AnnotationHub: A client package for retrieving data from the AnnotationHub web service

Marc Carlson

October 14, 2013

1 AnnotationHub Objects

The *AnnotationHub* package provides a client interface to resources stored at the AnnotationHub web service.

```
> library(AnnotationHub)
```

The *AnnotationHub* package is straightforward to use. The 1st thing you need to do to make use of it is to create an `AnnotationHub` object like this:

```
> ah = AnnotationHub()
```

Now at this point you have already done everything you need in order to get annotations. If you know exactly what the resource you want is called (and where it can be found), you could get it right now by just tab completing to it using the `$` operator.

Lets suppose that you knowd the following is the path to your data:

`ah$goldenpath.hg19.encodeDCC.wgEncodeUwTfbs.wgEncodeUwTfbsMcf7CtcfStdPkRep1. narrowPeak_0.0.1.RData`

Simply tab completing to the above path (followed by hitting enter), as demonstrated below will actually retrieve an object and then assign its contents to a local variable called res.

```
> res <- ah$goldenpath.hg19.encodeDCC.wgEncodeUwTfbs.wgEncodeUwTfbsMcf7CtcfStdPkRep1.
```

As you can see it's pretty easy to get data out using `AnnotationHub` objects. The rest of this vignette is mostly about helping you to make sure you are accessing the version of `AnnotationHub` objects that you intend to use and also about making sure that you can filter down the huge number of objects to the few that you are really interested in.

2 Configuring `AnnotationHub` objects

When you create the `AnnotationHub` object, it will set up the object for you with some default settings. If you look at the object you will see some helpful information about it.

```
> ah

class: AnnotationHub
length: 8674
filters: none
hubUrl: http://annotationhub.bioconductor.org/ah
snapshotVersion: 2.13; snapshotDate: 2013-06-29
hubCache: /home/biocbuild/.AnnotationHub
```

By default, you can see that the `AnnotationHub` object is set to the latest snapshotData and a snapshot version that matches the version of Bioconductor that you are using. You can also learn about these data with the appropriate methods.

```
> snapshotVersion(ah)

[1] "2.13"

> snapshotDate(ah)

[1] "2013-06-29"
```

If you are interested in using an older version of a snapshot, you can list previous versions with the `possibleDates` like this:

```
> pd <- possibleDates(ah)
> pd

 [1] "2013-03-20" "2013-03-21" "2013-03-22" "2013-03-27" "2013-04-05"
 [6] "2013-04-30" "2013-06-24" "2013-06-25" "2013-06-26" "2013-06-27"
[11] "2013-06-28" "2013-06-29"
```

And then you can set the dates like this:

```
> snapshotDate(ah) <- pd[1]
```

3 Exploring and setting filters for `AnnotationHub`

If you are interested in how many annotation resources are currently available for your `AnnotationHub` object, you can just take the length like this:

```
> length(ah)
```

```
[1] 8674
```

Similarly, there are also methods to show the resource names, or even the full set of resource URLs for available resources.

```
> names <- head(names(ah),n=1)

> names
```

```
dbSNP.organisms.human_9606.VCF.ByChromosome.01.12156.ASW.RData
```

```
> urls <- head(snapshotUrls(ah),n=1)

> urls
```

```
http://annotationhub.bioconductor.org/ah/dbSNP/organisms/human_
9606/VCF/ByChromosome/01-12156-ASW.RData
```

For humans, the number of resources available is going to be overwhelming. How should we cut this data set down to size? For this task, we introduce filters. Every `AnnotationHub` object contains a list of filters that can be configured to control which resources it can return. By default this list is empty, which means you get everything.

```
> filters(ah)
```

```
list()
```

How can we learn which things are available for filtering? For this we have defined `columns` and `keytypes` methods, which will list all the kinds of data that can be filtered on.

```
> columns(ah)
```

```
 [1] "BiocVersion"            "Coordinate_1_based"    "DataProvider"
 [4] "Description"            "Genome"                "Maintainer"
 [7] "RDataClass"             "RDataDateAdded"        "RDataLastModifiedDate"
[10] "RDataPath"              "RDataSize"             "RDataVersion"
[13] "Recipe"                 "RecipeArgs"            "SourceFile"
[16] "SourceSize"             "SourceUrl"             "SourceVersion"
[19] "Species"                "Tags"                  "TaxonomyId"
[22] "Title"
```

```
> keytypes(ah)
```

```
 [1] "BiocVersion"            "Coordinate_1_based"    "DataProvider"
 [4] "Description"            "Genome"                "Maintainer"
 [7] "RDataClass"             "RDataDateAdded"        "RDataLastModifiedDate"
[10] "RDataPath"              "RDataSize"             "RDataVersion"
[13] "Recipe"                 "RecipeArgs"            "SourceFile"
[16] "SourceSize"             "SourceUrl"             "SourceVersion"
[19] "Species"                "Tags"                  "TaxonomyId"
[22] "Title"
```

Once we know which things can be used to filter on, we can extract values that these things can be required to match. For this task, we have defined a `key` method.

```
> head(keys(ah, keytype="Species"))
```

```
[1] "Ailuropoda melanoleuca" "Anolis carolinensis"    "Anopheles gambiae"
[4] "Apis mellifera"         "Aplysia californica"    "Bos taurus"
```

Now we are able to construct and assign a filter to our `AnnotationHub` object. Lets set it up to only find resources from humans.

```
> filters(ah) <- list(Species="Homo sapiens")
```

And now if we look we will see that our `AnnotationHub` object is only exposing resources from Homo sapiens.

```
> length(ah)
```

```
[1] 5324
```

```
> names <- head(names(ah),n=1)
> names
```

goldenpath.hg19.encodeDCC.wgEncodeCshlShortRnaSeq.wgEncodeCshlShortRnaSeqK562Chrom
bedRnaElements_0.0.1.RData

```
> urls <- head(snapshotUrls(ah),n=1)
```

```
> urls
```

http://annotationhub.bioconductor.org/ah/goldenpath/hg19/encodeDCC/
wgEncodeCshlShortRnaSeq/wgEncodeCshlShortRnaSeqK562ChromatinShorttotalTapContigs.
bedRnaElements_0.0.1.RData

4 Using `AnnotationHub` to retrieve data

So now that we have our `AnnotationHub` object configured to expose only the data for humans how would we go about getting that data downloaded? As mentioned above, we can use the `$` operator and tab completion to pull down a data source of interest like this.

ah$goldenpath.hg19.encodeDCC.wgEncodeUwTfbs.wgEncodeUwTfbsMcf7CtcfStdPkRep1.
narrowPeak_0.0.1.RData

Just by using tab completion like this:

```
> res <- ah$goldenpath.hg19.encodeDCC.wgEncodeUwTfbs.wgEncodeUwTfbsMcf7CtcfStdPkRep1.
```

And once you have done this, you can look at the object stored in res and use it etc.. Any dependencies that you need to use this kind of object should automatically try to load at this time.

```
> res
```

```
GRanges with 82163 ranges and 6 metadata columns:
          seqnames                 ranges strand   |        name     score
             <Rle>              <IRanges>  <Rle>   | <character> <integer>
      [1]     chr1       [237640, 237790]      *   |           .         0
      [2]     chr1       [544660, 544810]      *   |           .         0
      [3]     chr1       [567480, 567630]      *   |           .         0
      [4]     chr1       [569820, 569970]      *   |           .         0
      [5]     chr1       [714200, 714350]      *   |           .         0
      ...      ...                    ...    ... ...         ...       ...
  [82159]     chrX [154764540, 154764690]      *   |           .         0
  [82160]     chrX [154807400, 154807550]      *   |           .         0
  [82161]     chrX [154881060, 154881210]      *   |           .         0
  [82162]     chrX [154892100, 154892250]      *   |           .         0
  [82163]     chrX [154916040, 154916190]      *   |           .         0
         signalValue    pValue    qValue      peak
           <numeric> <numeric> <numeric> <integer>
      [1]         30  26.89200        -1        -1
      [2]          6   8.16393        -1        -1
      [3]        100  56.71760        -1        -1
      [4]         85  49.65350        -1        -1
      [5]         17  13.18360        -1        -1
      ...        ...       ...       ...       ...
  [82159]         26   25.2917        -1        -1
  [82160]         22   27.6521        -1        -1
  [82161]         17   16.4194        -1        -1
  [82162]         72  101.6090        -1        -1
  [82163]         32   32.5209        -1        -1
  ---
  seqlengths:
        chr1     chr10     chr11     chr12 ...      chr8      chr9      chrX
   249250621 135534747 135006516 133851895 ... 146364022 141213431 155270560
```

Also, since you have previously downloaded this object at the start of this vignette, the 2nd time it should pull this object from a local cache that *AnnotationHub* will have created for you. This is a feature of *AnnotationHub* that is meant to provide better performance by removing the need to pull a large amount of data from a distant server every time. However, this does not mean that once you have used `AnnotationHub` to retrieve data that you no longer need to have internet access. This is because whenever you create a `AnnotationHub` object, it needs to talk to the metadata server to learn about things like the latest available version etc. So if you intend to access your objects on the plane you will need to either save them to a convenient location or else take note of where your local cache is located so that you can load them up manually later.

5 Session Information

```
R version 3.0.2 (2013-09-25)
Platform: x86_64-unknown-linux-gnu (64-bit)
```

```
locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        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] parallel  stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] GenomicRanges_1.14.0 XVector_0.2.0        AnnotationHub_1.2.0
[4] IRanges_1.20.0       BiocGenerics_0.8.0

loaded via a namespace (and not attached):
[1] AnnotationDbi_1.24.0 Biobase_2.22.0       BiocInstaller_1.12.0
[4] DBI_0.2-7            RSQLite_0.11.4       rjson_0.2.13
[7] stats4_3.0.2         tools_3.0.2
```