rapids-singlecell

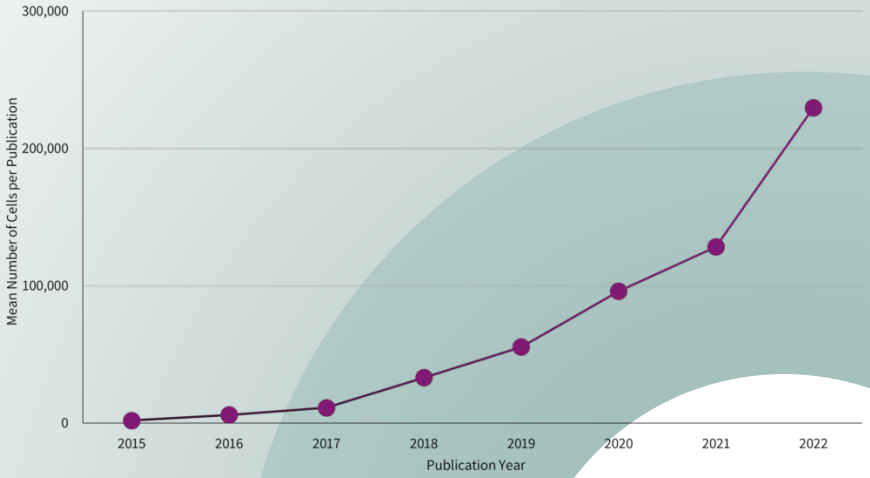
Getting your analysis done faster

severin.dicks@helmholtz-munich.de
Severin Dicks
@Severin1991
github.com/Intron7

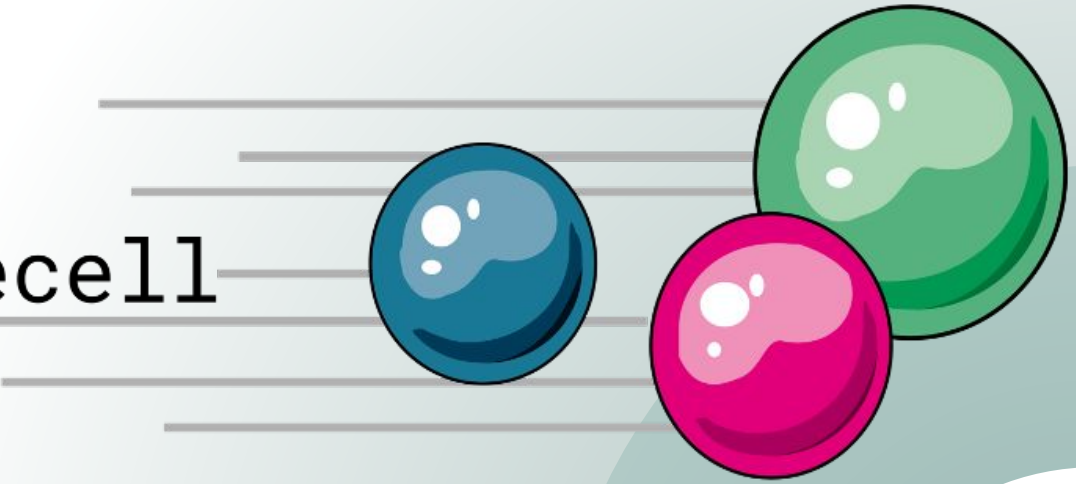
NUMFOCUS
OPEN CODE = BETTER SCIENCE

What’s the need for another Package?

- Python is slow
- Python doesn’t scale well
- People want to not do coffee breaks all day
- Maybe you want to work with collaborators in real time
- Singlecell analysis need a lot of iterations



rapids-singlecell



What is rapids-singlecell?

- GPU accelerated singlecell analysis
- Near drop-in replacement for scanpy
- more than 30x Speedup over the CPU
- fully compatible with the scverse ecosystem

Tech Stack RSC

CuPy

RAPIDS

How do I get RSC?

- PyPi
- Conda/YAML
- Docker

- Pitfalls:
  - Wrong CUDA Stack
  - Issues with CuPy
- Solution:
  - Talk to your IT
  - Know you CUDA Version
  - Use CuPy Cuda specific wheels

What’s Supported?

scanpy.pp

- filter cells / filter genes
- calculate_qc_metrics
- normalize_total
- log1p
- highly_variable_genes
- regress_out
- scale
- PCA
- Neighbors
- Harmony

seurat
cellranger
seurat_v3
pearson_residuals
poisson_gene_selection (from scVI)

scanpy.tl

- Leiden
- Louvain
- TSNE
- UMAP
- Draw Graph
- Kernel Density
- Diffusion Maps
- Score Cells
- Rank Gene Groups with logreg

decoupler

- different statistical methods to extract biological activities

- run_mlm
- run_wsum
- run_ulm
- run_aucell

squidpy

- Spatial Autocorrelation
  - Moran’s I
  - Geary’s C
- Ligrec (receptor-ligand analysis)

Get

Utilities for AnnData Handling

- aggregate
  - scanpy.get.aggregate
  - 'count_nonzero', 'mean', 'sum', 'var'
  - output dense and sparse
- anndata_to_GPU & anndata_to_CPU
- X_to_GPU & X_to_CPU

Who uses/used rapids-singlecell?

Memory Management

Performance Tuning with RMM

- Memory management has an impact on performance
- Managed Memory:
  - Oversubscription
  - more Data
  - less performant
- Pool Allocation:
  - Maximal Performance
  - more memory usage.

Limitations

- Not every function is ported to GPU yet
- Int32 indptr 2^31-1 nnz
- GPUs might be slower for small Datasets
- Memory Allocators
- Rapids Managed unified Memory doesn’t allow for limitless oversubscription of VRAM

Outlook

Multi GPU with Dask

- Normalize
- Log1p
- Basic qc
- Scale
- PCA (dense from CUML +Sparse)
- HVG (Cellranger, Seurat)

Documentation

Github

Contributions are Welcome

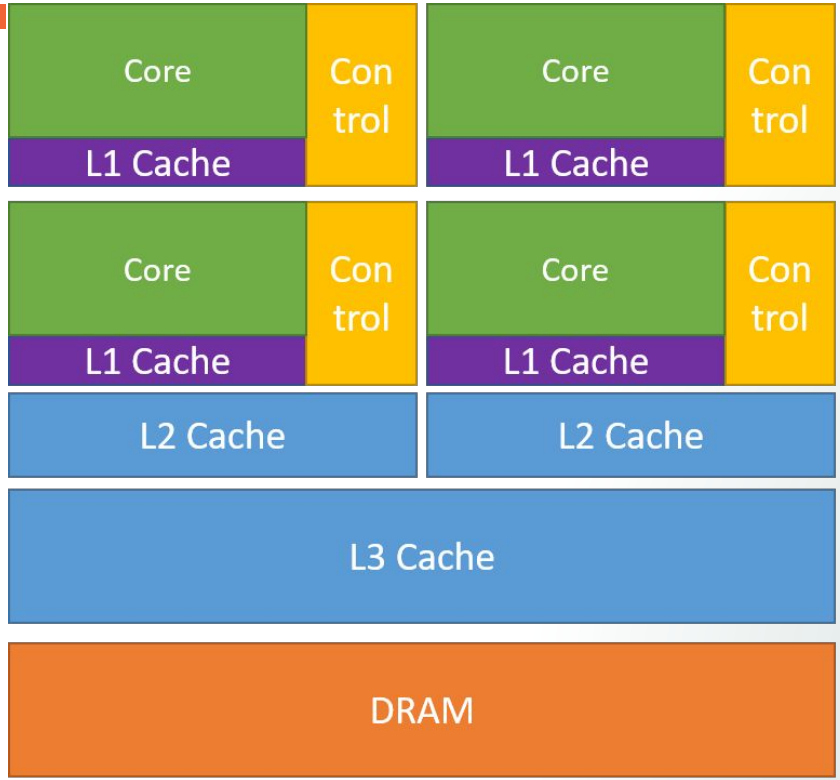
- Tell me what’s missing
- General improvements
- Start contributing yourself

Acknowledgments

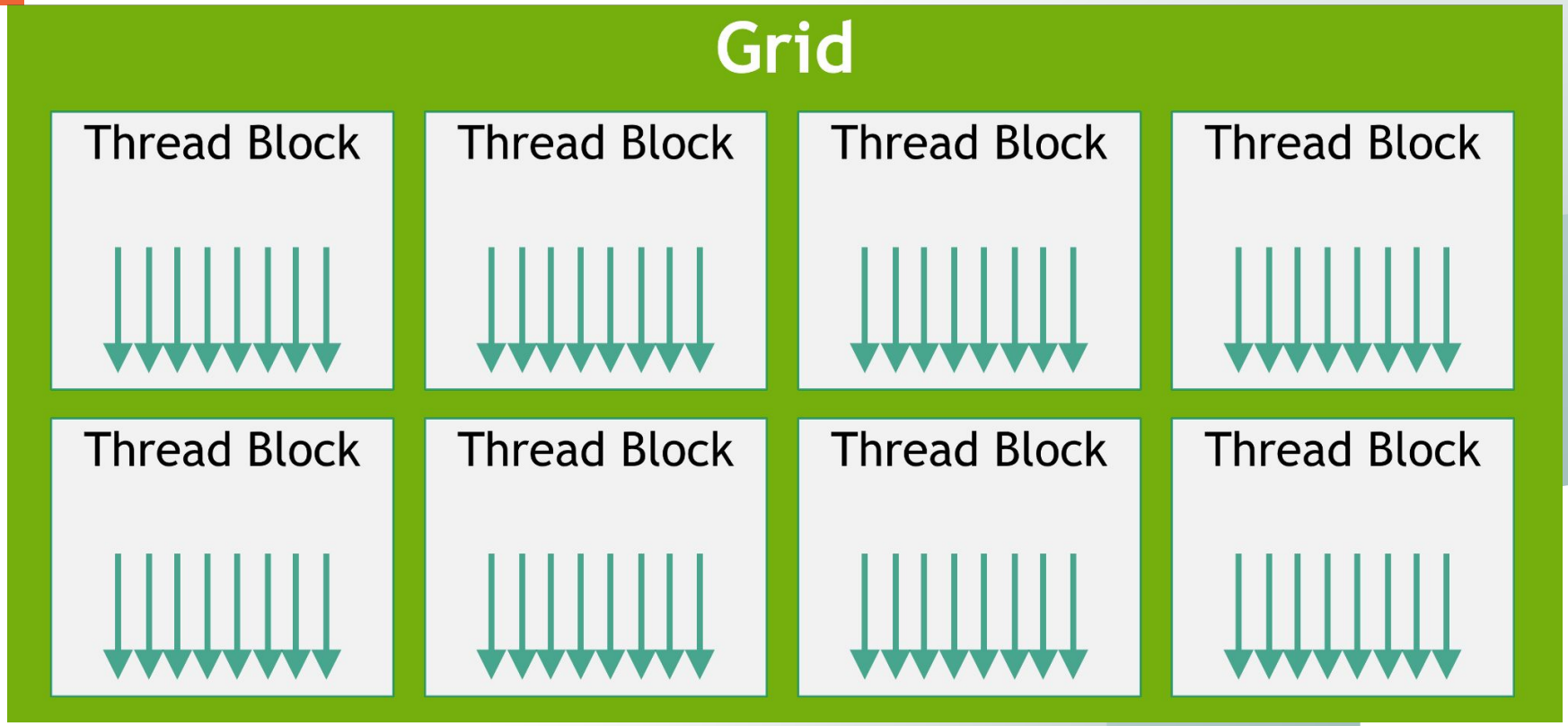
- scverse
  - Isaac Virshup
  - Lukas Heumos
  - Philipp Angerer
  - Ilan Gold
- Nvidia
  - Taurean Dyer
  - Corey Nolet
- CZI
- Fabian Theis
- The Users

Accelerate your Package

GPU Computing



GPU Computing



GPU Computing

Array API

CuPy

PyTorch

GPU Computing

Rapids

Get Nerdy

Kernels

- Numba
- CuPy

```
_mul_kernel_csr = r"""
(const int *indptr, {0} *data,
                int nrows, int tsum) {
    int row = blockDim.x * blockIdx.x + threadIdx.x;

    if(row >= nrows)
        return;

    {0} scale = 0.0;
    int start_idx = indptr[row];
    int stop_idx = indptr[row+1];

    for(int i = start_idx; i < stop_idx; i++)
        scale += data[i];

    if(scale > 0.0) {
        scale = tsum / scale;
        for(int i = start_idx; i < stop_idx; i++)
            data[i] *= scale;
    }
  }
"""
```

Quick Introduction to CSR Matrix