain file was downloaded from UCSC, available here.