Scaling R and Bioconductor to support methods for single-cell genomic analysis

Peter Hickey

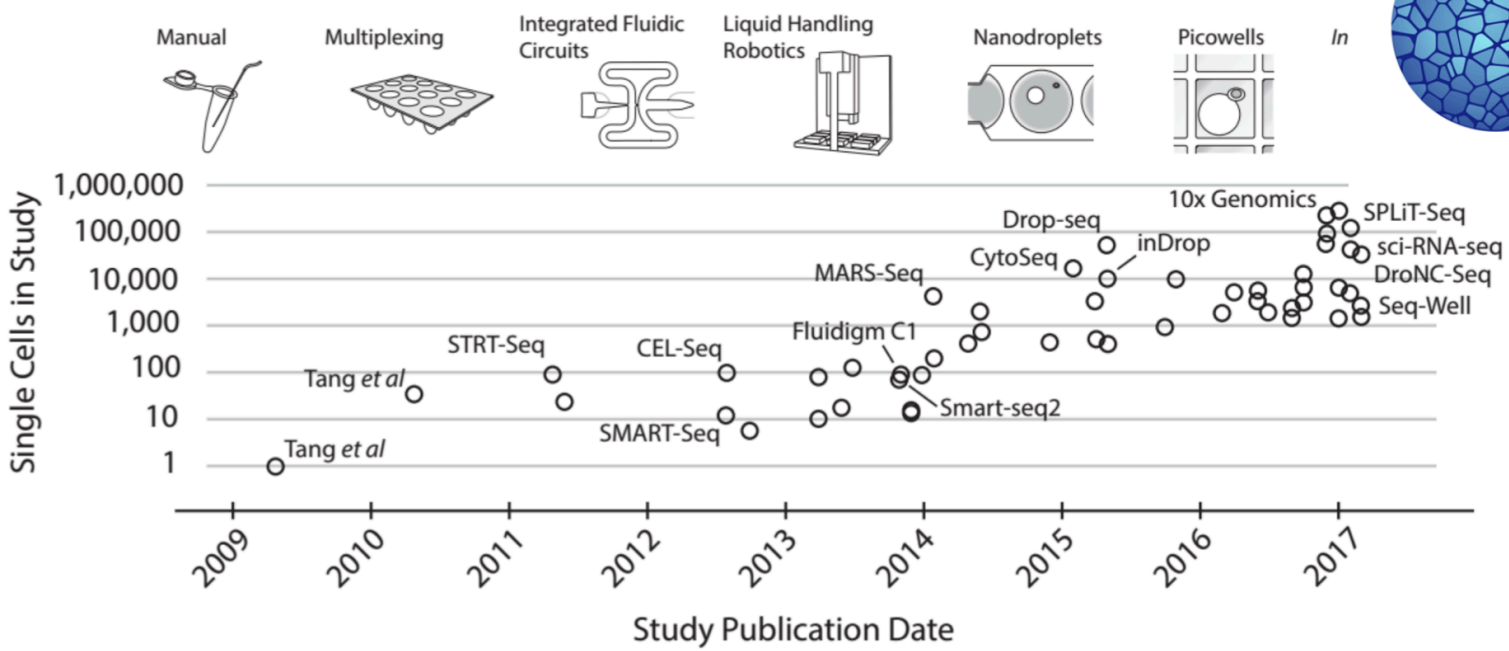
Department of Biostatistics

Johns Hopkins University

@PeteHaitch

www.peterhickey.org

Oh my, what big data you have!



HUMAN CELL ATLAS

80 GB

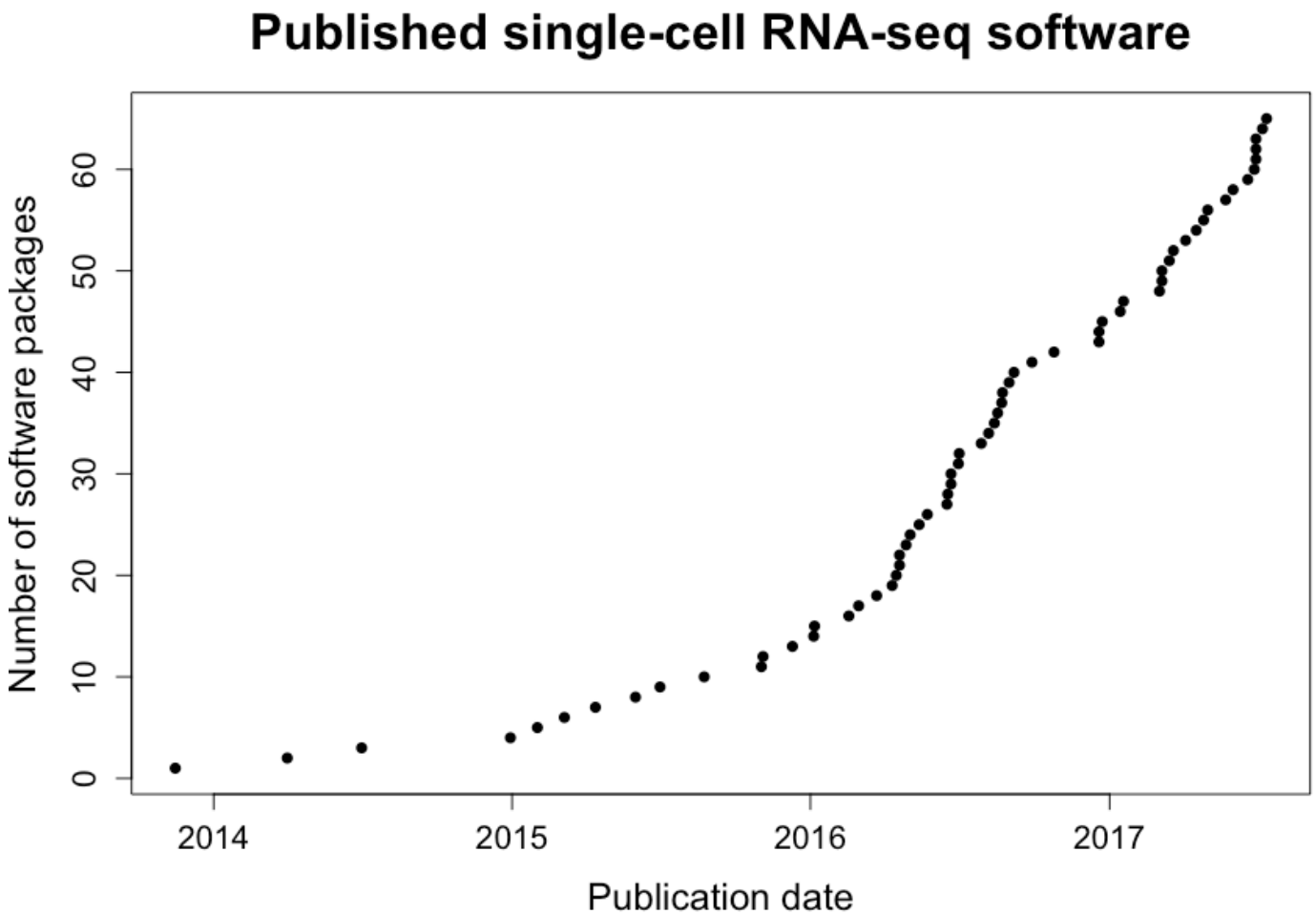
800 MB

8 MB

Not just single-cell data

Svensson V, Vento-Tormo R, Teichmann SA. Moore's Law in Single Cell Transcriptomics, *arXiv*, 2017. Available: http://arxiv.org/abs/1704.01379

More data, more software



+ 42 preprints + 15 without publication

Data from Luke Zappia (https://github.com/Oshlack/scRNA-tools)

More data, more software

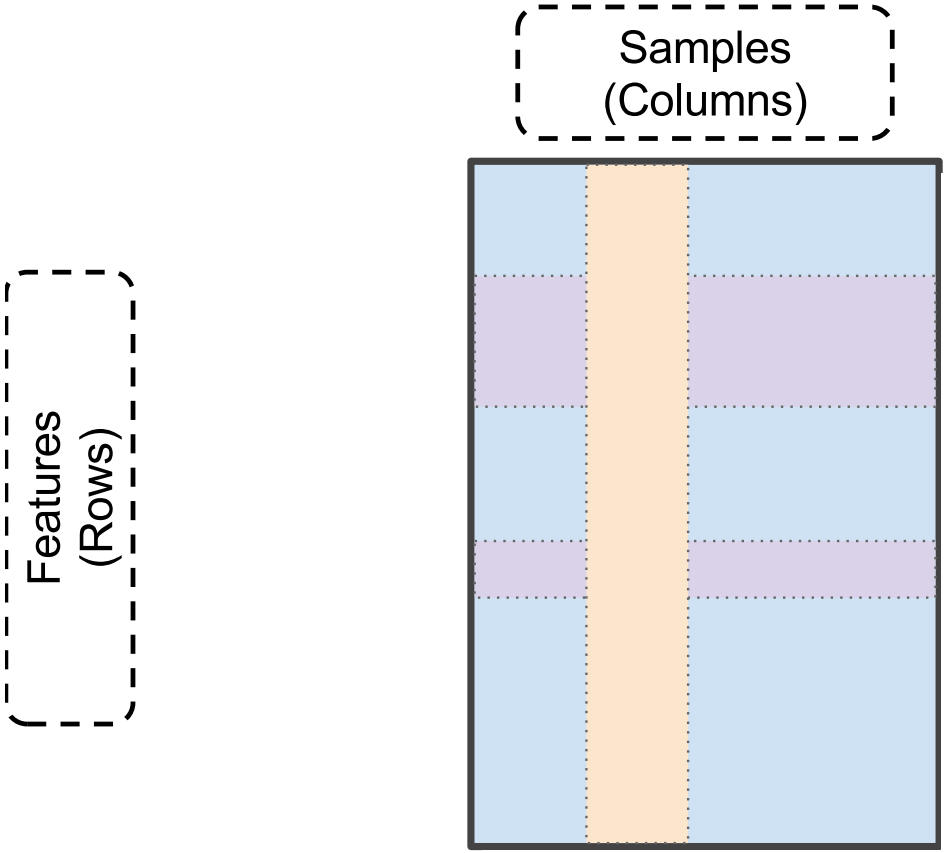
- https://github.com/seandavi/awesome-single-cell
  - > 80 software packages
- https://github.com/Oshlack/scRNA-tools
  - Spreadsheet with description of > 120 software packages
- Even within Bioconductor, lots of data structures


SUCH OPTIONS
MUCH CHOICES
DOING ME A CONFUSION
WHY YOU NO PLAY NICE TOGETHER?
imgflip.com

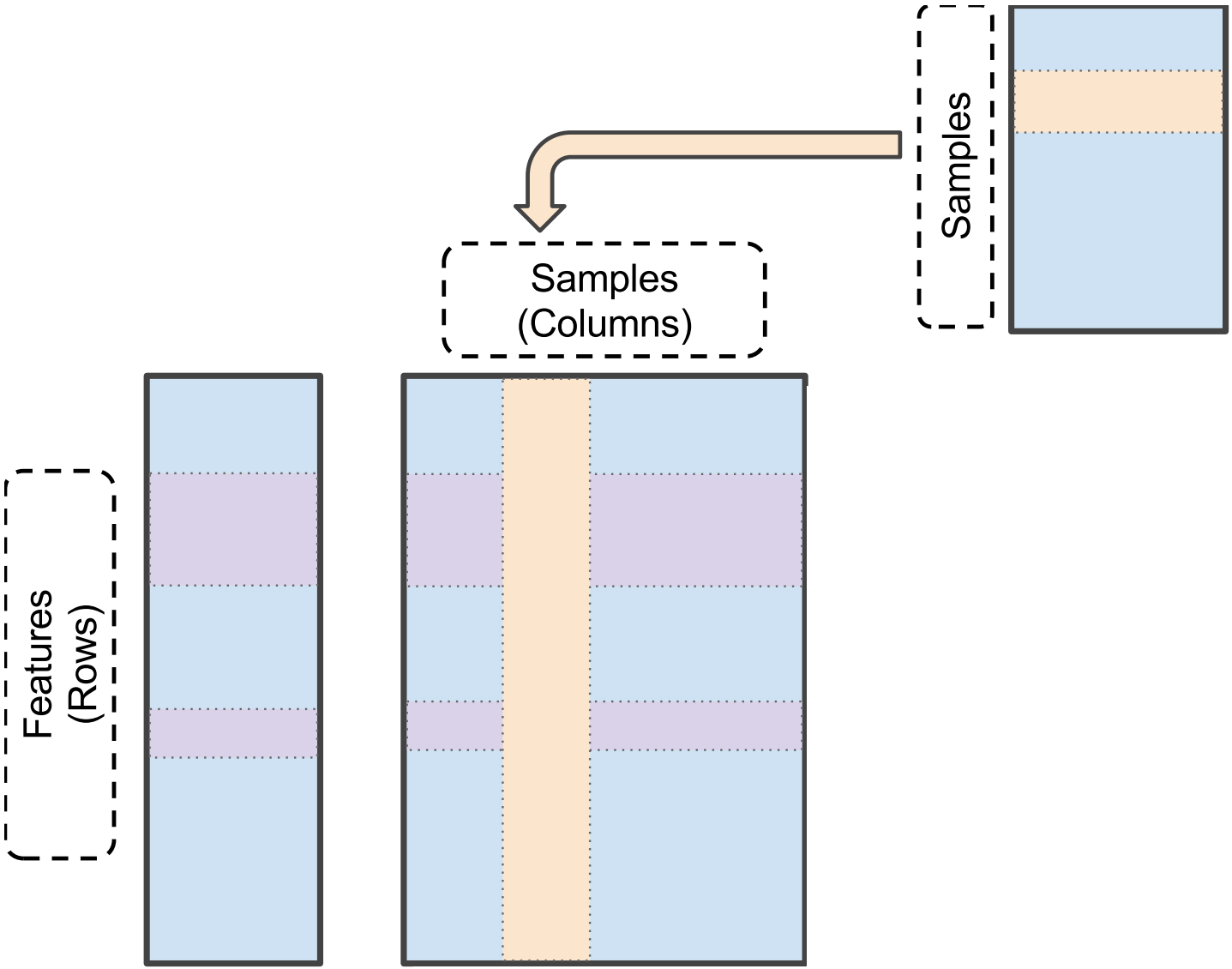
SingleCellExperiment: a Bioconductor class for single-cell data

- Extends SummarizedExperiment

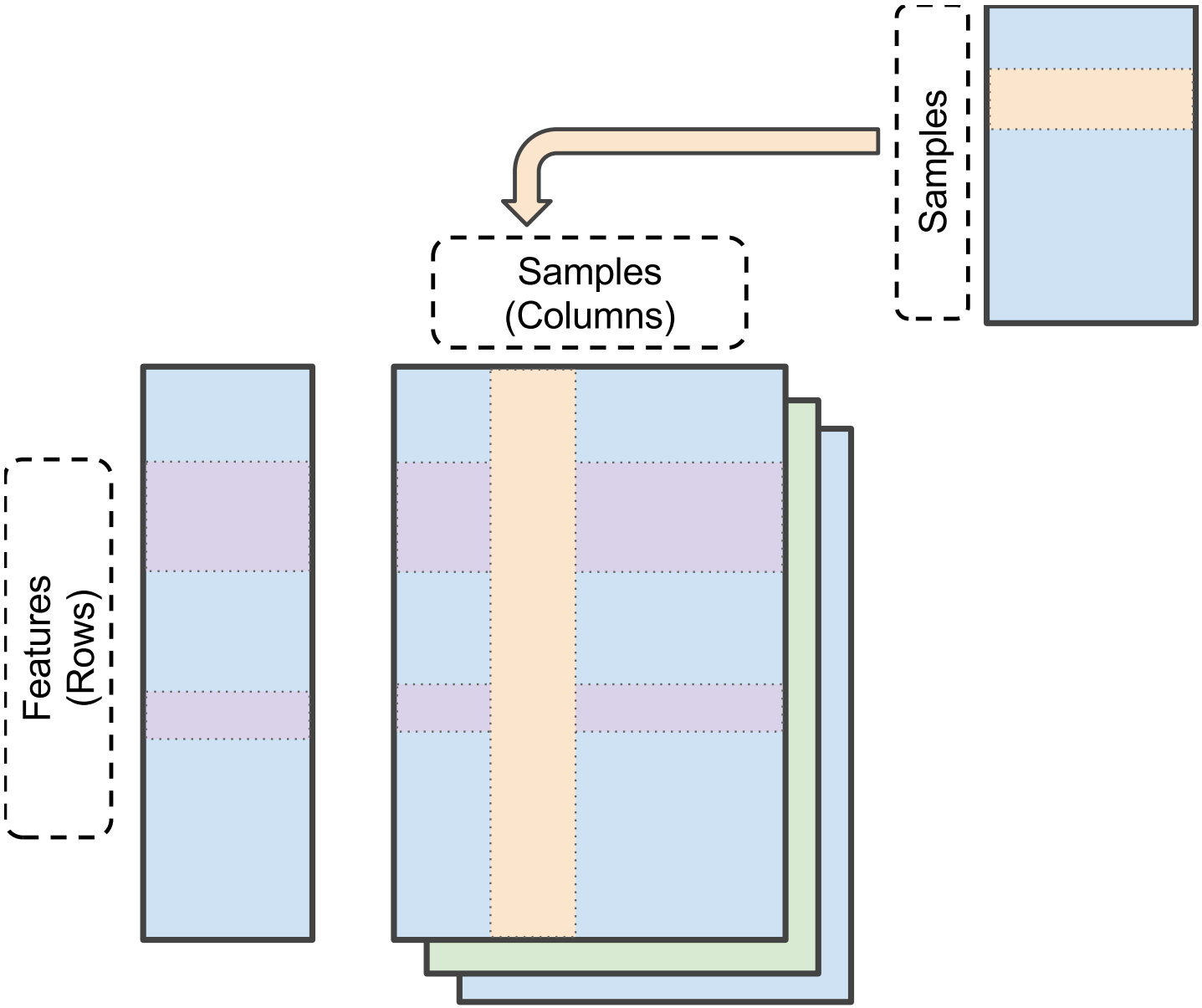
SummarizedExperiment



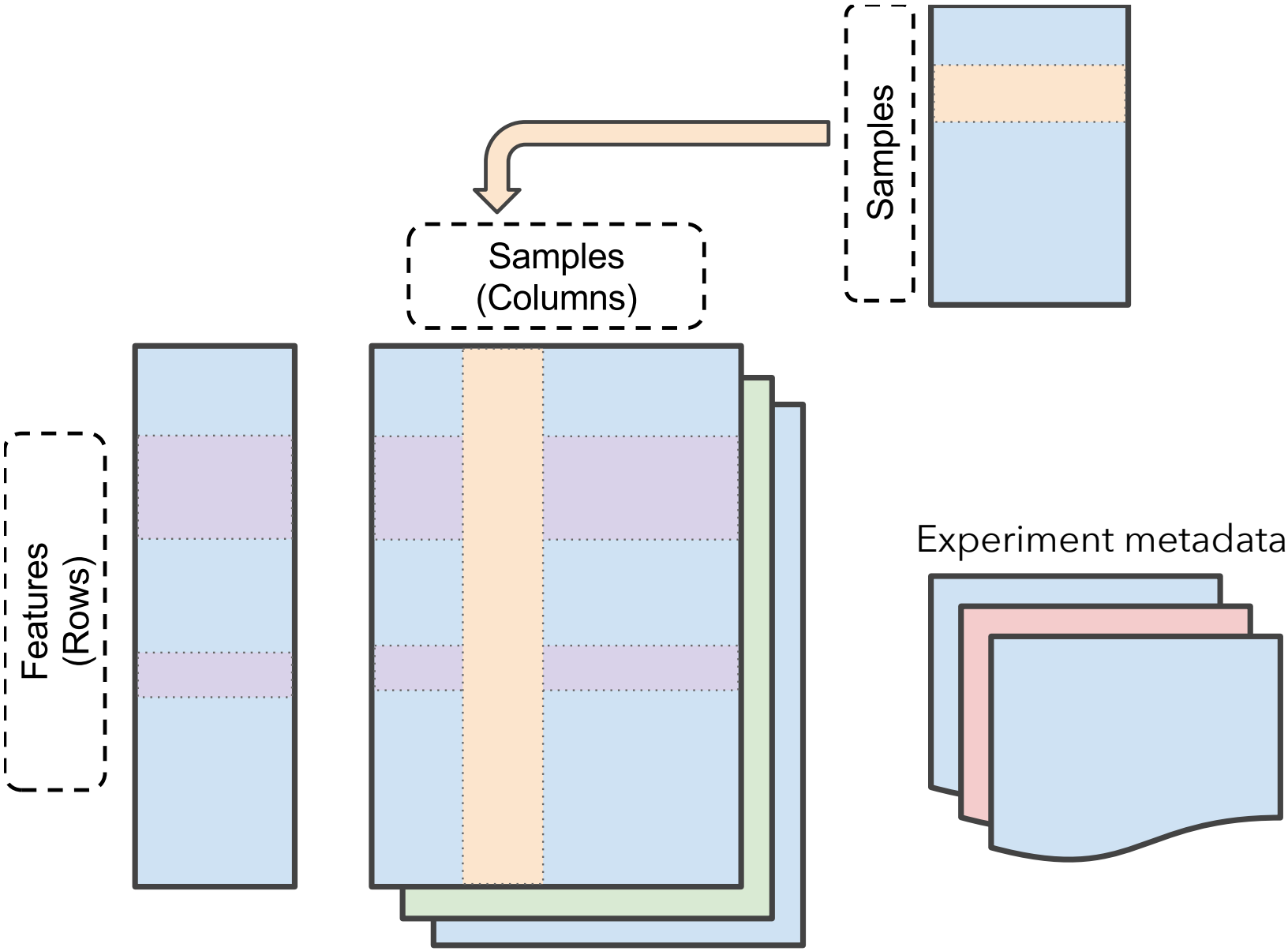
SummarizedExperiment



SummarizedExperiment



SummarizedExperiment



SingleCellExperiment: a Bioconductor class for single-cell data

- Davide Risso & Aaron Lun
- Extends SummarizedExperiment
- Adds slots for common single-cell data and operations
  - Spike-ins
  - Dimensionality reductions
- Available on Bioconductor devel branch
- Popular single-cell analysis packages are migrating to add support
  - scater
  - scran
  - MAST
  - zinbwave

That's all lovely, but I've got **BIG DATA**

- Yeah, sorta
- Most single-cell genomics data are **sparse data**
- 10X Genomics 1 million neuron scRNA-seq
- HDF5 file
- 30,000 rows (genes), 1.3 million columns (cells)
- 93% zero
- 136 GB as an ordinary *matrix*
- Sparse *Matrix*
  - Limited to $< 2^{31} - 1$ non-zero elements
  - Integer matrix stored as double

Instructions to open the file in Python are here. We do not recommend loading the file into R, due to the file size and the lack of 64 bit integers support in R.

Aaron Lun demonstrated analysis on desktop with 8 GB RAM

DelayedArray: For all your array-like needs

- Hervé Pagès
- *DelayedArray* is to <u>arrays</u> as *tibble* is to <u>tables</u>
- Familiar matrix API
  - `[`
  - `dim()`
  - `t()`
  - `log()`
  - …
- But operations are *delayed* until data are explicitly *realised*
- Data can be stored in a variety of backends
- Works as an *assay* in a SummarizedExperiment (and derived classes)

Backends

- In-memory
  - *matrix* (**base**)
  - *Matrix* (**Matrix**)
  - *RleArray* (**DelayedArray**)
    - Rle = run length encoding
- On-disk
  - *HDF5* (**HDF5Array**)
    - Data are in a HDF5 file, keep it in an HDF5 file
  - *matter* (**matterArray**)
    - Kylie A. Bemis (Northeastern University)

Backends

Class/backend	Package	Size in memory	Size on disk
DelayedArray with matrix	base	800 MB	0 MB
DelayedArray with dgCMatrix	Matrix	951 MB	0 MB
RleMatrix (solid)	DelayedArray	1001 MB	0 MB
RleMatrix (chunked)	DelayedArray	634 MB	0 MB
HDF5Array (default compression)	HDF5Array	< 10 kB	165 MB
matter	matterArray	< 10 kB	800 MB

- Fairly straightforward to add new backends

DelayedMatrixStats

- Me
- Inspired by matrixStats (Henrik Bengsston, CRAN)
- Functions for columns and rows operations on DelayedMatrix (2D DelayedArray) objects
  - `colSums2(), rowSums2()`
  - `colMeans2(), rowMeans2()`
  - `colSds(), rowSds()`
  - `colLogSumExps(), rowLogSumsExps()`
  - … (33 more methods)
- **Idea:** Support matrixStats API for DelayedMatrix and derived classes
- **Aim 1:** General methods to work on arbitrary DelayedMatrix
- **Aim 2:** Optimised methods for specific backends

beachmat

- Aaron Lun, Mike Smith, Hervé Pagès
- Unified C++ API for (most) DelayedMatrix backends
  - `get_col(), get_row()`
  - `set_col(), set_row()`
  - Currently: matrix, Matrix, RleMatrix, HDF5Matrix

restfulSE

- Vincent Carey
- Proof-of-concept
- HDF5 server backed SummarizedExperiment
  - Data live on remote server, stored in HDF5 file
  - RESTful API
  - Data returned as binary (better) or JSON
  - No server-side computation (yet)

Key points

- Starting point for a lot of genomics data analysis is a array of numbers
- Bioconductor strength is semantically rich data structures for array-like data
  - SummarizedExperiment -> SingleCellExperiment
- Assay data doesn't have to be an ordinary *array*
- Supporting general array-like data with DelayedArray and different backends
- DelayedMatrixStats, beachmat, HDF5 Server
- R/BioC's strength is supporting interactive exploratory data analysis, rich data structures, interoperability

Links and contact

- **Packages**:
  - https://bioconductor.org/packages/SingleCellExperiment/
  - https://bioconductor.org/packages/DelayedArray/
  - https://bioconductor.org/packages/HDF5Array/
  - https://bioconductor.org/packages/beachmat/
  - https://bioconductor.org/packages/matter/
  - https://github.com/PeteHaitch/matterArray
  - https://github.com/PeteHaitch/DelayedMatrixStats
  - https://github.com/vjcitn/restfulSE
- **Slides**: http://peterhickey.org/presentations/
- **GitHub & Twitter**: @PeteHaitch