Reading HDF5

Jeffrey Leek

May 18, 2016

HDF5

- Used for storing large data sets
- Supports storing a range of data types
- Hierarchical data format
- *groups* containing zero or more data sets and metadata
- Have a *group header* with group name and list of attributes
- Have a *group symbol table* with a list of objects in group
- *datasets* multidimensional array of data elements with metadata
- Have a *header* with name, datatype, dataspace, and storage layout
- Have a *data array* with the data

`http://www.hdfgroup.org/`

R HDF5 package

```
source("http://bioconductor.org/biocLite.R")
biocLite("rhdf5")
```

```
library(rhdf5)
created = h5createFile("example.h5")
created
```

```
## [1] TRUE
```

- This will install packages from Bioconductor `http://bioconductor.org/`, primarily used for genomics but also has good "big data" packages
- Can be used to interface with hdf5 data sets.
- This lecture is modeled very closely on the rhdf5 tutorial that can be found here `http://www.bioconductor.org/packages/release/bioc/vignettes/rhdf5/inst/doc/rhdf5.pdf`

Create groups

```
created = h5createGroup("example.h5","foo")
created = h5createGroup("example.h5","baa")
created = h5createGroup("example.h5","foo/foobaa")
h5ls("example.h5")
```

```
##   group   name     otype dclass dim
## 0     /    baa H5I_GROUP
## 1     /    foo H5I_GROUP
## 2  /foo foobaa H5I_GROUP
```

Write to groups

```
A = matrix(1:10,nr=5,nc=2)
h5write(A, "example.h5","foo/A")
B = array(seq(0.1,2.0,by=0.1),dim=c(5,2,2))
attr(B, "scale") <- "liter"
h5write(B, "example.h5","foo/foobaa/B")
h5ls("example.h5")
```

```
##          group   name       otype  dclass       dim
## 0            /    baa   H5I_GROUP                  
## 1            /    foo   H5I_GROUP                  
## 2         /foo      A H5I_DATASET INTEGER     5 x 2
## 3         /foo foobaa   H5I_GROUP                  
## 4 /foo/foobaa      B H5I_DATASET   FLOAT 5 x 2 x 2
```

Write a data set

```
df = data.frame(1L:5L,seq(0,1,length.out=5),
  c("ab","cde","fghi","a","s"), stringsAsFactors=FALSE)
h5write(df, "example.h5","df")
h5ls("example.h5")
```

```
##           group   name       otype   dclass       dim
## 0             /    baa   H5I_GROUP                   
## 1             /     df H5I_DATASET COMPOUND         5
## 2             /    foo   H5I_GROUP                   
## 3          /foo      A H5I_DATASET  INTEGER     5 x 2
## 4          /foo foobaa   H5I_GROUP                   
## 5 /foo/foobaa      B H5I_DATASET    FLOAT 5 x 2 x 2
```

Reading data

```
readA = h5read("example.h5","foo/A")
readB = h5read("example.h5","foo/foobaa/B")
readdf= h5read("example.h5","df")
readA
```

```
##      [,1] [,2]
## [1,]    1    6
## [2,]    2    7
## [3,]    3    8
## [4,]    4    9
## [5,]    5   10
```

Writing and reading chunks

```
h5write(c(12,13,14),"example.h5","foo/A",index=list(1:3,1))
h5read("example.h5","foo/A")
```

```
##      [,1] [,2]
## [1,]   12    6
## [2,]   13    7
## [3,]   14    8
## [4,]    4    9
## [5,]    5   10
```

Notes and further resources

- hdf5 can be used to optimize reading/writing from disc in R
- The rhdf5 tutorial:
- `http://www.bioconductor.org/packages/release/bioc/vignettes/rhdf5/inst/doc/rhdf5.pdf`
- The HDF group has informaton on HDF5 in general `http://www.hdfgroup.org/HDF5/`