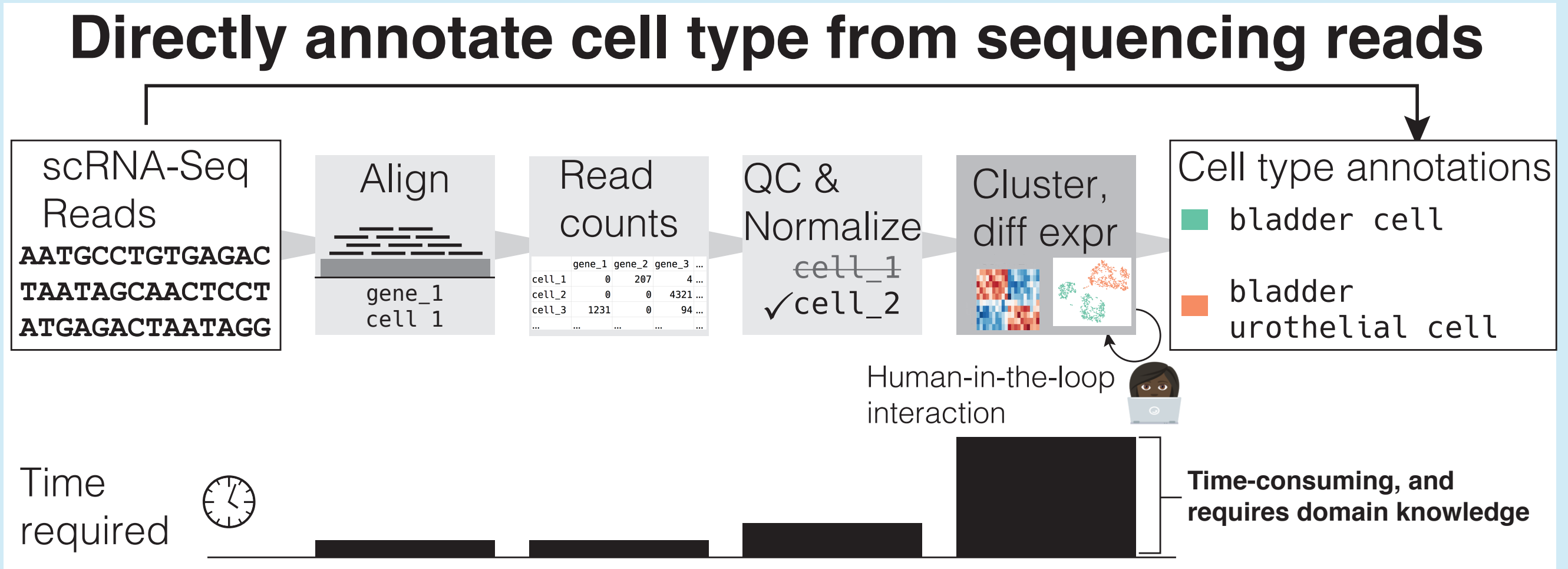
Fast approximate cell type identification via MinHash sketches of k-mers in single cell RNA-seq

CHAN ZUCKERBERG BIOHUB
Olga B. Botvinnik
Data Sciences, Chan Zuckerberg Biohub, 499 Illinois St, San Francisco, CA 94555

olgabot
olga.botvinnik@czbiohub.org
http://github.com/czbiohub/kmer-hashing

The Problem: Cell type annotation is laborious, time-intensive, and requires extensive domain knowledge.

The Solution: Compress scRNA-seq reads into *k*-mer signatures for fast lookup against an annotated cell type database.



Directly annotate cell type from sequencing reads

Abstract

Single-cell RNA-seq (scRNA-Seq) has emerged as a dominant technology for identifying novel and known cell types in organisms. However, current methods for assigning cell types involve multiple data transformations, each requiring manual parameter tuning. Often, an approximate cell designation is sufficient to identify outlier cells which do not closely match previously identified cells. We apply MinHash sketches to sparsely represent the k-mer content of each cell as a "signature." A key advantage of MinHashing is the addition of new signatures merely appends to the database and does not require re-clustering. Using the Tabula Muris data, we show that signatures approximate gene expression space and recover assigned cell ontology classes. We compare cell signatures found by both 3'-end and full transcript single-cell library preparation techniques, and perform cross-species comparison via translated protein k-mers. Thus, MinHash sketches are a fast method to identify potential cell types, build a compendium of known cell types, and identify outlier cells.

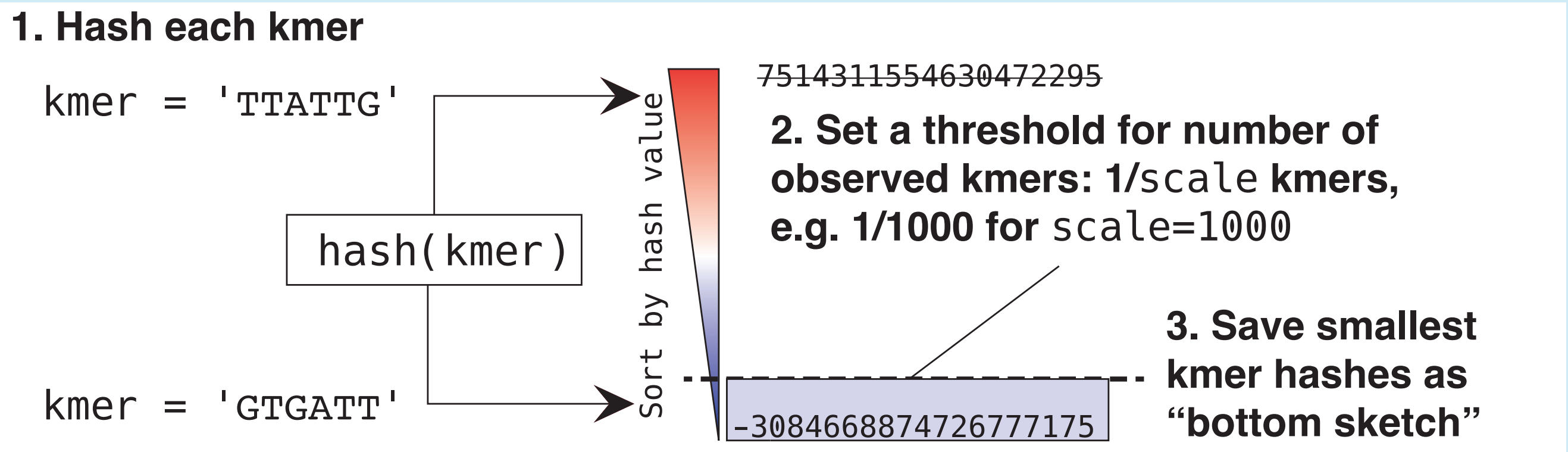
Introduction

Currently, assigning cell type identities to a single-cell RNA-seq dataset can take several days of processing and iteration, even for a technical domain expert. What if there was a faster way to identify cell types? There are several methods for "aligning" gene counts matrices across datasets, but this still requires the full processing pipeline. Additionally, if the datasets are in different but related species, aligning gene expression across is difficult outside of 1-1 orthologue mapping. Enter MinHash Sketches (Broder 1997), which approximate the sequencing data as a compressed set of kmers. MinHashing has been applied in the past to metagenomics to identify microbial species (Ondov et al. 2016; Titus Brown and Irber 2016; Koslicki and Falush 2016).

Methods

We cut up each cell's sequencing reads into a k-mers, then save only a randomly chosen subset of the k-mers using a MinHash Sketch. This is similar to transcript equivalency classes (Bray et al. 2016; Patro et al. 2017), but the goal is not to cluster cells and potentially discover novel cell types, but rather identify known cell types.

How to compute a MinHash Sketch



1. Hash each kmer

`kmer = 'TTATTG'`

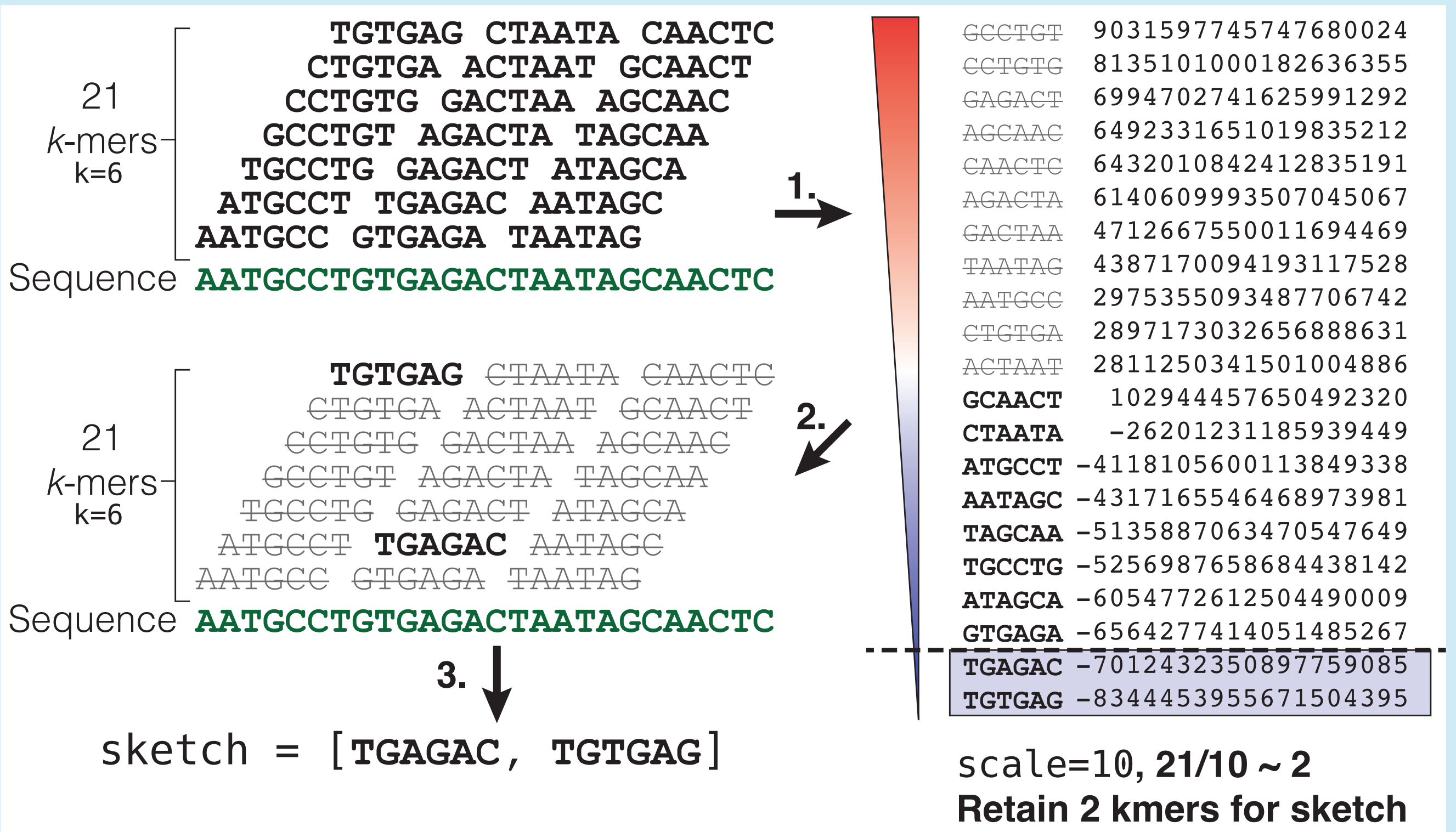
`hash(kmer)`

`kmer = 'GTGATT'`

2. Set a threshold for number of observed kmers: 1/`scale` kmers, e.g. 1/1000 for `scale=1000`

3. Save smallest kmer hashes as "bottom sketch"

Worked example of MinHash sketch, with k=6, `scale=10`



21 *k*-mers k=6

TGTGAG CTAATA CAACTC
CTGTGA ACTAAT GCAACT
CCTGTG GACTAA AGCAAC
GCCTGT AGACTA TAGCAA
TGCCTG GAGACT ATAGCA
ATGCCT TGAGAC AATAGC
AATGCC GTGAGA TAATAG

Sequence AATGCCTGTGAGACTAATAGCAACTC

1.

~~GCCTGT~~ 9031597745747680024
~~CCTGTG~~ 8135101000182636355
~~GAGACT~~ 6994702741625991292
~~AGCAAC~~ 6492331651019835212
~~CAACTC~~ 6432010842412835191
~~AGACTA~~ 6140609993507045067
~~GACTAA~~ 4712667550011694469
~~TAATAG~~ 4387170094193117528
~~AATGCC~~ 2975355093487706742
~~CTGTGA~~ 2897173032656888631
~~ACTAAT~~ 2811250341501004886
GCAACT 1029444576504923 20
CTAATA -2620123118593944 9
ATGCCT -4118105600113849338
AATAGC -4317165546468973981
TAGCAA -5135887063470547649
TGCCTG -5256987658684438142
ATAGCA -6054772612504490009
GTGAGA -6564277414051485267
TGAGAC -7012432350897759085
TGTGAG -8344453955671504395

2.

21 *k*-mers k=6

TGTGAG ~~CTAATA CAACTC~~
~~CTGTGA ACTAAT GCAACT~~
~~CCTGTG GACTAA AGCAAC~~
~~GCCTGT AGACTA TAGCAA~~
~~TGCCTG GAGACT ATAGCA~~
~~ATGCCT~~ TGAGAC ~~AATAGC~~
~~AATGCC GTGAGA TAATAG~~

Sequence AATGCCTGTGAGACTAATAGCAACTC

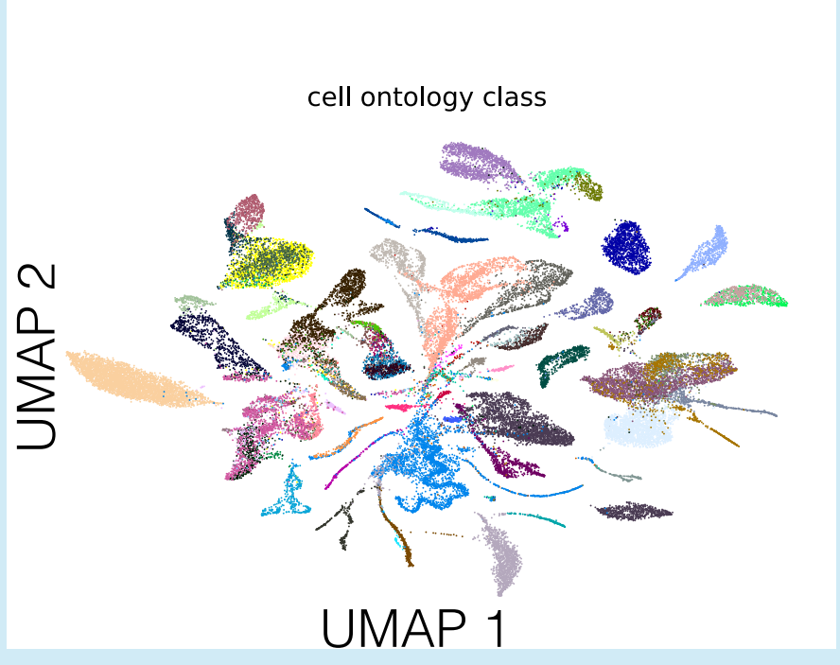
3.

`sketch = [TGAGAC, TGTGAG]`

`scale=10`, **21/10 ~ 2**
Retain 2 kmers for sketch

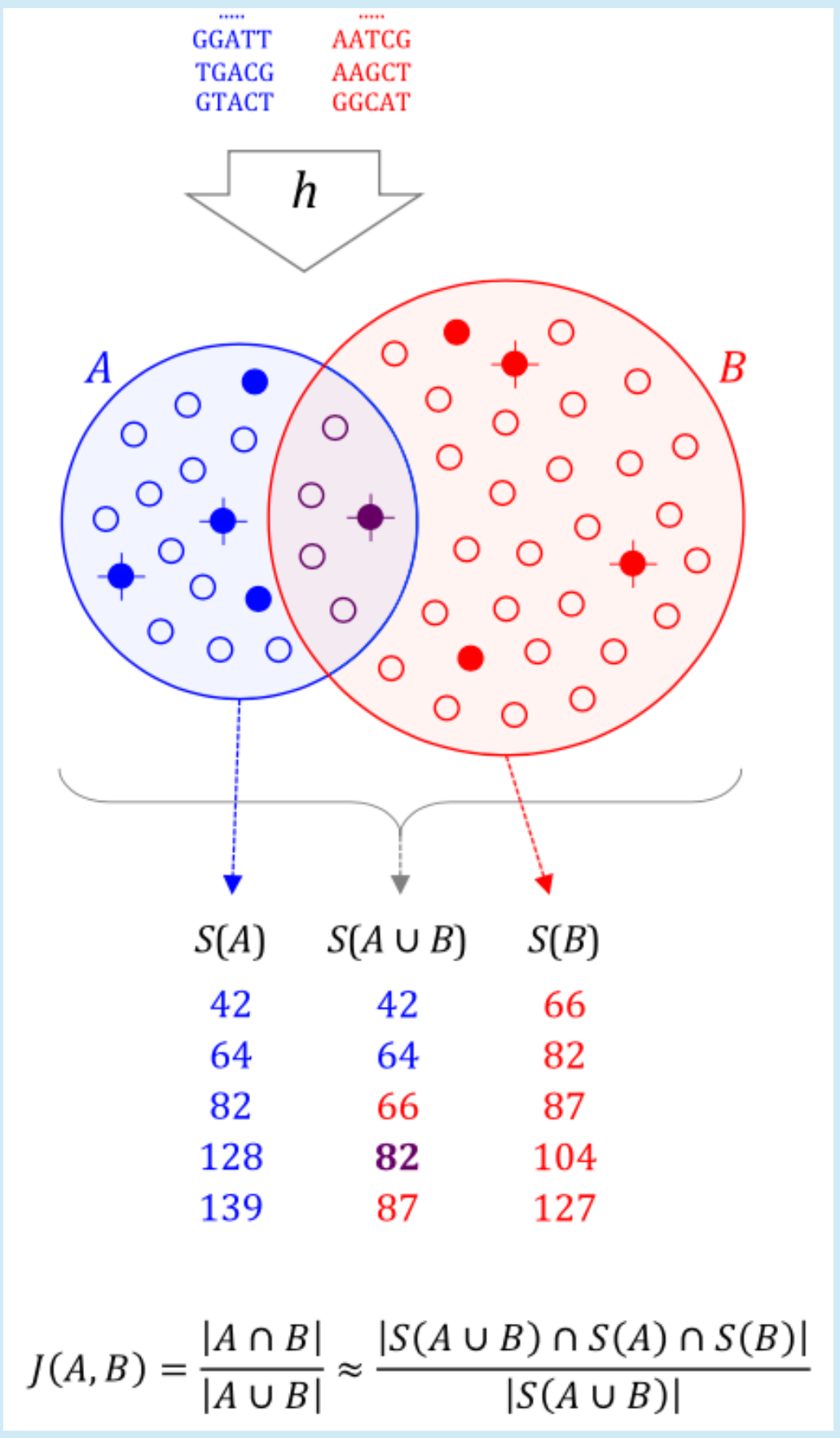
Results

Dataset: Tabula Muris (Wyss-Coray et al 2018) - SmartSeq2 single-cell RNA-seq of 50,000 cells of 20 mouse organs



UMAP projection of Tabula Muris dataset.

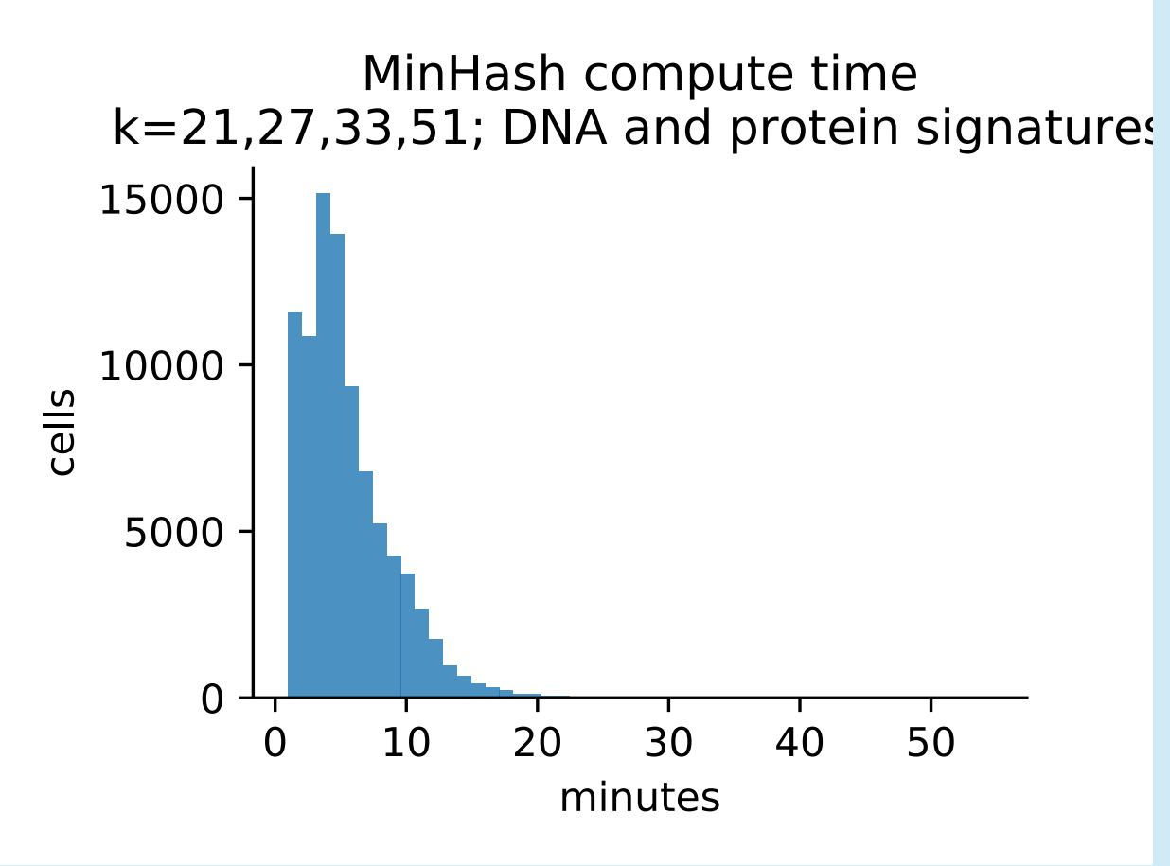
Software:
github.com/dib-lab/sourmash

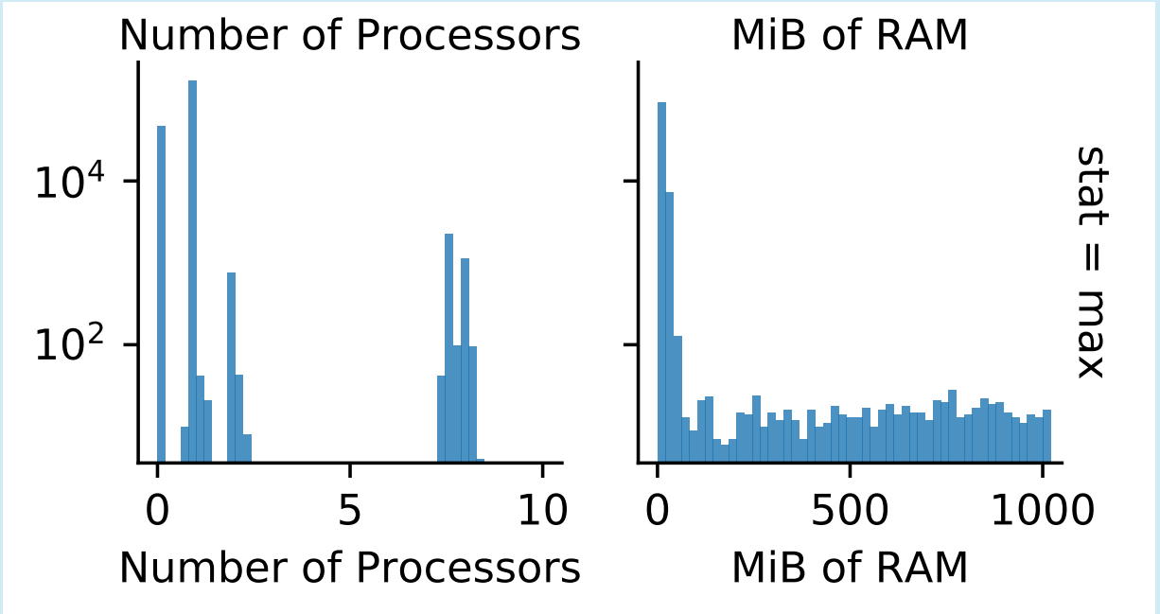


$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \approx \frac{|S(A \cup B) \cap S(A) \cap S(B)|}{|S(A \cup B)|}$$

Computation of approximate Jaccard similarity using MinHash from Ondov et al. 2016.

How fast is it? ~5min per cell signature computation





CPU and RAM usage by "sourmash compute" on single cells for `ksize=21, 27, 33, 51` and `scaled=1000.`

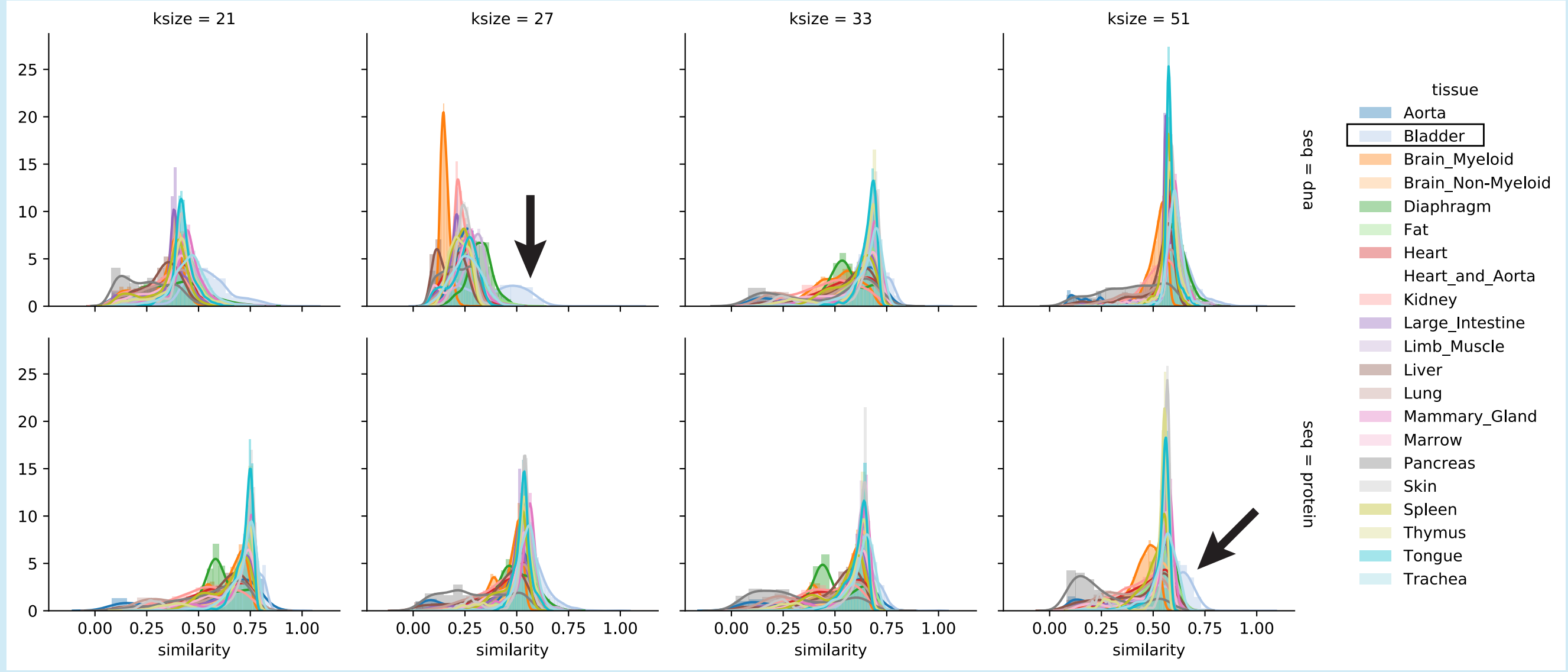
Total compute and data copying time by "sourmash compute" on single cells for `ksize=21, 27, 33, 51` and `scaled=1000.`

Self-lookup: "bladder cell" similarity to other organs

```
sourmash search --num-results 10 --csv  --ksize 21 --dna \
    /mnt/data/maca-facs-sourmash_compute_all/A1-B000610-3_56_F-1-1.sig \
    /mnt/data/sourmash_databases/tabula-muris-dna-k21.sbt.json


32210 matches; showing first 10:
similarity   match
----------   -----
100.0%       cell_ontology_class:bladder_cell|tissue:Bladder|subtissue...
 89.4%       cell_ontology_class:bladder_cell|tissue:Bladder|subtissue...
 87.9%       cell_ontology_class:bladder_cell|tissue:Bladder|subtissue...
 87.7%       cell_ontology_class:bladder_cell|tissue:Bladder|subtissue...
```

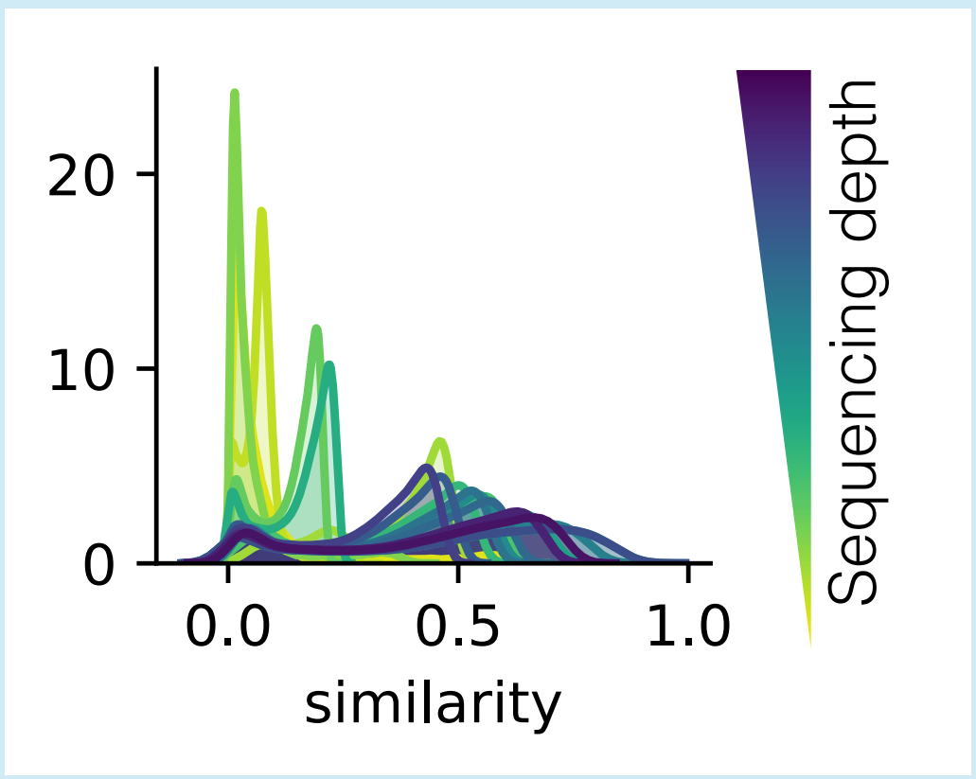
Top hits for bladder cell within sourmash database for k=21. 100% hit is to itself.



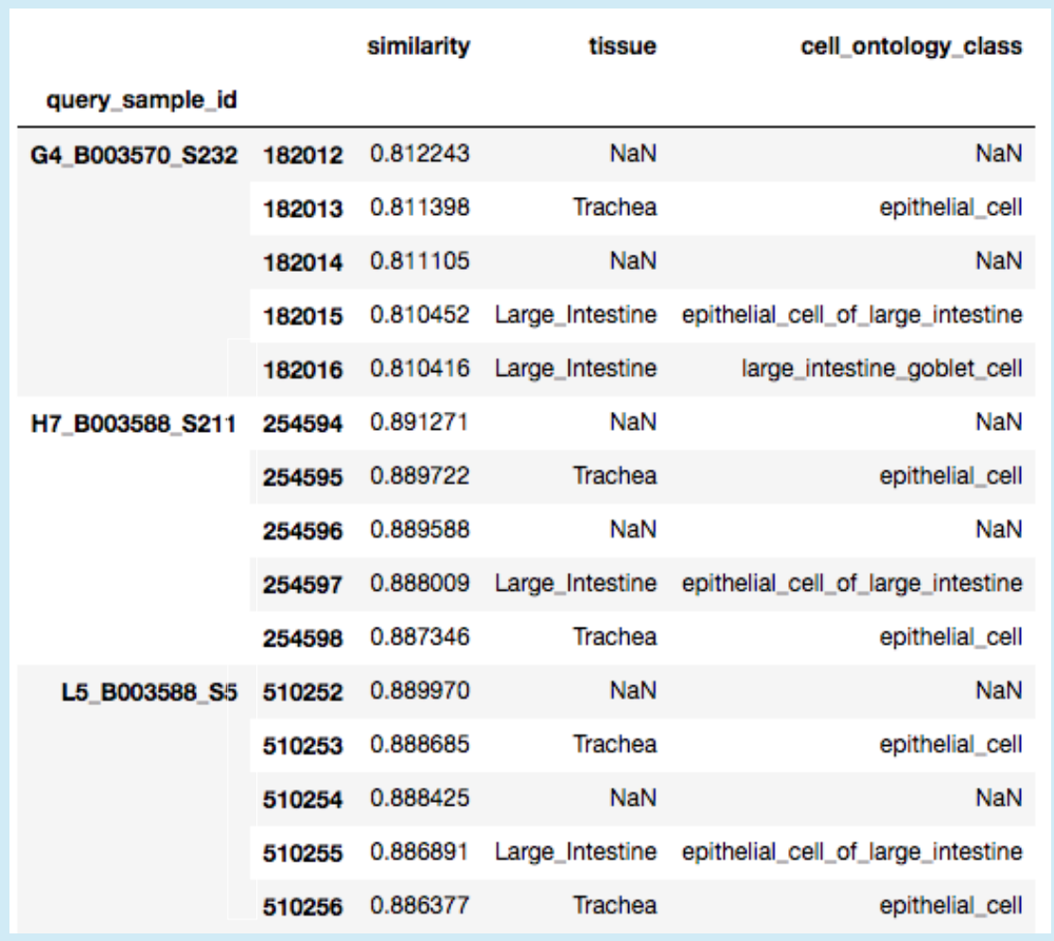
Distribution of cell-cell similarity relative to a cell with cell ontology class "bladder cell", across all of Tabula Muris.

Conclusion: k=27 best for DNA comparison, k=51 best for protein comparison

In progress: "LiftOver" annotations between species via protein signatures



Distribution of similarity relative to human lung cancer cells in Tabula Muris data. Similarity is extremely dependent on sequencing depth.



query_sample_id		similarity	tissue	cell_ontology_class
G4_B003570_S232	182012	0.812243	NaN	NaN
	182013	0.811398	Trachea	epithelial_cell
	182014	0.811105	NaN	NaN
	182015	0.810452	Large_Intestine	epithelial_cell_of_large_intestine
	182016	0.810416	Large_Intestine	large_intestine_goblet_cell
H7_B003588_S211	254594	0.891271	NaN	NaN
	254595	0.889722	Trachea	epithelial_cell
	254596	0.889588	NaN	NaN
	254597	0.888009	Large_Intestine	epithelial_cell_of_large_intestine
	254598	0.887346	Trachea	epithelial_cell
L5_B003588_S5	510252	0.889970	NaN	NaN
	510253	0.888685	Trachea	epithelial_cell
	510254	0.888425	NaN	NaN
	510255	0.886891	Large_Intestine	epithelial_cell_of_large_intestine
	510256	0.886377	Trachea	epithelial_cell

Top hits for human lung cancer cells by querying Tabula Muris data. NaN indicates samples that were filtered out in intial Tabula Muris QC and do not have an annotation.

Pros and Cons

Pros

- Fast to compute a MinHash signature per cell (5m)
- No need to align, renormalize, etc for a new dataset
- Self-lookup finds itself perfectly
- Can "LiftOver" cell type annotations between species via protein k-mers

Cons

- Looking up cell in a database is takes too long (1-4 hours), can be improved by more efficient search tree data structures
- Since k-mers are hashed, don't retain the actual sequence that differs between cells, so still need to perform differential expression to find differential genes
- Cannot find unlabeled cell types
- The method is only as good as the training data

Future Work

- Simulate RNA seq profiles differing by a known N number of transcripts
- Used fixed number of hashes rather than 1/1000 scaling
- To address depth-dependence, ignore abundance of kmers when comparing
- Speed up cell lookup - Implement AllSome (Sun et al. 2018) or Split Sequence Bloom Trees (Solomon and Kingsford 2018)
  - Both are GPL-3 licensed, which is incompatible with sourmash's and kmer-hashing's BSD-3 clause licenses
- Encode kmer signatures into Cell Ontology graph

References

Bray, Nicolas L., Harold Pimentel, Páll Melsted, and Lior Pachter. 2016. "Near-Optimal Probabilistic RNA-Seq Quantification." Nature Biotechnology 34 (5): 525–27. https://doi.org/10.1038/nbt.3519.

Broder, A. Z. 1997. "On the Resemblance and Containment of Documents." In Proceedings. Compression and Complexity of SEQUENCES 1997 (Cat. No.97TB100171), 21–29. https://doi.org/10.1109/SEQUEN.1997.666900.

Koslicki, David, and Daniel Falush. 2016. "MetaPalette: A K-Mer Painting Approach for Metagenomic Taxonomic Profiling and Quantification of Novel Strain Variation." mSystems 1 (3). https://doi.org/10.1128/mSystems.00020-16.

Ondov, Brian D., Todd J. Treangen, Páll Melsted, Adam B. Mallonee, Nicholas H. Bergman, Sergey Koren, and Adam M. Phillippy. 2016. "Mash: Fast Genome and Metagenome Distance Estimation Using MinHash." Genome Biology 17 (1): 132. https://doi.org/10.1186/s13059-016-0997-x.

Patro, Rob, Geet Duggal, Michael I. Love, Rafael A. Irizarry, and Carl Kingsford. 2017. "Salmon Provides Fast and Bias-Aware Quantification of Transcript Expression." Nature Methods 14 (4): 417–19. https://doi.org/10.1038/nmeth.4197.

Solomon, Brad, and Carl Kingsford. 2018. "Improved Search of Large Transcriptomic Sequencing Databases Using Split Sequence Bloom Trees." Journal of Computational Biology: A Journal of Computational Molecular Cell Biology 25 (7): 755–65. https://doi.org/10.1089/cmb.2017.0265.

Sun, Chen, Robert S. Harris, Rayan Chikhi, and Paul Medvedev. 2018. "AllSome Sequence Bloom Trees." Journal of Computational Biology: A Journal of Computational Molecular Cell Biology 25 (5): 467–79. https://doi.org/10.1089/cmb.2017.0258.

Titus Brown, C., and Luiz Irber. 2016. "Sourmash: A Library for MinHash Sketching of DNA." The Journal of Open Source Software 1 (5): 27. https://doi.org/10.21105/joss.00027.

Wyss-Coray, T., S. Darmanis, and Tabula Muris Consortium. 2018. "Single-Cell Transcriptomic Characterization of 20 Organs and Tissues from Individual Mice Creates a Tabula Muris." bioRxiv. https://www.biorxiv.org/content/early/2018/03/29/237446.abstract.

Acknowledgements

Thank you to the Luiz Irber (@luizirber) and C. Titus Brown (@ctb) at UC Davis for accepting my pull requests to the sourmash package for min hash signatures (github.com/dib-lab/sourmash/), and to Marcus D Sherman (@betteridiot) at University of Michigan for fixing 10x bam reading bugs in bamnostic (github.com/betteridiot/bamnostic). Thank you to Marius A. Eriksen (@mariusae) and Prasad Gopal (@prasadgopal) at Grail for answering my annoying questions and helping debug cloud computing workflows (github.com/grailbio/reflow).

Outside of CZ Biohub, this project would not have been possible without the generous help of open source software contributors. (@username) denotes a GitHub username.

Thank you to the Data Sciences and Darmanis teams at CZ Biohub, especially Josh Batson, James Webber, Angela Oliveira Pisco, Spyros Darmanis, Ashley Maynard, Lincoln Harris.