SomaScan.db

November 11, 2026

`DataObjects` *SomaScan Menu Character Vectors*

Description

The `somascan_menu` object provides users with a quick and easy method for retrieving analytes available in SomaLogic's SomaScan assay menus. The object is available whenever the `SomaScan.db` package is loaded.

Format

A list containing the following elements:

v4.0_analytes A character vector of SomaScan analytes from V4.0 of the SomaScan menu (commonly referred to as "5k"). The V4.0 vector contains 4,966 elements in `SeqId` format (e.g. `1234-56`).

v4.1_analytes A character vector of SomaScan analytes from V4.1 of the SomaScan menu (commonly referred to as "7k"). The V4.1 vector contains 7,267 elements in `SeqId` format (e.g. `1234-56`).

v5.0_analytes A character vector of SomaScan analytes from V5.0 of the SomaScan menu (commonly referred to as "11k"). The V5.0 vector contains 10,731 elements in `SeqId` format (e.g. `1234-56`).

Data Description

The `somascan_menu` object is a list of character vectors of analytes available in each SomaScan menu. The currently available vectors in the `somascan_menu` object correspond to the V4.0 ("5k"), V4.1 ("7k"), and V5.0 ("11k") menus, and are named accordingly. Older menus (ex. the 5k menu) contain fewer analytes than more recent releases (ex. the 11k menu), and each subsequent release introduces new analytes targeting additional protein epitopes. Consequently, the 7k menu contains all analytes present in the 5k menu, but the 5k menu does not contain all analytes in the 7k menu.

Source

https://github.com/SomaLogic/SomaLogic-Data

SomaLogic Operating Co., Inc.

Examples

```
# View the structure of the list and its elements (i.e. SomaScan menus)
summary(somascan_menu)

# Preview each element in the list
str(somascan_menu)
```

SomaDb-class *The SomaDb S4 class*

Description

The SomaDb class is a subclass of the ChipDb (and, by extension, the AnnotationDb) class defined in the AnnotationDbi package. It is designed to hold annotations for SomaScan data.

Value

`keys`, `columns`, and `keytypes` each return a character vector or possible values. `select` returns a data.frame and `mapIds` returns a named vector.

SomaScanALIAS2PROBE *Map between Common Gene Symbol Identifiers and SomaScan SeqIds*

Description

`SomaScanALIAS2PROBE` is an R object that provides mappings between common gene symbol identifiers and manufacturer identifiers, i.e. the SomaScan `SeqId`.

Details

Each gene symbol is mapped to a named vector of `SeqIds`. The name of each vector element represents the gene symbol, and the vector contains all manufacturer identifiers that are found for that symbol. An `NA` is reported for any gene symbol that cannot be mapped to a `SeqId`.

This mapping includes **all** gene symbols, including those which are already listed in the `SomaScanSYMBOL` map. The `SomaScanSYMBOL` map is meant to only list *official* gene symbols, while the `SomaScanALIAS` maps are meant to store all *used* symbols.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
# Convert the object to a list
xx <- as.list(SomaScanALIAS2PROBE)
if(length(xx) > 0){
    # Get the probe identifiers for the first two aliases
    xx[1:2]
    # Get the first value
    xx[[1]]
}
```

SomaScan.db *Bioconductor annotation data package for SomaScan*

Description

Welcome to the SomaScan.db annotation package. The purpose of this package is to provide detailed annotation information for SomaLogic's SomaScan platform. This package is updated biannually.

Objects in this package are accessed using the select() interface. See this package's vignettes for more details, or reference ?AnnotationDbi::select from the AnnotationDbi package.

Value

A platform-centric annotation package.

See Also

AnnotationDb-class for use of keys(), columns() and select().

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.
columns(SomaScan.db)

## Bimap interface:
# The 'old style' of interacting with these objects is manipulation as
# bimaps. While this approach is still available we strongly encourage the
# use of select().
ls("package:SomaScan.db")
```

SomaScanENSEMBL *Map Ensembl gene accession numbers with Entrez Gene identifiers*

Description

SomaScanENSEMBL is an R object that contains mappings between SeqIds and Ensembl gene accession numbers.

Details

This object is a simple mapping of SeqIds to Ensembl gene accession numbers.

For most species, this mapping is a combination of SeqIds to Ensembl IDs from *both* NCBI and Ensembl. Users who wish to only use mappings from NCBI are encouraged to see the ncbi2ensembl table in the appropriate organism package. Users who wish to only use mappings from Ensembl are encouraged to see the ensembl2ncbi table which, like ncbi2ensembl, is found in the appropriate organism package. These mappings are based upon the Ensembl table, which contains data from both of these sources in an effort to maximize the chances of finding a match.

Mappings were based on data provided by both of these sources:

- http://www.ensembl.org/biomart/martview/
- ftp://ftp.ncbi.nlm.nih.gov/gene/DATA

Value

A Bimap object of the ProbeAnnDbBimap class.

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanENSEMBL
# Get the entrez gene IDs that are mapped to an Ensembl ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes][1:300])
if(length(xx) > 0) {
    # Get the Ensembl gene IDs for the first five genes
    xx[1:5]
    # Get the first value
    xx[[1]]
}

# For the reverse map ENSEMBL2PROBE:
x <- SomaScanENSEMBL2PROBE
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes][1:300])
if(length(xx) > 0){
    # Gets the entrez gene IDs for the first five Ensembl IDs
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanENTREZID *Map between SomaScan SeqIds and Entrez gene identifiers*

Description

SomaScanENTREZID is an R object that provides mappings between SeqIds and Entrez Gene identifiers.

Details

Each SeqId is mapped to a vector of Entrez Gene identifiers. An NA is assigned to those SeqIds that can not be mapped to an Entrez Gene identifier at this time.

If a given SeqId can be mapped to different Entrez Gene identifiers from various sources, we attempt to select the common identifiers. If a concensus cannot be determined, we select the smallest identifier.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanENTREZID
# Get the probe identifiers that are mapped to an ENTREZ Gene ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the ENTREZID for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanENZYME *Map between SomaScan SeqIds and Enzyme Commission (EC) numbers*

Description

SomaScanENZYME is an R object that provides mappings between SeqIds and EC numbers. SomaScanENZYME2PROBE is an R object that maps Enzyme Commission (EC) numbers to SeqIds.

Details

When the SomaScanENZYME mapping viewed as a list, each manufacturer identifier maps to a named vector containing the EC number that corresponds to the enzyme produced by that gene. The names correspond to the SeqIds. If this information is unknown, the vector will contain an NA.

For the SomaScanENZYME2PROBE object, each EC number maps to a named vector containing all of the SeqIds that correspond to the gene that produces that enzyme. The name of the vector corresponds to the EC number.

EC numbers are assigned by the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (http://www.chem.qmw.ac.uk/iubmb/enzyme/) to allow enzymes to be identified.

An EC number is of the format EC x.y.z.w, where x, y, z, and w are numeric values. In SomaScanENZYME2PROBE, the "EC" prefix is dropped from the Enzyme Commission numbers.

EC numbers have corresponding names that describe the functions of enzymes in such a way that EC x is a more general description than EC x.y, that in turn is a more general description than EC x.y.z. The top level EC numbers and names are listed below:

- EC 1 oxidoreductases
- EC 2 transferases
- EC 3 hydrolases
- EC 4 lyases
- EC 5 isomerases
- EC 6 ligases

The EC name for a given EC number can be viewed at http://www.chem.qmul.ac.uk/iupac/jcbn/index.html#6

Mappings between SeqIds and enzyme identifiers were obtained using files provided by KEGG GENOME (ftp://ftp.genome.jp/pub/kegg/genomes), with a date stamp from the source of: 2011-Mar15

Value

A Bimap object of the ProbeAnnDbBimap class.

References

ftp://ftp.genome.ad.jp/pub/kegg/pathways

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanENZYME
# Get the probe identifiers that are mapped to an EC number
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:3])
if(length(xx) > 0) {
    # Get the ENZYME for the first five probes
```

```
    xx[1:5]
    # Get the first one
    xx[[1]]
}

# Now convert SomaScanENZYME2PROBE to a list to see inside
x <- SomaScanENZYME2PROBE
mapped_probes <- mappedkeys(x)
## convert to a list
xx <- as.list(x[mapped_probes][1:3])
if(length(xx) > 0){
    # Get the probe identifiers for the first five enzyme
    #commission numbers
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanGENENAME *Map between SomaScan SeqIds and gene names*

Description

SomaScanGENENAME is an R object that maps SeqId to the corresponding gene name.

Details

Each SeqId maps to a named vector containing the gene name. The vector name corresponds to the SeqId. If the gene name is unknown, the vector will contain an NA.

Gene names currently include both the *official* (validated by a nomenclature committee) and *preferred* names (interim selected for display) for genes. Efforts are being made to differentiate the two by adding a name to the vector.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanGENENAME
# Get the probe identifiers that are mapped to a gene name
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the GENENAME for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanGO *Map between SomaScan SeqIds and Gene Ontology (GO) identifiers*

Description

SomaScanGO is an R object that provides mappings between SeqIds and the GO identifiers that they are directly associated with. This mapping and its reverse mapping (SomaScanGO2PROBE) do *not* associate the child terms from the GO ontology with the gene. Only the directly evidenced terms are represented here.

SomaScanGO2ALLPROBES is an R object that provides mappings between a given GO identifier and all of the manufacturer identifiers annotated at that GO term or to one of its child nodes in the GO ontology. Thus, this mapping is much larger and more inclusive than SomaScanGO2PROBE.

Details

If SomaScanGO is cast as a list, each SeqId is mapped to a list of lists. The names of the outer list are GO identifiers. Each inner list consists of three named elements: GOID, Ontology, and Evidence.

The **GOID** element matches the GO identifier named in the outer list and is included for convenience when processing the data using lapply().

The **Ontology** element indicates which of the three Gene Ontology categories this identifier belongs to. The categories are biological process (BP), cellular component (CC), and molecular function (MF).

The **Evidence** element contains a code indicating what kind of evidence supports the association of the GO identifier to the SeqId. Some of the evidence codes in use include:

- IMP: inferred from mutant phenotype
- IGI: inferred from genetic interaction
- IPI: inferred from physical interaction

- ISS: inferred from sequence similarity
- IDA: inferred from direct assay
- IEP: inferred from expression pattern
- IEA: inferred from electronic annotation
- TAS: traceable author statement
- NAS: non-traceable author statement
- ND: no biological data available
- IC: inferred by curator

A more complete listing of evidence codes can be found at http://www.geneontology.org/GO.evidence.shtml

If SomaScanGO2ALLPROBES or SomaScanGO2PROBE is cast as a list, each GO term maps to a named vector of SeqIds and evidence codes. A GO identifier may be mapped to the same manufacturer identifier more than once but the evidence code can be different. Mappings between Gene Ontology identifiers, Gene Ontology terms, and other information are available in a separate data package named GO.

Whenever any of these mappings are cast as a data.frame, all the results will be output in an appropriate tabular form.

Mappings between SeqIds and GO information were obtained through their mappings to SeqIds. NA valuess are assigned to SeqIds that cannot be mapped to any Gene Ontology information.

All mappings were based on data provided by Gene Ontology (http://current.geneontology.org/ontology/go-basic.obo), with a date stamp from the source of: 2021-09-01

Value

A Bimap object of the ProbeAnnDbBimap class.

References

ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/

See Also

- SomaScanGO2ALLPROBES
- AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanGO
# Get the manufacturer identifiers that are mapped to a GO ID
mapped_genes <- mappedkeys(x)
# Convert to a list
```

```
xx <- as.list(x[mapped_genes][1:300])
if(length(xx) > 0) {
    # Try the first one
    got <- xx[[1]]
    got[[1]][["GOID"]]
    got[[1]][["Ontology"]]
    got[[1]][["Evidence"]]
}
# For the reverse map:
x <- SomaScanGO2PROBE
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes][1:300])
if(length(xx) > 0){
    # Gets the manufacturer ids for the top 2nd and 3nd GO identifiers
    goids <- xx[2:3]
    # Gets the manufacturer ids for the first element of goids
    goids[[1]]
    # Evidence code for the mappings
    names(goids[[1]])
}

x <- SomaScanGO2ALLPROBES
mapped_genes <- mappedkeys(x)
# Convert SomaScanGO2ALLPROBES to a list
xx <- as.list(x[mapped_genes][1:300])
if(length(xx) > 0){
# Gets the manufacturer identifiers for the top 2nd and 3nd GO identifiers
    goids <- xx[2:3]
    # Gets all the manufacturer identifiers for the first element of goids
    goids[[1]]
    # Evidence code for the mappings
    names(goids[[1]])
}
```

SomaScanMAP *Map between SomaScan SeqIds and cytogenetic maps/bands*

Description

SomaScanMAP is an R object that provides mappings between SeqIds and cytoband locations.

Details

Each SeqId is mapped to a vector of cytoband locations. The vector length may be one or longer, if there are multiple reported chromosomal locations for a given gene. An NA is reported for any SeqId that cannot be mapped to a cytoband at this time.

Cytogenetic bands for most higher organisms are labeled p1, p2, p3, q1, q2, q3 (p and q are the p and q arms), etc., counting from the centromere out toward the telomeres. At higher resolutions,

sub-bands can be seen within the bands. The sub-bands are also numbered from the centromere out toward the telomere. Thus, a label of 7q31.2 indicates that the band is on chromosome 7, q arm, band 3, sub-band 1, and sub-sub-band 2.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://www.ncbi.nlm.nih.gov

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanMAP
# Get the probe identifiers that are mapped to any cytoband
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the MAP for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanOMIM *Map between SomaScan SeqIds and Mendelian Inheritance in Man (MIM) identifiers*

Description

SomaScanOMIM is an R object that provides mappings between manufacturer identifiers and OMIM identifiers.

Details

Each SeqIds is mapped to a vector of OMIM identifiers. The vector length may be one or longer, depending on how many OMIM identifiers the SeqIds maps to. An NA is reported for any SeqIds that cannot be mapped to an OMIM identifier at this time.

OMIM is based upon the book Mendelian Inheritance in Man (V. A. McKusick) and focuses primarily on inherited or heritable genetic diseases. It contains textual information, pictures, and reference information that can be searched using various terms, among which the MIM number is one.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanOMIM
# Get the probe identifiers that are mapped to a OMIM ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the OMIM for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanORGANISM *The Organism information for SomaScan*

Description

SomaScanORGANISM is an R object that contains a single item: a character string that names the organism for which SomaScan was built. SomaScanORGPKG is an R object that contains a character vector with the name of the organism package that a chip package depends on for its gene-centric annotations.

Details

SomaScanORGANISM provides a simple way to programmatically extract the organism name. SomaScanORGPKG provides a simple way to programmatically extract the name of the parent organism package. The parent organism package is a strict dependency for chip packages, as this is where the gene-centric information is ultimately extracted from. The full package name will always be this string plus the extension ".db". But most programmatic access will not require this extension, so it is more convenient to leave it out.

Value

A Bimap object of the ProbeAnnDbBimap class.

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
SomaScanORGANISM
SomaScanORGPKG
```

SomaScanPATH *Mappings between probe identifiers and KEGG pathway identifiers*

Description

KEGG (Kyoto Encyclopedia of Genes and Genomes) maintains pathway data for various organisms.

SomaScanPATH maps probe identifiers to the identifiers used by KEGG for pathways in which the genes represented by the probe identifiers are involved.

SomaScanPATH2PROBE is an R object that provides mappings between KEGG identifiers and SeqIds.

Details

Each KEGG pathway has a name and identifier. The pathway name for a given pathway identifier can be obtained using the KEGG data package that can downloaded from Bioconductor (http://www.bioconductor.org).

Graphic presentations of pathways are searchable at http://www.genome.ad.jp/kegg/pathway.html by using pathway identifiers as keys.

Mappings were based on data provided by KEGG GENOME (ftp://ftp.genome.jp/pub/kegg/genomes), with a date stamp from the source of: 2011-Mar15

Value

A Bimap object of the ProbeAnnDbBimap class.

References

http://www.genome.ad.jp/kegg/

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanPATH
# Get the probe identifiers that are mapped to a KEGG pathway ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the PATH for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}

x <- SomaScanPATH
mapped_probes <- mappedkeys(x)
# Now convert the SomaScanPATH2PROBE object to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0){
    # Get the probe identifiers for the first two pathway identifiers
    xx[1:2]
    # Get the first value
    xx[[1]]
}
```

SomaScanPMID *Maps between SomaScan SeqIds and PubMed identifiers*

Description

SomaScanPMID is an R object that provides mappings between SeqIds and PubMed identifiers.

SomaScanPMID2PROBE is an R object that provides mappings between PubMed identifiers and SeqIDs.

Details

When SomaScanPMID is viewed as a list, each SeqId is mapped to a named vector of PubMed identifiers. The name associated with each vector corresponds to the SeqId. The length of the vector may be one or greater, depending on how many PubMed identifiers a given SeqId is mapped to. An NA is reported for any SeqId that cannot be mapped to a PubMed identifier.

When SomaScanPMID2PROBE is viewed as a list, each PubMed identifier is mapped to a named vector of SeqIds. The name represents the PubMed identifier and the vector contains all SeqIds that are represented by that PubMed identifier. The length of the vector may be one or longer, depending on how many SeqIds are mapped to a given PubMed identifier.

Titles, abstracts, and possibly full texts of articles can be obtained from PubMed by providing a valid PubMed identifier. The pubmed() function of the annotate package can also be used for the same purpose.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanPMID
# Get the probe identifiers that are mapped to any PubMed ID
mapped_probes <- mappedkeys(x)
```

```
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0){
    # Get the PubMed identifiers for the first two probe identifiers
    xx[1:2]
    # Get the first one
    xx[[1]]
    if(interactive() && !is.null(xx[[1]]) && !is.na(xx[[1]])
    && require(annotate)){
        # Get article information as XML files
        xmls <- pubmed(xx[[1]], disp = "data")
        # View article information using a browser
        pubmed(xx[[1]], disp = "browser")
    }
}

x <- SomaScanPMID2PROBE
mapped_probes <- mappedkeys(x)
# Now convert the reverse map object SomaScanPMID2PROBE to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0){
    # Get the probe identifiers for the first two PubMed identifiers
    xx[1:2]
    # Get the first one
    xx[[1]]
    if(interactive() && require(annotate)){
        # Get article information as XML files for a PubMed id
        xmls <- pubmed(names(xx)[1], disp = "data")
        # View article information using a browser
        pubmed(names(xx)[1], disp = "browser")
    }
}
```

SomaScanREFSEQ *Map between SomaScan SeqIds and RefSeq identifiers*

Description

SomaScanREFSEQ is an R object that provides mappings between SeqIds and RefSeq identifiers.

Details

Each manufacturer identifier is mapped to a named vector of RefSeq identifiers. The name represents the manufacturer identifier, and the vector contains all RefSeq identifiers that can be mapped to that SeqId. The length of the vector may be one or greater, depending on how many RefSeq identifiers a given SeqId can be mapped to. An NA is reported for any SeqId that cannot be mapped to a RefSeq identifier at this time.

RefSeq identifiers differ in format according to the type of record. See the identifier descriptions below, where XXXXX is a sequence of integers:

- NG_XXXXX: RefSeq accessions for genomic region (nucleotide) records
- NM_XXXXX: RefSeq accessions for mRNA records
- NC_XXXXX: RefSeq accessions for chromosome records
- NP_XXXXX: RefSeq accessions for protein records
- XR_XXXXX: RefSeq accessions for model RNAs that are not associated with protein products
- XM_XXXXX: RefSeq accessions for model mRNA records
- XP_XXXXX: RefSeq accessions for model protein records

NCBI (https://www.ncbi.nlm.nih.gov/RefSeq/) allows users to query the RefSeq database using RefSeq identifiers.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://www.ncbi.nlm.nih.gov https://www.ncbi.nlm.nih.gov/RefSeq/

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanREFSEQ
# Get the probe identifiers that are mapped to any RefSeq ID
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the REFSEQ for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScanSYMBOL *Map between SomaScan SeqIds and Gene Symbols*

Description

SomaScanSYMBOL is an R object that provides mappings between SeqIds and gene abbreviations.

Details

Each SeqId is mapped to an abbreviation (symbol) for the corresponding gene. An NA is reported if there is no known abbreviation for a given gene.

Symbols typically consist of 3-6 letters that define either a single gene (ABC) or multiple genes (ABC1, ABC2, ABC3). Gene symbols can be used as key words to query public databases such as Entrez Gene.

Mappings were based on data provided by Entrez Gene (ftp://ftp.ncbi.nlm.nih.gov/gene/DATA), with a date stamp from the source of: 2021-Sep13

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanSYMBOL
# Get the probe identifiers that are mapped to a gene symbol
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
    # Get the SYMBOL for the first five probes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaTARGETFULLNAME *Map between SomaScan SeqIds and the full name of the protein target*

Description

SomaScanTARGETFULLNAME is an R object that provides mappings between manufacturer identifiers and the names of their protein targets.

Details

Each manufacturer identifier is mapped to the full name of the SOMAmer protein target. An NA is reported if there is no target name information for a given SOMAmer reagent.

The full protein target name is typically written in sentence case without using gene symbol abbreviations, however gene symbols can still occasionally be found in some target names. This information can also be found in an ADAT file with the accompanying target name (an abbreviated version of the full name found in this mapping).

Mappings were based on data provided by: SomaLogic (https://somalogic.com/somascan-menu/), with a date stamp from the source of: 2023-Oct

Value

A Bimap object of the ProbeAnnDbBimap class.

References

https://somalogic.com/somascan-menu/

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanTARGETFULLNAME
# Get the probe identifiers that are mapped to a gene symbol
mapped_probes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_probes][1:300])
if(length(xx) > 0) {
  # Get the TARGETFULLNAME for the first five probes
  xx[1:5]
  # Get the first value
  xx[[1]]
}
```

SomaScanUNIPROT *Map between Uniprot accession numbers and SomaScan SeqIds*

Description

SomaScanUNIPROT is an R object that contains mappings between SeqIds and Uniprot accession numbers.

Details

This object is a simple mapping of SeqIds to Uniprot accession numbers.

Human mappings were based on data provided by NCBI, with an exception for fly, which required retrieving the data from Ensembl (http://www.ensembl.org/biomart/martview/).

Value

A Bimap object of the ProbeAnnDbBimap class.

See Also

AnnotationDb-class for use of the select() interface.

Examples

```
## select() interface:
## Objects in this package can be accessed using the select() interface
## from the AnnotationDbi package. See ?select for details.

## Bimap interface:
x <- SomaScanUNIPROT
# Get the entrez gene IDs that are mapped to an Uniprot ID
mapped_genes <- mappedkeys(x)
# Convert to a list
xx <- as.list(x[mapped_genes][1:300])
if(length(xx) > 0) {
    # Get the Uniprot IDs for the first five genes
    xx[1:5]
    # Get the first value
    xx[[1]]
}
```

SomaScan_dbconn *Collect information about the package annotation DB*

Description

Some convenience functions for getting a connection object to (or collecting information about) the package annotation DB.

Usage

```
SomaScan_dbconn()
SomaScan_dbfile()
SomaScan_dbschema(file="", show.indices=FALSE)
SomaScan_dbInfo()
```

Arguments

file	A connection, or a character string naming the file to print to (see the file argument of the cat function for the details).
show.indices	The CREATE INDEX statements are not shown by default. Use show.indices=TRUE to get them.

Details

SomaScan_dbconn returns a connection object to the package annotation DB. IMPORTANT: Don't call dbDisconnect on the connection object returned by SomaScan_dbconn or you will break all the AnnDbObj objects defined in this package!

SomaScan_dbfile returns the path (character string) to the package annotation DB (this is an SQLite file).

SomaScan_dbschema prints the schema definition of the package annotation DB.

SomaScan_dbInfo prints other information about the package annotation DB.

Value

SomaScan_dbconn: a DBIConnection object representing an open connection to the package annotation DB.

SomaScan_dbfile: a character string with the path to the package annotation DB.

SomaScan_dbschema: none (invisible NULL).

SomaScan_dbInfo: none (invisible NULL).

See Also

dbGetQuery, dbConnect, dbconn, dbfile, dbschema, dbInfo

Examples

```
library(DBI)
## Count the number of rows in the "probes" table:
dbGetQuery(SomaScan_dbconn(), "SELECT COUNT(*) FROM probes")

SomaScan_dbschema()

SomaScan_dbInfo()
```

`addTargetFullName` *Add Target Full Name information to data frame*

Description

Retrieve the Target Full Name for a given SeqId (or set of SeqIds) and add that information to a data frame. The Target Full Name can be found in an ADAT file and represents the full name of the protein target of a primary SOMAmer reagent.

Note: You can retrieve a list of Target Full Name values for a given set of SeqIds with `mget(seqids, SomaScanTARGETFULLNAME)`, where `seqids` is a character vector of SeqIds. To retrieve a data frame of *all* Target Full Name values present in the package, use `toTable(SomaScanTARGETFULLNAME)`.

Usage

```
addTargetFullName(data)
```

Arguments

`data`	a data frame generated by a `select` query. This data frame can have any number of columns, but **must** contain a PROBEID column. This will be used to match the Target Full Name.

Value

The input data frame with one additional column called "TARGETFULLNAME".

Examples

```
seqs <- withr::with_seed(123, sample(keys(SomaScan.db), 5))
res <- select(SomaScan.db, keys = seqs, columns = "UNIPROT")
addTargetFullName(res)
```

keys,SomaDb-method *The* keys *method for SomaDb objects*

Description

keys returns the full set of keys for the annotation database. A database "key" is a unique value that identifies a single row of the database. By default, keys will return a vector of all primary database keys. The primary keys for SomaScan.db are the SomaLogic sequence identifiers, also called SeqIds.

If the keytype argument is specified, keys will return the keys that are available for that keytype, rather than the SeqIds (see examples).

Usage

```
## S4 method for signature 'SomaDb'
keys(x, keytype, ...)
```

Arguments

x	the AnnotationDb object. But in practice this will mean an object derived from an AnnotationDb object such as a OrgDb or ChipDb object.
keytype	the keytype that matches the keys used. For the select methods, this is used to indicate the kind of ID being used with the keys argument. For the keys method this is used to indicate which kind of keys are desired from keys
...	other arguments. These include: **pattern:** the pattern to match (used by keys) **column:** the column to search on. This is used by keys and is for when the thing you want to pattern match is different from the keytype, or when you want to simply want to get keys that have a value for the thing specified by the column argument. **fuzzy:** TRUE or FALSE value. Use fuzzy matching? (this is used with pattern by the keys method)

Value

A character vector of all available database keys.

Author(s)

Amanda Hiser

Examples

```
# Preview the primary database keys
keys(SomaScan.db) |> head()

# View a different type of database key, e.g. gene symbols
keys(SomaScan.db, keytype = "SYMBOL") |> head()
```

---

keytypes,SomaDb-method

The keytypes *method for SomaDb objects*

Description

keytypes produces a vector of annotation categories that can be used as input keys for the keytype= argument of the select, mapIds, and keys methods. The keytype also informs the query of the format of the search key. For example, when searching the database for annotations corresponding to the gene "NOTCH3", the key "NOTCH3" has the keytype of "SYMBOL". Specifying keytype="SYMBOL" ensures that the gene symbol is not confused for a different identifier ("key") type.

The default keytype ("PROBEID"), corresponds to the SomaLogic SeqId. For more information on database keys, see ?keys.

Usage

```
## S4 method for signature 'SomaDb'
keytypes(x)
```

Arguments

x	the AnnotationDb object. But in practice this will mean an object derived from an AnnotationDb object such as a OrgDb or ChipDb object.

Value

A character vector of all available database keytypes.

Author(s)

Amanda Hiser

Examples

```
# Retrieve a full list of available keytypes
keytypes(SomaScan.db)

# Specify a keytype other than PROBEID for select
select(SomaScan.db, keys = "NOTCH3", keytype = "SYMBOL",
       columns = "PROBEID")
```

mapIds,SomaDb-method *The* mapIds *method for* SomaDb *objects*

Description

mapIds will retrieve SomaScan annotations (as a named vector) based on the parameters provided by the keys, columns, and keytype arguments. The default keytype is "PROBEID", e.g. the SomaLogic SeqId; this value will be used to tie all annotations back to a SomaScan-specific identifier.

Usage

```
## S4 method for signature 'SomaDb'
mapIds(
  x,
  keys,
  column,
  keytype,
  menu = NULL,
  match = FALSE,
  ...,
  multiVals = c("filter", "asNA", "first", "list", "CharacterList")
)
```

Arguments

x	the AnnotationDb object. But in practice this will mean an object derived from an AnnotationDb object such as a OrgDb or ChipDb object.
keys	the keys to select records for from the database. All possible keys are returned by using the keys method.
column	the column to search on (for mapIds). Different from columns in that it can only have a single element for the value
keytype	the keytype that matches the keys used. For the select methods, this is used to indicate the kind of ID being used with the keys argument. For the keys method this is used to indicate which kind of keys are desired from keys
menu	a character string identifying a SomaScan menu version (optional). Possible options include: "5k","7k", or "11k", as well as the version numbers for those menus ("v4.0", "v4.1", or "v5.0", respectively). May only be used when keytype = "PROBEID". This argument will filter the keys to the specified menu and only return data associated with analytes present in that menu. By default, all annotations from all analytes are available.
match	a logical (TRUE/FALSE). Must be used with the "SYMBOL", "ALIAS", or "GENENAME" keytypes only. If true, the character string provided for keys will be used as a search term. The string will be used to match symbols that also start with that string (ex. a key of "CASP1" will return annotations for both the CASP10 & CASP14 genes).

`...` Arguments passed on to `AnnotationDbi::mapIds`

`multiVals` What should `mapIds` do when there are multiple values that could be returned? Options include:

first: This value means that when there are multiple matches only the 1st thing that comes back will be returned. This is the default behavior

list: This will just returns a list object to the end user

filter: This will remove all elements that contain multiple matches and will therefore return a shorter vector than what came in whenever some of the keys match more than one value

asNA: This will return an NA value whenever there are multiple matches

CharacterList: This just returns a SimpleCharacterList object

FUN: You can also supply a function to the `multiVals` argument for custom behaviors. The function must take a single argument and return a single value. This function will be applied to all the elements and will serve a 'rule' that for which thing to keep when there is more than one element. So for example this example function will always grab the last element in each result: `last <- function(x){x[[length(x)]]}`

Details

`mapIds` is similar to `select` in that it can be used to retrieve annotation information from the database. However, users should be aware that if they call `mapIds` and request columns that have multiple matches for the specified keys (ex. GO or other pathway-related terms), `mapIds` will return a named character vector with (by default) *only one* result for each possible match. This result in some results being truncated. Make sure to specify the appropriate `multiVals` parameter for each query.

Value

A named character vector containing the retrieved annotations. Missing values will be returned as `NA`.

Author(s)

Amanda Hiser

Examples

```
# Retrieve a set of example keys
keys <- head(keys(SomaScan.db))
keys

# Only 1 result is returned by default; missing values are returned as `NA`
mapIds(SomaScan.db, keys = keys, column = "UNIPROT")

# Specify `multiVals` to return all results as a list
mapIds(SomaScan.db, keys = keys, column = "UNIPROT", multiVals = "list")
```

`select,SomaDb-method` *The* `select` *method for* `SomaDb` *objects*

Description

`select` is the primary data retrieval method for the `SomaScan.db` database. `select` will retrieve a data frame of SomaScan annotations based on the parameters provided by the `keys`, `columns`, and `keytype` arguments. The default keytype is "PROBEID", e.g. the SomaLogic `SeqId`; this value is used to tie all annotations back to a SomaScan-specific identifier.

Usage

```
## S4 method for signature 'SomaDb'
select(x, keys, columns, keytype, menu = NULL, match = FALSE, ...)
```

Arguments

`x`	the `AnnotationDb` object. But in practice this will mean an object derived from an `AnnotationDb` object such as a `OrgDb` or `ChipDb` object.
`keys`	the keys to select records for from the database. All possible keys are returned by using the `keys` method.
`columns`	the columns or kinds of things that can be retrieved from the database. As with `keys`, all possible columns are returned by using the `columns` method.
`keytype`	the keytype that matches the keys used. For the `select` methods, this is used to indicate the kind of ID being used with the keys argument. For the `keys` method this is used to indicate which kind of keys are desired from `keys`
`menu`	a character string identifying a SomaScan menu version (optional). Possible options include: `"5k"`,`"7k"`, or `"11k"`, as well as the version numbers for those menus (`"v4.0"`, `"v4.1"`, or `"v5.0"`, respectively). May only be used when `keytype = "PROBEID"`. This argument will filter the keys to the specified menu and only return data associated with analytes present in that menu. By default, all annotations from all analytes are available.
`match`	a logical (TRUE/FALSE). Must be used with the "SYMBOL", "ALIAS", or "GENENAME" keytypes only. If true, the character string provided for `keys` will be used as a search term. The string will be used to match symbols that also start with that string (ex. a key of "CASP1" will return annotations for both the CASP10 & CASP14 genes).
`...`	Arguments passed on to `AnnotationDbi::select`

Details

Users should be aware that if they call `select` and request columns that have multiple matches for the provided keys (e.g. GO terms), `select` will return a `data.frame` with one row *for each possible match*. This can have a multiplicative effect and result in a large number of returned values. In general, if a user needs to retrieve a column that has a many-to-one relationship to the original keys, it is best to extract data from that column in its own query.

Value

A `data.frame` containing the retrieved annotations.

Author(s)

Amanda Hiser

Examples

```
# Retrieve a set of example keys
keys <- head(keys(SomaScan.db))
keys

# Look up the gene symbol and gene type for all example keys
select(SomaScan.db, keys = keys, columns = c("SYMBOL", "GENETYPE"))

# Look up SomaScan SeqIds & proteins associated with a gene of interest
select(SomaScan.db, keys = "NOTCH3", keytype = "SYMBOL",
      columns = c("PROBEID", "UNIPROT"))
```

Index

∗ **datasets**
DataObjects, 1
SomaScan.db, 3
SomaScan_dbconn, 22
SomaScanALIAS2PROBE, 2
SomaScanENSEMBL, 4
SomaScanENTREZID, 5
SomaScanENZYME, 6
SomaScanGENENAME, 8
SomaScanGO, 9
SomaScanMAP, 11
SomaScanOMIM, 12
SomaScanORGANISM, 14
SomaScanPATH, 14
SomaScanPMID, 16
SomaScanREFSEQ, 17
SomaScanSYMBOL, 19
SomaScanUNIPROT, 21
SomaTARGETFULLNAME, 20
∗ **data**
DataObjects, 1
∗ **utilities**
SomaScan_dbconn, 22
.SomaDb *(SomaDb-class)*, 2

addTargetFullName, 23
AnnDbObj, *22*
AnnotationDbi::mapIds, *27*
AnnotationDbi::select, *28*

Bimap, *3, 4*, *6–8*, *10*, *12–16*, *18–21*

cat, *22*

DataObjects, 1
dbconn, *22*
dbConnect, *22*
dbDisconnect, *22*
dbfile, *22*
dbGetQuery, *22*
dbInfo, *22*
dbschema, *22*

keys *(keys,SomaDb-method)*, 24
keys,SomaDb-method, 24
keytypes *(keytypes,SomaDb-method)*, 25
keytypes,SomaDb-method, 25

mapIds *(mapIds,SomaDb-method)*, 26
mapIds,SomaDb-method, 26

probe_list *(DataObjects)*, 1

select *(select,SomaDb-method)*, 28
select,SomaDb-method, 28
SomaDb-class, 2
SomaScan *(SomaScan.db)*, 3
SomaScan.db, 3
somascan_analytes *(DataObjects)*, 1
SomaScan_dbconn, 22
SomaScan_dbfile *(SomaScan_dbconn)*, 22
SomaScan_dbInfo *(SomaScan_dbconn)*, 22
SomaScan_dbschema *(SomaScan_dbconn)*, 22
somascan_menu *(DataObjects)*, 1
SomaScanALIAS2PROBE, 2
SomaScanENSEMBL, 4
SomaScanENSEMBL2PROBE *(SomaScanENSEMBL)*, 4
SomaScanENTREZID, 5
SomaScanENZYME, 6
SomaScanENZYME2PROBE *(SomaScanENZYME)*, 6
SomaScanGENENAME, 8
SomaScanGO, 9
SomaScanGO2ALLPROBES, *10*
SomaScanGO2ALLPROBES *(SomaScanGO)*, 9
SomaScanGO2PROBE *(SomaScanGO)*, 9
SomaScanLOCUSID *(SomaScanENTREZID)*, 5
SomaScanMAP, 11
SomaScanOMIM, 12
SomaScanORGANISM, 14

SomaScanORGPKG *(*SomaScanORGANISM*)*, 14
SomaScanPATH, 14
SomaScanPATH2PROBE *(*SomaScanPATH*)*, 14
SomaScanPMID, 16
SomaScanPMID2PROBE *(*SomaScanPMID*)*, 16
SomaScanREFSEQ, 17
SomaScanSYMBOL, 19
SomaScanTARGETFULLNAME
*(*SomaTARGETFULLNAME*)*, 20
SomaScanUNIPROT, 21
SomaScanUNIPROT2PROBE
*(*SomaScanUNIPROT*)*, 21
SomaTARGETFULLNAME, 20