SNPlocs.Hsapiens.dbSNP144.GRCh37

May 14, 2025

`getSNPlocs` *Accessing the SNPs stored in SNPlocs.Hsapiens.dbSNP144.GRCh37*

Description

Functions for accessing the SNPs stored in the SNPlocs.Hsapiens.dbSNP144.GRCh37 package.

WARNING: All the functions described in this man page are defunct and will be removed at some point in the future. See `?snpcount` in the **BSgenome** software package for the new preferred way to access the data stored in this package.

Usage

```
## Count and load all the SNPs for a given chromosome:
getSNPcount()
getSNPlocs(seqname, as.GRanges=FALSE, caching=TRUE)

## Extract SNP information for a set of rs ids:
rsid2loc(rsids, caching=TRUE)
rsid2alleles(rsids, caching=TRUE)
rsidsToGRanges(rsids, caching=TRUE)
```

Arguments

`seqname`	The name of the sequence for which to get the SNP locations and alleles. If `as.GRanges` is `FALSE`, only one sequence can be specified (i.e. `seqname` must be a single string). If `as.GRanges` is `TRUE`, an arbitrary number of sequences can be specified (i.e. `seqname` can be a character vector of arbitrary length).
`as.GRanges`	`TRUE` or `FALSE`. If `TRUE`, then the SNP locations and alleles are returned in a GRanges object. Otherwise (the default), they are returned in a data frame (see below).
`caching`	Should the loaded SNPs be cached in memory for faster further retrieval but at the cost of increased memory usage?
`rsids`	A vector of rs ids. Can be integer or character vector, with or without the `"rs"` prefix. NAs are not allowed.

Details

See SNPlocs.Hsapiens.dbSNP144.GRCh37 for general information about this package.

The SNP data are split by chromosome (1-22, X, Y, MT) i.e. the package contains one data set per chromosome, each of them being a serialized data frame with 1 row per SNP and the 2 following columns:

- `loc`: The 1-based location of the SNP relative to the first base at the 5' end of the plus strand of the reference sequence.
- `alleles`: A raw vector with no NAs which can be converted into a character vector containing the alleles for each SNP represented by an IUPAC nucleotide ambiguity code (see `?IUPAC_CODE_MAP` in the Biostrings package for more information).

Note that those data sets are not intended to be used directly but the user should instead use the `getSNPcount` and `getSNPlocs` convenience wrappers for loading the SNP data. When used with `as.GRanges=FALSE` (the default), `getSNPlocs` returns a data frame with 1 row per SNP and the 3 following columns:

- `RefSNP_id`: RefSNP ID (aka "rs id") with `"rs"` prefix removed. Character vector with no NAs and no duplicates.
- `alleles_as_ambig`: A character vector with no NAs containing the alleles for each SNP represented by an IUPAC nucleotide ambiguity code.
- `loc`: Same as for the 2-col serialized data frame described previously.

Value

`getSNPcount` returns a named integer vector containing the number of SNPs for each sequence in the reference genome.

By default (`as.GRanges=FALSE`), `getSNPlocs` returns the 3-col data frame described above containing the SNP data for the specified chromosome. Otherwise (`as.GRanges=TRUE`), it returns a GRanges object with extra columns `"RefSNP_id"` and `"alleles_as_ambig"`. Note that all the elements (genomic ranges) in this GRanges object have their strand set to `"+"` and that all the sequence lengths are set to `NA`.

`rsid2loc` and `rsid2alleles` both return a named vector (integer vector for the former, character vector for the latter) where each (name, value) pair corresponds to a supplied rs id. For both functions the name in (name, value) is the chromosome of the rs id. The value in (name, value) is the position of the rs id on the chromosome for `rsid2loc`, and a single IUPAC code representing the associated alleles for `rsid2alleles`.

`rsidsToGRanges` returns a GRanges object similar to the one returned by `getSNPlocs` (when used with `as.GRanges=TRUE`) and where each element corresponds to a supplied rs id.

Author(s)

H. Pagès

See Also

- snpcount in the **BSgenome** software package for the new preferred way to access the data stored in this package.
- SNPlocs.Hsapiens.dbSNP144.GRCh37

SNPlocs.Hsapiens.dbSNP144.GRCh37

The SNPlocs.Hsapiens.dbSNP144.GRCh37 package

Description

SNP positions and alleles for Homo sapiens extracted from NCBI dbSNP Build 144. The source data files used for this package were created by NCBI on May 29-30, 2015, and contain SNPs mapped to reference genome GRCh37.p13 (a patched version of GRCh37 that doesn't alter chromosomes 1-22, X, Y, MT).

Details

SNPs from dbSNP were filtered to keep only those satisfying the 3 following criteria:

1. The SNP is a single-base substitution i.e. its class is *snp*. Other classes supported by dbSNP are: *in-del*, *heterozygous*, *microsatellite*, *named-locus*, *no-variation*, *mixed*, and *multinucleotide-polymorphism*. These SNPs are NOT included in **SNPlocs.Hsapiens.dbSNP144.GRCh37** but are available in separate package **XtraSNPlocs.Hsapiens.dbSNP144.GRCh37**.
2. The SNP is marked as notwithdrawn.
3. A *single* position on the reference genome (GRCh37.p13) is reported for the SNP, and this position is on chromosome 1-22, X, Y, or MT.

SNPlocs packages always store the alleles corresponding to the *plus* strand, whatever the strand reported by dbSNP is (which is achieved by storing the complement of the alleles reported by dbSNP for SNPs located on the minus strand). In other words, in a SNPlocs package, all the SNPs are considered to be on the plus strand and everything is reported with respect to that strand.

Note

WARNING: The SNPs in this package are mapped to reference genome GRCh37.p13. Note that the GRCh37.p13 genome is a patched version of GRCh37. However the patch doesn't alter chromosomes 1-22, X, Y, MT. GRCh37 itself is the same as the hg19 genome from UCSC *except* for the mitochondrion chromosome. Therefore, the SNPs in this package can be "injected" in BSgenome.Hsapiens.UCSC.hg19 and will land at the correct position but this injection will exclude chrM (i.e. nothing will be injected in that sequence).

See ?injectSNPs in the **BSgenome** software package for more information about the SNP injection mechanism.

See https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.25/ for more information about the GRCh37.p13 assembly.

See http://genome.ucsc.edu/cgi-bin/hgGateway?db=hg19 for the UCSC Genome Browser based on the hg19 assembly. Note that chromosomes 1-22, X, and Y in hg19 and GRCh37.p13 are the same except that they are named differently (no chr prefix in GRCh37.p13).

Author(s)

H. Pagès

References

SNP Home at NCBI: https://www.ncbi.nlm.nih.gov/snp

dbSNP Human BUILD 144 announcement: https://www.ncbi.nlm.nih.gov/mailman/pipermail/dbsnp-announce/2015q2/000163.html

See Also

- The **XtraSNPlocs.Hsapiens.dbSNP144.GRCh37** package for SNPs of class other than *snp*.
- snpcount in the **BSgenome** software package for how to access the data stored in this package.
- IUPAC_CODE_MAP in the **Biostrings** package.
- The GRanges class in the **GenomicRanges** package.
- injectSNPs in the **BSgenome** software package for SNP injection.
- The **VariantAnnotation** software package to annotate variants with respect to location and amino acid coding.

Examples

```
## ---------------------------------------------------------------------
## A. BASIC USAGE
## ---------------------------------------------------------------------
snps <- SNPlocs.Hsapiens.dbSNP144.GRCh37
snpcount(snps)

## Get the positions and alleles of all SNPs on chromosome 22:
chr22_snps <- snpsBySeqname(snps, "22")
chr22_snps

## Get the positions and alleles of all SNPs on chromosomes 22 and MT:
snpsBySeqname(snps, c("22", "MT"))

## ---------------------------------------------------------------------
## B. EXTRACT SNP INFORMATION FOR A SET OF RS IDS
## ---------------------------------------------------------------------
my_rsids <- c("rs2639606", "rs75264089", "rs73396229", "rs55871206",
              "rs10932221", "rs56219727", "rs73709730", "rs55838886",
              "rs3734153", "rs79381275", "rs1516535")
my_snps <- snpsById(snps, my_rsids)
my_snps

## Translate the IUPAC ambiguity codes used to represent the alleles
## into nucleotides:
IUPAC_CODE_MAP[mcols(my_snps)$alleles_as_ambig]

## ---------------------------------------------------------------------
```

```
## C. INJECTION IN THE REFERENCE GENOME
## ---------------------------------------------------------------------
library(BSgenome.Hsapiens.UCSC.hg19)
genome <- BSgenome.Hsapiens.UCSC.hg19
genome

genome2 <- injectSNPs(genome, "SNPlocs.Hsapiens.dbSNP144.GRCh37")
genome2  # note the additional line "with SNPs injected from..."

alphabetFrequency(genome$chr22)
alphabetFrequency(genome2$chr22)

## Get the number of nucleotides that were modified by this injection:
neditAt(genome$chr22, genome2$chr22)  # 1835910

## ---------------------------------------------------------------------
## D. SOME BASIC QUALITY CONTROL (WITH SURPRISING RESULTS!)
## ---------------------------------------------------------------------

## Note that dbSNP can assign distinct ids to SNPs located at the same
## position:
any(duplicated(mcols(chr22_snps)$RefSNP_id))  # rs ids are all distinct...
any(duplicated(chr22_snps))  # but some positions are repeated!

which(duplicated(chr22_snps))[1:2]  # 14, 20
chr22_snps[12:15]  # rs2186463 and rs146752890 share the same position
                   # (16050612) and alleles (S, i.e. C/G)

## Also note that not all SNP alleles are consistent with the GRCh37
## genomic sequences, that is, the alleles reported for a given SNP are
## not necessarily compatible with the nucleotide found at the SNP
## position in GRCh37. For example, to get the number of inconsistent
## SNPs on chr1:
chr1_snps <- snpsBySeqname(snps, "1")
chr1_alleles <- mcols(chr1_snps)$alleles_as_ambig
chr1_alleles <- DNAString(paste(chr1_alleles, collapse=""))
nchar(chr1_alleles)  # 10608552 SNPs on chr1
neditAt(genome$chr1[pos(chr1_snps)], chr1_alleles, fixed=FALSE)
## ==> 5724 SNPs (0.054%) are inconsistent with GRCh37 chr1!
```

Index