Package ‘gwascat’

March 13, 2024

Title representing and modeling data in the EMBL-EBI GWAS catalog

Version 2.34.0

Author VJ Carey <stvjc@channing.harvard.edu>

Description Represent and model data in the EMBL-EBI GWAS catalog.

Enhances SNPlocs.Hsapiens.dbSNP144.GRCh37

Depends R (>= 3.5.0), methods

Imports S4Vectors (>= 0.9.25), IRanges, GenomeInfoDb, GenomicRanges (>= 1.29.6), GenomicFeatures, readr, Biostrings, AnnotationDbi, BiocFileCache, snpStats, VariantAnnotation, AnnotationHub

Suggests DO.db, DT, knitr, RBGL, testthat, rmarkdown, dplyr, Gviz, Rsamtools, rtracklayer, graph, ggbiom, DelayedArray, TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.eg.db, BiocStyle

VignetteBuilder knitr

Maintainer VJ Carey <stvjc@channing.harvard.edu>

License Artistic-2.0

LazyData no

biocViews Genetics

RoxygenNote 7.2.1

Encoding UTF-8

git_url https://git.bioconductor.org/packages/gwascat

git_branch RELEASE_3_18

git_last_commit 9252d33

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-03-13
### R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>as_GRanges</td>
<td>2</td>
</tr>
<tr>
<td>bindcadd_snv</td>
<td>3</td>
</tr>
<tr>
<td>chklloc</td>
<td>4</td>
</tr>
<tr>
<td>ebicat_2020_04_30</td>
<td>5</td>
</tr>
<tr>
<td>g17SM</td>
<td>5</td>
</tr>
<tr>
<td>getRsids</td>
<td>5</td>
</tr>
<tr>
<td>getRsids,gwaswloc-method</td>
<td>6</td>
</tr>
<tr>
<td>getTraits</td>
<td>6</td>
</tr>
<tr>
<td>getTraits,gwaswloc-method</td>
<td>7</td>
</tr>
<tr>
<td>get_cached_gwascat</td>
<td>7</td>
</tr>
<tr>
<td>gg17N</td>
<td>8</td>
</tr>
<tr>
<td>gr6.0_hg38</td>
<td>8</td>
</tr>
<tr>
<td>gw6.rs_17</td>
<td>9</td>
</tr>
<tr>
<td>gwascat_from_AHub</td>
<td>9</td>
</tr>
<tr>
<td>gwastagger</td>
<td>10</td>
</tr>
<tr>
<td>gwaswloc-class</td>
<td>10</td>
</tr>
<tr>
<td>gwcat_snapshot</td>
<td>10</td>
</tr>
<tr>
<td>gwcex2gviz</td>
<td>11</td>
</tr>
<tr>
<td>ldtagr</td>
<td>12</td>
</tr>
<tr>
<td>locon6</td>
<td>13</td>
</tr>
<tr>
<td>locs4trait</td>
<td>13</td>
</tr>
<tr>
<td>low17</td>
<td>14</td>
</tr>
<tr>
<td>makeCurrentGwascat</td>
<td>14</td>
</tr>
<tr>
<td>obo2graphNEL</td>
<td>15</td>
</tr>
<tr>
<td>process_gwas_dataframe</td>
<td>17</td>
</tr>
<tr>
<td>riskyAlleleCount</td>
<td>17</td>
</tr>
<tr>
<td>si.hs.37</td>
<td>18</td>
</tr>
<tr>
<td>si.hs.38</td>
<td>18</td>
</tr>
<tr>
<td>subsetByChromosome</td>
<td>19</td>
</tr>
<tr>
<td>subsetByChromosome,gwaswloc-method</td>
<td>19</td>
</tr>
<tr>
<td>subsetByTraits</td>
<td>20</td>
</tr>
<tr>
<td>subsetByTraits,gwaswloc-method</td>
<td>20</td>
</tr>
<tr>
<td>topTraits</td>
<td>21</td>
</tr>
<tr>
<td>traitsManh</td>
<td>21</td>
</tr>
<tr>
<td>[.gwaswloc,ANY,ANY,ANY-method</td>
<td>23</td>
</tr>
</tbody>
</table>

### Description

produce a GRanges from gwascat tibble
bindcadd_snv

Usage

```r
as_GRanges(
  x,
  short = TRUE,
  for_short = c("PUBMEDID", "DATE", "DISEASE/TRAIT", "SNPS"),
  genome_tag = "GRCh38"
)
```

Arguments

- **x**: a tibble from `get_cached_gwascat()`
- **short**: logical(1) if TRUE only keep selected columns in mcols
- **for_short**: character() column names to keep in mcols
- **genome_tag**: character(1) defaults to "GRCh38"

Description

bind CADD scores of Kircher et al. 2014 to a GRanges instance; by default will use HTTP access at UW

Usage

```r
bindcadd_snv(
  gr,
  fn = "http://krishna.gs.washington.edu/download/CADD/v1.0/1000G.tsv.gz"
)
```

Arguments

- **gr**: query ranges to which CADD scores should be bound
- **fn**: path to Tabix-indexed bgzipped TSV of CADD as distributed at krishna.gs.washington.edu on 1 April 2014

Details

joins CADD fields at addresses that match query; the CADD fields for query ranges that are not matched are set to NA

Value

GRanges instance with additional fields as obtained in the CADD resource
chklocs

Note

This software developed in part with support from Genentech, Inc.

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

References

M Kircher, DM Witten, P Jain, BJ O’Roak, GM Cooper, J Shendure, A general framework for estimating the relative pathogenicity of human genetic variants, Nature Genetics Feb 2014, PMID 24487276

Examples

```r
## Not run:
data(ebicat_2020_04_30)
g2 = as(ebicat_2020_04_30, "GRanges")
# would need to lift over here
bindcadd_snv( g2[which(seqnames(g2)=="chr2")][1:20] )
## End(Not run)
```

chklocs

return TRUE if all named SNPs with locations in both the SNPlocs package and the gwascat agree

Description

return TRUE if all named SNPs with locations in both the SNPlocs package and the gwascat agree

Usage

```r
chklocs(chrtag = "20", gwwl = gwrngs19)
```

Arguments

- `chrtag`: character, chromosome identifier
- `gwwl`: instance of `gwaswloc`
ebicat_2020_04_30

**ebicat_2020_04_30**  
*serialized gwaswloc instance from april 30 2020, sample of 50000 records*

**Description**  
serialized gwaswloc instance from april 30 2020, sample of 50000 records

**Usage**  
`ebicat_2020_04_30`

**Format**  
gwaswloc instance

**g17SM**  
*SnpMatrix instance from chr17*

**Description**  
SnpMatrix instance from chr17

**Usage**  
g17SM

**Format**  
snpStats SnpMatrix instance

**getRsids**  
*generic snp name retrieval*

**Description**  
generic snp name retrieval

**Usage**  
`getRsids(x)`

**Arguments**  
x  
gwaswloc
getRsids, gwaswloc-method

*specific snp name retrieval*

---

**Description**

specific snp name retrieval

**Usage**

```r
## S4 method for signature 'gwaswloc'
getRsids(x)
```

**Arguments**

- `x`  
gwaswloc

---

getTraits  

*generic trait retrieval*

---

**Description**

generic trait retrieval

**Usage**

```
getTraits(x)
```

**Arguments**

- `x`  
gwaswloc
getTraits, gwaswloc-method

Description

specific trait retrieval

Usage

```r
## S4 method for signature 'gwaswloc'
getTraits(x)
```

Arguments

- `x`: gwaswloc

get_cached_gwascat

Description

use BiocFileCache to retrieve and keep an image of the tsv file distributed by EBI

Usage

```r
get_cached_gwascat(
  url = "http://www.ebi.ac.uk/gwas/api/search/downloads/alternative",
  cache = BiocFileCache::BiocFileCache(),
  refresh = FALSE,
  ...
)
```

Arguments

- `url`: character(1) url to use
- `cache`: BiocFileCache::BiocFileCache instance
- `refresh`: logical(1) force download and recaching
- `...`: passed to bfcadd

Value

a tibble as produced by readr::read_tsv, with attributes extractDate (as recorded in cache as 'access_time'), and problems (a tibble returned by read_tsv).
Note

If query of cache with 'ebi.ac.uk/gwas' returns 0-row tibble, will populate cache with bfcadd. Uses readr::read_tsv on cache content to return tibble. The etag field does not seem to be used at EBI, thus user must check for updates.

---

**gg17N**

*genotype matrix from chr17 1000 genomes*

---

**Description**

genotype matrix from chr17 1000 genomes

**Usage**

```r
gg17N
```

**Format**

matrix

**Examples**

```r
data(gg17N)
gg17N[1:4,1:4]
```

---

**gr6.0_hg38**

*image of locon6 in GRanges, lifted over to hg38*

---

**Description**

image of locon6 in GRanges, lifted over to hg38

**Usage**

```r
gr6.0_hg38
```

**Format**

GRanges instance
**gw6.rs_17**

character vector of rs numbers for SNP on chr17

**Description**

character vector of rs numbers for SNP on chr17

**Usage**

gw6.rs_17

**Format**

character vector

---

**gwascat_from_AHub**

grab an image of EBI GWAS catalog from AnnotationHub

**Description**

grab an image of EBI GWAS catalog from AnnotationHub

**Usage**

gwascat_from_AHub(tag = "AH91571", simple = FALSE, fixNonASCII = TRUE)

**Arguments**

- **tag** character(1) defaults to "AH91571" which is the 3.30.2021 image
- **simple** logical(1) if TRUE, just returns data.frame as retrieved from EBI; defaults to FALSE
- **fixNonASCII** logical(1) if TRUE, use iconv to identify and eliminate non-ASCII content

**Value**

If 'simple', a data.frame is returned based on TSV data produced by EBI. Otherwise, non-ASCII content is processed according to the value of 'fixNonASCII' and a 'gwaswloc' instance is returned, which has a concise show method. This can be coerced to a simple GRanges instance with as(..., "GRanges"). The reference build is GRCh38.

**Examples**

gwcat = gwascat_from_AHub()
gwcat
gwastagger

**Description**

GRanges with LD information on 9998 SNP

**Usage**

gwastagger

**Format**

GRanges
gwaswloc-class: container for gwas hit data and GRanges for addresses

---

gwcat_snapshot

**Description**

use AnnotationHub snapshot as basis for gwaswloc structure creation

**Usage**

`gwcat_snapshot(x, fixNonASCII = TRUE)`

**Arguments**

- `x`: inherits from data.frame, with columns consistent with EBI table
- `fixNonASCII`: logical(1) if TRUE, use iconv to replace non-ASCII character, important for CMD check but perhaps not important for applied use
gwcex2gviz

Examples

```r
ah = AnnotationHub::AnnotationHub()
entitytab = AnnotationHub::query(ah, "gwascatData")
cand = names(entitytab)[1]
stopifnot(nchar(cand)>0)
tab = ah[[cand]]
gww = gwcex2gviz(tab)
gww
```

---

**gwcex2gviz**

Prepare salient components of GWAS catalog for rendering with Gviz

### Description

Prepare salient components of GWAS catalog for rendering with Gviz

### Usage

```r
gwcex2gviz(
  basegr,
  contextGR = GRanges(seqnames = "chr17", IRanges::IRanges(start = 37500000, width = 1e+06)),
  txrefobj = TxDb.Hsapiens.UCSC.hg19.knownGene::TxDb.Hsapiens.UCSC.hg19.knownGene,
  genome = "hg19",
  genesymobj = org.Hs.eg.db::org.Hs.eg.db,
  plot.it = TRUE,
  maxmlp = 25
)
```

### Arguments

- **basegr**: gwaswloc instance containing information about GWAS in catalog
- **contextGR**: A GRanges instance delimiting the visualization in genomic coordinates
- **txrefobj**: a TxDb instance
- **genome**: character tag like ‘hg19’
- **genesymobj**: an OrgDb instance
- **plot.it**: logical, if FALSE, just return list
- **maxmlp**: maximum value of -10 log p – winsorization of all larger values is performed, modifying the contents of Pvalue\_mlogp in the elementMetadata for the call

### Examples

```r
data(ebicat_2020_04_30)
# GenomeInfoDb::seqlevelsStyle(ebicat_2020_04_30) = "UCSC" # no more
GenomeInfoDb::seqlevels(ebicat_2020_04_30) = paste0("chr", GenomeInfoDb::seqlevels(ebicat_2020_04_30))
gwcex2gviz(ebicat_2020_04_30)
```
ldtagr

expand a list of variants by including those in a VCF with LD exceeding some threshold; uses snpStats ld()

Usage

```r
ldtagr(
  snprng,
  tf,
  samples,
  genome = "hg19",
  lbmaf = 0.05,
  lbR2 = 0.8,
  radius = 1e+05
)
```

Arguments

- `snprng`: a named GRanges for a single SNP. The name must correspond to the name that will be assigned by genotypeToSnpMatrix (from VariantTools) to the corresponding column of a SnpMatrix.
- `tf`: TabixFile instance pointing to a bgzipped tabix-indexed VCF file
- `samples`: a vector of sample identifiers, if excluded, all samples used
- `genome`: tag like 'hg19'
- `lbmaf`: lower bound on variant MAF to allow consideration
- `lbR2`: lower bound on R squared for regarding SNP to be incorporated
- `radius`: radius of search in bp around the input range

Value

a GRanges with names corresponding to 'new' variants and mcols fields 'paramRangeID' (base variant input) and 'R2'

Note

slow but safe approach. probably a matrix method could be substituted using the nice sparse approach already in snpStats

Author(s)

VJ Carey
Examples

cand = GenomicRanges::GRanges("1", IRanges::IRanges(113038694, width=1))
names(cand) = "rs883593"
requireNamespace("VariantAnnotation")
expath = dir(system.file("vcf", package="gwascat"), patt=".*exon.*gz\$", full=TRUE)
tf = Rsamtools::TabixFile(expath)
ldtagr(cand, tf, lbR2 = .8)

Description

location data for 10000 SNP

Usage

locon6

Format

data.frame, coordinates are hg19

locs4trait

get locations for SNP affecting a selected trait

Description

get locations for SNP affecting a selected trait

Usage

locs4trait(gwwl, trait, tag = "DISEASE/TRAIT")

Arguments

gwwl instance of {gwaswloc}
trait character, name of trait
tag character, name of field to be used for trait enumeration
makeCurrentGwascat

10w17

SnpMatrix instance from chr17

Description

SnpMatrix instance from chr17

Usage

10w17

Format

snpStats SnpMatrix instance

makeCurrentGwascat

read NHGRI GWAS catalog table and construct associated GRanges instance records for which clear genomic position cannot be determined are dropped from the ranges instance an effort is made to use reasonable data types for GRanges metadata, so some qualifying characters such as (EA) in Risk allele frequency field will simply be omitted during coercion of contents of that field to numeric.

Description

read NHGRI GWAS catalog table and construct associated GRanges instance records for which clear genomic position cannot be determined are dropped from the ranges instance an effort is made to use reasonable data types for GRanges metadata, so some qualifying characters such as (EA) in Risk allele frequency field will simply be omitted during coercion of contents of that field to numeric.

Usage

makeCurrentGwascat(
    table.url = "http://www.ebi.ac.uk/gwas/api/search/downloads/alternative",
    fixNonASCII = TRUE,
    genome = "GRCh38",
    withOnt = TRUE
)
obo2graphNEL

Arguments

table.url  string identifying the .txt file curated at EBI/EMBL
fixNonASCII  logical, if TRUE, non-ASCII characters as identified by iconv will be replaced by asterisk
genome  character string: 'GRCh38' is default and yields current image as provided by EMBL/EBI; 'GRCh37' yields a realtime liftOver to hg19 coordinates, via AnnotationHub storage of the chain files. Any other value yields an error.
withOnt  logical indicating whether 'alternative' (ontology-present, includes repetition of loci with one:many ontological mapping) or 'full' (ontology-absent, one record per locus report) version of distributed table

Value

a slightly extended GRanges instance, with class name 'gwaswloc'; the purpose of the introduction of this class is to support a concise show method that does not produce very long lines owing to large numbers of fields in the mcols component.

Note

'readr::read_tsv' records problems when some records have field contents that are inconsistent with the column specification. This information can be retrieved from the metadata slot of the returned object, as noted in a message produced when this function is run.

Author(s)

VJ Carey

Examples

```r
# if you have good internet access
if (interactive()) {
  newcatr = makeCurrentGwascat()
  newcatr
}
```

obo2graphNEL  convert a typical OBO text file to a graphNEL instance (using Term elements)

Description

convert a typical OBO text file to a graphNEL instance (using Term elements)
Usage

obo2graphNEL(
    obo = "human-phenotype-ontology.obo",
    kill = "\[Typedef\]",
    killTrailSp = TRUE
)

Arguments

obo string naming a file in OBO format
kill entity types to be excluded from processing – probably this should be in a 'keep' form, but for now this works.
killTrailSp In the textual version of EFO ca. Aug 2015, there is a trailing blank in the tag field defining EFO:0000001, which is not present in references to this term. Set this to TRUE to eliminate this, or graphNEL construction will fail to validate.

Details

Very rudimentary list and grep operations are used to retain sufficient information to map the DAG to a graphNEL, using formal term identifiers as node names and 'is-a' relationships as edges, and term names and other metadata are assigned to nodeData components.

Value

a graphNEL instance

Note

The OBO for Human Disease ontology is serialized as text with this package.

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

References

For use with human disease ontology, http://www.obofoundry.org/cgi-bin/detail.cgi?id=disease_ontology

Examples

data(efo.obo.g)
requireNamespace("graph")
hn = graph::nodes(efo.obo.g)[1:5]
hn
graph::nodeData(efo.obo.g, hn[5])
process_gwas_dataframe

convert GWAS catalog data.frame to gwaswloc, a GRanges extension with simple show method

Description

convert GWAS catalog data.frame to gwaswloc, a GRanges extension with simple show method

Usage

process_gwas_dataframe(df)

Arguments

df data.frame

riskyAlleleCount given a matrix of subjects x SNP calls, count number of risky alleles

Description

given a matrix of subjects x SNP calls, count number of risky alleles for various conditions, relative to NHGRI GWAS catalog

Usage

riskyAlleleCount(
callmat,
matIsAB = TRUE,
chr,
gwwl,
snpap = "SNPlocs.Hsapiens.dbSNP144.GRCh37",
gencode = c("A/A", "A/B", "B/B")
)

Arguments

callmat matrix with subjects as rows, SNPs as columns; entries can be generic A/A, A/B, B/B, or specific nucleotide calls
matIsAB logical, FALSE if nucleotide codes are present, TRUE if generic call codes are present; in the latter case, gwascat::ABmat2nuc will be run
chr code for chromosome, should work with the SNP annotation getSNPlocs function, so likely "ch[nn]"
gwwl an instance of {gwaswloc}
snpap name of a Bioconductor SNPlocs.Hsapiens.dbSNP.* package
gencode codes used for generic SNP call
Value

matrix with rows corresponding to subjects, columns corresponding to SNP

Examples

```r
# Not run:
data(gg17N) # translated from GGdata chr 17 calls using ABmat2nuc
data(ebicat37)
library(GenomeInfoDb)
seqlevelsStyle(ebicat37) = "UCSC"
h17 = riskyAlleleCount(gg17N, matIsAB=FALSE, chr="ch17", gwwl=ebicat37)
h17[1:5,1:5]
[table(as.numeric(h17))]

## End(Not run)
```

---

si.hs.37  Seqinfo for GRCh37

Description

Seqinfo for GRCh37

Usage

```r
si.hs.37
```

Format

```
GenomeInfoDb Seqinfo instance
```

---

si.hs.38  Seqinfo for GRCh38

Description

Seqinfo for GRCh38

Usage

```r
si.hs.38
```

Format

```
GenomeInfoDb Seqinfo instance
```
subsetByChromosome  generic trait subsetting

Description

generic trait subsetting

Usage

subsetByChromosome(x, ch)

Arguments

x  gwaswloc
ch  character vector of chromosomes

subsetByChromosome,gwaswloc-method  specific trait subsetting

Description

specific trait subsetting

Usage

## S4 method for signature 'gwaswloc'
subsetByChromosome(x, ch)

Arguments

x  gwaswloc
ch  character vector of chromosomes
### subsetByTraits

**Description**

generic trait subsetting

**Usage**

```r
subsetByTraits(x, tr)
```

**Arguments**

- `x`: gwaswloc
- `tr`: character vector of traits

---

### subsetByTraits,gwaswloc-method

**Description**

specific trait subsetting

**Usage**

```r
## S4 method for signature 'gwaswloc'
subsetByTraits(x, tr)
```

**Arguments**

- `x`: gwaswloc
- `tr`: character vector of traits
topTraits

operations on GWAS catalog

Description

operations on GWAS catalog

Usage

topTraits(gwwl, n = 10, tag = "DISEASE/TRAIT")

Arguments

gwwl instance of {gwaswloc}
n numeric, number of traits to report
tag character, name of field to be used for trait enumeration

Value

topTraits returns a character vector of most frequently occurring traits in the database
locs4trait returns a {gwaswloc} object with records defining associations to the specified trait
chklocs returns a logical that is TRUE when the asserted locations of SNP in the GWAS catalog agree with the locations given in the dbSNP package SNPlocs.Hsapiens.dbSNP144.GRCh37

Author(s)

VJ Carey <stvjc@channing.harvard.edu>

Examples

data(ebicat_2020_04_30)
topTraits(ebicat_2020_04_30)

traitsManh use ggbio facilities to display GWAS results for selected traits in genomic coordinates

Description

use ggbio facilities to display GWAS results for selected traits in genomic coordinates
traitsManh

Usage

traitsManh(
  gwr,
  selr = GRanges(seqnames = "chr17", IRanges(3e+07, 5e+07)),
  traits = c("Asthma", "Parkinson's disease", "Height", "Crohn's disease"),
  truncmlp = 25,
  ...
)

Arguments

gwr GRanges instance as managed by the gwaswloc container design, with Disease.Trait and Pvalue\_mlog among elementMetadata columns

selr A GRanges instance to restrict the gwr for visualization. Not tested for noncontiguous regions.

traits Character vector of traits to be exhibited; GWAS results with traits not among these will be labeled "other".

truncmlp Maximum value of -log10 p to be displayed; in the raw data this ranges to the hundreds and can cause bad compression.

... not currently used

Details

uses a ggbio autoplot

Value

autoplot value

Note

An xlab is added, concatenating genome tag with seqnames tag.

Author(s)

VJ Carey <stvjcarey@channing.harvard.edu>

Examples

# do a p-value truncation if you want to reduce compression
## Not run: # ggbio July 2018
data(ebicat_2020_04_30)
library(GenomeInfoDb)
seqlevelsStyle(ebicat_2020_04_30) = "UCSC"
traitsManh(ebicat_2020_04_30)

## End(Not run)
### Description

extractor for gwaswloc

#### Usage

```r
## S4 method for signature 'gwaswloc,ANY,ANY,ANY'
x[i, j, ..., drop = FALSE]
```

#### Arguments

- `x` gwaswloc
- `i` index
- `j` index
- `...` addtl indices
- `drop` logical(1)
Index

* datasets
  ebicat_2020_04_30, 5
  g17SM, 5
gg17N, 8
  gr6.0_hg38, 8
gw6.rs_17, 9
gwastagger, 10
  locon6, 13
  low17, 14
  si.hs.37, 18
  si.hs.38, 18

* graphics
  gwex2gviz, 11
  traitsManh, 21

* models
  bindcadd_snv, 3
  ldtagr, 12
  makeCurrentGwascat, 14
  obo2graphNEL, 15
  riskyAlleleCount, 17
  topTraits, 21
  traitsManh, 21
  [,gwaswloc,ANY,ANY,ANY-method, 23
  as_GRanges, 2
  bindcadd_snv, 3
  chklocs, 4
  ebicat_2020_04_30, 5
efo.obo.g (obo2graphNEL). 15
  g17SM, 5
  get_cached_gwascat, 7
  getRsids, 5
  getRsids,gwaswloc-method, 6
  getTraits, 6
  getTraits,gwaswloc-method, 7
  gg17N, 8
  gr6.0_hg38, 8
gw6.rs_17, 9
  gwascat_from_AHub, 9
gwastagger, 10
gwaswloc-class, 10
gwcat_snapshot, 10
gwex2gviz, 11
  ldtagr, 12
  locon6, 13
  locs4trait, 13
  low17, 14
  makeCurrentGwascat, 14
  node2uri (obo2graphNEL), 15
  obo2graphNEL, 15
  process_gwas_dataframe, 17
  riskyAlleleCount, 17
  si.hs.37, 18
  si.hs.38, 18
  subsetByChromosome, 19
  subsetByChromosome,gwaswloc-method, 19
  subsetByTraits, 20
  subsetByTraits,gwaswloc-method, 20
  topTraits, 21
  traitsManh, 21
  uri2node (obo2graphNEL), 15