UniProt.ws: A package for retrieving data from the UniProt web service

Marc Carlson

August 21, 2022

1 Configuring `uniport.ws`

The *UniProt.ws* package provides a `select` interface to the UniProt web service.

```
suppressPackageStartupMessages({
    library(UniProt.ws)
})
up <- UniProt.ws(taxId=9606)
```

If you already know about the select interface, you can immediately learn about the various methods for this object by just looking it's the help page.

```
help("UniProt.ws")
```

When you load the *UniProt.ws* package, it creates a `UniProt.ws` object. If you look at the object you will see some helpful information about it.

```
up
```

```
## UniProt.ws interface object:
## Taxonomy ID: 9606
## Species name: Homo sapiens (Human)
## List species with 'availableUniprotSpecies()'
```

By default, you can see that the `UniProt.ws` object is set to retrieve records from Homo sapiens. But you can change that of course. In order to change it, you first need to look up the appropriate taxonomy ID for the species that you are interested in. Uniprot provides support for over 20 thousand species, so there are a few to choose from! In order to make this easier, we have provided the helper function `availableUniprotSpecies` which will list all the supported species along with their taxonomy ids. When you call the `availableUniprotSpecies` function, it's recommended that you make use of the pattern argument to limit your queries like this:

```
availableUniprotSpecies(pattern="musculus")
```

```
##       kingdom Taxon Node  Official (scientific) name
## ANTMS       E     520121         Anthocoris musculus
## ANTMU       E     208057        Anthoscopus musculus
## APOMU       E     238007             Apomys musculus
## BAIMU       E     213557            Baiomys musculus
## BALMU       E       9771       Balaenoptera musculus
## BLEMU       E     197864        Blepharisma musculus
```

```
## MOUSE       E      10090               Mus musculus
## MUSMB       E      35531     Mus musculus bactrianus
## MUSMC       E      10091      Mus musculus castaneus
## MUSMM       E      57486     Mus musculus molossinus
## POVM1       V    1891730 Mus musculus polyomavirus 1
```

Once you have learned the taxonomy ID for the species of interest, you can then change the taxonomy id for the `UniProt.ws` object using `taxId` setter or by calling the constructor for `UniProt.ws`

```
mouseUp <- UniProt.ws(10090)
mouseUp
```

```
## UniProt.ws interface object:
## Taxonomy ID: 10090
## Species name: Mus musculus (Mouse)
## List species with 'availableUniprotSpecies()'
```

As you can see the species is different for the `mouseUp` new object.

2 Using `UniProt.ws`

Once you are safisfied that you have an `uniport.ws` that is using the appropriate organsims, you can make use of the standard set of methods in a `select` interface. Specifically: `columns`, `keytypes`, `keys` and `select`.

You will probably notice that there are a large number of columns that can be retrieved.

```
head(keytypes(up))
```

```
## [1] "Allergome"     "ArachnoServer" "Araport"       "BioCyc"
## [5] "BioGRID"       "BioMuta"
```

And most (but not all) of these fields can also be used as keytypes.

```
head(columns(up))
```

```
## [1] "absorption"              "accession"
## [3] "annotation_score"        "cc_activity_regulation"
## [5] "cc_allergen"             "cc_alternative_products"
```

If necessary you can also look up the keys of a given type. But please be warned that the web service is slow at this particular kind of lookup. So if you really want to do this kind of operation you are probably going to want to save the result to your R session.

```
egs <- keys(up, "GeneID")
```

Finally, you can loop up whatever combinations of columns, keytypes and keys that you need when using `select`.

Note. 'ENTREZ_GENE' is now 'GeneID'

```
keys <- c("1","2")
columns <- c("xref_pdb", "xref_hgnc", "sequence")
```

```
kt <- "GeneID"
res <- select(up, keys, columns, kt)
res
```

```
##   From  Entry                                                    PDB    HGNC
## 1    1 P04217                                                  <NA> HGNC:5;
## 2    1 V9HWD8                                                  <NA>    <NA>
## 3    2 P01023 1BV8;2P9R;6TAV;7O7M;7O7N;7O7O;7O7P;7O7Q;7O7R;7O7S; HGNC:7;
##
## 1
## 2
## 3 MGKNKLLHPSLVLLLLVLLPTDASVSGKPQYMVLVPSLLHTETTEKGCVLLSYLNETVTVSASLESVRGNRSLFTDLEAENDVLHCVAFAVPKSSSNEEVMFL
```

sessionInfo()

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /home/biocbuild/bbs-3.15-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.15-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB              LC_COLLATE=C
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] UniProt.ws_2.36.5   BiocGenerics_0.42.0 RSQLite_2.2.16
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.9             prettyunits_1.1.1      png_0.1-7
##  [4] Biostrings_2.64.1      assertthat_0.2.1       digest_0.6.29
##  [7] utf8_1.2.2             BiocFileCache_2.4.0    mime_0.12
## [10] R6_2.5.1               GenomeInfoDb_1.32.3    stats4_4.2.1
## [13] evaluate_0.16          highr_0.9              httr_1.4.4
## [16] pillar_1.8.1           zlibbioc_1.42.0        rlang_1.0.4
## [19] progress_1.2.2         curl_4.3.2             blob_1.2.3
## [22] S4Vectors_0.34.0       DT_0.24                rmarkdown_2.15
## [25] stringr_1.4.1          htmlwidgets_1.5.4      RCurl_1.98-1.8
## [28] bit_4.0.4              shiny_1.7.2            compiler_4.2.1
```

```
## [31] httpuv_1.6.5           xfun_0.32              pkgconfig_2.0.3
## [34] htmltools_0.5.3        cellxgenedp_1.0.0      tidyselect_1.1.2
## [37] KEGGREST_1.36.3        tibble_3.1.8           GenomeInfoDbData_1.2.8
## [40] IRanges_2.30.1         fansi_1.0.3            dbplyr_2.2.1
## [43] crayon_1.5.1           dplyr_1.0.9            later_1.3.0
## [46] rappdirs_0.3.3         bitops_1.0-7           jsonlite_1.8.0
## [49] xtable_1.8-4           lifecycle_1.0.1        DBI_1.1.3
## [52] magrittr_2.0.3         cli_3.3.0              stringi_1.7.8
## [55] cachem_1.0.6           XVector_0.36.0         promises_1.2.0.1
## [58] filelock_1.0.2         ellipsis_0.3.2         generics_0.1.3
## [61] vctrs_0.4.1            httpcache_1.2.0        BiocStyle_2.24.0
## [64] tools_4.2.1            bit64_4.0.5            Biobase_2.56.0
## [67] glue_1.6.2             purrr_0.3.4            hms_1.1.2
## [70] parallel_4.2.1         fastmap_1.1.0          yaml_2.3.5
## [73] AnnotationDbi_1.58.0   BiocManager_1.30.18    memoise_2.0.1
## [76] knitr_1.39
```