rhdf5: HDF5 in the Bioconductor ecosystem

Mike L. Smith
@grimbough
● Statistical programming language & environment

● Great for interactive data exploration & rapid prototyping

  > plot(1:10)

  > mean(1:10)

  [1] 5.5

● 10,000s of addon “packages”
  ○ CRAN, Github, etc
  ○ Cover a huge range of topics and application areas
  ○ Easy to install (most of the time)
- Additional R package repository with specific focus on biological research
- Has more / different rules than CRAN!
  - intention is to make better software & improve user experience
  - package review, minimum documentation requirements, daily CI testing, ...
- Strong emphasis on code reuse and modularisation within the ecosystem
  - Core infrastructure implemented once and used by everyone
  - e.g. reading specific file types, classes representing common data types
- HDF5 falls into this category
rhdf5 package
Three connected packages
Rhdf5lib

- Distributes static HDF5 library (currently 1.10.6)
- Ensures consistent version for users
- Ensures consistent installation instructions and toolchain
- Compiles on Linux and Mac, pre-compiled for Windows
rhdf5filters

- Distributes several dynamic filters
  - bzip2
  - lzf
  - blosc
    - blosclz, lz4, lz4hc, snappy, zstd, zlib

- Sets HDF5_PLUGIN_PATH environment variable in R session

- Can be used by external programs too
rhdf5

- Provides “high” and “low” level interfaces with C-API
- Reasonable coverage at “low level” with H5X() functions
  - Mapping to C interface
- “High level” functions for common operations - h5x() 
  - Wrappers with default choices made
rhdf5 - C-API mapping

```c
fid <- H5Fcreate( name = "/my/special/file.h5" )
sid <- H5Screate_simple( c(2,1) )
did <- H5Dcreate( fid, "A", "H5T_STD_I32LE", sid )
H5Dwrite(did, 1L:2L, h5spaceMem = sid, h5spaceFile = sid)
H5Dclose( did )
H5Sclose( sid )
H5Fclose( fid )
```
rhdf5 - wrapper functions

```r
h5createFile( file = "/my/special/file.h5" )
h5write( file = "/my/special/file.h5", obj = 1L:2L, name = "A" )
h5read( file = "/my/special/file.h5", name = "A" )
```

## [1] 1 2
Example use case:
Single-cell sequencing
It’s all about a counts matrix

Single-cell RNA-Seq (scRNA-Seq)

Tissue (e.g. tumor) → Isolate and sequence individual cells → Gene 1

Stephanie Hicks - Welcome to the World of Single-Cell RNA-Sequencing
It’s all about a counts matrix

Single-cell RNA-Seq (scRNA-Seq)

Tissue (e.g. tumor) → Isolate and sequence individual cells → Gene 1

<table>
<thead>
<tr>
<th>Gene</th>
<th>Cell 1</th>
<th>Cell 2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene 1</td>
<td>18</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Gene 2</td>
<td>1010</td>
<td>506</td>
<td></td>
</tr>
<tr>
<td>Gene 3</td>
<td>0</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>Gene 4</td>
<td>22</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Stephanie Hicks - Welcome to the World of Single-Cell RNA-Seqencing
It's all about a counts matrix

Single-cell RNA-Seq (scRNA-Seq)

Tissue (e.g. tumor) → Isolate and sequence individual cells → Gene 1 Cell 1

Read Counts

<table>
<thead>
<tr>
<th>Gene</th>
<th>Cell 1</th>
<th>Cell 2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene 1</td>
<td>18</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Gene 2</td>
<td>1010</td>
<td>506</td>
<td></td>
</tr>
<tr>
<td>Gene 3</td>
<td>0</td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>Gene 4</td>
<td>22</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Bioconductor defines a common class for this data
Counts matrices

- Data are typically sparse ( > 90% zeros )
- Number of genes & cells vary a lot
- Small datasets can be represented in memory
  - Either dense or sparse representations
- Large datasets (30,000 genes, > 1,000,000 cells) need another solution
  - HDF5 backed on-disk arrays
HDF5Array package provides familiar R interface to on-disk arrays

- Drop-in replacement for in-memory arrays
- Points to a single HDF5 dataset
- Upstream analysis packages don’t (necessarily) care
- In practice algorithms probably need to be optimised - many are

```
M1 <- HDF5Array(
  file = "~/my/special/file.h5",
  name = "counts" )

M1[1:10, ]
mean( M1 )
```
On-disk single-cell software stack

- Rhdf5lib
- rhdf5
- HDF5Array
- SingleCellExperiment

- zinbwave
- scater
- scran
- scone
- scPipe
- DropletUtils
- ...

C / C++ Library  R Interface  Counts Matrix  Complete SC Dataset  Analysis Tools
Thanks to EMBL Huber Lab & BioC community!

https://bioconductor.org/packages/rhdf5