# Genomic regions of interest TSV File only for canonical chromosomes

# All the following data are combined in one unique file Genome_Regions_data.tsv

ChrStart\tEnd\tRegion\tGenomeVersion

## Segmental Duplication regions dataset

Downloaded on 25/04/2025 from https://genome.ucsc.edu/cgi-bin/hgTables

SegmentalDups_GRCh37.bed from :
https://genome.ucsc.edu/cgi-bin/hgTables?hgsid=2529510726_A566RApY6cLEYg7x3NCXAq93ABrZ&clade=mammal&org=Human&db=hg19&hgta_group=allTracks&hgta_track=genomicSuperDups&hgta_table=0&hgta_regionType=genome&position=chr7%3A155%2C592%2C223-155%2C605%2C565&hgta_outputType=bed&hgta_outFileName=SegmentalDups_GRCh37.bed

SegmentalDups_GRCh38.bed from :
https://genome.ucsc.edu/cgi-bin/hgTables?hgsid=2529510726_A566RApY6cLEYg7x3NCXAq93ABrZ&clade=mammal&org=&db=hg38&hgta_group=allTracks&hgta_track=genomicSuperDups&hgta_table=genomicSuperDups&hgta_regionType=genome&position=&hgta_outputType=bed&hgta_outFileName=SegmentalDups_GRCh38.bed

```
cut -f1-3 SegmentalDups_GRCh38.bed |
                  sort -k1,1 -k2,2n |
               bedtools merge -i - |
               awk 'BEGIN {OFS="\t"} {print $0, "segmentaldup",
"GRCh38"}' | awk '$1 ~ /^chr([1-9]|1[0-9]|2[0-2]|X|Y)$/'> \
               merged_SegmentalDups_GRCh38.bed

cut -f1-3 SegmentalDups_GRCh37.bed |
                  sort -k1,1 -k2,2n |
               bedtools merge -i - |
               awk 'BEGIN {OFS="\t"} {print $0, "segmentaldup",
"GRCh37"}' | awk '$1 ~ /^chr([1-9]|1[0-9]|2[0-2]|X|Y)$/'> \
               merged_SegmentalDups_GRCh37.bed
```

## PAR (Pseudoautosomal Region) regions dataset

From https://www.ncbi.nlm.nih.gov/grc/human on 25/04/2025

GRCh37.p13
chrX 60001 2699520 PAR1 GRCh37
chrX 154931044 155260560 PAR2 GRCh37

GRCh38.p14
chrX 10001 2781479 PAR1 GRCh38
chrX 155701383 156030895 PAR2 GRCh38

## X-Transpose region dataset

From Timothy H Webster, Madeline Couse, Bruno M Grande, Eric Karlins, Tanya N Phung, Phillip A Richmond, Whitney Whitford, Melissa A Wilson, Identifying, understanding, and correcting technical artifacts on the sex chromosomes in next-generation sequencing data, GigaScience, Volume 8, Issue 7, July 2019, giz074, https://doi.org/10.1093/gigascience/giz074

"We define the XTR on the X chromosome as beginning at the start of DXS1217 and ending at the end of DXS3"

GRCh37.p13
https://grch37.ensembl.org/Homo_sapiens/Marker/Details?m=sWXD902
https://grch37.ensembl.org/Homo_sapiens/Marker/Details?m=sWXD298

DXS1217 chromosome X:88395845-88396079 GRCh37
DXS3 chromosome X:92582890-92583067 GRCh37

GRCh38.p14
https://useast.ensembl.org/Homo_sapiens/Marker/Details?db=core;m=DXS1217;r=X:89140845-89141079
https://useast.ensembl.org/Homo_sapiens/Marker/Details?db=core;m=DXS3;r=X:93327891-93328068

DXS1217 chromosome X:89140845-89141079 GRCh38
DXS3 chromosome X:93327891-93328068 GRCh38

### XTR region coordinates

chrX 88395845 92583067 XTR GRCh37
chrX 89140845 93328068 XTR GRCh38

## Telomeric and Centromeric regions dataset

Downloaded on 25/04/2025 from https://genome.ucsc.edu/cgi-bin/hgTables

ChromosomeBand_GRCh37.tsv from :
https://genome.ucsc.edu/cgi-bin/hgTables?hgsid=2529613476_dkAUVDEoH74j8LaCc6nSM9DQngP5&clade=mammal&org=Human&db=hg19&hgta_group=map&hgta_track=cytoBand&hgta_table=0&hgta_regionType=genome&position=chr7%3A155%2C592%2C223-155%2C605%2C565&hgta_outputType=primaryTable&hgta_outFileName=ChromosomeBand ChromosomeBand_GRCh37.tsv

ChromosomeBand_GRCh38.tsv from :
https://genome.ucsc.edu/cgi-bin/hgTables?hgsid=2529613476_dkAUVDEoH74j8LaCc6nSM9DQngP5&clade=mammal&org=Human&db=hg38&hgta_group=map&hgta_track=cytoBand&hgta_table=0&hgta_regionType=genome&position=chr7%3A155%2C592%2C223-155%2C605%2C565&hgta_outputType=primaryTable&hgta_outFileName=ChromosomeBand_GRCh38.tsv

## Formatting Code, example on GRCh37

**Get first and last bands for each chromosome (after skipping header), then filter for 'gneg' (telomeric regions), keep only canonical chromosomes, and format the output with a "telomere" label.**

```
genome_version=GRCh37
awk 'NR > 1' ChromosomeBand_${genome_version}.tsv | sort -k1,1 -k2,2n |
\
awk '
{
    chrom=$1
    if (chrom != prev_chrom) {
        if (NR > 2) print last_line
        print $0
        prev_chrom = chrom
    }
    last_line = $0
}
END {
    print last_line
}' | grep gneg |
awk '$1 ~ /^chr([1-9]|1[0-9]|2[0-2]|X|Y)$/' |
                                  cut -f1-3 |
                        bedtools merge -i - |
awk -v gv="${genome_version}" 'BEGIN {OFS="\t"} {print $0, "telomere",
gv}' > telomere_${genome_version}.tsv
```

**Extract centromeric regions (gieStain == "acen"), restrict to canonical chromosomes, and format with a "centromere" label.**

```
grep acen ChromosomeBand_${genome_version}.tsv | awk '$1 ~ /^chr([1-
9]|1[0-9]|2[0-2]|X|Y)$/' | cut -f1-3 | bedtools merge -i - | awk -v
gv="${genome_version}" 'BEGIN {OFS="\t"} {print $0, "centromere", gv}' >
centromere_${genome_version}.tsv
```

**Combine centromere and telomere regions into a unified, sorted file.**

```
cat centromere_${genome_version}.tsv telomere_${genome_version}.tsv |
sort -k1,1 -k2,2n > regions_${genome_version}.tsv
```

Getting Transcript Coordinates from GTF file:

```
curl https://ftp.ensembl.org/pub/release-
113/gtf/homo_sapiens/Homo_sapiens.GRCh38.113.gtf.gz >
Homo_sapiens.GRCh38.113.gtf.gz

duckdb -c "
    CREATE TABLE tbl AS (SELECT * FROM
read_csv('Homo_sapiens.GRCh38.113.gtf.gz', delim = '\t', all_varchar =
true));
    ALTER TABLE tbl ADD COLUMN transcript_id VARCHAR;
    UPDATE tbl SET transcript_id = REGEXP_EXTRACT(string_split(column8,
';')[3], 'transcript_id \"(.*?)\"', 1);

    COPY(SELECT column0::VARCHAR AS Chr,
                column3::INTEGER AS Start,
                column4::INTEGER AS Stop,
                transcript_id FROM tbl WHERE(column2 = 'transcript'))
    TO transcript_coords_38.parquet;
    "


curl https://ftp.ensembl.org/pub/grch37/release-
113/gtf/homo_sapiens/Homo_sapiens.GRCh37.87.chr.gtf.gz >
Homo_sapiens.GRCh37.87.gtf.gz

duckdb -c "
    CREATE TABLE tbl AS (SELECT * FROM
read_csv('Homo_sapiens.GRCh37.87.gtf.gz', delim = '\t', all_varchar =
true));
    ALTER TABLE tbl ADD COLUMN transcript_id VARCHAR;
    UPDATE tbl SET transcript_id = REGEXP_EXTRACT(string_split(column8,
';')[3], 'transcript_id \"(.*?)\"', 1);

    COPY(SELECT column0::VARCHAR AS Chr,
                column3::INTEGER AS Start,
                column4::INTEGER AS Stop,
                transcript_id FROM tbl WHERE(column2 = 'transcript'))
    TO transcript_coords_37.parquet;
    "
```