SNPlocs.Hsapiens.dbSNP.20101109

October 17, 2011

`SNPlocs.Hsapiens.dbSNP.20101109`
SNP locations and alleles for Homo sapiens (dbSNP Build 132)

Description

SNP locations and alleles for Homo sapiens extracted from dbSNP Build 132.

Usage

```
## Datasets:
data(SNPcount)
data(ch1_snplocs)
data(ch2_snplocs)
data(ch3_snplocs)
data(ch4_snplocs)
data(ch5_snplocs)
data(ch6_snplocs)
data(ch7_snplocs)
data(ch8_snplocs)
data(ch9_snplocs)
data(ch10_snplocs)
data(ch11_snplocs)
data(ch12_snplocs)
data(ch13_snplocs)
data(ch14_snplocs)
data(ch15_snplocs)
data(ch16_snplocs)
data(ch17_snplocs)
data(ch18_snplocs)
data(ch19_snplocs)
data(ch20_snplocs)
data(ch21_snplocs)
data(ch22_snplocs)
data(chX_snplocs)
data(chY_snplocs)
data(chMT_snplocs)
```

```
## Convenience wrappers for loading the SNP data:
getSNPcount()
getSNPlocs(seqname, as.GRanges=FALSE)
```

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).

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) is reported for the SNP.

In this package, the SNP data are split by chromosome 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 following columns:

- `RefSNP_id`: RefSNP ID. Character vector with no NAs and no duplicates.
- `alleles_as_ambig`: The alleles for a given SNP are represented by an IUPAC nucleotide ambiguity code (see `?IUPAC_CODE_MAP` in the Biostrings package for more information). Character vector with no NAs.
- `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.

`getSNPcount` and `getSNPlocs` are just convenience wrappers for loading the SNP data.

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 data frame containing the SNP data for the specified chromosome (description of this data frame is given above). 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`.

Note

The source data files used for this package were created by the dbSNP Development Team at NCBI on 9 November 2010.

WARNING: The SNPs in this package are mapped to reference genome GRCh37. Note that the GRCh37 genome 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 but this injection will exclude chrM (i.e. nothing will be injected in that sequence).

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

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

dbSNP Build 132 announcement: http://www.ncbi.nlm.nih.gov/mailman/pipermail/dbsnp-announce/2010q4/000097.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

BSgenome-class, injectSNPs, GRanges-class, findOverlaps, IUPAC_CODE_MAP

Examples

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

getSNPcount()
ch22snps <- getSNPlocs("ch22")
dim(ch22snps)
colnames(ch22snps)
head(ch22snps)

## Get the SNP locations and alleles as a GRanges object:
getSNPlocs(c("ch22", "chMT"), as.GRanges=TRUE)

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

library(BSgenome.Hsapiens.UCSC.hg19)
```

```
## Inject the SNPs in hg19:
Hs2 <- injectSNPs(Hsapiens, "SNPlocs.Hsapiens.dbSNP.20101109")
Hs2
alphabetFrequency(unmasked(Hs2$chr1))
alphabetFrequency(unmasked(Hsapiens$chr1))
## Get the number of nucleotides that were modified by this injection:
neditAt(unmasked(Hs2$chr1), unmasked(Hsapiens$chr1))

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

## Note that dbSNP can assign distinct ids to SNPs that are located at
## the same position:
any(duplicated(ch22snps$RefSNP_id))  # ids are all distinct...
any(duplicated(ch22snps$loc))  # but some locations are repeated!

## 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$alleles_as_ambig, collapse="")
nchar(all_alleles)  # 1849438 SNPs on chr1
neditAt(all_alleles, unmasked(Hsapiens$chr1)[ch1snps$loc], fixed=FALSE)
## ==> 3181 SNPs (0.17%) are inconsistent with hg19 chr1!

## Finally, let's check that no SNP falls in an assembly gap:
agaps <- masks(Hsapiens$chr1)$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:
length(findOverlaps(ch1snps$loc, agaps))  # 0
```

Index