Bioconductor Packages For Cached File Management

BiocFileCache, AnnotationHub, ExperimentHub

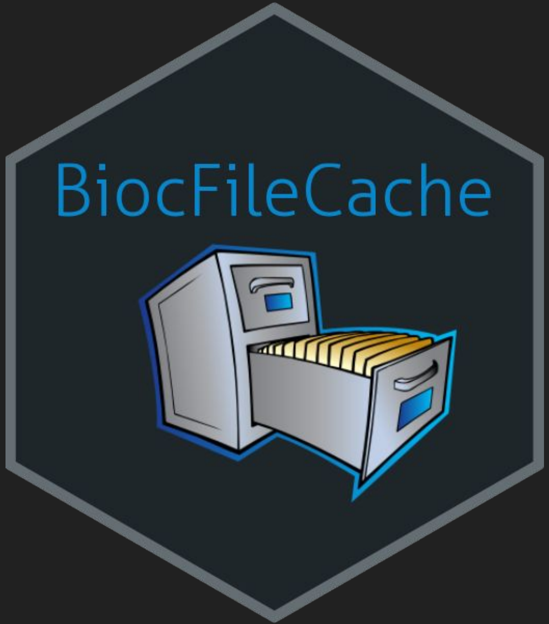
Bioconductor Packages For Cached File Management

BiocFileCache, AnnotationHub, ExperimentHub

https://docs.google.com/presentation/d/1dxAmnk_M-wOzoCD0Wod2XB94eavPJJUNIuISiEH3cL8/edit?usp=sharing

BiocFileCache

Local File Management



Motivation:

It can be time consuming to download remote resource from the web. Let's design a way to check a local resource to see if it needs to be updated or not.

Motivation:

Let’s also have a way to better organize local files

BiocFileCache( )

- creates a cache object
- sqlite database backend
- add ‘resources’ (files) to the cache object to track

Cache Info:

- bfccache ( )
- length ( )
- show ( )
- bfcinfo ( )

Adding Resources:

- bfcadd( )
- bfcnew ( )

Removing Resources:

- bfcremove ( )
- bfcsync ( )

Investigating Resources:

- bfcquerycols ( )
- bfcquery ( )
- bfccount ( )
- bfcrid ( )
- bfcpath ( )
- bfcrpath ( )
- [

Web Resources:

- bfcneedsupdate ( )
- bfcdownload ( )

Updating Resources:

- bfcupdate ( )
- [[

MetaData:

- bfcmetalist ( )
- bfcmeta ( )
- bfcmeta ( ) <-
- bfcmetaremove ( )

Export/Import Cache:

- importbfc ( )
- exportbfc ( )
- makeBiocFileCacheFromDataFrame( )

Clean/Remove Cache:

- cleanbfc ( )
- removebfc ( )

Example:

```
> BiocFileCache()
class: BiocFileCache
bfccache: /home/lori/.cache/BiocFileCache
bfccount: 0
For more information see: bfcinfo() or bfcquery()
```

```
> bfcadd(rname="Wiki", fpath="https://en.wikipedia.org/wiki/Bioconductor")
  |======================================================================| 100%
                                                          BFC1 
"/home/lori/.cache/BiocFileCache/282e8be47f6_Bioconductor"
```

Example:

```
> pathToSave = bfcnew(rname="My RDS File", ext=".rds")


> pathToSave
                                                                BFC2
  "/home/lori/.cache/BiocFileCache/2feb30a96058_2feb30a96058.rds"


> bfcinfo()
# A tibble: 2 x 10
  rid   rname  create_time access_time rpath rtype fpath last_modified_t… etag
  <chr> <chr>  <chr>       <chr>       <chr> <chr> <chr> <chr>            <chr>
1 BFC1  Wiki   2018-07-12… 2018-07-12… /hom… web   http… 2018-07-07 07:1… NA
2 BFC2  My RD… 2018-07-12… 2018-07-12… /hom… rela… 388d… NA               NA
# ... with 1 more variable: expires <chr>


> saveRDS(myObj, file=pathToSave)
```

Example:

```
> bfcneedsupdate()
 BFC1
TRUE
```

Utilizes functions from httr to capture Expires, Last-modified time, and Etag

1. HEAD()
2. cache_info()

```
> library(httr)

> cache_info(HEAD("https://en.wikipedia.org/wiki/Bioconductor"))
<cache_info>  https://en.wikipedia.org/wiki/Bioconductor
  Cacheable:     TRUE
  Expires:       Thu, 12 Jul 2018 13:37:06 GMT <expired>
  Last-Modified: Sat, 07 Jul 2018 07:13:52 GMT
  Etag:


> cache_info(HEAD("https://bioconductor.org/packages/3.8/data/annotation/src/contrib/PANTHER.db_1.0.4.tar.gz"))
<cache_info>  https://bioconductor.org/packages/3.8/data/annotation/src/contrib/PANTHER.db_1.0.4.tar.gz
  Cacheable:     TRUE
  Last-Modified: Wed, 27 Sep 2017 17:09:56 GMT
  Etag:          "608b685-55a2edc70632a"
```

Example:

```
> bfcquery(query="RDS")
# A tibble: 1 x 10
  rid   rname  create_time access_time rpath rtype fpath last_modified_t… etag
  <chr> <chr>  <chr>       <chr>       <chr> <chr> <chr>            <dbl> <chr>
1 BFC2  My RD… 2018-07-12… 2018-07-12… /hom… rela… 388d…                NA NA
# ... with 1 more variable: expires <dbl>


> bfcrid(bfcquery(query="RDS"))
[1] "BFC2"

> bfcrpath(rids="BFC2")

                                                            BFC2
  "/home/lori/.cache/BiocFileCache/2feb30a96058_2feb30a96058.rds"


> readRDS(bfcrpath(rids="BFC2"))
```

Example:

```
# data.frame or tibble

> meta = data.frame(rid="BFC2", info="pipeLine project X", numSamples=2000)

> bfc = BiocFileCache()

> bfcmeta(bfc, name="pipeLineXmeta") <- meta
> bfcmetalist()
[1] "pipeLineXmeta"


> library(dplyr)
> bfcinfo(bfc) %>% select(rid, rname, info, numSamples)
# A tibble: 2 x 4
    rid         rname                   info numSamples
  <chr>         <chr>                  <chr>      <dbl>
1  BFC1          Wiki                   <NA>         NA
2  BFC2 My RData File pipeLine project X       2000
```

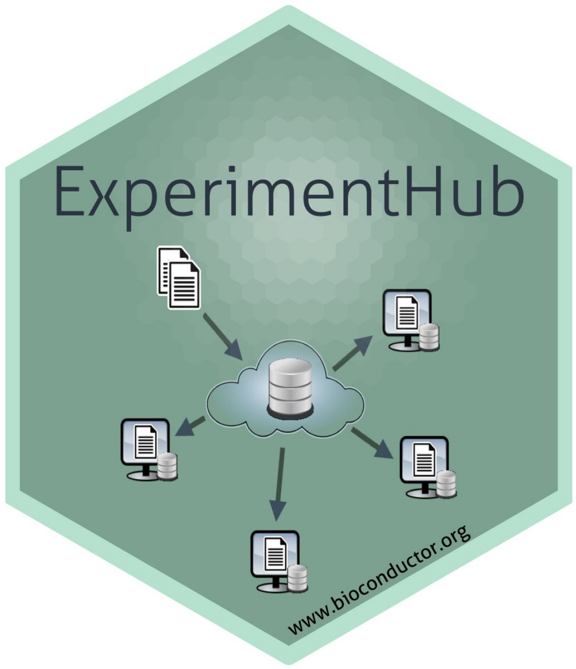
Example:

```
> bfcquery(query="project X", field="info")
# A tibble: 1 x 12
  rid   rname  create_time access_time rpath rtype fpath last_modified_t… etag
  <chr> <chr>  <chr>       <chr>       <chr> <chr> <chr>            <dbl> <chr>
1 BFC2  My RD… 2018-07-12… 2018-07-12… /hom… rela… 388d…               NA NA
# ... with 3 more variables: expires <dbl>, info <chr>, numSamples <dbl>


> bfcquerycols()
 [1] "rid"                "rname"              "create_time"
 [4] "access_time"        "rpath"              "rtype"
 [7] "fpath"              "last_modified_time" "etag"
[10] "expires"            "info"               "numSamples"
```

Implementations

1. Directly
2. Many package started using in the backend to manage package data

AnnotationHub/ExperimentHub



AnnotationHub

AnnotationHub is a package that allows us to query and download many different annotation objects, without having to explicitly install them.

AnnotationHub( )

- creates a hub object
- sqlite database backend
- Files are stored remotely and downloaded as needed
  - Bioconductor AWS S3 Buckets
  - After downloaded, cached for quick access for future runs
  - Uses BiocFileCache to manage individual files

Example:

```
> hub = AnnotationHub()
snapshotDate(): 2019-07-10


> hub
AnnotationHub with 46429 records
# snapshotDate(): 2019-07-10
# $dataprovider: BroadInstitute, Ensembl, UCSC, ftp://ftp.ncbi.nlm.nih.gov/g...
# $species: Homo sapiens, Mus musculus, Drosophila melanogaster, Bos taurus,...
# $rdataclass: GRanges, BigWigFile, FaFile, TwoBitFile, Rle, ChainFile, OrgD...
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["AH2"]]'
```

Querying AnnotationHub

Finding the 'right' resource on AnnotationHub is like using Google - a well posed query is necessary to find what you are after. Useful queries are based on:

- Data provider
- Data class
- Species
- Data source
- … other metadata column information …

Example:

```
> names(mcols(hub))
 [1] "title"                "dataprovider"         "species"             
 [4] "taxonomyid"           "genome"               "description"         
 [7] "coordinate_1_based"   "maintainer"           "rdatadateadded"      
[10] "preparerclass"        "tags"                 "rdataclass"          
[13] "rdatapath"            "sourceurl"            "sourcetype"          
```

Example:

```
> length(unique(hub$species))
[1] 2296


> head(unique(hub$species))
[1] "Homo sapiens" "Vicugna pacos"    "Dasypus novemcinctus"
[4] "Otolemur garnettii" "Papio hamadryas"     "Papio anubis"

> length(unique(hub$rdataclass))
[1] 20


> unique(hub$rdataclass)
 [1] "GRanges"          "data.frame"        "Inparanoid8Db"     “TwoBitFile"
 [5] "ChainFile"        "SQLiteConnection"  "biopax"            "BigWigFile"
 [9] "AAStringSet"      "MSnSet"            "mzRpwiz"           "mzRident"
[13] "list"             "TxDb"              "Rle"               "EnsDb"
[17] "VcfFile"          “igraph”            “sqlite"            "OrgDb"
```

Example:

```
> qry <- query(hub, c("Homo sapien", "ensembl", “GRanges"))

> qry
AnnotationHub with 72 records
# snapshotDate(): 2019-07-10
# $dataprovider: Ensembl, UCSC
# $species: Homo sapiens
# $rdataclass: GRanges
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["AH5046"]]'

            title
  AH5046  | Ensembl Genes
  AH5160  | Ensembl Genes
  AH5311  | Ensembl Genes
  AH5424  | Ensembl Genes
  AH5435  | Ensembl EST Genes
  ...       ...
  AH68821 | Homo_sapiens.GRCh38.95.gtf
  AH69458 | Homo_sapiens.GRCh38.96.abinitio.gtf
  AH69459 | Homo_sapiens.GRCh38.96.chr.gtf
  AH69460 | Homo_sapiens.GRCh38.96.chr_patch_hapl_scaff.gtf
  AH69461 | Homo_sapiens.GRCh38.96.gtf
```

Example:

```
> hub["AH50377"]
AnnotationHub with 1 record
# snapshotDate(): 2019-07-10
# names(): AH50377
# $dataprovider: Ensembl
# $species: Homo sapiens
# $rdataclass: GRanges
# $rdatadateadded: 2016-01-25
# $title: Homo_sapiens.GRCh38.83.gtf
# $description: Gene Annotation for Homo sapiens
# $taxonomyid: 9606
# $genome: GRCh38
# $sourcetype: GTF
# $sourceurl: ftp://ftp.ensembl.org/pub/release-83/gtf/homo_sapiens/Homo_sap...
# $sourcesize: 45686084
# $tags:  c("GTF", "ensembl", "Gene", "Transcript", "Annotation")
# retrieve record with 'object[["AH50377"]]'
```

Example:

```
> whatIwant = hub[["AH50377"]]
downloading 1 resources
retrieving 1 resource
  |======================================================================| 100%
loading from cache
    'AH50377 : 57107'

> summary(whatIwant)
[1] "GRanges object with 2569150 ranges and 26 metadata columns"
```

```
# We talked about GRanges and TxDb; you could use as is or convert to a TxDb object
```

```
> GRCh38TxDb <- makeTxDbFromGRanges(whatIwant)
[1] "GRanges object with 2569150 ranges and 26 metadata columns"

> class(GRCh38TxDb)
[1] "TxDb"
```

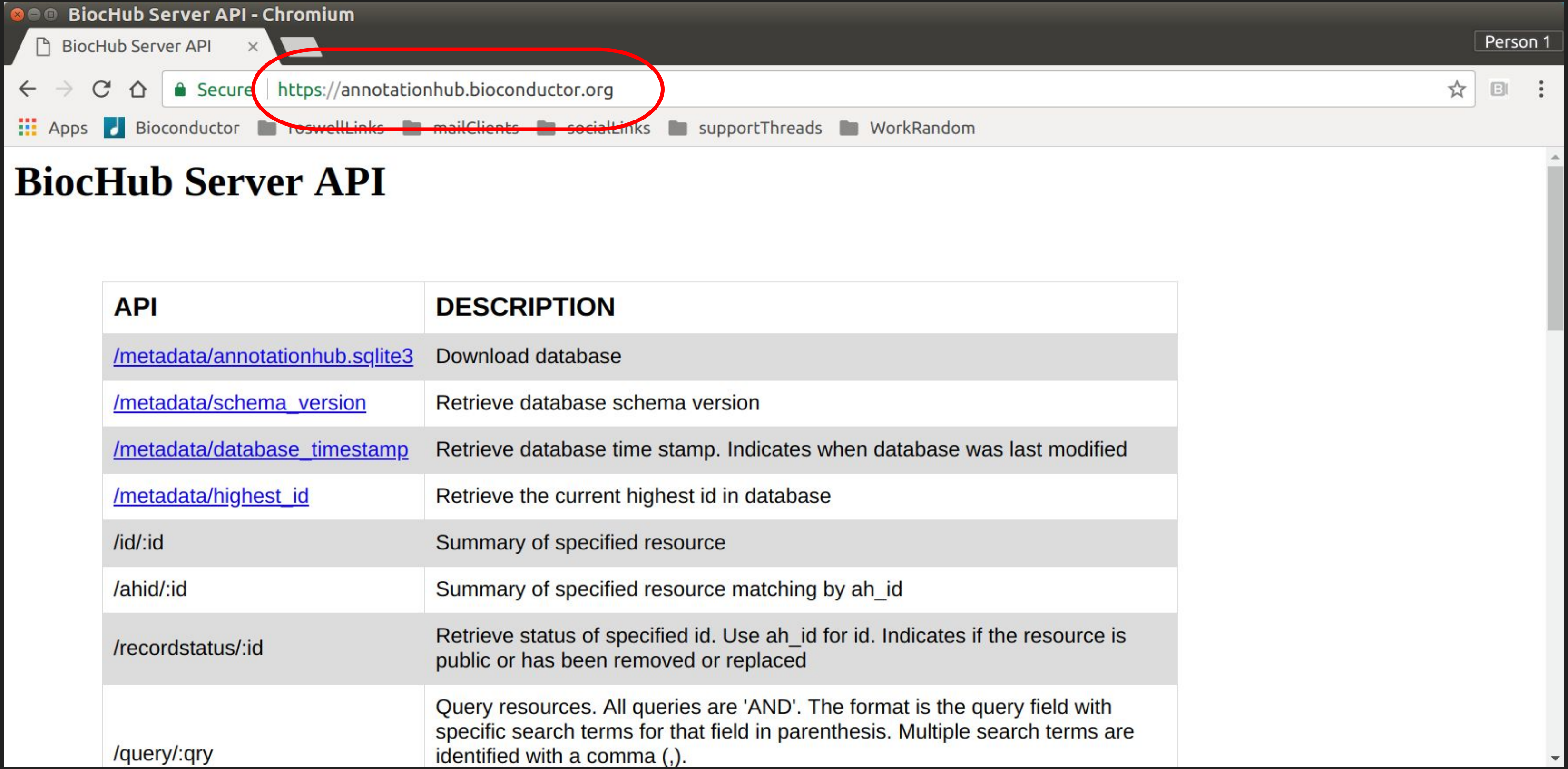
Example:

```
> gr = hub[["AH50377"]]
downloading 1 resources
retrieving 1 resource
  |======================================================================| 100%
loading from cache
    'AH50377 : 57107'



> gr = hub[["AH50377"]]
downloading 0 resources
loading from cache
    'AH50377 : 57107'
```

Example:

```
> getInfoOnIds(hub, "AH50377")
          ah_id fetch_id                      title rdataclass status
268089 AH50377    57107 Homo_sapiens.GRCh38.83.gtf    GRanges Public
       biocversion rdatadateadded rdatadateremoved file_size
268089         3.2     2016-01-25             <NA>  21737279


> subset(hub, species == "Homo sapiens" & genome=="GRCh38" & rdataclass=="VcfFile")
AnnotationHub with 4 records
# snapshotDate(): 2018-06-27
# $dataprovider: dbSNP
# $species: Homo sapiens
# $rdataclass: VcfFile
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["AH57960"]]'

            title
  AH57960 | clinvar_20160203.vcf.gz
  AH57961 | clinvar_20160203_papu.vcf.gz
  AH57962 | common_and_clinical_20160203.vcf.gz
  AH57963 | common_no_known_medical_impact_20160203.vcf.gz
```

Hub API



BiocHub Server API

API	DESCRIPTION
/metadata/annotationhub.sqlite3	Download database
/metadata/schema_version	Retrieve database schema version
/metadata/database_timestamp	Retrieve database time stamp. Indicates when database was last modified
/metadata/highest_id	Retrieve the current highest id in database
/id/:id	Summary of specified resource
/ahid/:id	Summary of specified resource matching by ah_id
/recordstatus/:id	Retrieve status of specified id. Use ah_id for id. Indicates if the resource is public or has been removed or replaced
/query/:qry	Query resources. All queries are 'AND'. The format is the query field with specific search terms for that field in parenthesis. Multiple search terms are identified with a comma (,).

ExperimentHub

ExperimentHub( )

- creates a hub object
- sqlite database backend
- Files are stored remotely and downloaded as needed
    - Bioconductor AWS S3 Buckets
    - After downloaded, cached for quick access for future runs

ExperimentHub data is associated with a Bioconductor package!

Example:

```
> eh = ExperimentHub()
snapshotDate(): 2019-07-10

> length(eh)
[1] 2377

> eh
ExperimentHub with 2377 records
# snapshotDate(): 2019-07-10
# $dataprovider: Eli and Edythe L. Broad Institute of Harvard and MIT, NA, D...
# $species: Homo Sapiens, Homo sapien, Homo sapiens, Mus musculus, Mus Muscu...
# $rdataclass: ExpressionSet, SummarizedExperiment, RaggedExperiment, DataFr...
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH1"]]'
```

Example:

```
> names(mcols(eh))
 [1] "title"               "dataprovider"        "species"
 [4] "taxonomyid"          "genome"              "description"
 [7] "coordinate_1_based" "maintainer"          "rdatadateadded"
[10] "preparerclass"       "tags"                "rdataclass"
[13] "rdatapath"           "sourceurl"           "sourcetype"


> length(unique(eh$preparerclass))
[1] 44

 > head(unique(eh$preparerclass), 15)
 [1] "GSE62944"                "alpineData"                "CellMapperData"
 [4] "HumanAffyData"           "curatedMetagenomicData" "SeqSQC"
 [7] "restfulSEData"           "curatedTCGAData"           "HarmonizedTCGAData"
[10] "HMP16SData"              "TENxBrainData"             "MetaGxOvarian"
[13] "CLLmethylation"          "tissueTreg"                "MetaGxBreast"
```

Example:

```
> query(eh, "TENxBrainData")
ExperimentHub with 4 records
# snapshotDate(): 2018-06-29
# $dataprovider: 10X Genomics
# $species: Mus musculus
# $rdataclass: character
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH1039"]]'

           title
  EH1039 | Brain scRNA-seq data, 'RLE-compressed'
  EH1040 | Brain scRNA-seq data, 'rectangular'
  EH1041 | Brain scRNA-seq data, sample (column) annotation
  EH1042 | Brain scRNA-seq data, gene (row) annotation
```

Example:

```
> query(eh, c("Mus musculus", "rna-seq"))
ExperimentHub with 158 records

# snapshotDate(): 2019-07-10
# $dataprovider: Jonathan Griffiths, Sten Linnarsson, Michael Cole, Robinson...
# $species: Mus musculus
# $rdataclass: character, SingleCellExperiment, SummarizedBenchmark, Summari...
# additional mcols(): taxonomyid, genome, description,
#   coordinate_1_based, maintainer, rdatadateadded, preparerclass, tags,
#   rdatapath, sourceurl, sourcetype
# retrieve records with, e.g., 'object[["EH1039"]]'

           title

  EH1039 | Brain scRNA-seq data, 'HDF5-based 10X Genomics' format
  EH1040 | Brain scRNA-seq data, 'dense matrix' format
  EH1041 | Brain scRNA-seq data, sample (column) annotation
  EH1042 | Brain scRNA-seq data, gene (row) annotation
  EH1074 | RNA-seq data from tissue Tregs (RPKM values)
  ...      ...
  EH2691 | Macosko retina colData
  EH2694 | Nestorowa HSC counts
  EH2695 | Nestorowa HSC colData
  EH2696 | Shekhar retina counts
  EH2697 | Shekhar retina colData
```

Example:

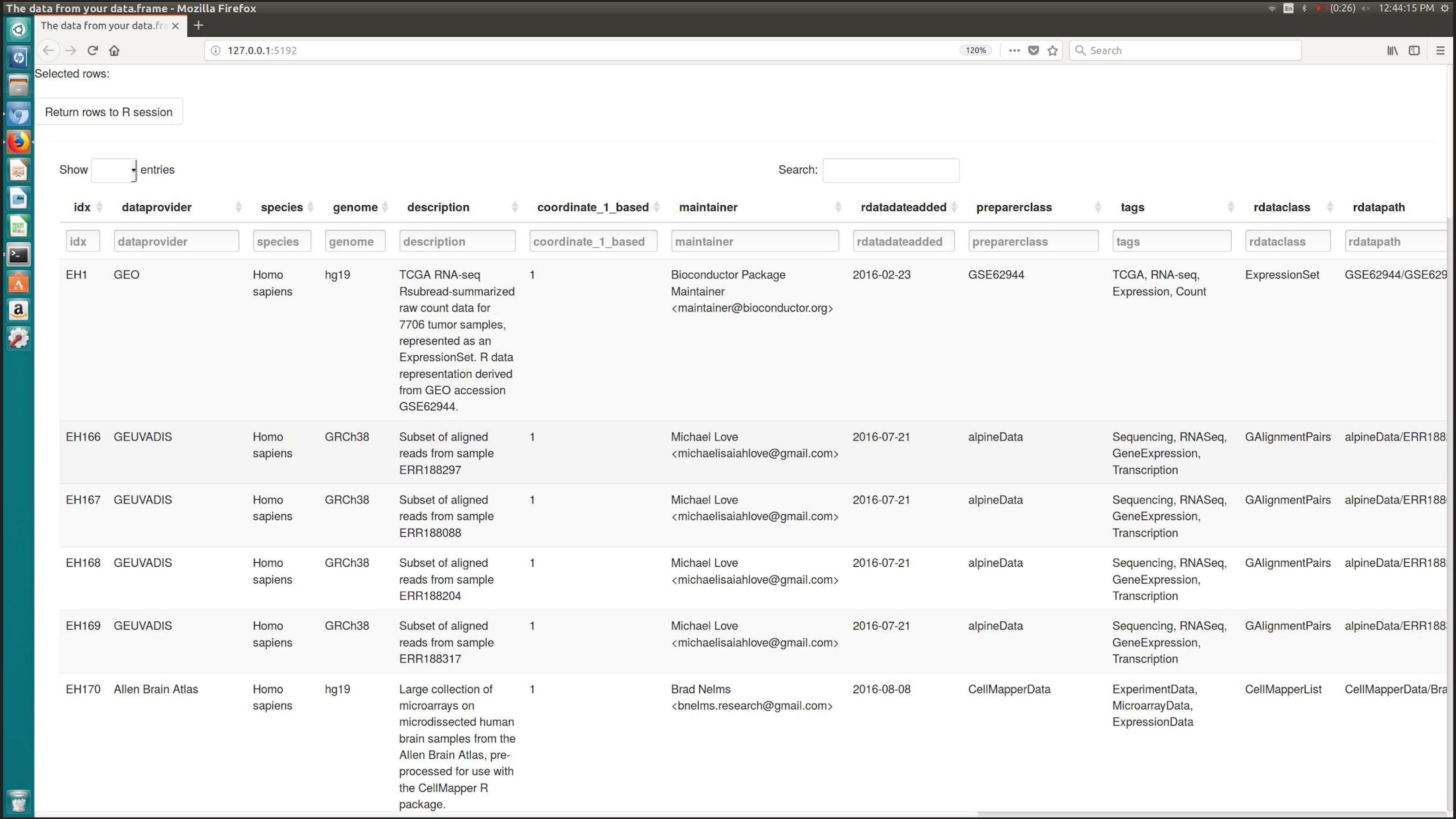
```
> # query(eh, c("Mus musculus", "rna-seq"))
```

```
> package(query(eh, c("Mus musculus", "rna-seq")))
         EH1039              EH1040              EH1041              EH1042
"TENxBrainData"     "TENxBrainData"     "TENxBrainData"     "TENxBrainData"
         EH1074              EH1075              EH1433              EH1508
   "tissueTreg"        "tissueTreg"          "allenpvc" "DuoClustering2018"
```

```
> unique(package(query(eh,c("Mus musculus", "rna-seq"))))

[1] "TENxBrainData"         "tissueTreg"            "allenpvc"

[4] "DuoClustering2018"     "benchmarkfdrData2019"  "scRNAseq"

[7] "MouseGastrulationData"
```

Example:

```
> display(eh)

Loading required package: shiny

Listening on http://127.0.0.1:5192
```



Selected rows:

idx	dataprovider	species	genome	description	coordinate_1_based	maintainer	rdatadateadded	preparerclass	tags	rdataclass	rdatapath
idx	dataprovider	species	genome	description	coordinate_1_based	maintainer	rdatadateadded	preparerclass	tags	rdataclass	rdatapath
EH1	GEO	Homo sapiens	hg19	TCGA RNA-seq Rsubread-summarized raw count data for 7706 tumor samples, represented as an ExpressionSet. R data representation derived from GEO accession GSE62944.	1	Bioconductor Package Maintainer <maintainer@bioconductor.org>	2016-02-23	GSE62944	TCGA, RNA-seq, Expression, Count	ExpressionSet	GSE62944/GSE629
EH166	GEUVADIS	Homo sapiens	GRCh38	Subset of aligned reads from sample ERR188297	1	Michael Love <michaelisaiahlove@gmail.com>	2016-07-21	alpineData	Sequencing, RNASeq, GeneExpression, Transcription	GAlignmentPairs	alpineData/ERR188
EH167	GEUVADIS	Homo sapiens	GRCh38	Subset of aligned reads from sample ERR188088	1	Michael Love <michaelisaiahlove@gmail.com>	2016-07-21	alpineData	Sequencing, RNASeq, GeneExpression, Transcription	GAlignmentPairs	alpineData/ERR188
EH168	GEUVADIS	Homo sapiens	GRCh38	Subset of aligned reads from sample ERR188204	1	Michael Love <michaelisaiahlove@gmail.com>	2016-07-21	alpineData	Sequencing, RNASeq, GeneExpression, Transcription	GAlignmentPairs	alpineData/ERR188
EH169	GEUVADIS	Homo sapiens	GRCh38	Subset of aligned reads from sample ERR188317	1	Michael Love <michaelisaiahlove@gmail.com>	2016-07-21	alpineData	Sequencing, RNASeq, GeneExpression, Transcription	GAlignmentPairs	alpineData/ERR188
EH170	Allen Brain Atlas	Homo sapiens	hg19	Large collection of microarrays on microdissected human brain samples from the Allen Brain Atlas, pre-processed for use with the CellMapper R package.	1	Brad Nelms <bnelms.research@gmail.com>	2016-08-08	CellMapperData	ExperimentData, MicroarrayData, ExpressionData	CellMapperList	CellMapperData/Bra

What’s the advantage?
From a user perspective:

Public Accessible data!

Easy access to either more data or a second set of validation data

What’s the advantage?
From a developer perspective:

Keeps the Package Lightweight!

Only download data as needed

Make large files accessible as simple objects

Resource are documented through package documentation

Packages utilize the Hub to manage files ...
User is none the wiser...

```
> library(curatedTCGAData)

## discovery

> curatedTCGAData(diseaseCode = "*", assays = "*", dry.run = TRUE)

> curatedTCGAData(diseaseCode = "COAD", assays = "RPPA*", dry.run = TRUE)


## access

> gbm <- curatedTCGAData("GBM", "RPPA*", FALSE)


## use

> experiments(gbm)[["GBM_RPPAArray-20160128"]]
```