*NAR Genomics and Bioinformatics*, 2025, **7**, lqaf044
https://doi.org/10.1093/nargab/lqaf044
**Pipelines and Workflows for Biological Data Analysis**

# Fast analysis of Spatial Transcriptomics (FaST): an ultra lightweight and fast pipeline for the analysis of high resolution spatial transcriptomics

**Valerio Fulci** *

Dipartimento di Medicina Molecolare. Università di Roma “La Sapienza”, Viale Regina Elena, 291 Rome, Italy
*To whom correspondence should be addressed. Email: valerio.fulci@uniroma1.it

## Abstract

Recently, several protocols repurposing the Illumina flow cells or DNA nanoballs as an RNA capture device for spatial transcriptomics have been reported. These protocols yield high volumes of sequencing data which are usually analyzed through the use of high-performance computing clusters. I report Fast analysis of Spatial Transcriptomic (FaST), a novel pipeline for the analysis of subcellular resolution spatial transcriptomics datasets based on barcoding. FaST is compatible with OpenST, seq-scope, Stereo-seq, and potentially other protocols. It allows full reconstruction of the spatially resolved transcriptome, including cell segmentation, of datasets consisting of >500 M million reads in as little as 1 h on a standard multi core workstation with 32 Gb of RAM. The FaST pipeline returns RNA segmented Spatial Transcriptomics datasets suitable for subsequent analysis through commonly used packages (e.g scanpy or seurat). Notably, the pipeline I present relies on the spateo-release package for RNA segmentation and does not require hematoxylin/eosin or any other imaging procedure to guide cell segmentation. Nevertheless, integration with other software for imaging-guided cell segmentation is still possible. FaST is publicly available on github (https://github.com/flcvlr/FaST)

## Introduction

Spatially resolved transcriptomics (ST) has recently emerged as a powerful technique to investigate the spatial patterns of gene expression in tissues, in physiologic as well as pathologic contexts [1]. While the earliest approaches yielded gene expression information with a medium resolution (~100 μm), more recent protocols proved able to faithfully record spatially resolved gene expression at submicrometric resolution, allowing analysis of these datasets at cellular and subcellular level. In particular, a series of recent works has independently reported that Illumina flow cells or DNA nanoballs may be used as an RNA capture device to yield ST datasets at a resolution of ~0.6 μm, closely resembling the resolution power of optical microscopy and faithfully recapitulating the expected histological features of the analyzed tissues [2–5]. A typical ST experiment on a 10–15 mm$^2$ specimen will yield 0.5–1 billion reads posing a considerable challenge to rapid and efficient downstream analysis. Importantly, Illumina flow cells are organized as an array of “tiles” each with a surface of ~1 mm$^2$. A typical ST experiment will involve 15–20 such “tiles”. While recently published papers mainly focus on “proof of principle” specimens, with a deep focus on a single or few samples, unleashing ST potential will require analysis of large cohorts of samples, suggesting the need for a lightweight, reproducible, and fast analysis pipeline that could be routinely applied in both small scale and large scale dataset to provide a univocal method for barcode capture based submicrometric resolution ST datasets.

One of the major issues posed by analysis of high resolutions ST is represented by the cell segmentation step of the analysis. Earliest approaches have leveraged tissue staining techniques highlighting cell nuclei (e.g. DAPI (4′,6-diamidino-2-phenylindole) and H&E (Hematoxylin and Eosin)) thus providing compelling evidence that the density and identity of the captured RNAs faithfully mirrors the composition of the analyzed tissues at single cell level [2, 4]. However, such approaches require tissue staining, image acquisition and image data analysis, resulting in complicated and poorly scalable protocols both in terms of wet lab and bioinformatic analysis. Indeed, several recent bioinformatic approaches have been developed to perform segmentation of ST data analysis into putative single cells guided in part or completely by RNA density as assessed by ST [6–9].

I describe the Fast analysis of ST (FaST) pipeline, a simple yet rigorous algorithm with a low memory footprint, which allows very quick analysis of large ST datasets yielding RNA segmented datasets in anndata [10] formats suitable for downstream analysis with widespreadly used single cell transcriptome data analysis tools [11, 12].

The data I present show that RNA segmented “pseudo cells” recapitulate faithfully the results obtained by image based cell segmentation, both in terms of cell type identification, cell number, and specificity of gene expression.

FaST has been written largely in bash and perl and by design has a minimal set of requirements, to ensure long term reliability with minimal maintenance of the pipeline.

**Received:** August 5, 2024. **Revised:** February 27, 2025. **Editorial Decision:** April 2, 2025. **Accepted:** April 3, 2025

## Materials and methods

### Flowcell barcode map preparation

FaST reads the HDMI fastq file obtained in the first round of sequencing and outputs a "flow cell barcode map" consisting of a set of files (one for each tile in the flow cell) each listing the HDMI barcodes associated with $x$ and $y$ coordinates as reported in the header of the reads. If identical barcodes occur within a tile, those are discarded. Identical barcodes across different tiles are kept and later handled during sample analysis.

At the same time, an index consisting of a random sample of the barcodes from each tile is saved to allow fast retrieval of tiles in each experiment during data analysis.

### Sample fastq reads preprocessing

FaST reads the whole set of R1 reads (containing spatial barcodes). To identify the tiles of the Illumina flow cell used for RNA capture, FaST compares the whole set of R1 reads with the Illumina flowcell barcode map index generated with FaST-map, retaining tiles for which at least 10% of the indexed barcodes are present in the R1 file.

Following the identification of the tiles used in the experiment, only barcodes mapping to these tiles are retained, and all reads corresponding to a different barcode are discarded. Barcodes mapping to multiple tiles in the flow cell, but mapping to one and only one barcode within the selected tiles are retained. Any other ambiguous barcode (i.e. matching more than one barcode in the selected tiles) will be discarded.

Following the R1 reads analysis and tile/barcode selection, R2 reads will be collected and converted into a unaligned BAM file with the following attributes as BAM tags:

- R1 barcode: BC:Z: tag
- Tile name: TL:Z:tag
- UMI: MI:Z:tag
- Hashed X Y coordinates within tile: CO:Z:tag

### Reads alignment

FaST uses a single alignment step with STAR [13]. The STAR index is generated by the FaST-reference script. During alignment polyA tail will be clipped by STAR. All BAM tags will be retained and reads with a valid alignment will be output in BAM format.

### Digital gene expression

The BAM file is then split for parallel processing tile by tile, piping reads belonging to the same tile (based on the TL:Z: tag of the read) to parallel processes. Each process will:

1. Remove and count reads mapping to rRNA and PhiX (selected based on the chromosome names) to output PhiX and rRNA mapping reads statistics.
2. A custom perl script parses genomic intervals overlapped by each read, setting the subcellular localization to cytoplasmic if the gene is a mitochondrial gene and to nuclear if the read overlaps any intron of the gene body. Due to the lack of introns in mitochondrial genes, these two classifications will be necessarily mutually exclusive for reads mapping to a single gene body. If all mappings of the same read map to the same gene body, an output will be generated, consisting of the gene name, the barcode, the UMI and the $x$ and $y$ coordinates of the barcode and the putative subcellular localization.
3. A custom perl script loads all the data of each tile into a sparse "barcodes x gene counts" matrix and writes MatriMarket format output and a text file in the spateo-release input format.

### RNA-based cell segmentation

Cell segmentation starts with a list of nuclear localized transcripts including microRNA host genes and APEX-seq [14] determined nuclear transcripts to obtain a putative nuclear mask applying the EM + BP algorithm. Then the intron counts are used to perform a second round of segmentation and resulting masks are joined with those obtained in the previous step. Finally, whole cells are segmented using the entire matrix of all reads and joined with the masks obtained in the previous step. Segmented cell counts and spatial coordinates are exported in Anndata format [10].

### Benchmarking

Benchmarks were performed either in a "workstation" or in an "HPC" hardware.

The "workstation" hardware is equipped with a 12th Gen Intel® Core™ i9-12900K processor (16 CPU cores, 8 of which are multithreading cores, for a total of 24 threads) and 32 Gb of RAM. All jobs were launched with a 24 threads setting in this hardware.

A single node of the HPC hardware used for benchmarking is equipped with 2 AMD EPYC™ 7452 processors (32 CPUs, 64 threads each) and 256 Gb of RAM. All jobs were launched with a 32 threads setting in this hardware. This "HPC" enforces a rule making available no more than 2 Gb of RAM for each thread, resulting in a maximum of 64 Gb of RAM when 32 threads are reserved through slurm. To ensure that spacemake and NovaScope jobs requiring >64 Gb of RAM were not killed, a considerably higher number of threads was reserved, although the actual jobs were launched on no more than 32 threads. Due to the limited memory footprint of FaST, this was not needed when running FaST.

### Downstream analysis

Anndata object were analyzed with scanpy 1.10.1 [12], filtering with identical parameters (min counts/cell = 200; max counts/cell = 3000, mitochondrial counts % < 20, minimum number of cells expressing a gene to retain it = 10) for all datasets. Data were normalized and log transformed. Leiden algorithm was used for clustering.

## Results

### The FaST pipeline

FaST is mainly written in perl and bash and has a very limited set of dependencies (STAR, samtools, bedtools, and the python packages spateo-release and scanpy). The first steps of the pipeline consist in the collection of the spatial barcodes of the experiment ("read 1"), the selection of the "tiles" of the Illumina flow cell corresponding to that set of barcodes and the mapping of the reads to the reference genome with STAR. Additionally, several data, including spatial coordinates, putative subcellular localization, UMI, and barcode, are attached to each read as BAM tags, thus avoiding time consuming down-

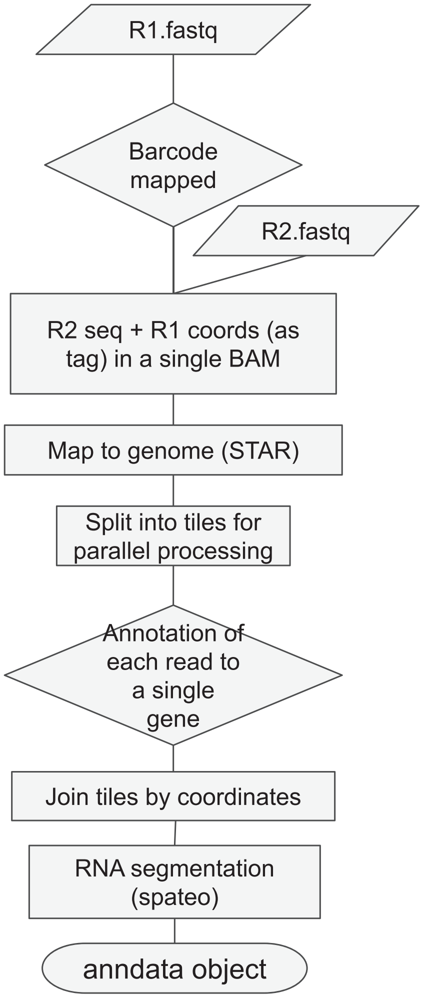
R1.fastq
Barcode mapped
R2.fastq
R2 seq + R1 coords (as tag) in a single BAM
Map to genome (STAR)
Split into tiles for parallel processing
Annotation of each read to a single gene
Join tiles by coordinates
RNA segmentation (spateo)
anndata object

**Figure 1.** Flowchart representing the major steps of the FaST pipeline.

stream steps to re-match barcode, spatial coordinates, and read alignment data (Fig. 1).

Downstream of reads alignment on the reference genome using STAR [15], reads are parsed by custom scripts which are launched in parallel on each tile of the Illumina flow cell or on subsections of the Stereo-seq capture area used in the experiment. Reads are first compared with Gencode annotation [16] and checked for overlap with one and only one gene. Reads spanning an exon/exon junction or a mitochondrial gene are labeled as cytoplasmic, while reads overlapping intron sequences or mapping to microRNA host genes are labeled as nuclear. A further tag corresponding to the gene is added in this step.

Reads mapping to the same gene and carrying the same barcode and UMI are collapsed and used to build a "puck"-level sparse count matrix for each tile. These data can be easily imported into Seurat, Scanpy, or other similar tools and converted in further formats to be fed to a variety of downstream cell segmentation packages, including OpenST.

The sparse matrix is then used by FaST to generate the input file for the spateo package for each tile. Depending on the capture area, RNA segmentation is performed either in parallel on groups of tiles or in a single step and the RNA segmented tiles are finally rejoined into a single anndata object.

RNA segmentation leverages a pool of nuclearly localized transcripts, as observed through Apex-seq [14].

To achieve a quick processing of large amounts of data, FaST, with the notable exception of reads alignment using STAR and segmentation using Spateo, does not rely on preexisting functions, but is an original, *de novo* implementation tailored to barcode-based ST data analysis, implemented in pure Perl. For this purpose, the following data structures have been developed:

- A dedicated nucleotide sequence hash data structure to store the barcodes in RAM, with a minimal memory footprint. The structure is spread on 16 threads, so that 16 cores can act in parallel on different, independent slices of this database. The hashed structure allows duplicate identification, collapsing identical barcodes and rejection of ambiguous barcodes (i.e. same sequence and different coordinates) in quasi-linear time, as no sorting and a very small number of pairwise comparisons are required in the process.
- A subsampling strategy to the identification of the tiles (out of the ∼3500 tiles in the Illumina flowcell) allows FaST to correctly identify the tiles ∼1 min after barcode collection has been completed.
- A dedicated data structure for the mapping to the annotation, which relies on genomic coordinates to quickly retrieve annotation features in the same genomic bin as the read under exam. This allows quick annotation without sorting the BAM file by coordinates, allowing this step to proceed in linear time complexity.
- A dedicated hash table structure for UMI collection at each coordinate of each tile, thus relieving the need for lexicographic sorting of barcodes, with the only pairwise comparison being the one between UMIs falling in the same "puck," thus allowing to collect the entire Digital Gene Expression matrix in linear time complexity.

The overall linear time complexity of FaST is confirmed by the benchmarks reported in Tables 1 and 2, which clearly show that FaST runs in a time that increases roughly linearly with the number of reads processed.

## FaST can quickly process large amounts of data with minimal hardware requirements

Aiming to obtain a lightweight software for the analysis of high resolution ST repurposing Illumina flowcells as RNA capture devices, I benchmarked my pipeline on a small workstation equipped with a 12th Gen Intel® Core™ i9-12900K processor (16 CPU cores, 8 of which are multithreading cores, for a total of 24 threads) and 32 Gb of RAM. Currently, such workstations can be purchased for around 1000 Euros each. To validate my pipeline I chose to analyze the samples listed in Table 1.

FaST could successfully analyze both sample GSM7990097 (596 millions reads; elapsed wall time: 47 min) and GSM7990102 (1.2 billion reads; elapsed wall time 1 h and 56 min).

To ascertain if the limited hardware could also accommodate analysis of significantly larger datasets, I also tested FaST on the recently published NovaScope dataset [3], finding that FaST can handle equally well these data (∼1.8 billion reads, datasets N3_B08C_v2b, N3_B08C_v2bu

**Table 1.** List of datasets used to validate FaST with overview of time requirements on a workstation*

| Dataset | Protocol | GEO # | Reads (M) | Wall time (mapping + DGE) | Wall time (segmentation) | Wall time (full analysis) |
|---|---|---|---|---|---|---|
| Mouse head e13 | OpenST | GSM7990097 | 569.6 | 0:32:54 | 0:15:40 | 0:48:34 |
| Metastatic Lymphnode S4 | OpenST | GSM7990102 (SRR27331444, SRR27331446) | 1.258 | 1:15:31 | 0:40:57 | 1:56:28 |
| Mouse liver | Seq-scope | N/A | 1781.9 | 1:59:12 | 1:01:22 | 3:00:42 |
| Mouse head | Stereo-seq | N/A | 776.9 | 1:08:23 | 1:26:07 | 2:34:30 |

*All these tests were carried out on a workstation equipped with an 12th Gen Intel® Core™ i9-12900K and 32 Gb of RAM.

**Table 2.** Benchmarking data for preliminary Tasks performed by FaST on a workstation*

| Task | Fastq file size (Gb) | Wall time (mm:ss) | User time (s) | %CPU | RAM (max, Gb) |
|---|---|---|---|---|---|
| Extract barcode coordinates for entire tile** | 400 | 3:21:50 | 24 125.16 | 224% | 0.005 760 |
| Generate reference (hg38) | N/A | 14:08 | 9817.73 | 1170% | 26.9 |
| Generate reference (mm39) | N/A | 9.44 | 5263.40 | 911% | 26.9 |

*All these tests were carried out on a workstation equipped with an 12th Gen Intel® Core™ i9-12900K and 32 Gb of RAM.
**This task is performed only once to obtain a barcode map of an entire S4 flowcell, sufficient to perform at least 100 different experiments.

N3_B08C_v2c, N3_B08C_v2cu) with an elapsed wall time 3 h and 46 min and a memory footprint of 27.8 Gb (Table 1 and Supplementary Table S1).

To compare FaST with existing alternative pipelines, I compared the performance of FaST, NovaScope [3], and spacemake [17] in an high-performance computing environment. This comparison has been limited to the mapping step, as neither spacemake nor NovaScope perform cell segmentation of data, which is performed instead by the OpenST [4] python package or the Ficture [9] package, respectively.

In an HPC environment FaST proved to run slightly less fast (this is expected, as single processors on HPC tend to have a lower frequency compared to small workstations), with a wall time of 1 h and 44 min for sample GSM7990097 (memory footprint of 23.4 Gb), 1 h and 44 min for seq-scope dataset N3_B08C_v2cu (memory footprint of 23.2 Gb), and 2 h and 25 min for the mapping of sample GSM7990102, (memory footprint of 28.9 Gb).

On the one hand, NovaScope required, for sample N3_B08C_v2cu, 2 h and 46 min (memory footprint 28.7 Gb). Larger datasets (Table 3) up to 1.3 billion reads highlight a roughly linear increase in the processing time for NovaScope (Supplementary Figs S1 and S2), while FaST, in an HPC environment, scales quite well, provided that data are split in independent fastq files.

On the other hand, spacemake required 96 Gb of RAM to successfully carry out mapping of sample GSM7990097 with a wall time of 16 h and 145.5 Gb for sample GSM7990102, with a wall time of 19 h and 40 min (Supplementary Figs S1 and S2). I did not test spacemake on larger datasets.

## FaST analysis yields the expected number of cells and the expected patterns within tissue

Full analysis of dataset GSM7990097 (open ST protocol, mouse e13 head) consisting of 596 million raw reads) with the FaST pipeline yielded a total of 55 615 cells, with a mean of 763 UMIs per cells, accounting for 42.4 Millions UMIs (Fig. 2). The overall organization is extremely similar to the one reported in the GEO deposited data through H&E segmentation, which yielded a comparable number of cells (49 043). The Leiden clustering of the cells suggests that the overall tissue organization is correctly represented (Fig. 2). In particular, in both plots the three major clusters identified in the forebrain (clusters 1 and 2 and 9 in panel A; clusters 1, 2, and 7 in panel D), have a similar pattern.

As an example of a human sample, I chose the section 4 of a metastatic lymph node (GEO GSM7990102). My pipeline yielded a total of 110 724 cells, which included 99 M UMIs, with an average of 893 UMIs per cell (Fig. 3). We compared our findings with the H&E segmented dataset deposited on GEO in anndata format. Applying the identical thresholds for analysis, H&E segmentation yielded 79 577 cells on the same sample, with a mean of 1130 counts/cell, accounting for 90 M UMIs. In this case FaST identified a significantly higher number of cells, most of which are localized in the cancerous part of the lymphnode (metastatic tumor cells, cluster "0" in Fig. 3A). In this case, a slightly better resolution seems to be achieved by FaST, which clearly identifies keratin pearls (cluster 5, Fig. 3A) as a specific cell type. On the other hand, the Spacemake/OpenST protocol only identifies a single major population of cancerous cells (cluster 1, Fig. 3B). However, the spacemake/OpenST protocol seems to capture a higher diversity in the lymphoid part of the tissue (clusters 1, 4, and 9 in Fig. 3 A; clusters 0, 4, 5, 6, and 7, Fig. 3B). UMAP projections (Fig. 3C and D) confirm that both protocols yield a neat segregation of epithelial cancer cells (panel C:upper left, clusters 0 and 5; panel D: right side, cluster 2) vs lymphoid cells. Plots of individual genes (Supplementary Fig. S3) highlight the overall agreement in the spatial pattern of gene expression between FaST and OpenST and confirm the above mentioned differences in the segmentation of the cancerous tissue and the lymph node between the two algorithms.

Finally, FaST also supports Stereo-seq datasets, which relies also on a barcode-based spatial mapping of captured transcript at submicrometric resolution. To validate FaST on Stereo-seq, I chose one of the demo dataset available through the STOmics website, a section of an adult mouse head. Results are depicted in Fig. 4. For this dataset, I used three of the six available libraries obtained from this sample, for a total of 0.9 billion reads. To analyse this dataset on a i9-12900k

**Table 3.** Comparison with existing alternatives on HPC* (mapping + DGE)

| | | Elapsed Time (hh:mm) | | | RSS (Gb) | | |
|---|---|---|---|---|---|---|---|
| Sample | Reads (M) | FaST | Spacemake | NovaScope | FaST | Spacemake | NovaScope |
| NovaScope N3_B08C_v2cu | 466.5 | 00:38 | N/A | 02:46 | 20.4 | N/A | 28.7 |
| Mouse head e13 | 569.6 | 01:01 | 16:08 | N/A | 21.0 | 96.4 | N/A |
| Metastatic lymphnode S4 | 782.7 | 01:44 | 19:39 | N/A | 23.2 | 145.5 | N/A |
| NovaScope N3_B08C_v2 b + cu | 922.1 | 00:45 | N/A | 04:58 | 20.6 | N/A | 46.3 |
| NovaScope N3_B08C_v2 b + bu + cu | 1353.9 | 01:00 | N/A | 05:39 | 20.7 | N/A | 61.6 |

*All tests were carried out on a HPC consisting of 12 nodes, each equipped with two AMD EPYC 7452 processors (32-core – 64 thread each) and 256 Gb of RAM. Jobs were allocated 32 cores on a single node. FaST was tested on all datasets. Spacemake was tested on Open-ST protocol datasets. NovaScope was tested on seq-scope protocol datasets.

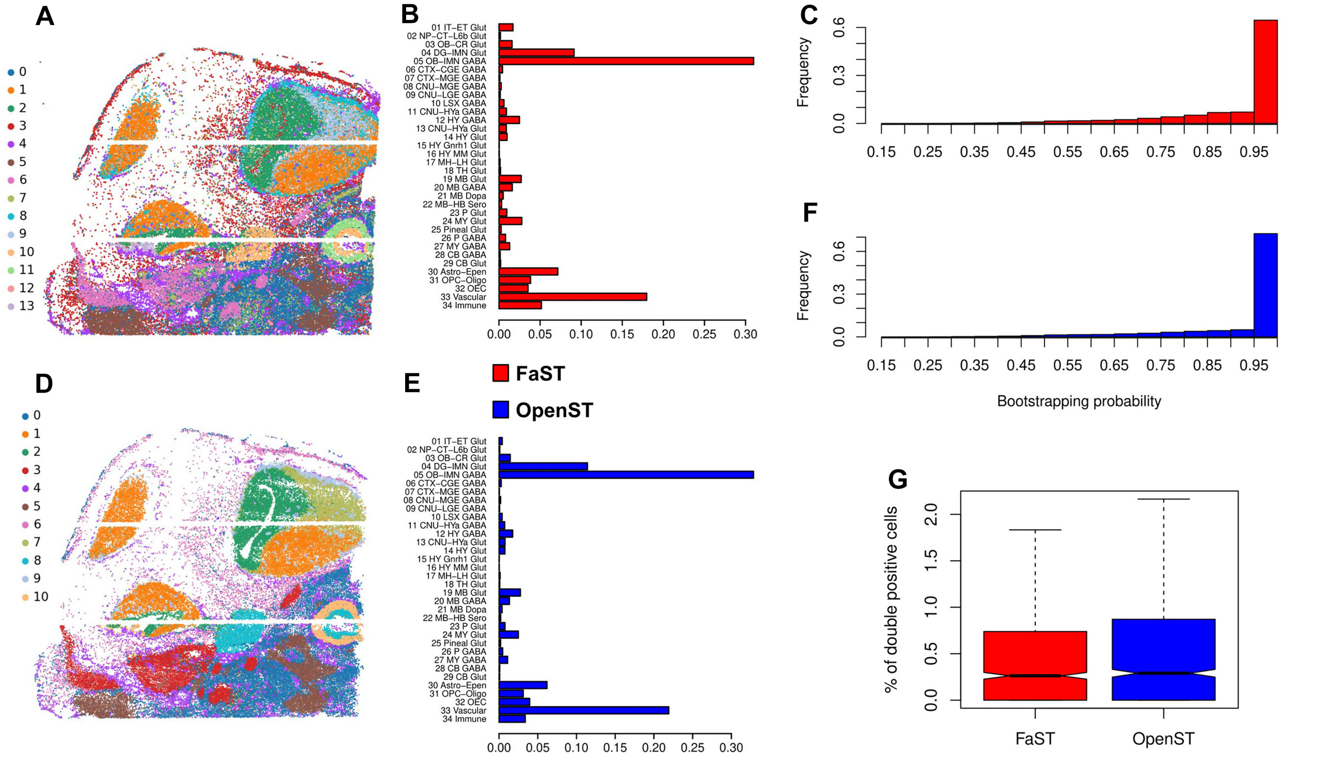

**Figure 2.** Analysis of sample GSM7990097 (mouse embryo head) using FaST (**A–C**) compared to GEO deposited processed data (**D–F**). (A) and (D): Segmented gene expression in anndata format was used to cluster "pseudo-cells" using scanpy with default parameters. Clustered cells were plotted in spatial coordinates. Panels (B) and (E): Composition of the sample as assessed by label transfer from the Allen Brain Atlas, mapping to the "class" category of the Allen Brain Atlas is reported. (C) and (F): Bootstrapping probability for the classification at the "subclass" level of the Hierachical tree of the Allen Brain Atlas (338 subclasses) for each cell in the datased. (FaST, panels A and C; Spacemake/OpenST panels B and D). (**G**) Percentage of cells with co-occurrences of putatively incompatible marker genes. Pairwise analysis of 62 marker genes was performed.

equipped workstation with 32 Gb of RAM, FaST required 2 h and 34 min, including preprocessing, mapping, DGE data summarization and segmentation.

## RNA segmented pseudo-cells recapitulate the tissue composition observed with H&E segmentation and scRNA-seq

FaST relies on RNA based segmentation which yields "pseudo cells." To assess the reliability of this method we looked at the clustering of the pseudo-cells as compared to the clustering of the H&E segmented cells, as obtained by the OpenST pipeline. The number of pseudo cells obtained by FaST is roughly comparable to the one obtained by H&E segmentation on both datasets tested. Importantly, the spatial organization of the cell clusters (obtained on scanpy, using default parameters) highlights that, with equal parameters, pseudo-cells yield a slightly larger number of clusters compared to H&E segmented cells. The spatial organization of the clusters observed with FaST mirrors the spatial organization of the clusters observed with OpenST, strongly suggesting that RNA segmented pseudo cells are a very close proxy for H&E segmented cells.

To further validate that the segmented cells obtained with FaST are equivalent to the ones predicted by OpenST, I projected both dataset onto the mouse Allen Brain Cell Atlas [18] using the online tool MapMyCells. This mapping is performed at single cell level, thus allowing to assess whether the segmented cells yielded by the two different methods are comparable. Firstly, I compared how the cells obtained by the two

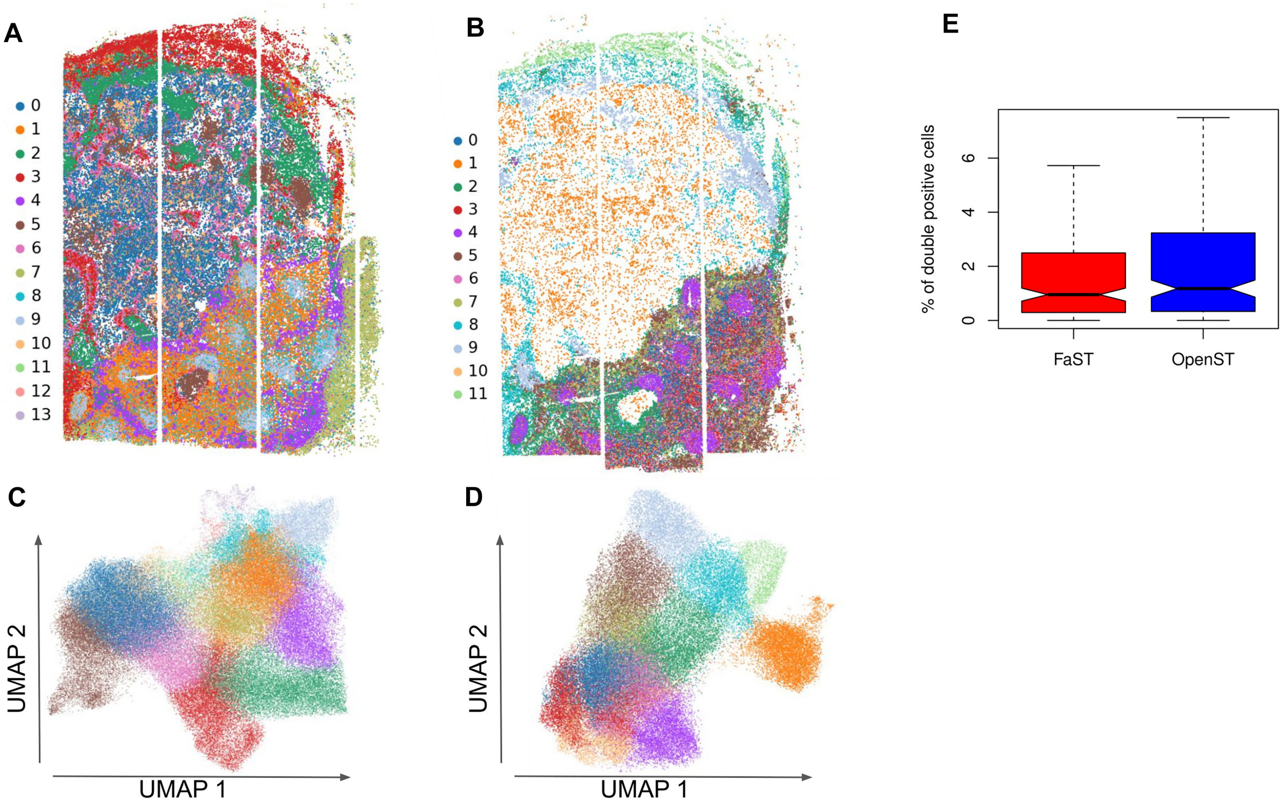

**Figure 3.** Analysis of sample GSM8990102 (metastatic lymphnode section #4) using FaST (**A** and **C**) compared to GEO deposited processed data from the same sample. (A) Spatial gene expression was reconstructed using FaST. Segmented gene expression in anndata format was used to cluster "pseudo-cells" using scanpy with default parameters. Clustered cells were plotted in spatial coordinates. (**B**) Segmented gene expression in anndata format (GEO) was used to cluster cells using scanpy with default parameters. Clustered cells were plotted in spatial coordinate. (C) UMAP plot of pseudo-cells obtained with FaST. (**D**) UMAP plot of segmented cells from GEO processed data (GSM8990102). Cluster colors do not correspond to the same cell types in the two different protocols. Colors were automatically assigned based on the abundance of each cluster in each output (FaST, panels A and C; Spacemake/OpenST panels B and D). (**E**) Percentage of cells double positive to markers specific of different cell types in FaST and OpenST segmented cells.

methods mapped to the 34 classes defined in the Atlas. The data reported in Fig. 2B and E show a very similar composition of the datasets. In fact, a Chi-squared test to assess whether any significant difference exists between the two distributions yielded a *P*-value of 0.2, suggesting that the hypothesis of any significant difference should be rejected. I then wanted to check how the cells were projected one level deeper, to the 338 different subclasses listed in the Atlas. In this case, I am reporting the Bootstrapping probability computed by the "MapMyCells" algorithm for the two datasets. A high Bootstrapping probability is an indication of a high confidence in the assignment of a "cell" to a subclass, and is expected to correlate with the quality of the segmentation. Also in this case, both datasets show a similar pattern, with overall very high bootstrapping probability scores (Fig. 2C and F), suggesting that in both cases the label transfer and the underlying segmentation were successful.

I next asked whether RNA based segmentation is prone to possible artifacts. A major concern in the field of ST cell segmentation is the generation of segmented cells consisting of fragments from different neighboring cells. To address this issue, I have systematically computed the pairwise co-occurrences of 62 different markers for 62 different cell types chosen from the Panglao Database [19], for a total of 1953 pairwise comparisons. For each cell type possibly present in the embryonic mouse head, I manually picked the best marker based on specificity and sensitivity scores reported in the PanglaoDB. The list of these 62 genes is available in Supplementary Table S2. As reported in Fig. 2G, both algorithms yielded a very low percentage of cells which were positive for incompatible marker genes: FaST yields an average rate of 0.87% double positives compared to 1.01% observed for OpenST output.

A similar analysis was performed on the lymphnode sample, using a set of 22 markers that I chose from the PanglaoDB manually picking the marker with the highest score for cell types belonging to the immune system or to the skin (Supplementary Table S3).

The results are reported in Fig. 3E, and, consistently with what I observed in the mouse head sample, both methods yield good segmentations, although FaST proves to be slightly less prone to segment cells which are positive for putatively incompatible markers. In this case the difference attains a borderline statistical significance according to the one tailed Wilcoxon sum rank test (*P*-value: 0.0317).

These data suggest that both H&E and RNA segmentation are very close to the "ground truth" and that FaST output does not contain a fraction of inappropriately segmented cells higher than the ones obtained with H&E staining based cell segmentation.

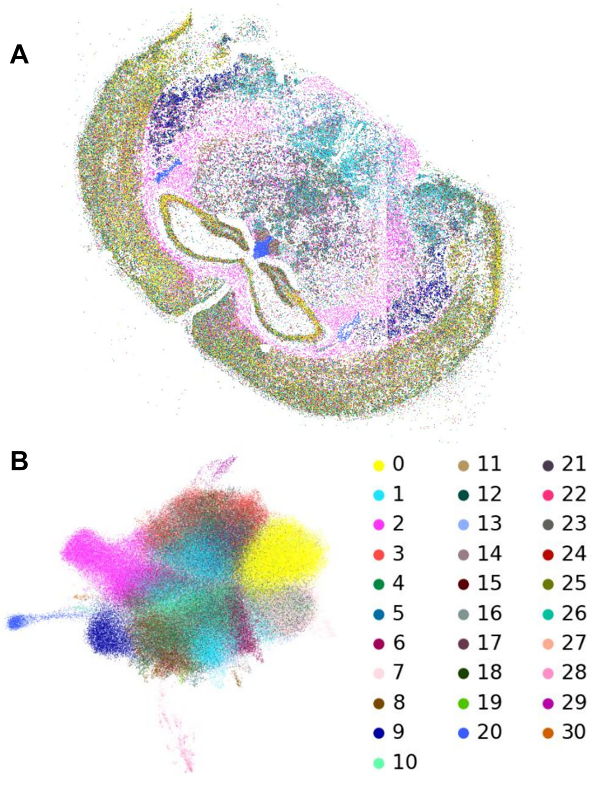

**Figure 4.** Analysis of the "mouse head" demo Stereo-seq dataset (STOmics, #C04042E3). (**A**) Spatial mapping of clustered segmented cells. (**B**) UMAP projection of the clusters identified.

## Discussion

The method I am reporting allows fast analysis of ST datasets corresponding to a tissue section with an area of ∼15 mm$^2$ in ∼1 h, significantly outperforming existing softwares both in terms of time and memory requirements, thus allowing analysis of such datasets using a small workstation.

Cell segmentation in the context of ST is still an active field of research, with several different approaches proposed thus far. It is worth mentioning that even tissue staining cell segmentation algorithms mainly aim at automating an otherwise very low-throughput task, yet all imaging based cell segmentation algorithms have a tradeoff in terms of accuracy. This is clearly reported in independent systematic comparative assessments of the currently available state of the art methods for staining-driven cell segmentation [20, 21]. These studies clearly show that accuracy of these methods is generally good and certainly very useful, but still not identical to the "ground truth."

The FaST pipeline relies on the spateo-release package [8] to perform a quick and informative cell segmentation based on RNA molecule mappings. The data presented show that, although a perfect overlap between H&E guided and RNA based segmentation is not observed, RNA segmentation faithfully recapitulates the cell types and localizations observed with image-based cell segmentation. The data I report also rule out the possibility that FaST might be more prone to "undersegmentation" artifacts (as assessed by counting cells double positive for incompatible marker genes) compared to existing softwares. The accuracy observed when mapping RNA segmented cells to the AllenBrain Atlas further support the reliability of the output of FaST.

Based on these considerations, I foresee that, by relieving the requirement for tissue staining and image capture and analysis, RNA based segmentation might soon attain sufficient accuracy to replace image based segmentation in most ST data analyses.

Indeed, RNA segmentation might have specific advantages in high resolution ST, as RNA segmented cells share the same coordinates of the barcodes naturally.

FaST pipeline however is modular, thus allowing users to only run mapping and DGE steps and afterwards continue analysis using the preferred (either staining-based or RNA-based) cell segmentation model instead of spateo-release.

Currently, FaST can handle ST protocols relying on either the Illumina flow or the BGI Stereo-seq cells as RNA capture devices. Possible future improvements may include extension to other ST platforms with submicrometer resolution.

FaST was not specifically designed to yield high performance in HPC environments (where the single processors generally have a lower performance compared to the processors of small workstations), however, given that in ST analysis setting the chronological sequence of the different steps of the analysis is relevant, and FaST takes advantage of parallelization whenever possible, my benchmarks confirm that FaST is in fact faster than other existing HPC designed ST analysis softwares also in an HPC computing environment.

## Acknowledgements

I would like to express my gratitude to Elena Splendiani for introducing me to OpenST protocol and spatial transcriptomics. I also want to thank Gianluca Moretti for stimulating discussion and constant feedback which greatly helped me to improve FaST.

*Author contributions*: Valerio Fulci (Conceptualization [lead], Data curation [lead], Investigation [lead], Methodology [lead], Software [lead], Validation [lead], Writing—original draft [lead], Writing—review & editing [lead])

## Supplementary data

Supplementary data is available at NAR Genomics & Bioinformatics online.

## Conflict of interest

None declared.

## Funding

This work was not supported by a dedicated financial support.

## Data availability

OpenST data are available through GEO (GSE251926) NovaScope data were obtained from this website https://seqscope.github.io/NovaScope/getting_started/access_data/). To obtain the 1.8 billion reads dataset the following samples were pooled together: N3_B08C_v2b; N3_B08C_v2bu; N3_B08C_v2a; N3_B08C_v2cu. FaST is available on github: https://github.com/flcvlr/FaST

A freeze of the version 0.4.1 of FaST is available on zenodo: https://doi.org/10.5281/zenodo.14532553

Genomic reference files were downloaded from GENCODE [16]. Version 46 (GRCh38) was used for human samples; version M36 (GRCm39) was used for mouse samples.

Human (URS0000ABD7E8_9606) and mouse (URS00026C81C6_10 090) 45S rRNA sequences were obtained from RNA central [22]. PhiX genome sequence (NC_001422.1) was obtained from NCBI Nucleotide.

## References

1. Moses L, Pachter L. Museum of spatial transcriptomics. *Nat Methods* 2022;**19**:534–46. https://doi.org/10.1038/s41592-022-01409-2
2. Cho C-S, Xi J, Si Y *et al.* Microscopic examination of spatial transcriptome using Seq-Scope. *Cell* 2021;**184**:3559–72. https://doi.org/10.1016/j.cell.2021.05.010
3. Kim Y, Cheng W, Cho C-S *et al.* Seq-scope protocol: repurposing illumina sequencing flow cells for high-resolution spatial transcriptomics. *Nature Protocols* 2025;**20**:643–89. http://dx.doi.org/10.1038/s41596-024-01065-0
4. Schott M, León-Periñán D, Splendiani E *et al.* Open-ST: high-resolution spatial transcriptomics in 3D. *Cell* 2024;**187**:3953–72. https://doi.org/10.1016/j.cell.2024.05.055
5. Chen A, Liao S, Cheng M *et al.* Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell* 2022;**185**:1777–92. https://doi.org/10.1016/j.cell.2022.04.003
6. Petukhov V, Xu RJ, Soldatov RA *et al.* Cell segmentation in imaging-based spatial transcriptomics. *Nat Biotechnol* 2022;**40**:345–54. https://doi.org/10.1038/s41587-021-01044-w
7. Wang Y, Wang W, Liu D *et al.* GeneSegNet: a deep learning framework for cell segmentation by integrating gene expression and imaging. *Genome Biol* 2023;**24**:235. https://doi.org/10.1186/s13059-023-03054-0
8. Qiu X, Zhu DY, Lu Y *et al.* Spatiotemporal modeling of molecular holograms. *Cell* 2024;**187**;7351–73. https://doi.org/10.1016/j.cell.2024.10.011
9. Si Y, Lee C, Hwang Y *et al.* FICTURE: scalable segmentation-free analysis of submicron-resolution spatial transcriptomics. *Nat Methods* 2024;**21**:1843–54. https://doi.org/10.1038/s41592-024-02415-2
10. Virshup I, Rybakov S, Theis FJ *et al.* anndata: annotated data. bioRxiv, https://doi.org/10.1101/2021.12.16.473007, 19 December 2021, preprint: not peer reviewed.
11. Hao Y, Stuart T, Kowalski MH *et al.* Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat Biotechnol* 2024;**42**:293–304. https://doi.org/10.1038/s41587-023-01767-y
12. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018;**19**:15. https://doi.org/10.1186/s13059-017-1382-0
13. Dobin A, Davis CA, Schlesinger F *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinforma Oxf Engl* 2013;**29**:15–21. https://doi.org/10.1093/bioinformatics/bts635
14. Fazal FM, Han S, Parker KR *et al.* Atlas of subcellular RNA localization revealed by APEX-Seq. *Cell* 2019;**178**:473–90. https://doi.org/10.1016/j.cell.2019.05.027
15. Dobin A, Gingeras TR. Mapping RNA-seq reads with STAR. *Curr Protoc Bioinformatics* 2015;**51**:11.14.1–11.14.19. https://doi.org/10.1002/0471250953.bi1114s51
16. Frankish A, Diekhans M, Ferreira A-M *et al.* GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res* 2019;**47**:D766–73. https://doi.org/10.1093/nar/gky955
17. Sztanka-Toth TR, Jens M, Karaiskos N *et al.* Spacemake: processing and analysis of large-scale spatial transcriptomics data. *GigaScience* 2022;**11**:giac064. https://doi.org/10.1093/gigascience/giac064
18. Yao Z, van Velthoven CTJ, Kunst M *et al.* A high-resolution transcriptomic and spatial atlas of cell types in the whole mouse brain. *Nature* 2023;**624**:317–32. https://doi.org/10.1038/s41586-023-06812-z
19. Franzén O, Gan L-M, Björkegren JLM. PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database* 2019;**2019**:baz046. https://doi.org/10.1093/database/baz046
20. Goyal V, Schaub NJ, Voss TC *et al.* Unbiased image segmentation assessment toolkit for quantitative differentiation of state-of-the-art algorithms and pipelines. *BMC Bioinf* 2023;**24**:388. https://doi.org/10.1186/s12859-023-05486-8
21. Wang Y, Zhao J, Xu H *et al.* A systematic evaluation of computation methods for cell segmentation. *Brief Bioinform* 2024; **25**:bbae407. https://doi.org/10.1093/bib/bbae407
22. The RNAcentral Consortium RNAcentral: a hub of information for non-coding RNA sequences. *Nucleic Acids Res* 2019;**47**:D221–9. https://doi.org/10.1093/nar/gky1034

**Received:** August 5, 2024. **Revised:** February 27, 2025. **Editorial Decision:** April 2, 2025. **Accepted:** April 3, 2025