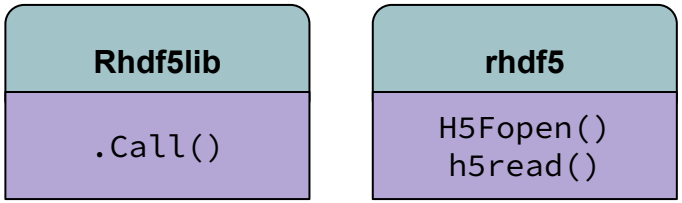
Updates to **rhdf5**

Mike Smith
@grimbough

Introduction to **rhdf5**

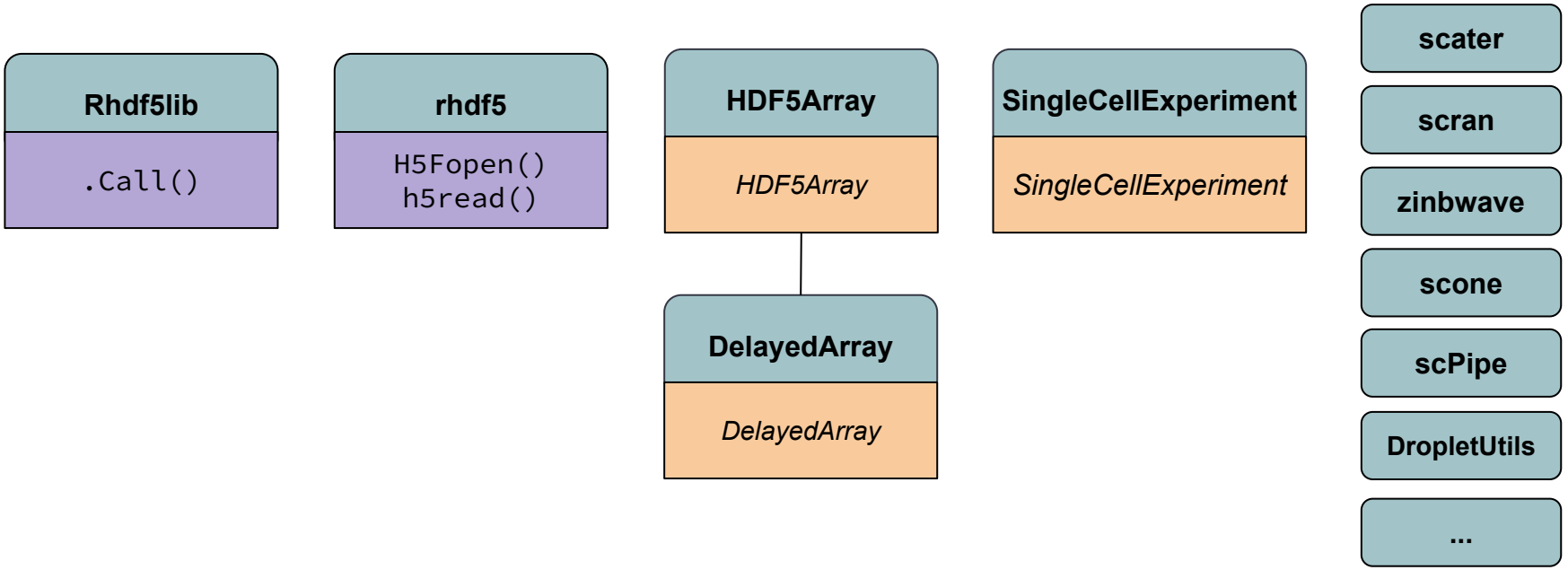
One of several packages providing this functionality

Introduction to **rhdf5**



C / C++ Library | R Interface | Counts Matrix | Complete SC Dataset | Analysis Tools

Introduction to **rhdf5**



C / C++ Library → R Interface → Counts Matrix → Complete SC Dataset → Analysis Tools

Update to underlying HDF5 library

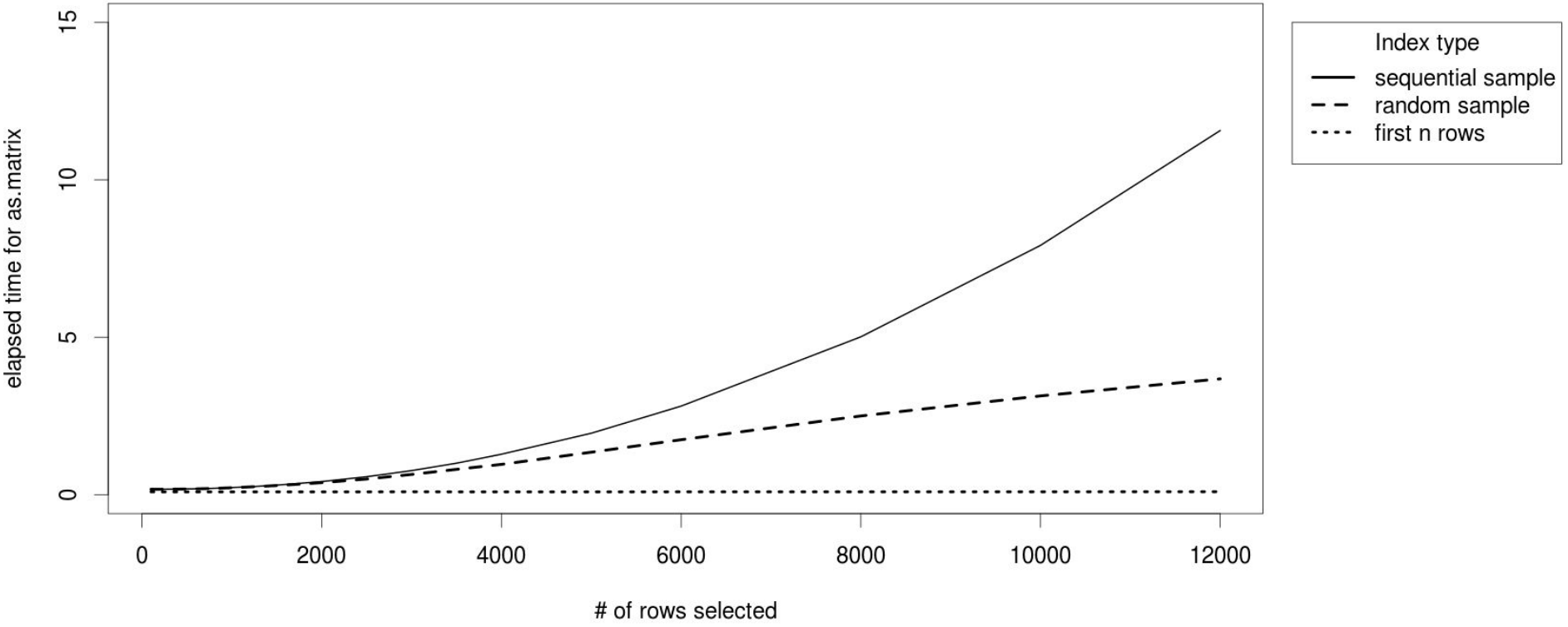
- Switch from HDF5 version 1.8 to 1.10
- Motivated by users unable to open files created by other software
  - https://support.bioconductor.org/p/109845
  - https://github.com/pachterlab/sleuth/issues/175
- Not a simple drop-in replacement!
  - *“The hid_t type was changed from 32-bit to a 64-bit value.”*
  - Change should be transparent to users
- Potential interesting new features e.g. SWMR

Performance improvements

- Low hanging fruit - classic R inefficiencies
  - Unnecessary reordering
  - Copying rather than preallocating

https://github.com/lgatto/MSnbase/issues/395

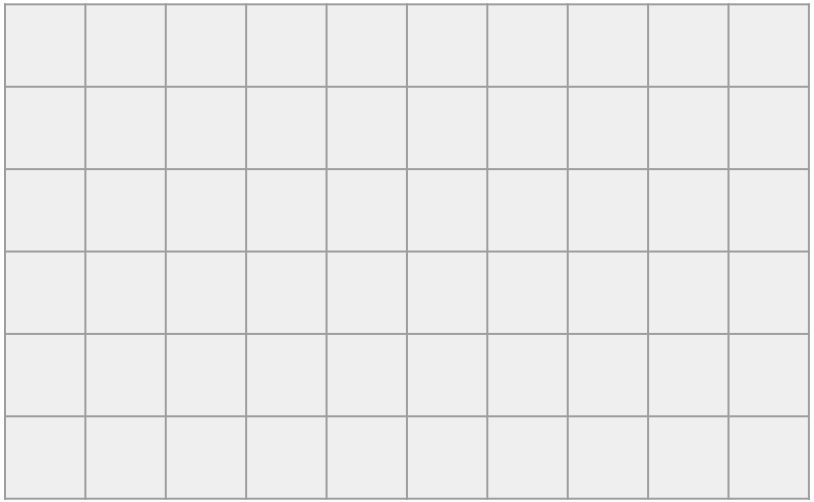
Performance improvements



https://github.com/grimbough/rhdf5/issues/31

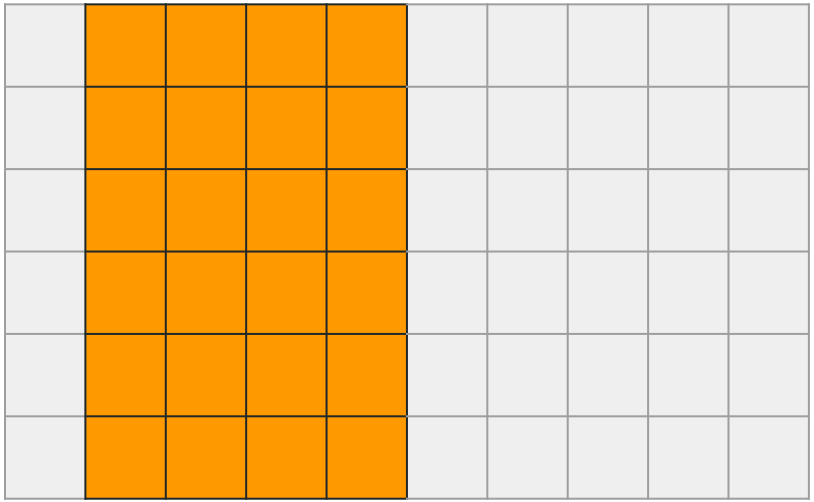
HDF5 Hyperslabs

- Regularly spaced selections of elements



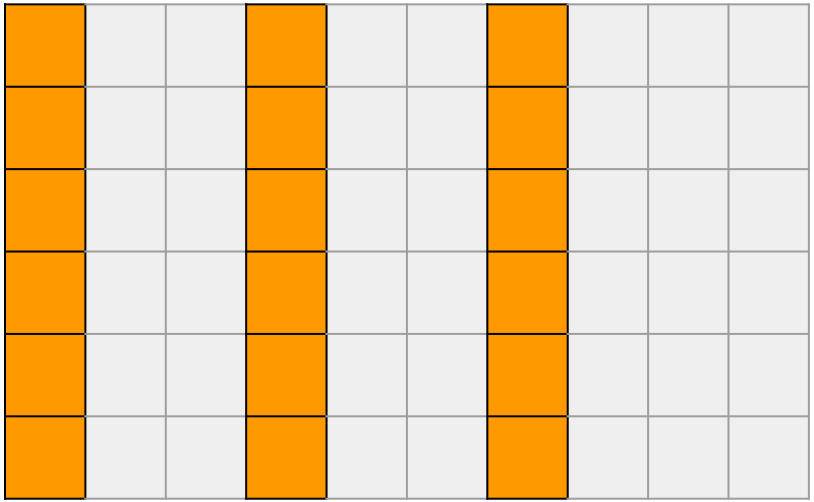
HDF5 Hyperslabs

- Regularly spaced selections of elements



HDF5 Hyperslabs

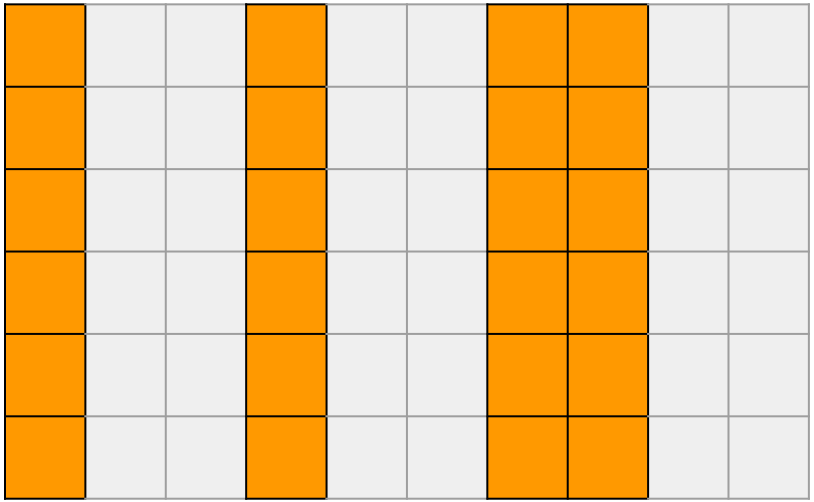
- Regularly spaced selections of elements



- Defined by *offset*, *count*, *stride* and *block*
- Available in **rhdf5**

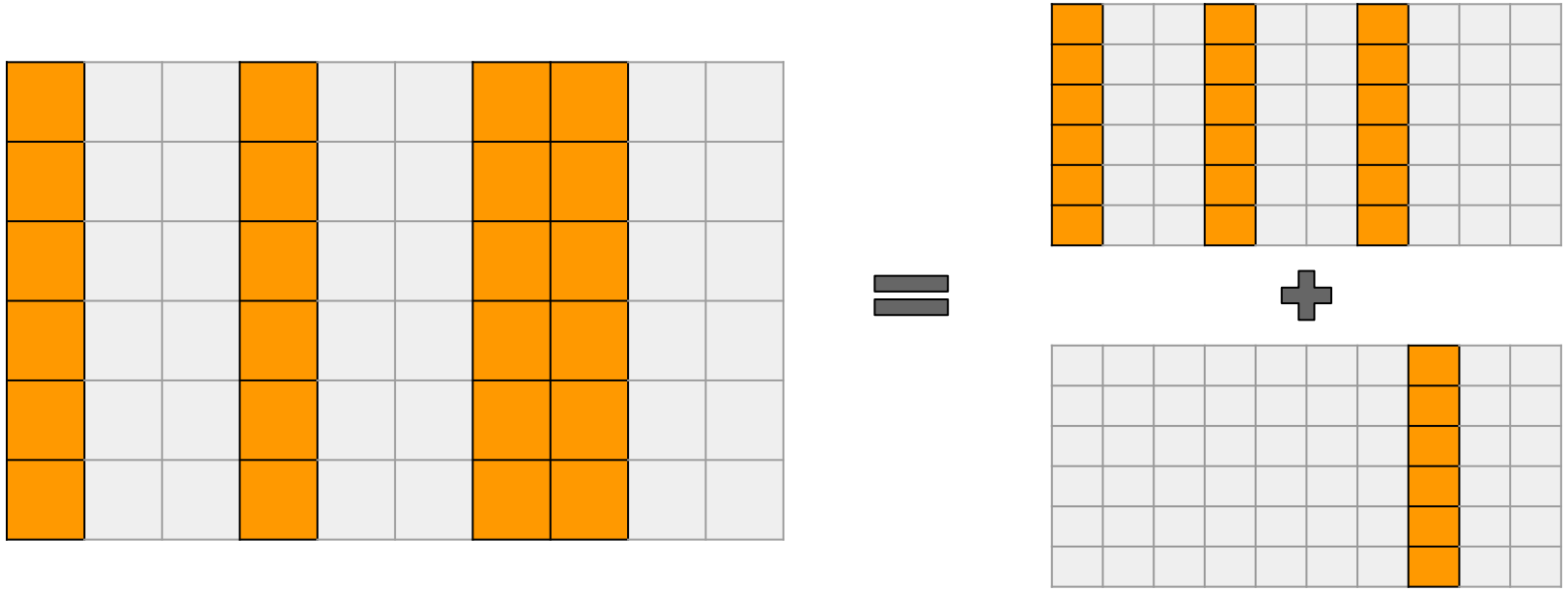
HDF5 Hyperslab Unions

- More complex selections require unions of hyperslabs



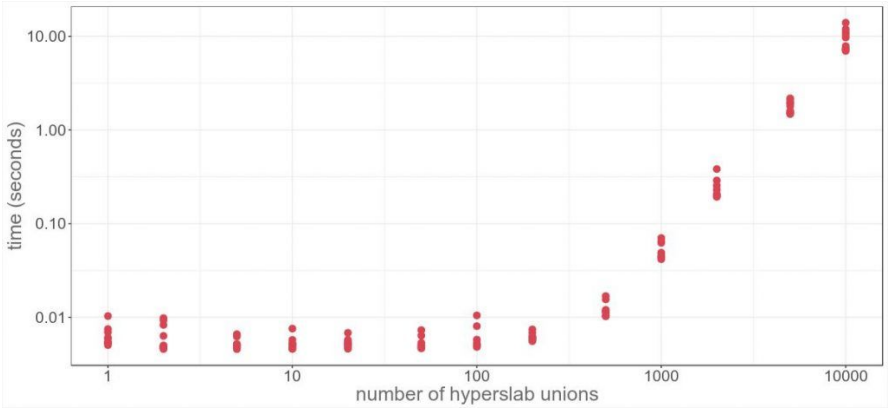
HDF5 Hyperslab Unions

- More complex selections require unions of hyperslabs



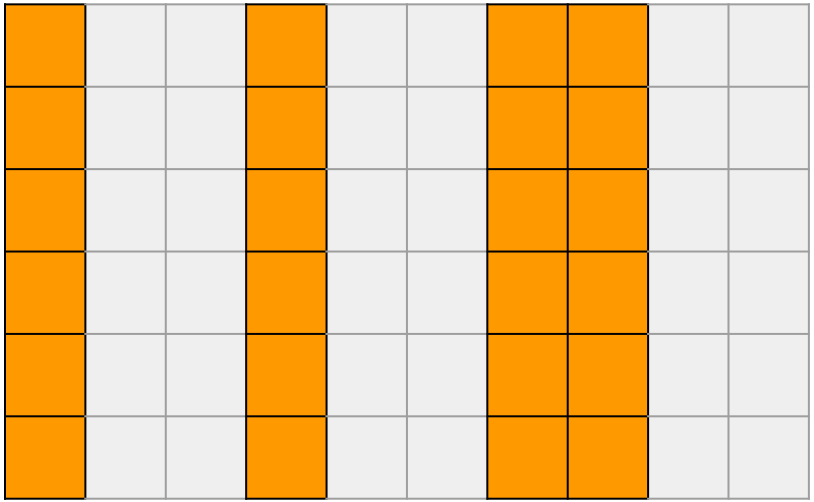
HDF5 Hyperslab Unions

- Performing many unions gets very slow
- Acknowledged by HDF5 group - but no solution suggested e.g. Union of non-consecutive hyperslabs is very slow



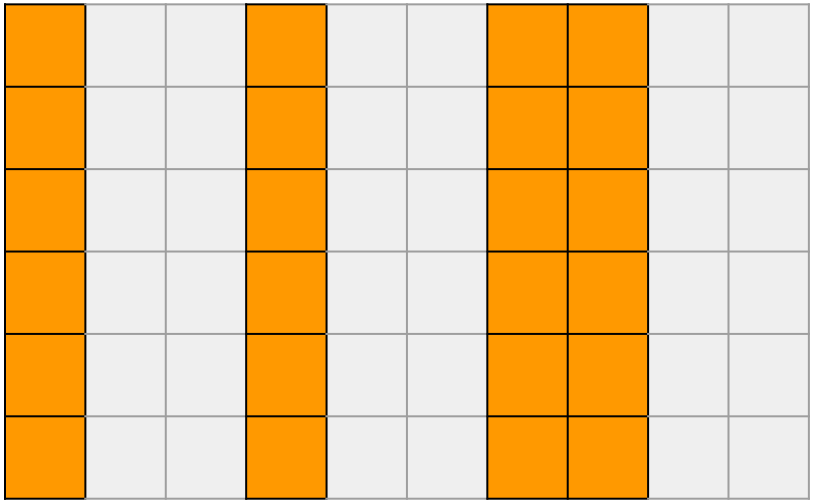
Selections with **rhdf5**

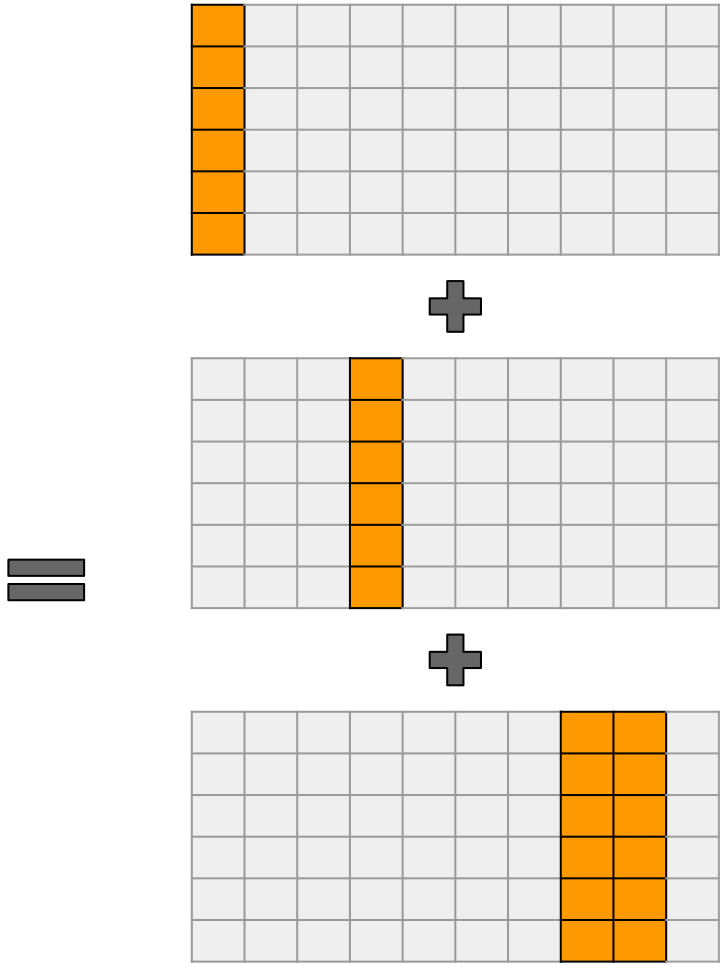
- In R it is more familiar to give an index
  e.g. `c(1, 4, 7, 8)`



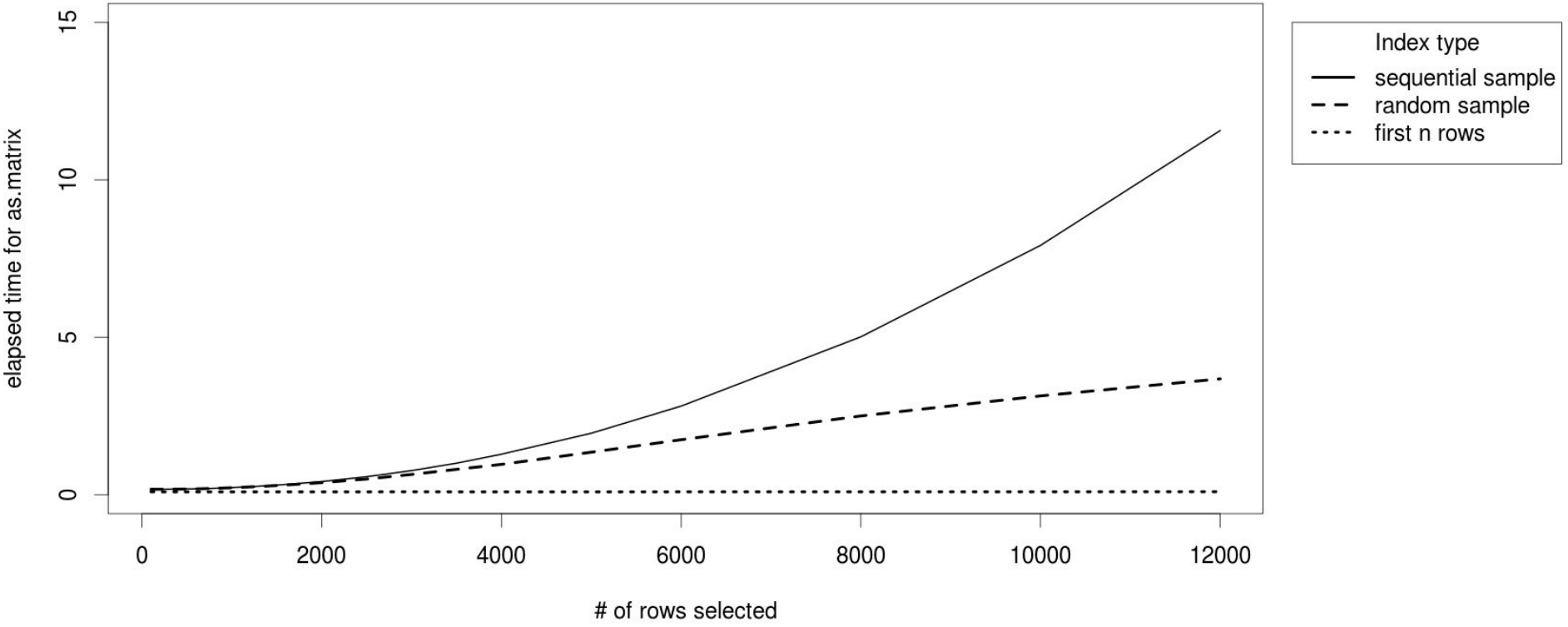
Selections with **rhdf5** (< 2.27.6)

- Existing rhdf5 implementation not efficient - many more unions than optimal



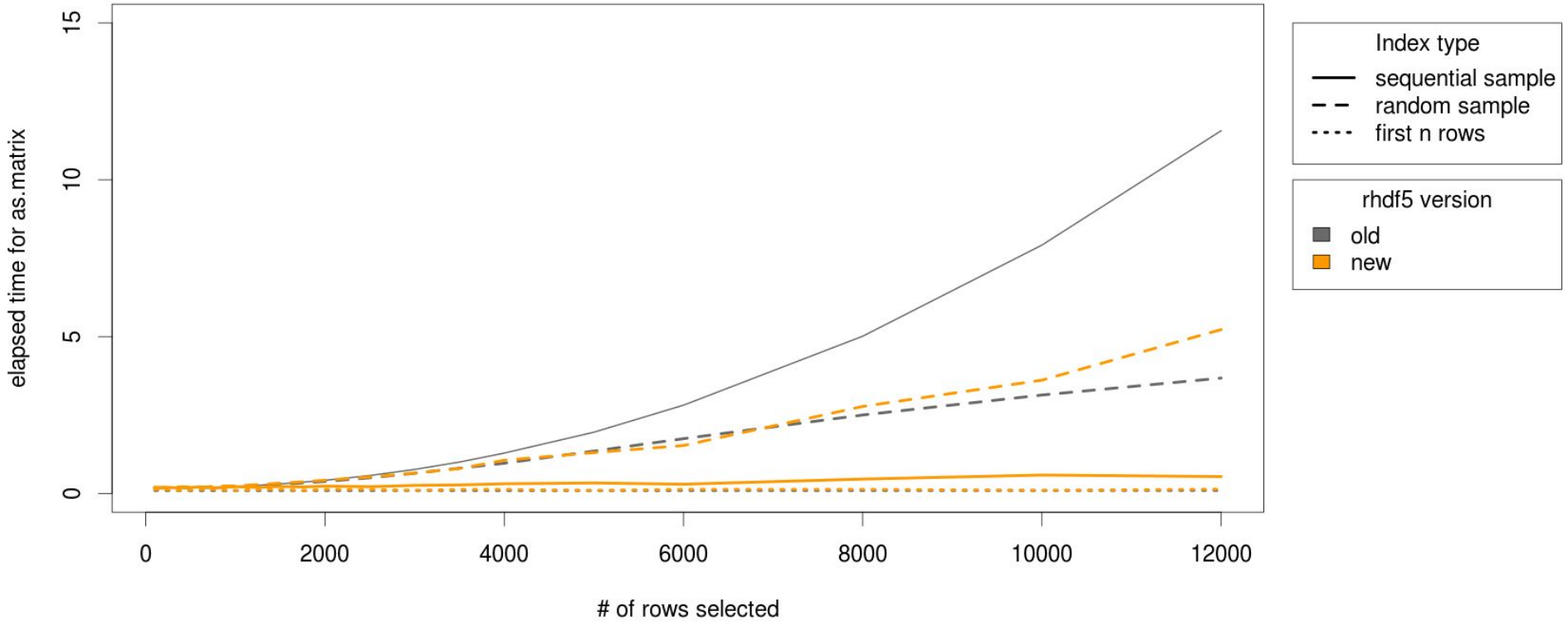


Improvements to **rhdf5** indexing



https://github.com/grimbough/rhdf5/issues/31

Improvements to **rhdf5** indexing



https://github.com/grimbough/rhdf5/issues/31

Other additions

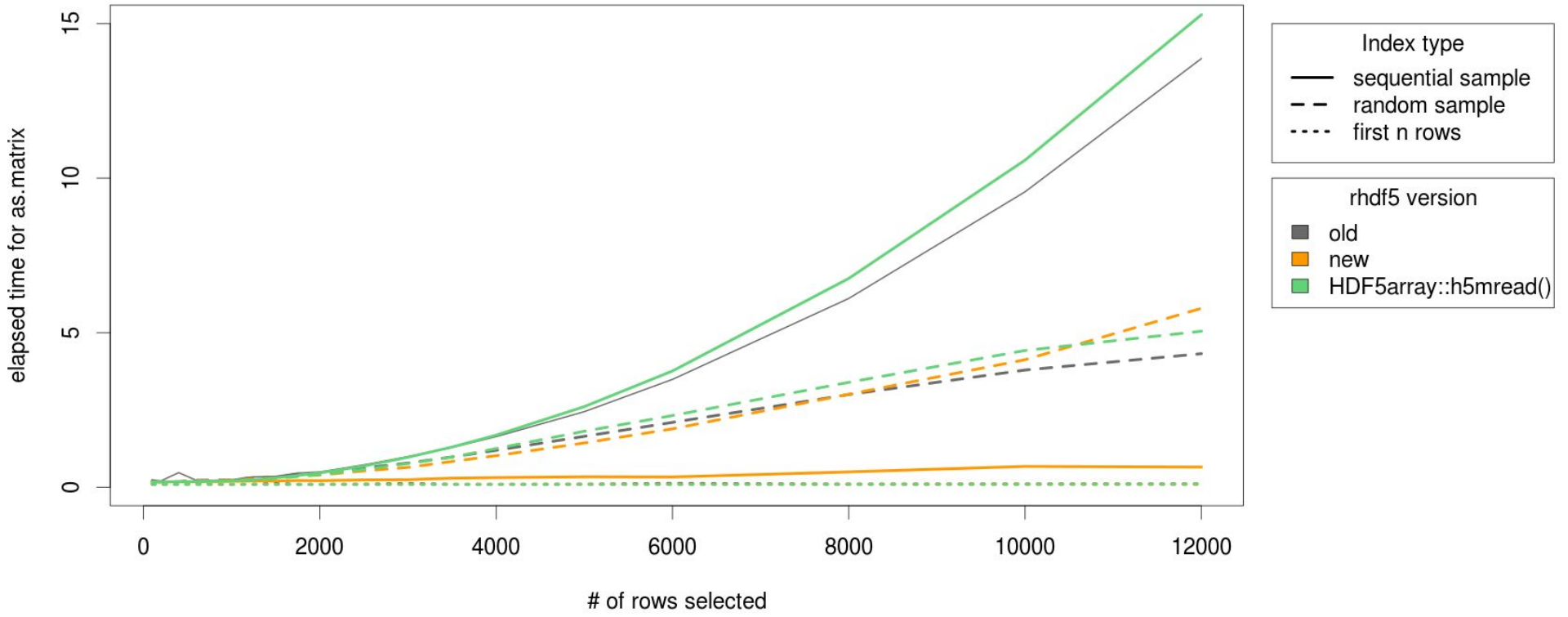
- Reading ‘long’ vectors e.g. Original 10x 1M Neuron h5 files
  - https://github.com/grimbough/rhdf5/issues/8
- Writing ‘large’ datasets
  - https://github.com/grimbough/rhdf5/issues/30 & #32
- Tests & settings for file locking issues on Lustre & Solaris
  - `h5testFileLocking()`
  - `h5enableFileLocking()`
  - `h5disableFileLocking()`

Expanding documentation

- Important to share knowledge / offer advice to users
  - Practical Tips vignette
  - Blog posts
- Other suggestions?

Acknowledgements

Wolfgang Huber
Martin Morgan
Vince Carey
Daniel van Twisk
Hervé Pagès
Levi Waldron
Aaron Lun
Mike Jiang


elapsed time for as.matrix
0
5
10
15
0
2000
4000
6000
8000
10000
12000
# of rows selected
Index type
sequential sample
random sample
first n rows
rhdf5 version
old
new
HDF5array::h5mread()