Image-based representation of massive spatial transcriptomics datasets

Stephan Preibisch[1], Nikos Karaiskos[2] & Nikolaus Rajewsky[2]

[1]Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA
[2]Systems Biology of Gene Regulatory Elements, Berlin Institute for Medical Systems Biology, Max Delbrück Center for Molecular Medicine in the Helmholtz Association, Berlin, Germany

Correspondence: preibischs@janelia.hhmi.org, rajewsky@mdc-berlin.de

Abstract

We present STIM, an imaging-based computational framework for exploring, visualizing, and processing high-throughput spatial sequencing datasets. STIM is built on the powerful ImgLib2, N5 and BigDataViewer (BDV) frameworks enabling transfer of computer vision techniques to datasets with irregular measurement-spacing and arbitrary spatial resolution, such as spatial transcriptomics data generated by multiplexed targeted hybridization or spatial sequencing technologies. We illustrate STIM's capabilities by representing, visualizing, and automatically registering publicly available spatial sequencing data from 14 serial sections of mouse brain tissue.

Main

Several recent technological breakthroughs have triggered the rapid development of numerous high-throughput spatial transcriptomics methods over the last few years. These fluorescent RNA hybridization-based[1,2] or array-based RNA capture, barcoding, and subsequent sequencing[3–8] techniques provide (typically at single-cell or subcellular resolution) molecular readouts within the native spatial context of a tissue, which is critical for understanding cellular interactions in healthy and diseased states[9]. Thus, we anticipate the production of a steeply growing number of spatial transcriptomics data sets important for basic life science as well as clinical/medical research and applications. The handling and analyses of these datasets pose several, currently unaddressed challenges: (a) size – spatial sequencing can easily generate gigabytes of data for a single tissue slice (b) heterogeneity – datasets greatly differ in the number of genes and transcripts captured, their spatial resolution, and tissue architecture (c) spatial transcriptomics data are often not spaced regularly (d) three-dimensional (3D) integration – data from slices need to be integrated into a 3D molecular map (e) access, analyses and visualization – the need to

easily share and interactively interrogate spatial transcriptomics data, and (f) flexibility and long-term availability – the need for an open source, community-based approach. Here, we show that powerful methods from the computer vision field that have been developed by a large scientific community for decades, can be repurposed to meet all of these challenges. Specifically, we present the "Spatial Transcriptomics Imaging Framework" (STIM), a computational, scalable and extendable toolkit based on ImgLib2[10] that allows efficient handling, processing (including integration of 2D data into 3D molecular maps), visualization and analysis of high-throughput spatial -omics datasets. We demonstrate the power of our approach by applying STIM to spatial sequencing data from adjacent slices of mouse brain and integrating them into a 3D molecular map of the cerebellum.

The generic, interface-driven design of ImgLib2[10] natively supports non-regularly spaced pixels and imposes no constraints on dataset size, dimensionality or type, which is highlighted by the fact that the largest biological image datasets ever acquired were reconstructed using ImgLib2, the ImgLib2-backed BDV[11,12] and the integrated N5 file format[13]. STIM makes use of these frameworks to provide fast, random and distributed read&write access, interactive visualization and efficient processing. First, STIM re-saves input datasets from standardized text- or comma-separated formats into an N5 container (**Fig 1a,b**) while optionally log-normalizing the data. Coordinate- and gene expression data can be loaded fast and memory-efficiently in blocks using the Imglib2-cache framework, which can be accessed as values or as rendered images (**Fig. 1b,c**). ImgLib2 provides nearest-neighbor and linear interpolation for mapping irregularly-spaced samples onto pixel grids necessary for visualization. To achieve a more realistic-looking rendering at arbitrary resolutions we implemented a rendering method based on Gaussian distributions (**Fig. 1, Methods, Supplementary Methods**). Spatial image filtering, also referred to as digital filtering, is an established, powerful technique to enhance certain aspects of a signal that is represented as discrete samples (e.g. an image) using mathematical operations[14]. While such filtering is mathematically directly applicable to irregularly-spaced data, it is not widely available as efficient implementations are more complex. To ease this barrier, we added a generic framework based on ImgLib2 using kd-trees for applying filters (e.g. Mean, Median, Gaussian) to irregularly-spaced data that can easily be extended (**Fig. 1c**). All operations are implemented virtually allowing interactive access to and rendering of the data using BDV.

Here, we demonstrate STIM's capabilities as follows. First, STIM can be used to filter the data and smoothen it, for instance by applying a Median filter or others (**Fig. 1c**). Second, tried-and-tested image registration techniques can be used to align datasets stemming from consecutive sections of the same tissue as described below (**Fig. 1d,e**). Third, STIM offers

an interactive visualization and exploration of the data through BDV in 2D and 3D, including the visualization of metadata -- such as cell type annotation -- together with gene expression in every spatial unit (**Supplementary Fig. 1**).

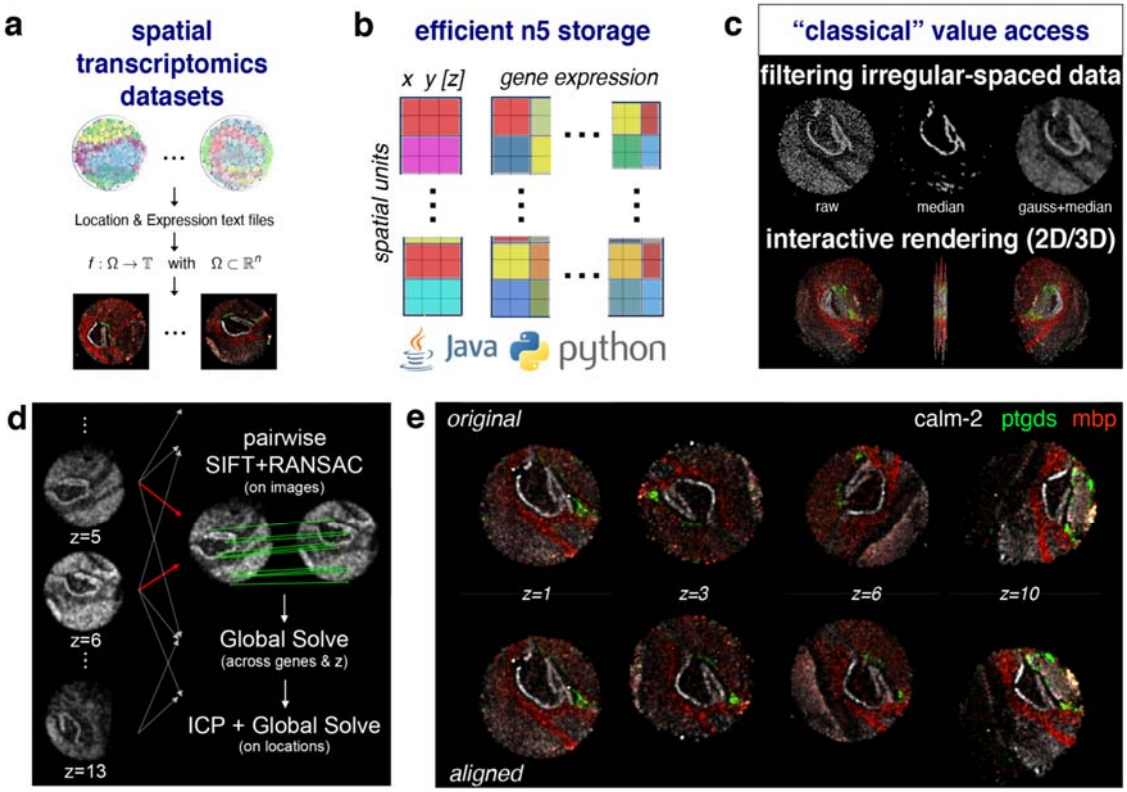

Figure 1: STIM unleashes the power of imaging techniques to analyze massive amounts of spatial transcriptomics datasets. ***(a)*** *Spatial transcriptomics datasets can be represented as images with the number of genes per spatial unit corresponding to a number of different colors.* ***(b)*** *Expression and spatial data is stored in N5 containers for efficiency and scalability and is accessible within the provided Java and Python frameworks.* ***(c)*** *STIM provides classical value access operations such as filtering irregular-spaced data, resulting in smoothed gene expression, or interactive rendering in 2D and 3D.* ***(d),*** *Schematic of how STIM aligns consecutive sections of spatial transcriptomics datasets.* ***(e),*** *Visualization of the spatial gene expression of three genes in four different sections of a published dataset. Top (bottom) row: spatial gene expression profiles in the original (aligned) puck orientations.*

To illustrate the power of applying computer vision techniques to sequencing datasets, we aligned a series of consecutive brain cerebellum sections using STIM that were published in [4] (**Fig. 2 / Supplementary Video 1**). Each of these 2D sections contains between 12,000 and 33,000 -cells and a median of ~50 quantified molecules at near-cellular resolution. To align the 13 sections in three dimensions we adapt an alignment strategy originally developed for the registration of large electron and light microscopy datasets[12,15,16]. We first apply Scale Invariant Feature Transform (SIFT)[17] in combination with Robust Sample

Consensus (RANSAC)[18] on pairs of sections (+-2 in the z-direction, **Fig. 1d**). For each section pair, we identify a set of corresponding points on a rigid 2D transformation across an automatically selected set of genes that show high entropy in both sections. To correctly identify as many corresponding points as possible we run SIFT independently for each gene with a low threshold for the minimal number of required points. RANSAC is then applied to all points again to identify those that agree on a common transformation across genes. Next, we globally minimize the distance between all corresponding image points across all sections, yielding a single 2D transformation for each section. In an optional refinement step, we use the Iterative Closest Point (ICP) algorithm[19] on locations of sequenced spots rather than rendered images, where neighboring points within a predefined radius showing most similar expression values are assigned to be correspondences. Finally, using all ICP correspondences, we globally solve again and identify a regularized affine transformation model for each section that is stored in the N5 container. The resulting tissue dataset can be rendered in 3D using STIM (**Fig. 2 / Supplementary Video 1**) and the STIM-explorer can be used to highlight the spatial expression of interactively selected genes on the whole tissue (**Supplementary Video 2**). Due to the robustness of SIFT, our alignment pipeline can readily be employed for similar technologies. To further demonstrate this, we used STIM to align serial adult mouse brain sections produced with the Visium platform (**Supplementary Fig. 2**). More generally, we anticipate STIM to process and stitch together consecutive tissue sections from the same tissue, regardless of the underlying spatial sequencing method. This enhances the information flow between the different sections and naturally enriches the molecular readouts.

STIM enables efficient, distributed access, processing, and visualization of large-scale spatial transcriptomics datasets. Irregularly-spaced data can be spatially filtered and accessed directly as values or rendered as images. STIM thereby acts as a bridge between the fields of computer vision and genomics, which we highlight by developing an automatic workflow for the alignment of sliced spatial transcriptomics datasets. We provide STIM as an extensible open-source framework available on GitHub (https://github.com/PreibischLab/STIM and https://github.com/rajewsky-lab/stimwrap) with interfaces in Java, Python and on the command line. We believe that these properties of STIM have the potential to enable the community to further unite the worlds of image analysis and genomics.

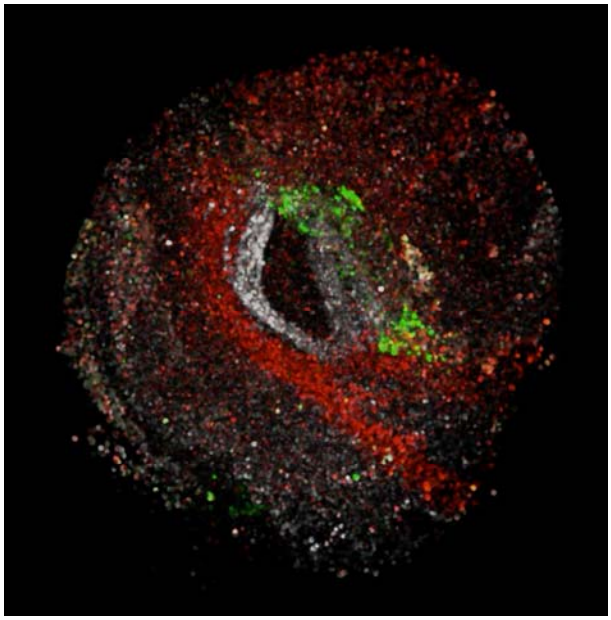

Figure 2 (Movie, submitted as Supplementary Movie 1): 3D rendering of the aligned SlideSeq dataset. *The movie shows gene calm-2 is shown in white, gene ptdgs in green, and gene mbp in red and highlights the 3D nature of the dataset after alignment when represented as an image.*

Methods

N5 storage and normalization

STIM ingests spatial sequencing data that is stored in (compressed) text files containing locations and barcodes of sequenced spots, expression levels, and optionally cell type predictions per dataset. The current framework supports 2D and 3D coordinates, and can be readily extended. Initially, STIM re-saves the set of text-file based datasets into a common N5 container that can be accessed as one common project, e.g. the image registration pipeline can be applied to an entire or parts of a project. By default, locations and expressions are stored with double precision and Gzip compression using a block length of 1024 for locations and a block size of 512x512 for expression values (Fig. 1b). Genes and the barcode list of each dataset together with their transformations are currently stored as metadata in the N5 container. The N5 data can be accessed in STIM/stimwrap or directly through Java and python N5 packages.

Optionally, the data can be normalized upon re-saving to N5 or at a later time. We have adopted here the standard library size normalization in log-space commonly used for scRNA-seq datasets. More specifically, if represents the raw count for gene in spatial unit , we normalize values as

$$d_{ij} \rightarrow d'_{ij} = \log_2\left(10^5 \times \frac{d_{ij}}{\sum_k d_{kj}} + 1\right)$$

where the dummy index is used for summing over all genes within a spatial unit.

Rendering of irregularly-spaced data

Image data is typically stored in n-dimensional, integer-based cartesian coordinate systems that are natively supported by most camera chips and display devices. In contrast, spatial sequencing data consist of measurements at arbitrary, floating-point precise locations, which stem from accurately localizing sequenced locations. To render such data, they need to be mapped to an integer-based cartesian coordinate system supported by standard display devices (e.g. to a 2048x1536 pixel grid), a problem that also occurs in other dataset types such as localization-based superresolution microscopy[20] and other disciplines such as astronomy[21]. Straight-forward, fast mapping can be achieved through nearest-neighbor interpolation using kd-trees. Resulting images are effectively Voronoi-tessellations with sharp, unnatural boundaries and artificial appearance where point densities are low towards the edges (**Suppl. Fig. 3A,B**). Distance-weighted interpolation creates more natural-looking images, which, however, still contain unnatural edges as either a number of points or maximal distance for interpolation needs to be defined (**Suppl. Fig. 3C**). Large values are able to create reasonable representations in areas of high point densities, but can still produce artificial structures, especially towards the edges of the dataset (**Suppl. Fig 3D**). To overcome these issues we represent each location as a Gaussian distribution, and each pixel is rendered as the sum of all overlapping distributions, normalized by their respective weights. In order not to create hard boundaries in areas with few locations, normalization is only performed if the sum of weights is bigger than one. This mapping is fast and produces representations of the data resembling naturally-looking images that can therefore easily be processed with computer vision tools and are additionally visually pleasing (**Fig. 1, Supplementary Video 2, Suppl. Fig. 3E**).

Filtering of irregularly-spaced data

Within STIM, we implemented a framework for spatial filtering of irregularly-spaced datasets based on kd-trees. We added mean filtering, median filtering, Gaussian filtering, as well as practical filters to hide single, isolated locations and to visualize the density of locations. Adding new filters is straight-forward, and typically requires the implementation `filter.RadiusSearchFilter` class, which already provides the kd-tree search and the location to be filtered.

Such basic filtering operations can for example help to smoothen noisy spatial sequencing data, to emphasize larger structures, or to identify edges (**Supplementary Video 2, Suppl. Fig. 3**).

Pairwise SIFT registration

In order to robustly identify corresponding points between pairs of two-dimensional serial sections of spatial sequencing datasets we first employ the Scale Invariant Feature Transform (SIFT) on images of renderings of individual genes.

First, we identify a set of genes (by default 100) that are expressed in both serial sections and show the highest combined standard deviations of their expression values, thus automatically selecting genes that are likely to show patterns that are helpful to perform an alignment. The user can additionally add genes that are known to create well-structured expression renderings. Second, we compute SIFT on all pairs of genes individually, using a low minimal number of corresponding points (inliers, by default 5) on a rigid model. Finally, we perform another RANSAC consensus across the points of all genes, requiring by default

at least 30 inliers. This combination of parameters was very robust in our tests and worked out of the box for SlideSeq as well as Visium datasets.

Global Optimization

In order to align more than two serial sections we first compute pairwise SIFT registrations between close-by sections (by default +-2). Using the set of all pairs of corresponding points we solve a global optimization that minimizes the distance between all corresponding points of all serial sections[12]. Potentially wrong pairs can be automatically identified and removed in the process as long as it happens infrequently[16] . By default, we use affine transformations regularized with rigid transformations for each serial section for the optimization process. All transformations are stored within the N5 metadata, thus all visualization tools of STIM will directly use these transformations.

ICP refinement

After a first round of global optimization based on the corresponding interest points identified by SIFT we optionally employ Iterative Closest Point (ICP) refinement. In contrast to the SIFT alignment step, ICP is performed on the actual coordinates of the sequenced locations. We again first compute pairwise ICP's between close-by serial sections using only the expression values of genes that yielded SIFT correspondences. The core of ICP is the assignment of nearest points as corresponding points in an iterative fashion. Here, we do not simply assign nearest points to each other, but those who show the most similar expression vector in the local vicinity (by default the median distance between all sequenced locations). We optionally support RANSAC filtering on the sets of corresponding points during each ICP iteration in order to identify a consensus update vector across all neighboring points. After all pairwise ICP matchings are performed, we re-solve the global optimization problem.

Data availability

All data analyzed within this work are publicly available. The datasets that were used for the alignment of mouse cerebellum sections were published in Slide-seq[4]. The 10X Visium datasets were downloaded from the 10X Genomics website.

Code availability

STIM and stimwrap are available on GitHub (https://github.com/PreibischLab/STIM and https://github.com/rajewsky-lab/stimwrap).

Contributions

Acknowledgements

S.P. was supported by the HFSP grant RGP0021/2018-102, MDC Berlin and HHMI Janelia. N.K. was supported by the DFG grants KA 5006/1-1 and RA 838/5-1. N.R. was supported by MDC Berlin and Charite.

Ethics declarations

Competing interests

This work is part of a larger patent application in which the authors are among the inventors. The patent application was submitted through the Technology Transfer Office of the Max-Delbrück Center (MDC), with the MDC being the patent applicant.

Bibliography

1. Lubeck, E., Coskun, A. F., Zhiyentayev, T., Ahmad, M. & Cai, L. Single-cell in situ RNA profiling by sequential hybridization. *Nat. Methods* **11**, 360–361 (2014).
2. Chen, K. H., Boettiger, A. N., Moffitt, J. R., Wang, S. & Zhuang, X. RNA imaging. Spatially resolved, highly multiplexed RNA profiling in single cells. *Science* **348**, aaa6090 (2015).
3. Ståhl, P. L. *et al.* Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science* **353**, 78–82 (2016).
4. Rodriques, S. G. *et al.* Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363**, 1463–1467 (2019).
5. 10X Genomics. Spatial Gene Expression - 10x Genomics. https://www.10xgenomics.com/products/spatial-gene-expression.
6. Cho, C.-S. *et al.* Microscopic examination of spatial transcriptome using Seq-Scope. *Cell* **184**, 3559-3572.e22 (2021).
7. Vickovic, S. *et al.* High-definition spatial transcriptomics for in situ tissue profiling. *Nat. Methods* **16**, 987–990 (2019).
8. Stickels, R. R. *et al.* Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2. *Nat. Biotechnol.* **39**, 313–319 (2021).
9. Rao, A., Barkley, D., França, G. S. & Yanai, I. Exploring tissue architecture using spatial transcriptomics. *Nature* **596**, 211–220 (2021).
10. Pietzsch, T., Preibisch, S., Tomancák, P. & Saalfeld, S. ImgLib2--generic image processing in Java. *Bioinformatics* **28**, 3009–3011 (2012).
11. Pietzsch, T., Saalfeld, S., Preibisch, S. & Tomancak, P. BigDataViewer: visualization

and processing for large image data sets. *Nat. Methods* **12**, 481–483 (2015).

12. Preibisch, S., Saalfeld, S. & Tomancak, P. Globally optimal stitching of tiled 3D microscopic image acquisitions. *Bioinformatics* **25**, 1463–1465 (2009).
13. GitHub - saalfeldlab/n5: Not HDF5. https://github.com/saalfeldlab/n5.
14. Oppenheim, A. V. & Schafer, R. W. *Digital Signal Processing*. 585 (Pearson, 1975).
15. Saalfeld, S., Fetter, R., Cardona, A. & Tomancak, P. Elastic volume reconstruction from series of ultra-thin microscopy sections. *Nat. Methods* **9**, 717–720 (2012).
16. Hörl, D. *et al.* BigStitcher: reconstructing high-resolution image datasets of cleared and expanded samples. *Nat. Methods* **16**, 870–874 (2019).
17. Lowe, D. G. Object recognition from local scale-invariant features. in *Proceedings of the Seventh IEEE International Conference on Computer Vision* 1150–1157 vol.2 (IEEE, 1999). doi:10.1109/ICCV.1999.790410.
18. Fischler, M. A. & Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **24**, 381–395 (1981).
19. Arun, K. S., Huang, T. S. & Blostein, S. D. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Anal. Mach. Intell.* **9**, 698–700 (1987).
20. Marin, Z. *et al.* PYMEVisualize: an open-source tool for exploring 3D super-resolution data. *Nat. Methods* **18**, 582–584 (2021).
21. Schaap, W. E. & van de Weygaert, R. Continuous Fields and Discrete Samples: Reconstruction through Delaunay Tessellations. *arXiv* (2000).