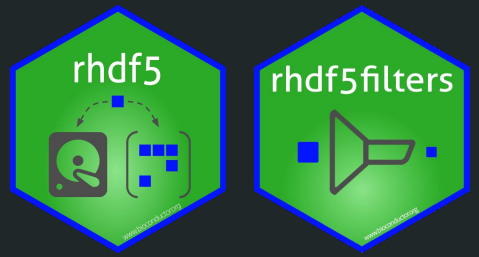
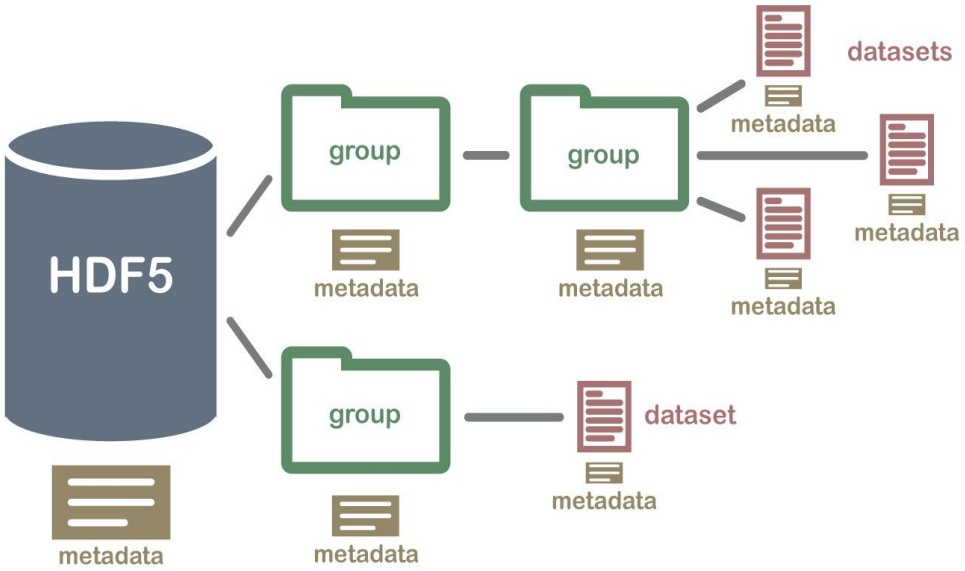
Benchmarking HDF5 Compression Filters in R

Mike L. Smith
@grimbough



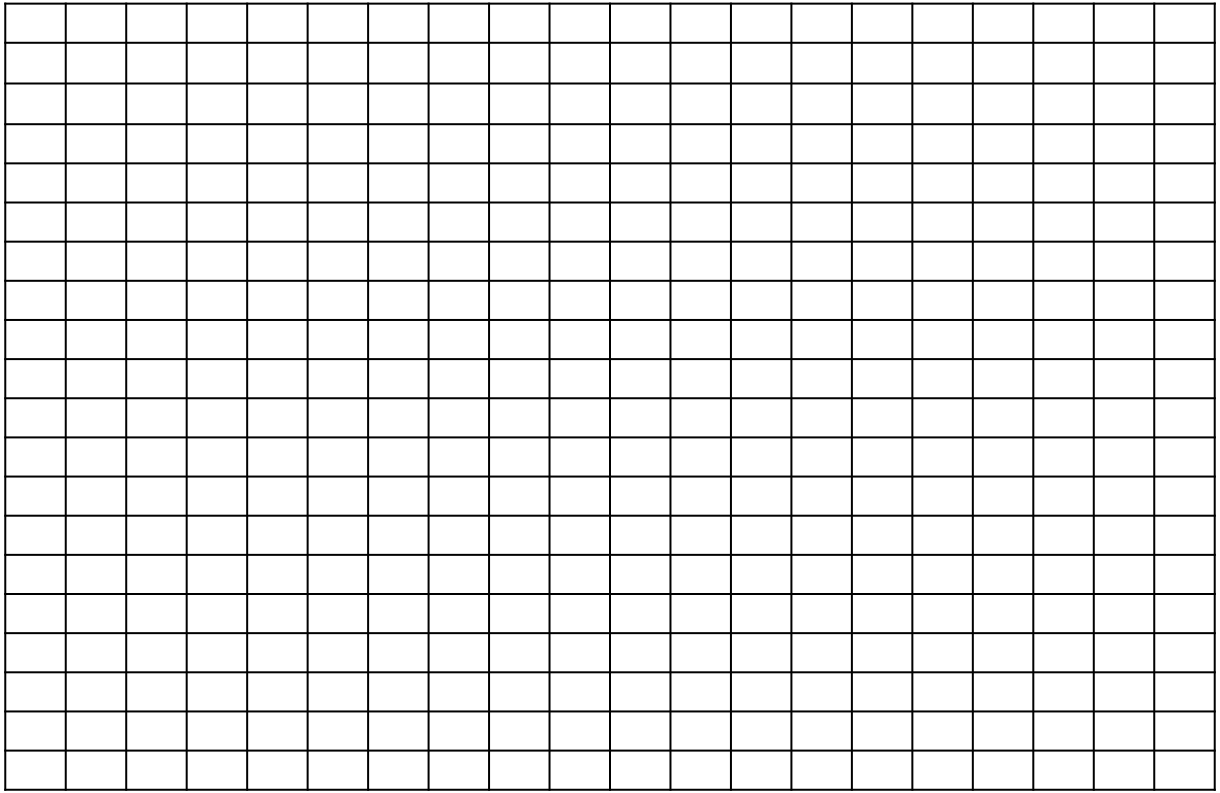
HDF5 is a file format for storing large, heterogenous, data



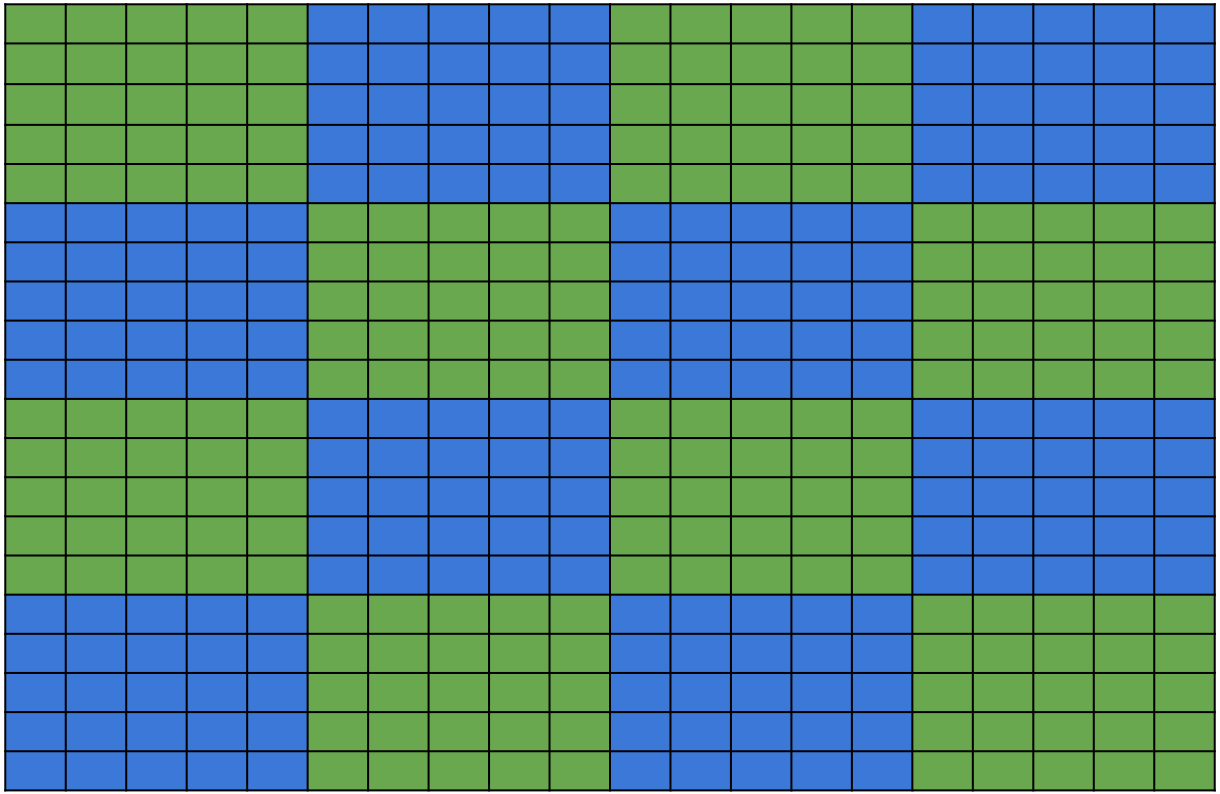
- Used in a variety of software, e.g:
  - DelayedArray
  - Kallisto
  - ONT sequencing
  - mz5 mass spec file
- Interfaces in many languages
  - C, Python, ...
  - **rhdf5** & **Rhdf5lib**
- Key features:
  - Hierarchical
  - Self describing
  - Efficient subsetting
  - Compressed

http://neondataskills.org/HDF5/About

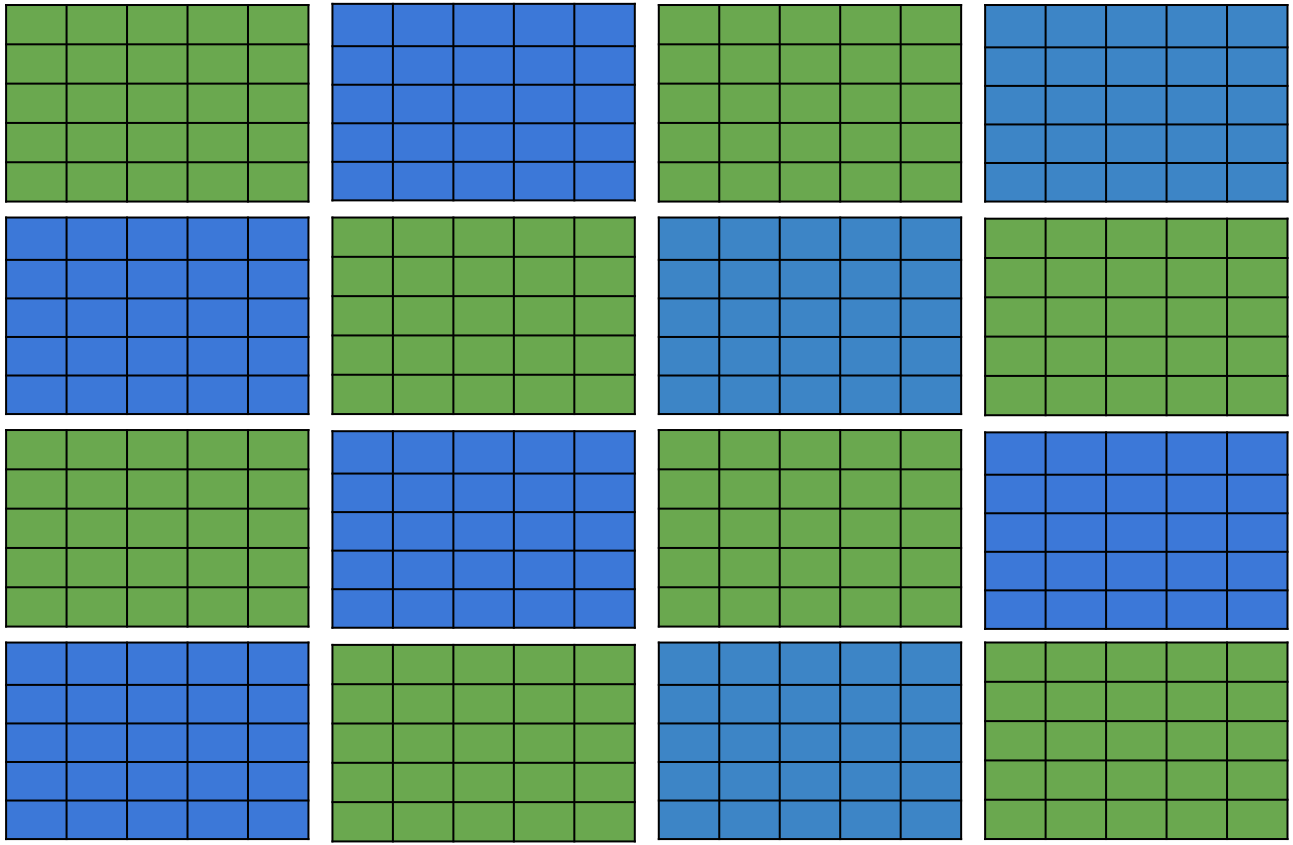
HDF5 datasets are not contiguous, but stored in chunks



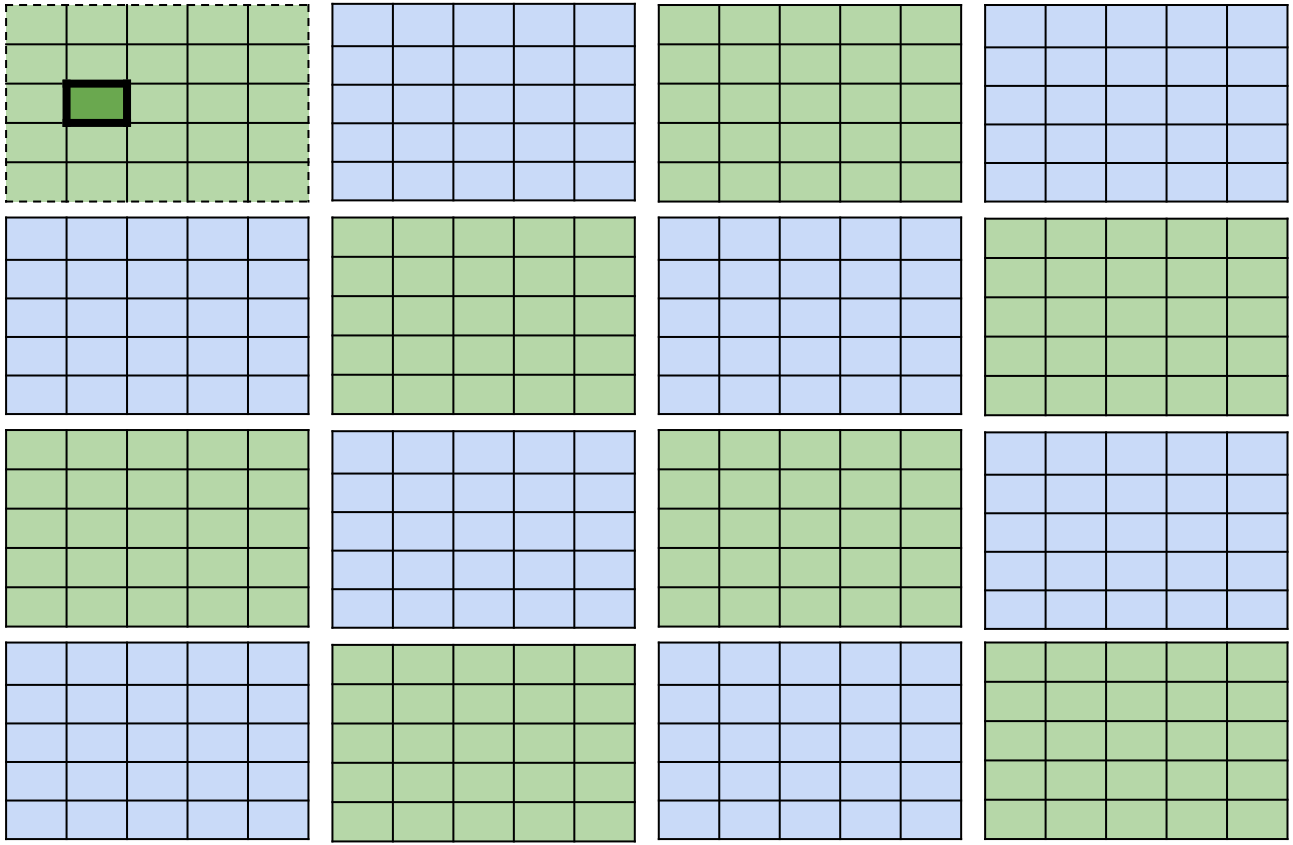
HDF5 datasets are not contiguous, but stored in chunks



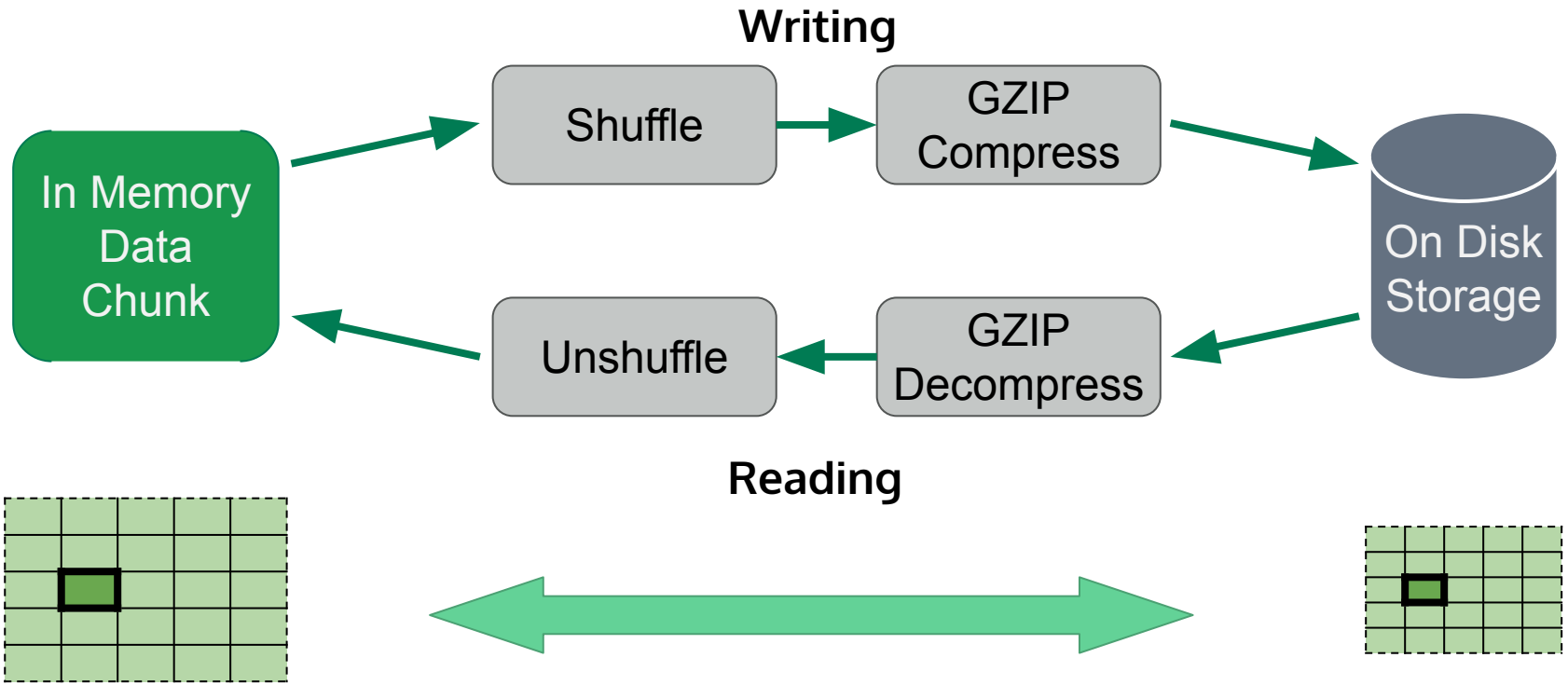
Chunks are stored separately on disk



Only read the chunks needed for a subset



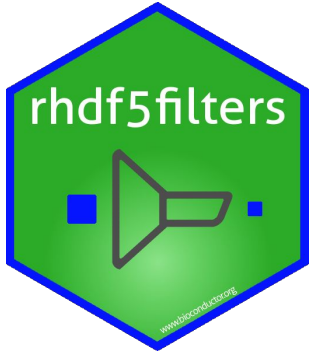
Chunks can be processed by filters - usually for compression



There are a number of compression filters available

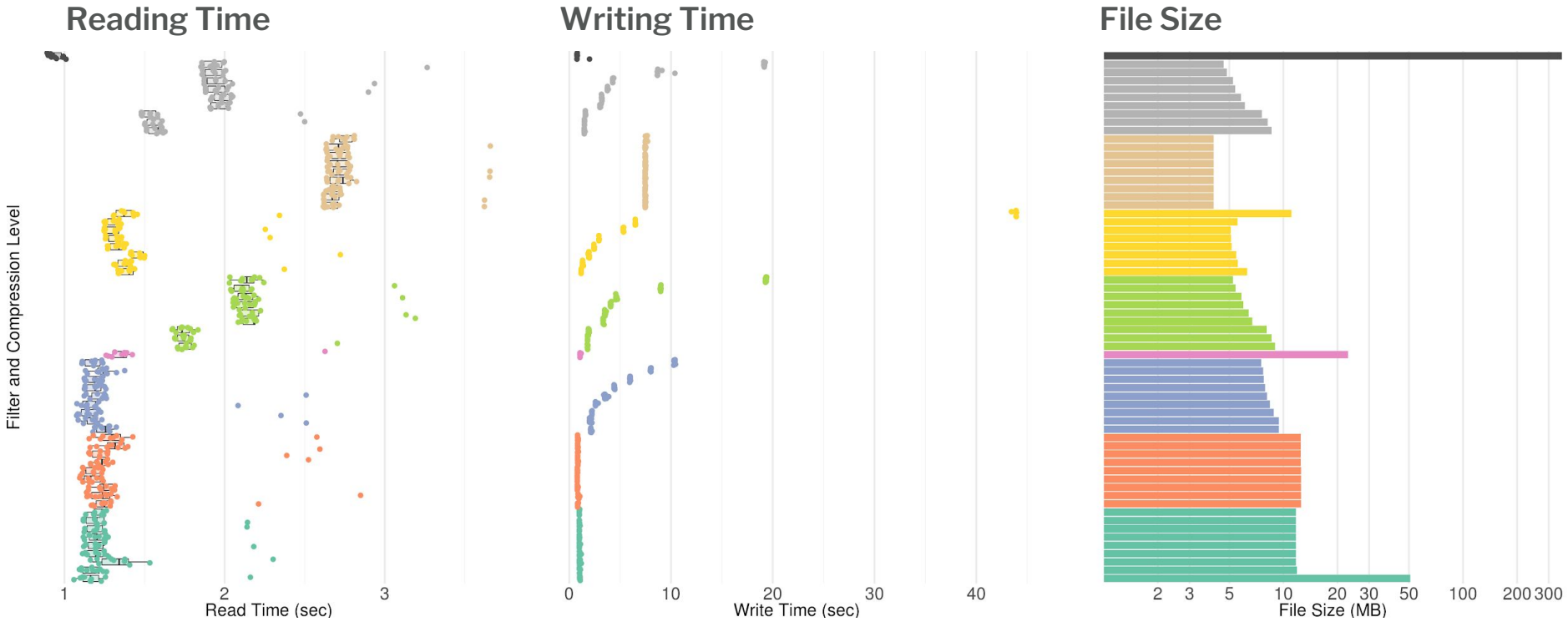
- Internal filters
  - HDF5 ships with support for GZIP and SZIP
- Dynamic filters
  - Third party tools can be made available at runtime
  - Wrap existing compression tool in small amount of C code
  - Provide location to HDF5 and they are loaded when required
  - Independent of the application(s) using them

rhdf5filters provides additional filters in R



- BLOSC meta compressor
- BZIP2
- Compiles C code on all platforms, including Windows
- Integrated with **rhdf5**
  - Writing: Supply argument to function
  - Reading: Used automatically if needed
- `msmith.de/rhdf5filters/`

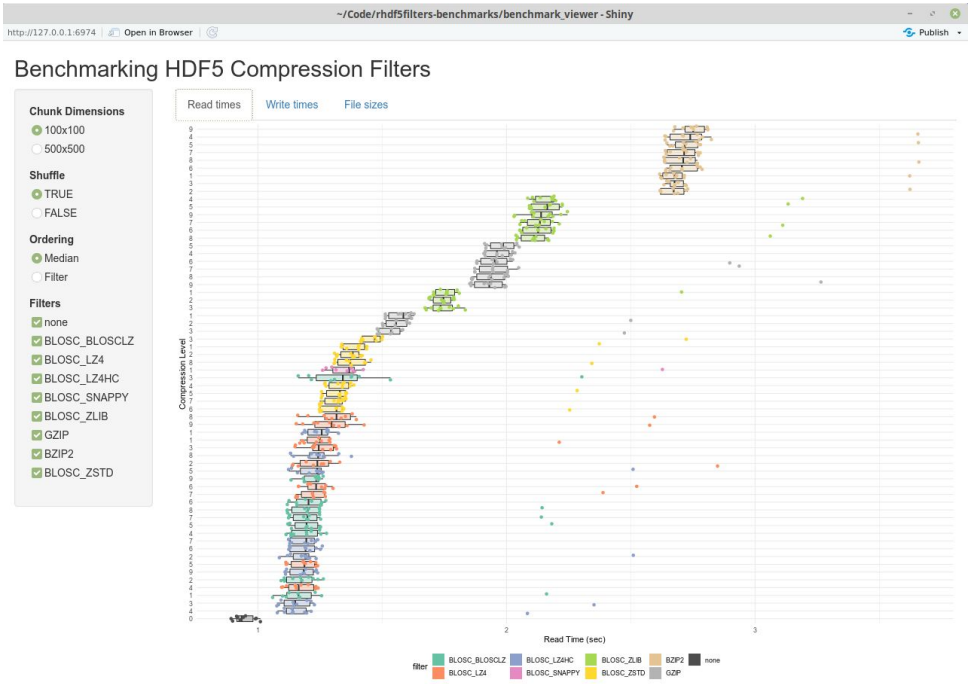
Filters & parameters have been benchmarked



You can explore the results with a shiny app

- **`msmith.de/rhdf5filters-benchmarks`**

- Scripts to run benchmarks also available
- Grateful for any contributions on both style and substance!



Thanks to EMBL Huber Lab & BioC community!

`msmith.de/rhdf5filters-benchmarks`

@grimbough