restfulSE – experiments with HDF5 server content wrapped in SummarizedExperiment

Vincent J. Carey, stvjc at channing.harvard.edu, Shweta Gopaulakrishnan, reshg at channing.harvard.edu

May 11, 2017

Contents

1 Introduction

Extensive human and computational effort is expended on downloading and managing large genomic data at site of analysis. Interoperable formats that are accessible via generic operations like those in RESTful APIs may help to improve cost-effectiveness of genome-scale analyses.

In this report we examine the use of HDF5 server as a back end for assay data, mediated through the RangedSummarizedExperiment API for interactive use.

A modest server configured to deliver HDF5 content via a RESTful API has been prepared and is used in this vignette.

2 Executive summary

We want to provide rapid access to array-like data. We'll work with the Banovich 450k data as there is a simple check against an in-memory representation.

```
library(restfulSE)
bigec2 = H5S_source("http://54.174.163.77:5000")
## analyzing groups for their links...
## done
bigec2
## HDF5 server domain:  http://54.174.163.77:5000
##  There are 10 groups.
##  Use groups(), links(), ..., to probe and access metadata.
##  Use dsmeta() to get information on datasets within groups.
##  Use [[ [dsname] ]]  to get a reference suitable for [i, j] subsetting.
dsmeta(bigec2)[1:2,] # two groups
## DataFrame with 2 rows and 3 columns
```

```
##    groupnum                        dsnames
##   <integer>                 <CharacterList>
## 1         1 tissues,assays,neurons100k,...
## 2         2
##                               grp.uuid
##                              <character>
## 1 c3ca306c-3020-11e7-806d-123feca22a06
## 2 c3df8476-3020-11e7-806d-123feca22a06
dsmeta(bigec2)[1,2][[1]] # all dataset candidates in group 1
## [1] "tissues"          "assays"           "neurons100k"
## [4] "neurons400k"      "tenx_100k_sorted"
```

We use double-bracket subscripting to grab a reference to a dataset from an H5S source.

```
banref = bigec2[["assays"]] # arbitrary name assigned long ago
banref
## H5S_dataset instance:
##   dsname intl.dim1 intl.dim2              created      type.base
## 1 assays        64    329469 2017-04-05T18:02:37Z H5T_IEEE_F64LE
```

We build a RESTfulSummarizedExperiment by combining an assay-free RangedSummarizedExperiment with this reference.

```
data(banoSEMeta)
rbano = RESTfulSummarizedExperiment(banoSEMeta, banref)
rbano
## class: RESTfulSummarizedExperiment
## dim: 329469 64
## metadata(0):
## assays(1): (served by HDF5Server)
## rownames(329469): cg00000029 cg00000165 ... ch.9.98989607R
##   ch.9.991104F
## rowData names(10): addressA addressB ... probeEnd probeTarget
## colnames(64): NA18498 NA18499 ... NA18489 NA18909
## colData names(35): title geo_accession ... data_row_count naid
```

We can update the SummarizedExperiment metadata through subsetting operations, and then extract the relevant assay data. The data are retrieved from the remote server with the assay method.

```
rbanoSub = rbano[5:8, c(3:9, 40:50)]
## Loading required package: Biostrings
## Loading required package: XVector
##
## Attaching package: 'Biostrings'
## The following object is masked from 'package:DelayedArray':
##
##     type
## The following object is masked from 'package:base':
##
##     strsplit
assay(rbanoSub)
##                 NA18501    NA18502    NA18516    NA18517    NA18519
## cg00000363  0.325433263  1.3778200  0.5966999 -1.0747716 -0.2686108
## cg00000622  0.003436888 -0.6684993 -1.2106348  0.4990709  0.4555032
## cg00000714 -1.184443665 -1.6540480 -0.1747294 -0.7111111  1.5458591
## cg00000734  0.153831565 -1.2992894  1.9039768  0.8735532 -0.1379805
##              NA18520    NA18855    NA18861    NA18862     NA18870
```

```
## cg00000363 1.2819188 -0.4633973  1.1110107  0.6474392  1.55193324
## cg00000622 0.8974313 -0.7700943  1.3585617 -1.0815012 -0.17871603
## cg00000714 0.5384043  1.0082528 -0.6950679  0.2298506 -0.01901198
## cg00000734 0.5042480  0.1923326  0.5121181 -1.0299498 -0.13733521
##               NA18871    NA19101    NA19102    NA19116    NA19119
## cg00000363 -1.1042549  0.1761046 -1.4397586 -0.2367100  0.6647072
## cg00000622 -0.2178474  0.6475906  0.3519086 -0.5933287  0.2022896
## cg00000714  1.7322850 -0.7990096  1.4408611  0.4633499  0.6167595
## cg00000734  0.5520922  0.5521181 -0.2285199 -0.2110932 -0.5053562
##               NA19137    NA19138      NA19140
## cg00000363  0.4485182 0.96669567  1.203765271
## cg00000622  0.2493717 0.07606248  0.958031578
## cg00000714  0.1988236 0.32574295 -0.008202908
## cg00000734 -0.6834287 1.18532042  0.319937329
```

3 10xGenomics example

We have used Martin Morgan's TENxGenomics package to create a dense HDF5 representation of the assay data, and placed it on the `bigec2` server. The metadata are available as `st100k` in this package; we have used EnsDb.Mmusculus.v79 to supply gene ranges where available; genes reported but without addresses are addressed at chr1:2 with width 0. The rows are sorted by genomic address within chromosomes.

```
txdat = bigec2[["tenx_100k_sorted"]]
data(st100k)
tenx100k = RESTfulSummarizedExperiment( st100k,
   txdat )
tenx100k
## class: RESTfulSummarizedExperiment
## dim: 27998 100000
## metadata(0):
## assays(1): (served by HDF5Server)
## rownames(27998): ENSMUSG00000109048 ENSMUSG00000109510 ...
##   ENSMUSG00000096768 ENSMUSG00000096850
## rowData names(6): gene_id gene_name ... seq_coord_system symbol
## colnames(100000): AAACCTGAGATAGGAG-1 AAACCTGAGCGGCTTC-1 ...
##   GACGTTAGTCATACTG-11 GACGTTAGTCCGTGAC-11
## colData names(4): Barcode Sequence Library Mouse
```

We will subset genes annotated to hippocampus development. Here are some related categories:

```
12092 GO:0021766                     hippocampus development
12096 GO:0021770           parahippocampal gyrus development
34609 GO:0097410      hippocampal interneuron differentiation
34631 GO:0097432 hippocampal pyramidal neuron differentiation
34656 GO:0097457                     hippocampal mossy fiber
35169 GO:0098686       hippocampal mossy fiber to CA3 synapse
42398 GO:1990026           hippocampal mossy fiber expansion
```

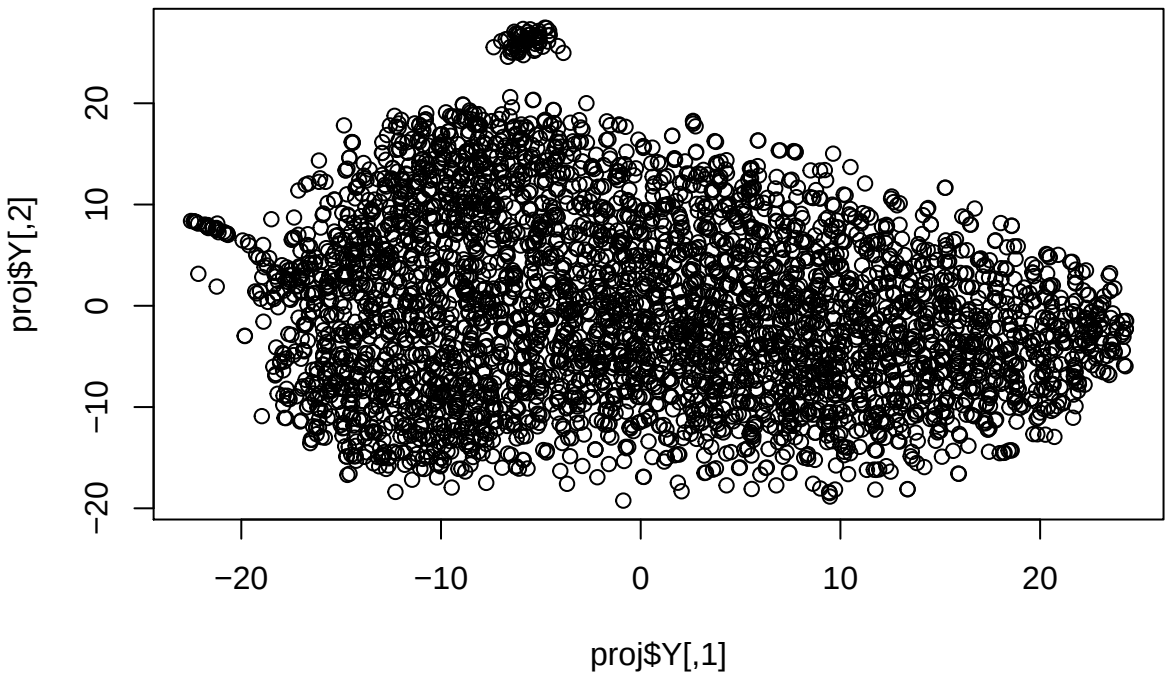
```
library(org.Mm.eg.db)
## Loading required package: AnnotationDbi
##
## Attaching package: 'AnnotationDbi'
## The following object is masked from 'package:restfulSE':
```

```
##
##     links
##
atab = select(org.Mm.eg.db, keys="GO:0021766", keytype="GO", columns="ENSEMBL")
## 'select()' returned 1:many mapping between keys and columns
hg = atab[,"ENSEMBL"]
length(hgok <- intersect(hg, rownames(tenx100k)))
## [1] 50
```

This is a very scattered collection of rows in the matrix. We acquire expression measures for genes annotated to hippocampus on 4000 samples. t-SNE is then used to project the log-transformed measures to the plane.

```
hipn = assay(tenx100k[hgok,1:4000])  # slow
d = dist(t(log(1+hipn)), method="manhattan")
proj = Rtsne(d)
```

```
plot(proj$Y)
```



4 Background

Banovich et al. published a subset of DNA methylation measures assembled on 64 samples of immortalized B-cells from the YRI HapMap cohort.

```
library(restfulSE)
data(banoSEMeta)
```

```
banoSEMeta
## class: RangedSummarizedExperiment
## dim: 329469 64
## metadata(0):
## assays(0):
## rownames(329469): cg00000029 cg00000165 ... ch.9.98989607R
##   ch.9.991104F
## rowData names(10): addressA addressB ... probeEnd probeTarget
## colnames(64): NA18498 NA18499 ... NA18489 NA18909
## colData names(35): title geo_accession ... data_row_count naid
```

The numerical data have been exported using H. Pages' saveHDF5SummarizedExperiment applied to the banovichSE SummarizedExperiment in the yriMulti package. The HDF5 component is simply copied into the server data space on the remote server.

5 Hierarchy of server resources

5.1 Server

Given the URL of a server running HDF5 server, we create an instance of `H5S_source`:

```
mys = H5S_source(serverURL="http://54.174.163.77:5000")
## analyzing groups for their links...
## done
mys
## HDF5 server domain:  http://54.174.163.77:5000
##  There are 10 groups.
##  Use groups(), links(), ..., to probe and access metadata.
##  Use dsmeta() to get information on datasets within groups.
##  Use [[ [dsname] ]]  to get a reference suitable for [i, j] subsetting.
```

5.2 Groups

The server identifies a collection of 'groups'. For the server we are working with, only one group, at the root, is of interest.

```
groups(mys)
## DataFrame with 10 rows and 2 columns
##                                    groups    nlinks
##                                <character> <integer>
## 1  c3ca306c-3020-11e7-806d-123feca22a06         7
## 2  c3df8476-3020-11e7-806d-123feca22a06         1
## 3  c3ca7f5e-3020-11e7-806d-123feca22a06        88
## 4  c3cab2c6-3020-11e7-806d-123feca22a06         1
## 5  cbf54056-3020-11e7-806d-123feca22a06         4
## 6  c8f49ff0-3020-11e7-806d-123feca22a06        28
## 7  c3e00f40-3020-11e7-806d-123feca22a06         1
## 8  c3ca6640-3020-11e7-806d-123feca22a06         1
## 9  c3e20b6a-3020-11e7-806d-123feca22a06         1
## 10 c3ca9926-3020-11e7-806d-123feca22a06         3
```

5.3 Links for a group

There is a class to hold the link set for any group:

```
lin1 = restfulSE::links(mys,1)
lin1
## HDF5 server link set for group c3ca306c-3020-11e7-806d-123feca22a06
##  There are 7 links.
##  Use targets([linkset]) to extract target URLs.
```