# SNPlocs.Hsapiens.dbSNP.20120608

April 10, 2015

---

`getSNPlocs` *Accessing the SNPs stored in SNPlocs.Hsapiens.dbSNP.20120608*

---

## Description

Functions for accessing the SNPs stored in the SNPlocs.Hsapiens.dbSNP.20120608 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.<br>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.dbSNP.20120608 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. Pages

## See Also

- SNPlocs.Hsapiens.dbSNP.20120608
- IUPAC_CODE_MAP
- GRanges-class
- BSgenome-class
- injectSNPs
- findOverlaps

## Examples

```
## ---------------------------------------------------------------------
## A. BASIC USAGE
## ---------------------------------------------------------------------

getSNPcount()

## Get the locations and alleles of all SNPs on chromosome 22:
ch22snps <- getSNPlocs("ch22")
dim(ch22snps)
colnames(ch22snps)
head(ch22snps)

## Get the locations and alleles of all SNPs on chromosomes 22 and MT
## as a GRanges object:
getSNPlocs(c("ch22", "chMT"), as.GRanges=TRUE)

## ---------------------------------------------------------------------
## B. EXTRACT SNP INFORMATION FOR A SET OF RS IDS...
## ---------------------------------------------------------------------

## ... and return it in a GRanges object:
myrsids <- c("rs2639606", "rs75264089", "rs73396229", "rs55871206",
             "rs10932221", "rs56219727", "rs73709730", "rs55838886",
             "rs3734153", "rs79381275", "rs1516535")
gr <- rsidsToGRanges(myrsids)
IUPAC_CODE_MAP[mcols(gr)$alleles_as_ambig]

## ---------------------------------------------------------------------
## C. INJECTION IN THE REFERENCE GENOME
## ---------------------------------------------------------------------

library(BSgenome.Hsapiens.UCSC.hg19)
BSgenome.Hsapiens.UCSC.hg19
## Note that the chromosome names in BSgenome.Hsapiens.UCSC.hg19
## are those used by UCSC and they differ from those used by dbSNP.

## Inject the SNPs in hg19 (injectSNPs() "knows" how to map dbSNP
## chromosome names to UCSC names):
genome <- BSgenome.Hsapiens.UCSC.hg19
genome2 <- injectSNPs(genome, "SNPlocs.Hsapiens.dbSNP.20120608")
```

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

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

## Note that dbSNP can assign distinct ids to SNPs located at the same
## position:
any(duplicated(ch22snps$RefSNP_id))  # rs ids are all distinct...
any(duplicated(ch22snps$loc))  # but some locations are repeated!
ch22snps <- ch22snps[order(ch22snps$loc), ]  # sort by location
which(duplicated(ch22snps$loc))[1:2]  # 368,370
ch22snps[365:372, ]  # rs75929351 and rs150543489 share the same location
                     # (16079244) and alleles (K, i.e. G/T)

## Also note that not all SNP alleles are consistent with the hg19 genome
## i.e. the alleles reported for a given SNP are not always compatible
## with the nucleotide found at the SNP location in hg19.
## For example, to get the number of inconsistent SNPs in chr1:
ch1snps <- getSNPlocs("ch1")
all_alleles <- paste(ch1snps$alleles_as_ambig, collapse="")
nchar(all_alleles)  # 3517088 SNPs on chr1
neditAt(all_alleles, unmasked(genome$chr1)[ch1snps$loc], fixed=FALSE)
## ==> 3039 SNPs (0.086%) are inconsistent with hg19 chr1!

## Finally, lets check that no SNP falls in an assembly gap:
agaps <- masks(genome$chr1)$AGAPS
agaps  # the assembly gaps
## Looping over the assembly gaps:
sapply(1:length(agaps),
       function(i)
           any(ch1snps$loc >= start(agaps)[i] &
               ch1snps$loc <= end(agaps)[i]))
## Or, in a more efficient way:
stopifnot(length(findOverlaps(ch1snps$loc, agaps)) == 0)
```

---

SNPlocs.Hsapiens.dbSNP.20120608 *The SNPlocs.Hsapiens.dbSNP.20120608 package*

---

## Description

This package contains SNP locations and alleles for Homo sapiens extracted from dbSNP Build 137.

## Details

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

- The SNP is a single-base substitution i.e. its type is "snp". Other types used by dbSNP are: "in-del", "mixed", "microsatellite", "named-locus", "multinucleotide-polymorphism", etc... All those SNPs were dropped.
- The SNP is marked as notwithdrawn.
- A *single* location on the reference genome (GRCh37.p5) is reported for the SNP, and this location is on chromosomes 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

The source data files used for this package were created by the dbSNP Development Team at NCBI on June 7-8, 2012.

WARNING: The SNPs in this package are mapped to reference genome GRCh37.p5. Note that the GRCh37.p5 genome is a patched version of GRCh37 but 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 they will land at the correct location but this injection will exclude chrM (i.e. nothing will be injected in that sequence).

See http://www.ncbi.nlm.nih.gov/genome/guide/human/release_notes.html for more information about the GRCh37.p5 assembly.

See http://www.ncbi.nlm.nih.gov/snp, the SNP Home at NCBI, for more information about dbSNP.

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

See http://genome.ucsc.edu/cgi-bin/hgGateway?clade=mammal&org=Human&db=hg19 for more information about the Human Feb. 2009 (GRCh37/hg19) assembly used by the UCSC Genome Browser.

## Author(s)

H. Pages

## References

Human genome at NCBI with details about the GRCh37.p5 assembly: http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/human/ http://www.ncbi.nlm.nih.gov/genome/guide/human/release_notes.html

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

dbSNP Build 137 announcements: http://www.ncbi.nlm.nih.gov/mailman/pipermail/dbsnp-announce/2012q2/thread.html About the Human Feb. 2009 (GRCh37/hg19) assembly used by the UCSC

Genome Browser: http://genome.ucsc.edu/cgi-bin/hgGateway?clade=mammal&org=Human&db=hg19

## See Also

- getSNPlocs for how to access the data stored in this package.
- injectSNPs in the BSgenome package for more information about SNP injection.
- The VariantAnnotation software package to annotate variants with respect to location and amino acid coding.

# Index