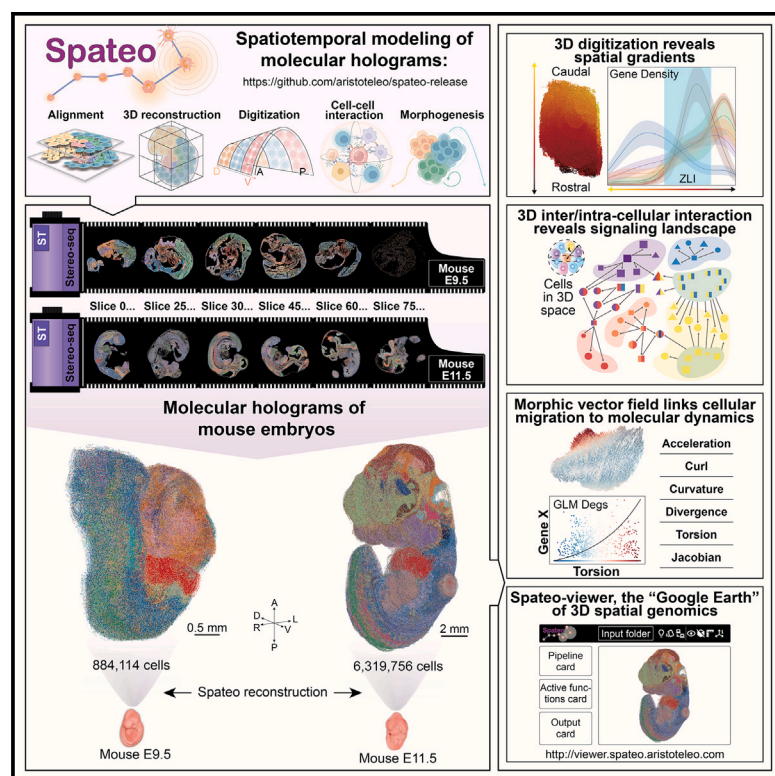# Spatiotemporal modeling of molecular holograms

## Graphical abstract



## Authors

Xiaojie Qiu, Daniel Y. Zhu, Yifan Lu, ..., Xun Xu, Shiping Liu, Yinqi Bai

## Correspondence

xiaojie@stanford.edu (X.Q.),
jyma2010@gmail.com (J.M.),
xuxun@genomics.cn (X.X.),
liushiping@genomics.cn (S.L.),
baiyinqi@genomics.com (Y.B.)

## In brief

Spateo is a comprehensive framework for 3D reconstruction and characterization of spatial gradients and cellular interactions at whole-organ and embryo levels, using spatial transcriptomics. Importantly, Spateo also introduces morphometric vector field analyses that connect macroscopic cell morphogenesis with microscopic molecular dynamics.

## Highlights

- Spateo reconstructs 3D maps and models spatiotemporal dynamics at the whole-embryo scale

- Reconstruction of the molecular holograms of mouse embryos at stages E9.5 and E11.5

- 3D digitization and cell communication reveal spatial gradients and signaling pathways

- Morphic vector field predicts molecular drivers of heart asymmetrical morphogenesis

# Cell

**CellPress**
OPEN ACCESS

## Article

# Spatiotemporal modeling of molecular holograms

Xiaojie Qiu,[1,7,8,28,29,30,*] Daniel Y. Zhu,[3,29] Yifan Lu,[1,7,8,9,29] Jiajun Yao,[2,4,10,29] Zehua Jing,[2,4,11,29] Kyung Hoi Min (Joseph),[12,29] Mengnan Cheng,[2,6,29] Hailin Pan,[6] Lulu Zuo,[6] Samuel King,[13] Qi Fang,[2,6] Huiwen Zheng,[2,11] Mingyue Wang,[2,14] Shuai Wang,[2,11] Qingquan Zhang,[25] Sichao Yu,[5] Sha Liao,[6,17,18] Chao Liu,[15] Xinchao Wu,[2,4,16] Yiwei Lai,[6] Shijie Hao,[2] Zhewei Zhang,[2,4,16] Liang Wu,[18] Yong Zhang,[15] Mei Li,[17] Zhencheng Tu,[2,11] Jinpei Lin,[2,4] Zhuoxuan Yang,[2,16] Yuxiang Li,[15] Ying Gu,[2,6,11] David Ellison,[27] Ao Chen,[6,17,18] Longqi Liu,[2,19,20] Jonathan S. Weissman,[5,22,23] Jiayi Ma,[9,*] Xun Xu,[2,11,21,*] Shiping Liu,[2,19,20,24,*] and Yinqi Bai[4,26,*]

[1]Department of Genetics, Stanford University School of Medicine, Stanford, CA, USA
[2]BGI Research, Hangzhou 310030, China
[3]Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA
[4]BGI Research, Sanya 572025, China
[5]Whitehead Institute for Biomedical Research, Cambridge, MA, USA
[6]BGI Research, Shenzhen 518083, China
[7]Basic Sciences and Engineering Initiative, Betty Irene Moore Children's Heart Center, Lucile Packard Children's Hospital, Stanford, CA, USA
[8]Department of Computer Science, Stanford University, Stanford, CA 94305, USA
[9]Electronic Information School, Wuhan University, Wuhan 430072, China
[10]College of Life Sciences, Northwest University, Xi'an 710069, China
[11]College of Life Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[12]Ginkgo Bioworks, The Innovation and Design Building, Boston, MA 02210, USA
[13]Department of Bioengineering, Stanford University School of Medicine, Stanford, CA, USA
[14]Shenzhen Key Laboratory of Gene Regulation and Systems Biology, School of Life Sciences, Southern University of Science and Technology, Shenzhen 518055, China
[15]BGI Research, Wuhan 430074, China
[16]School of Life Sciences, Southern University of Science and Technology, Shenzhen 518055, China
[17]STOmics Tech Co., Ltd, Shenzhen 518083, China
[18]BGI Research, Chongqing 401329, China
[19]Shenzhen Bay Laboratory, Shenzhen 518132, China
[20]Shenzhen Key Laboratory of Single-Cell Omics, BGI-Shenzhen, Shenzhen 518120, China
[21]Guangdong Provincial Key Laboratory of Genome Read and Write, BGI-Shenzhen, Shenzhen 518120, China
[22]Department of Biology and Howard Hughes Medical Institute, Massachusetts Institute of Technology, Cambridge, MA, USA
[23]Koch Institute for Integrative Cancer Research at MIT, MIT, Cambridge, MA, USA
[24]The Guangdong-Hong Kong Joint Laboratory on Immunological and Genetic Kidney Diseases, Guangzhou, Guangdong, China
[25]Department of Medicine, Division of Cardiology, University of California, San Diego, La Jolla, CA, USA
[26]Hainan Technology Innovation Center for Marine Biological Resources Utilization (Preparatory Period), BGI Research, Sanya 572025, China
[27]Infrastructure Solution Group, Lenovo
[28]Stanford Cardiovascular Institute, Stanford University, Stanford, CA, USA
[29]These authors contributed equally
[30]Lead contact
*Correspondence: xiaojie@stanford.edu (X.Q.), jyma2010@gmail.com (J.M.), xuxun@genomics.cn (X.X.), liushiping@genomics.cn (S.L.), baiyinqi@genomics.com (Y.B.)
https://doi.org/10.1016/j.cell.2024.10.011

## SUMMARY

Quantifying spatiotemporal dynamics during embryogenesis is crucial for understanding congenital diseases. We developed Spateo (https://github.com/aristoteo/spateo-release), a 3D spatiotemporal modeling framework, and applied it to a 3D mouse embryogenesis atlas at E9.5 and E11.5, capturing eight million cells. Spateo enables scalable, partial, non-rigid alignment, multi-slice refinement, and mesh correction to create molecular holograms of whole embryos. It introduces digitization methods to uncover multi-level biology from subcellular to whole organ, identifying expression gradients along orthogonal axes of emergent 3D structures, e.g., secondary organizers such as midbrain-hindbrain boundary (MHB). Spateo further jointly models intercellular and intracellular interaction to dissect signaling landscapes in 3D structures, including the zona limitans intrathalamica (ZLI). Lastly, Spateo introduces "morphometric vector fields" of cell migration and integrates spatial differential geometry to unveil molecular programs underlying asymmetrical murine heart organogenesis and others, bridging macroscopic changes with molecular dynamics. Thus, Spateo enables the study of organ ecology at a molecular level in 3D space over time.

**Cell**
Article

## INTRODUCTION

Knowledge of the precise spatial and temporal orchestration of gene expression and cell-cell interaction (CCI) at whole-embryo level during mammalian organogenesis will significantly advance developmental biology, and it holds enormous potential for human health.[1] Single-cell RNA sequencing (scRNA-seq) has enabled the creation of time-resolved cell atlases of *C. elegans*,[2,3] *Drosophila*,[4,5] mouse,[6,7] and human[8,9] embryogenesis. However, scRNA-seq approaches dissociate embryos into single cells, leading to a critical loss of spatial information.[10,11] The recent emergence of spatial transcriptomics (ST) methods[12–19] addressed this limitation, enabling spatiotemporal profiling of embryogenesis, yet whole-embryo-scale spatiotemporal studies for mammalian species pose great challenges; previously, these studies have been limited to a single time point[20] or early time points[21] or to a single organ,[22] given the limited field of view (FOV) of most ST platforms, but only until now for whole mouse embryos at E9.5 and E11.5.[23]

Furthermore, while tremendous progress has been made in computational analyses of ST with the development of methods for 3D reconstruction,[24] spatial domain digitization,[25] and CCI,[26–30] various challenges prevent their use for embryo-scale and 3D ST datasets. For example, existing methods for 3D alignment, such as probabilistic alignment of ST experiments (PASTE),[24] cannot align embryonic sections at large scale while accounting for missing regions across tissue sections and tissue deformation introduced during sample preparation. Additionally, existing methods for spatial domain digitization[25] are limited to 2D shapes but not to any 3D objects, and CCI models[26–30] are not designed to describe the many intercellular and intracellular interactions operating at different scales in 3D biological structures. Last but not least, modeling 4D spatiotemporal dynamics (e.g., morphogenesis) from time-resolved 3D ST data remains an open challenge.

To address these unmet gaps, in this study, we developed a unified framework, Spateo, which achieves 3D spatiotemporal modeling of the molecular hologram of the whole mouse embryos, leveraging a time-resolved whole mouse embryo cell atlas with a total of more than eight million cells for two critical time points (E9.5 and E11.5) of mouse organogenesis.[23] Note that the experimental protocol of the improved 3D ST profiling with the automated serial block-face imaging (SBFI) approach and the full characterization of this massive reference dataset will be reported in a separate study.[23] This study focuses on investigating several emergent 3D structures[1] that are characterized with rapid formation and maturation at this stage, including the four-chamber heart, spinal cord, and secondary organizers such as the zona limitans intrathalamica (ZLI) and midbrain-hindbrain boundary (MHB), which have specific gene expression patterns and signaling landscapes that govern cell fate of surrounding spatial regions.

Spateo features a coherent and unique set of 3D aware analytical approaches. First, Spateo introduces a scalable and powerful 3D reconstruction algorithm that is capable of accurately reconstructing 3D whole mouse embryos by allowing non-rigid, partial alignment; multi-slice refinement; and mesh correction using Gaussian process, variational Bayesian inference, and others. On the new 3D whole mouse datasets and several other 3D datasets, Spateo consistently outperforms PASTE,[24] PASTE2,[31] multi-omics single-cell optimal transport (Moscot),[32] spatial-linked alignment tool (SLAT),[33] STAlign,[34] and SPACEL[35] in terms of accuracy, robustness, and efficiency. Spateo also features a multi-scale spatial domain analyses framework applicable to defining specific spatial axes and study levels of biology ranging from single-cell to embryo level, using a generalized digitization approach by solving a potential function equation of spatial graph of single cells. We applied this approach to characterize gene expression variation across orthogonal axes of the ZLI and spinal cord, which would not otherwise be possible with 2D analyses. We identified a repertoire of signaling and transcription factors (TFs) such as *Slit2* and *Dbx1*, many of which regulate ZLI and spinal cord maturation. To probe the underlying cell-cell communication and intracellular signaling, we further developed an integrative framework that allows for integrative modeling of intercellular/intracellular interactions. We leveraged this framework to uncover and characterize the unique molecular networks underlying the organization of MHB and ZLI, involving key ligands of the *Wnt* family, *Bmp* family, and *Fgf8* and TFs such as *Foxd1*, *Cited*, and *Id1*, among others.

To connect macroscopic cellular morphogenesis with underlying microscopic molecular pathways, we further generalized the 3D alignment framework to allow for learning of a morphometric vector field that accurately predicts the 4D spatiotemporal migration path of cardiac cells from the 3D reconstructed heart, the earliest organ to appear during embryogenesis. We found that the asymmetrical migration of the heart may be related to atrium- and ventricle-specific *Notch* signaling; differentiation factors such as *Tbx2*, *Id2*, and *Tbx20*; or morphogenesis factors such as *Angpt1*, *Pitx2*, and *Hey2*. Going beyond spatiotemporal modeling of a single organ of the heart to multiple organs jointly, we next revealed the morphometric convergence of the *Drosophila* midgut and expansion of the hindgut and salivary gland, which is further used to reveal putative morphogenetic factors responsible for germband retraction, such as *otp*, *Abd-B*, and others. Lastly, to facilitate interactive exploration of massive, complex 3D datasets, we also developed Spateo-viewer, a versatile and powerful tool for 3D data visualization and manipulation that is freely accessible at http://viewer.spateo.aristoteleo.com/. Collectively, these integrative methods allow for deep investigations of 3D structures such as the murine secondary organizers in the brain (ZLI and MHB), the spinal cord, four-chamber heart, and whole *Drosophila* germband at breadth and resolution.

Spateo facilitates a shift in single-cell analysis—from the conventional, reductionist cell-centric focus to embracing the tissue, organ, and embryo as a whole—allowing for ultra-fine, multidimensional spatial and temporal examination of molecular mechanisms. Spateo is generally applicable to any sequencing-based or imaging-based ST readouts and can be used in conjunction with Dynamo,[36] a general framework for RNA velocity vector field analyses, to enable quantitative and predictive analyses of spatiotemporal kinetics of cell fate transitions. Extensive tutorials, workflows, and documentation are provided at https://spateo-release.readthedocs.io/en/latest/, and the open-source
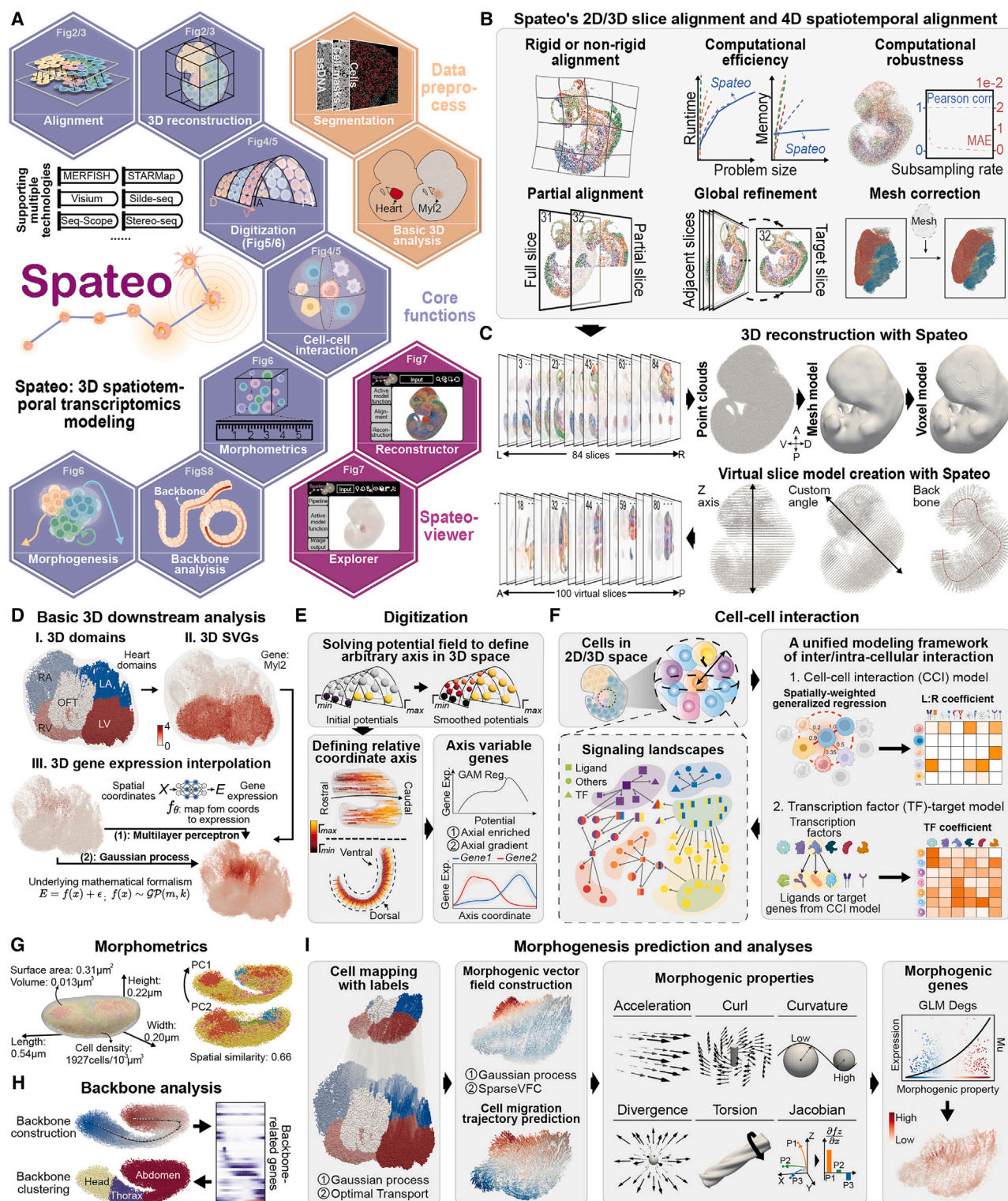
**Figure 1. Spateo, a versatile tool for 3D spatiotemporal transcriptomics modeling at single-cell resolution and whole-embryo scale, demonstrated on the mouse and *Drosophila* embryos**

(A) Three major functional modules of Spateo that enables 3D spatiotemporal transcriptomics modeling: data preprocessing (single-cell segmentation, basic 3D data exploration); core functions with a set of novel 3D aware modeling approaches (3D alignment, reconstruction, digitization, cell-cell interaction,

*(legend continued on next page)*

toolkit can be found at https://github.com/aristoteleo/spateo-release, where community contributions are welcome.

## RESULTS

### Spateo: A versatile tool for multi-level spatiotemporal modeling of 3D spatial transcriptomics

To embrace the arrival of the era of 3D ST, we developed a versatile tool for advanced 3D spatiotemporal modeling at whole-embryo scale with single-cell resolution. The Spateo tool chain has three major functional modules: data preprocessing, core functions, and Spateo-viewer (Figure 1A; Table S1). Spateo starts with a flexible application programming interface (API) that reads in different data formats from different 2D and 3D ST technologies (Figures 1A and S1A), which can then be converted into a single uniform data structure. The data preprocessing module consists mainly of functionalities for single-cell segmentation and basic 3D ST dataset exploration. Spateo's main core function module implements a comprehensive set of 3D spatiotemporal modeling approaches, introduced below, and these will be discussed in detail in subsequent sections. Lastly, we developed Spateo-viewer, the "Google Earth"-like web browser (http://viewer.spateo.aristoteleo.com/) for ST that allows for intuitive and interactive 3D data exploration and analyses.

In this study, we leveraged a 3D whole mouse cell atlas from E9.5 and E11.5 embryos[23] (Figures S1 and S2; Table S2), with a large number of sections and cells, an overall smaller embryonic section interval, and comparable RNA capture, compared with existing datasets from earlier mouse embryonic stages[21] or human gastrulating embryo[20] (Figures S1A–S1C). Additionally, we also leveraged another 3D *Drosophila* embryo dataset to demonstrate the generality of this approach.[37] To annotate this massive 3D mouse ST dataset with a total of more than 8 million cells, we first co-embedded this dataset with a well-annotated scRNA-seq atlas[38] to a latent space,[39] followed by using a hierarchical approach to transfer the cell-type label from the annotated scRNA-seq dataset to the Stereo-seq dataset via a trained classifier.[40] Across all cell types for both E9.5 and E11.5, we found that the Stereo-seq dataset has a high correspondence of matched markers and similar cell-type fractions with the scRNA-seq references from Qiu et al. (2024)[41] (Figures S1D and S1E). Overall, we show that the spatial distribution of specific cell types such as cardiomyocytes; hepatocytes in E9.5 embryo dataset; intermediate neuronal progenitors and ependymal cells in E11.5 dataset; and corre-

sponding marker genes such as *Myh6*, *Afp*, *Eomes*, and *Spon1* are enriched in corresponding locations in the embryo (Figure S1F).

To enable 3D ST modeling, a major hurdle is a powerful approach to allow for accurate, scalable, and robust alignment of 2D slices into 3D structure at a whole-embryo scale. In Spateo, we developed a 3D alignment framework (Figure 1B, refer to Figures 2 and 3) that allows for rigid, non-rigid, partial, pairwise alignment; multi-slice refinement (aligning multiple slices jointly all at once); and mesh correction (enhancing 3D reconstruction with exterior reference shape mesh, such as the Allen Mouse Brain common coordinate framework or CCF v3[42]—while having high computational efficiency at the same time. Once the 3D slices are aligned, Spateo can generate a point cloud model from which it can create a surface mesh model via marching cube algorithm and voxel model by voxelization of the mesh. Once the mesh and voxel models are created, virtual slices from arbitrary angles can be generated (Figure 1C, refer to Figures 3 and 4). Next, a series of 3D aware modeling strategies from Spateo can be used in downstream analysis (Figures 1D–1I). For relatively straightforward 3D analyses, Spateo can, for example, be used to identify 3D spatial domains, detect spatially variable genes in 3D, and to interpolate gene expression in 3D space via either an MLP-based or a Gaussian process-based approach (Figure 1D, refer to Figure 6). Importantly, we developed a series of advanced 3D aware approaches in Spateo, including 3D domain/surface digitization (Figure 1E, refer to Figures 4 and 5), CCIs (Figure 1F, refer to Figures 4 and 5), morphometric and volumetric measurements (Figure 1G, refer to Figure 6), morphometric backbone (Figure 1H, refer to Figure 6), morphogenesis predictions (Figure 1I, refer to Figure 6), and an interactive 3D data browser with Spateo-viewer (Figure 1A, refer to Figure 7). We will extensively describe each approach in the following sections.

Taken together, Spateo establishes a versatile tool for multi-scale spatiotemporal transcriptomics modeling, applicable to whole *Drosophila* and, more importantly, mouse embryo-scale datasets (Figures 2–7).

### Efficient, non-rigid alignment, multi-slice refinement, and mesh correction approaches to reconstruct 3D molecular hologram of whole mouse embryo

The sequential slicing and subsequent ST profiling at the whole-embryo level offer us a unique opportunity to reconstruct the molecular hologram of the entire 3D embryo structure. However, conventional sectioning and downstream library preparation[19–22] can

---

morphometric, backbone, and morphogenesis analyses); and Spatio-viewer, the "Google earth" of 3D spatial transcriptomics. Relevant sections are noted for each approach.

(B and C) Spateo's scalable, flexible, and powerful alignment (B) and 3D reconstruction (C) framework (see also Figures 2 and 3).

(D) Spateo allows for various basic 3D spatial data exploration.

(E and F) The schematic of 3D digitization (E) and cell-cell interaction (F) method of Spateo (see also Figures 4 and 5).

(G) Morphometric analysis quantifies the surface area, volume, height, length, width, and cell density of a reconstructed 3D object (see also Figure 6).

(H) Backbone analyses that use a principal curve to define the backbone of a 3D object, followed by identifying genes that significantly vary as a function of the backbone, and clustering backbone into different spatial domains (see also Figure S8).

(I) Morphometric vector learning and differential geometry analyses (see also Figure 6).

(J) Spateo-viewer is an online or standalone browser that allows intuitive and interactive exploration of 3D spatial transcriptomics (Figure 7).

(E)–(I) are designed for more advanced 3D downstream analyses.
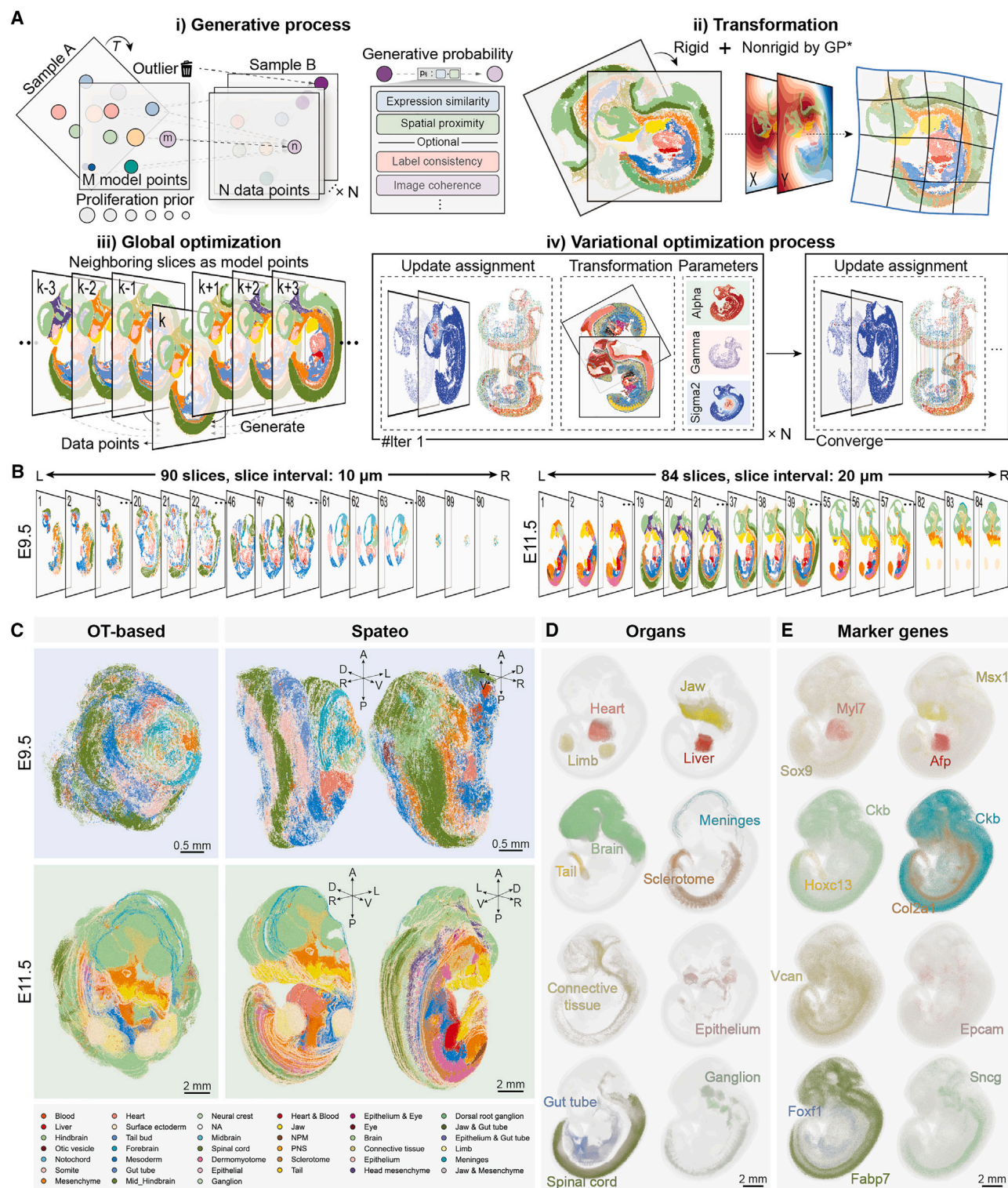
See also Figures S1 and S2.

**Figure 2. Spateo enables accurate, efficient, and scalable reconstruction of 3D molecular holograms of whole mouse embryos**

(A) The schematic of Spateo's probabilistic model and the associated variational optimization process for aligning spatial transcriptomic slices to create 3D whole-embryo models. For details of the 3D alignment model and optimization process, please refer to STAR Methods.

(B) Example mouse embryo slices at stages E9.5 and E11.5[23] after alignment using Spateo. Each point is a cell, colored by its region identity, with the region identity or spatial domain annotations obtained from Cheng et al.[23] The same as in (C) and (D).

*(legend continued on next page)*

rotate, transform, deform, and introduce missing regions in each profiled tissue section. While these issues are shown to be largely alleviated by the automated SBFI approach,[23] it is generally necessary to develop scalable and robust algorithms to reconstruct 3D structures to recover the relative spatial locations of single cells across different slices while allowing for local distortion within the same slice. Sophisticated methods such as PASTE[24] were recently developed to align spatial sections for 3D reconstruction. However, the unique features of whole-embryo datasets, such as the large number of sequential tissue sections (up to ~100 sections) and total cells (up to ~8 million cells), the missed regions, tissue deformation, and sectioning gap between consecutive slices, restrict the use of these approaches as they build upon rather inflexible formulation of all-cell to all-cell mapping across slices; rely on expensive optimal transport-based approaches; and perform sequential alignment of consecutive slices, which accumulate errors as the number of slices increases.

In Spateo, we introduced a powerful method that formulates the 3D alignment problem with a generative Gaussian process approach consisting of an initial pairwise alignment mode, followed by a multi-slice refinement mode to reconstruct 3D structures, scalable to the whole-embryo datasets with ~8 million cells, while addressing aforementioned challenges (Figure 2A). The generative process for pairwise alignment transforms the coordinate system from one sample (slice A) to another (slice B) (Figure 2Ai). We consider slice A as the model points that can generate the data points slice B through spatial transformation $\mathcal{T}$. We additionally include an outlier model to account for the commonly occurring partial overlapping between slices. The generative probabilities from model points to data points are very flexible and are expressed by both gene expression similarity and spatial proximity. Moreover, it can incorporate cell proliferation/apoptosis scores as a priori, such that more proliferative (apoptotic) cells will map to more cells with uniformly high (low) probability, and cells with the same label (such as cell type) and similar image features will be mapped across tissue sections. Furthermore, the generative model allows for both rigid and non-rigid transformation: where the rigid transformation is used to align the overall frame, while the non-rigid transformation aligns the local distortions, modeled by a Gaussian process[47] (Figure 2Aii). To reduce error accumulation during sequential alignment, we also designed a multi-slice refinement model by jointly considering multiple slices from the neighborhood of the slice of interest, either from the left or right, to refine the alignment (Figure 2Aiii). The underlying stochastic variational optimization process used by Spateo relies on coordinate ascent variational inference (CAVI)[48] to iteratively update the variational parameters (including the transformation as well as the posterior generative probabilities, etc.) until convergence, making the algorithm reliable, extremely fast and memory-efficient, and scalable to the whole mouse embryo dataset (Figure 2Aiv; Video S1; STAR Methods).

We next applied Spateo to the mouse embryo data,[23] reconstructing 3D molecular holograms of the entire mouse embryo at E9.5 and E11.5 (Figures 2B–2E). Specifically, we performed multi-slice refinement of the 90 slices and 84 slices from E9.5 and E11.5 embryos (Figure 2B), respectively, guided by an initial pairwise alignment across every two consecutive slices. Importantly, Spateo's alignment framework resulted in a noticeable improvement of the reconstructed 3D whole mouse embryo structures, compared with OT-based approaches (Figure 2C), revealing intricate 3D structures of every major organ as well as the spatial distribution of their corresponding marker genes within the whole embryo (Figures 2C–2E; Video S2, organs' annotations are token from Qiu et al.[23]).
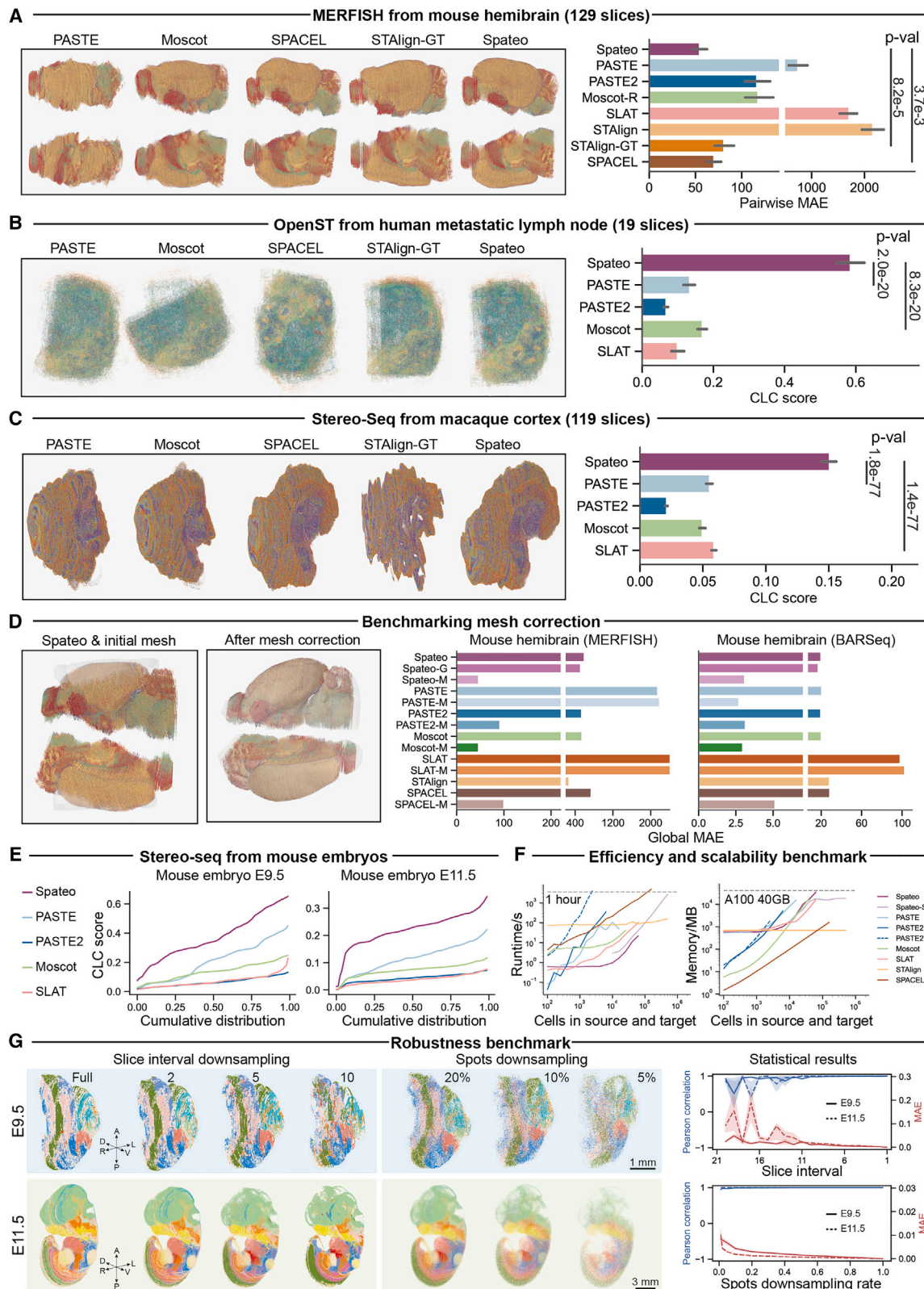
To qualitatively benchmark our 3D reconstruction ability, we first show that Spateo can reveal more intricate and smoother 3D embryo/tissue structures than PASTE[24] for human gastrulating embryo Stereo-seq data[20] and sc3D,[21] a 3D alignment method requiring cell-label information, for both E8.5 and E9.0 mouse embryo Slide-seq data[21] (Figures S3A and 3B). Next, we quantitatively benchmarked Spateo over alternative state-of-the-art algorithms (Figure 3), including PASTE,[24] PASTE2,[31] Moscot,[32] SLAT,[33] STAlign,[34] and SPACEL[35] on several 3D ST datasets with a considerable number of consecutive sections and cell numbers, i.e., mouse hemibrain MERFISH[37] (Figure 3A, 129 slices with 9.3 million cells) dataset, human metastatic lymph node OpenST[38] (Figure 3B, 19 slices with 1 million cells) dataset, macaque cortex Stereo-seq (Figure 3C, 119 slices with 30 million cells) dataset,[43] mouse forebrain hemisphere BARSeq[49] (Figure 3D, 40 slices with 1.2 million cells) dataset, and mouse embryos Stereo-seq[23] (Figure 3E, introduced above) dataset. These extensive benchmarks validate Spateo's leading edge in terms of pairwise alignment, multi-slice refinement, and mesh correction, either based on mean absolute error (MAE) when compared with the ground-truth Allen CCF v3 reference or on contextual label consistency score based on cell-type labels (see STAR Methods for details). We next evaluated the performance of Spateo as well as other methods on more intricate alignment cases such as partial alignment or non-rigid alignment. We first implemented a data simulation strategy using the STARMap Plus dataset (Figure S4Ai), where three different simulations are considered, i.e., non-rigid distortion, ratio crop, and manually crop (Figure S4Aii–S4Aiv). We then used this simulation strategy for three benchmarks: non-rigid, partial, and multi-slice refinement benchmarks. On the non-rigid alignment benchmark using the sagittal and well STARMap Plus datasets, Spateo had significantly lower pairwise MAE, which is consistently maintained under different distortion levels (Figure S4B, first row). When zooming into specific regions, we observed improved alignment across slides on intricate local structures

(C) Visualization of 3D reconstruction results of mouse embryos at stages E9.5 (top row) and E11.5 (bottom row), using the OT-based method (i.e., PASTE) and Spateo, respectively. The legend in the bottom is shared for both stages. Spateo but not OT-based methods accurately reconstruct 3D mouse embryo structures. Scale bars for E9.5 and E11.5 denote distances of 0.5 mm and 2 mm, respectively.

(D) The accurate 3D reconstruction by Spateo reveals intricate structures of every major organ of the whole embryo.

(E) Spatial 3D plot of the corresponding markers for each organ.

See also Figures S1 and S2.

**A** — MERFISH from mouse hemibrain (129 slices)

**B** — OpenST from human metastatic lymph node (19 slices)

**C** — Stereo-Seq from macaque cortex (119 slices)

**D** — Benchmarking mesh correction

**E** — Stereo-seq from mouse embryos

**F** — Efficiency and scalability benchmark

**G** — Robustness benchmark

*(legend on next page)*

(Figure S4B, second and third rows). Next, on the partial alignment benchmark (Figure S4C), while SPACEL, PASTE2, and STAlign tend to show apparent misalignment across slides, Spateo overcame this to correctly align overlapping regions while ignoring non-overlapping regions. Furthermore, Spateo consistently outperformed other methods in partial alignment under different overlapping ratios across sagittal and well datasets (Figure S4C). Finally, we demonstrated that Spateo's multi-slice refinement strategy can continue to improve the 3D reconstruction accuracy when multiple slices are involved (Figure S4D).

After establishing the accuracy of our algorithm, we further demonstrated that Spateo dramatically improves computational efficiency and scalability, compared with alternative tools (Figure 3F). Spateo is also robust to different downsampling strategies (Figure 3G), e.g., subsampling slices under different interval gaps or subsampling cells number at different ratios. Lastly, we validated the robustness of Spateo under a large range of values for all major parameters (Figure S4E).

In conclusion, leveraging its scalability, generative modeling framework that accounts for partial, non-rigid, multi-slice refinement, and mesh correction, Spateo is able to accurately create the 3D hologram of mouse embryos.

### A multi-scale, 3D-aware digitization and CCI framework to dissect intercellular and intracellular mechanisms of tissue organization

The whole 3D embryo (Figure 4A) dataset allows us to investigate emergent structures that were previously difficult to probe in their entirety, including the ZLI (Figure 5),[50,51] MHB (Figure S6), and spinal cord (Figure S6).[52] To comprehensively characterize 3D transcriptomic heterogeneity, powerful algorithms that can accommodate arbitrary 3D structures and enable detection of spatially polar genes across multiple scales (ranging from the subcellular level to the organ level) are critical. However, existing methods such as Belayer[25] are limited to 2D analyses and thus cannot be easily generalized to analyze the hollow 3D neural tube where the ZLI is located, the 3D ring of the ZLI itself, and the serpentine 3D spinal cord. Spateo introduces a digitization

method based on the concept of the graph potential function (STAR Methods) that empowers us to create arbitrary axes in any topologically complex 2D or 3D structure (Figure 4B, upper; STAR Methods).

Meanwhile, to uncover the signaling and regulatory mechanisms that result in gene expression variation, it is critical to dissect ligand-receptor or L:R interactions involved in cell-cell communications and associated downstream gene regulations. However, existing tools for CCI analysis mostly consider only the ligands and receptors that may be involved, without consideration of potential effects.[26–28] Those that do consider these effects frame the CCI events in terms of cell-type proximity instead of in terms of signaling molecules, limiting interpretation of possible molecular drivers,[29,30] and return global estimates,[45] despite heterogeneity from cell to cell that becomes more pronounced especially in patterned 3D structures with a large number of cells and cell types. Spateo utilizes spatially weighted modeling approaches to connect expression patterns to possible mechanistic L:R interactions while accounting for potential differences between tissue regions (Figure 4B, lower; STAR Methods), returning predictions for each cell. It additionally models the ligand-receptor and downstream gene expression as a function of TFs, thereby creating "TF-gene models," to connect intercellular interaction with intracellular interactions (Figures 4B–4D; STAR Methods).

Before we applied our digitization and CCI algorithms to the 3D mouse embryo dataset,[23] we first extensively validated them on simulations and several public datasets. Spateo's digitization was able to create more uniform "layers" and "columns" in two simulation cases (Figures 4E and 4F), representative of common biological structures, for example, layers of the brain. It also nicely digitized the inner-outer layer axis of the macaque cortex section with a complex topology (Figure 4G). The continuous axes generated by this method enable multi-scale investigation of gene expression polarity (Figures S5A–5D), revealing known rostral-caudal (R-C) gradients of functional markers in the mouse brain,[13] such as *Cntnap2*, *Epha7*, and *Nr2f1*[53] (Figures S5A and S5B), and enabling subcellular polarity analysis: e.g., identifying genes

---

**Figure 3. Spateo outperforms state-of-the-art 3D alignment approaches**

(A–C) Benchmarks demonstrated across a broad range of 3D spatial transcriptomics datasets, including (A) MERFISH mouse hemibrain data,[37] (B) OpenST human metastatic lymph node data,[38] and (C) Stereo-seq macaque cortex.[43] For each panel, we have 3D visualization of each dataset from different algorithms on the left, while benchmark metrics are on the right. 95% confidence interval is included for each bar, and *p* value shown on the right indicates the t test result comparing Spateo with other two leading performing algorithms. STAlign-GT, STAlign with pre-alignment; pairwise MAE, pairwise mean average error; CLC, contextual label consistency; see STAR Methods.

(D) Benchmarking the mesh correction algorithm. Left: the 3D reconstruction results after performing Spateo multi-slice refinement. Although the tissue structure is smooth, it may not match the actual structure. Middle: after incorporating the mesh correction, the original structure has been well recovered and aligned with Allen CCF v3.[42] Right: the statistical results of mesh correction compared with other methods on mouse hemibrain from MERFISH and BAR-seq. "-G", combined with multi-slice refinement; "-M", combined with mesh correction; global MAE, global mean average error; see STAR Methods.

(E) On the Stereo-seq E9.5 and E11.5 mouse embryo datasets, the cumulative distribution curves with respect to the CLC score are plotted indicating that Spateo's 3D alignment is much more accurate.

(F) Efficiency and scalability benchmark on Stereo-seq macaque cortex dataset. Spateo-S means using sparse calculation; PASTE2* means using the brute force search for overlap ratio by their original implementation; see more details in STAR Methods.

(G) Robustness benchmark on sampling interval (left) and sampling cells (middle). Left: the left column shows the reconstruction results of Spateo on the full embryo data (top: E9.5 embryo; bottom: E11.5 embryo). The right three columns exhibit Spateo's reconstruction when only 1/2, 1/5, and 1/10 slices are preserved, respectively. Middle: the columns from left to right present the reconstruction results of Spateo when only 20%, 10%, and 5% cells are retained, respectively. Right: line plots of Pearson correlation (blue, left y axis labels) and mean absolute error or MAE (red, right y axis labels) reconstructed on complete and downsampled data by Spateo. Solid line corresponds to E9.5 data and dashed line to E11.5 data. 95% confidence interval is included for each line. Scale bars for E9.5 and E11.5 denote distances of 1 mm and 3 mm, respectively.
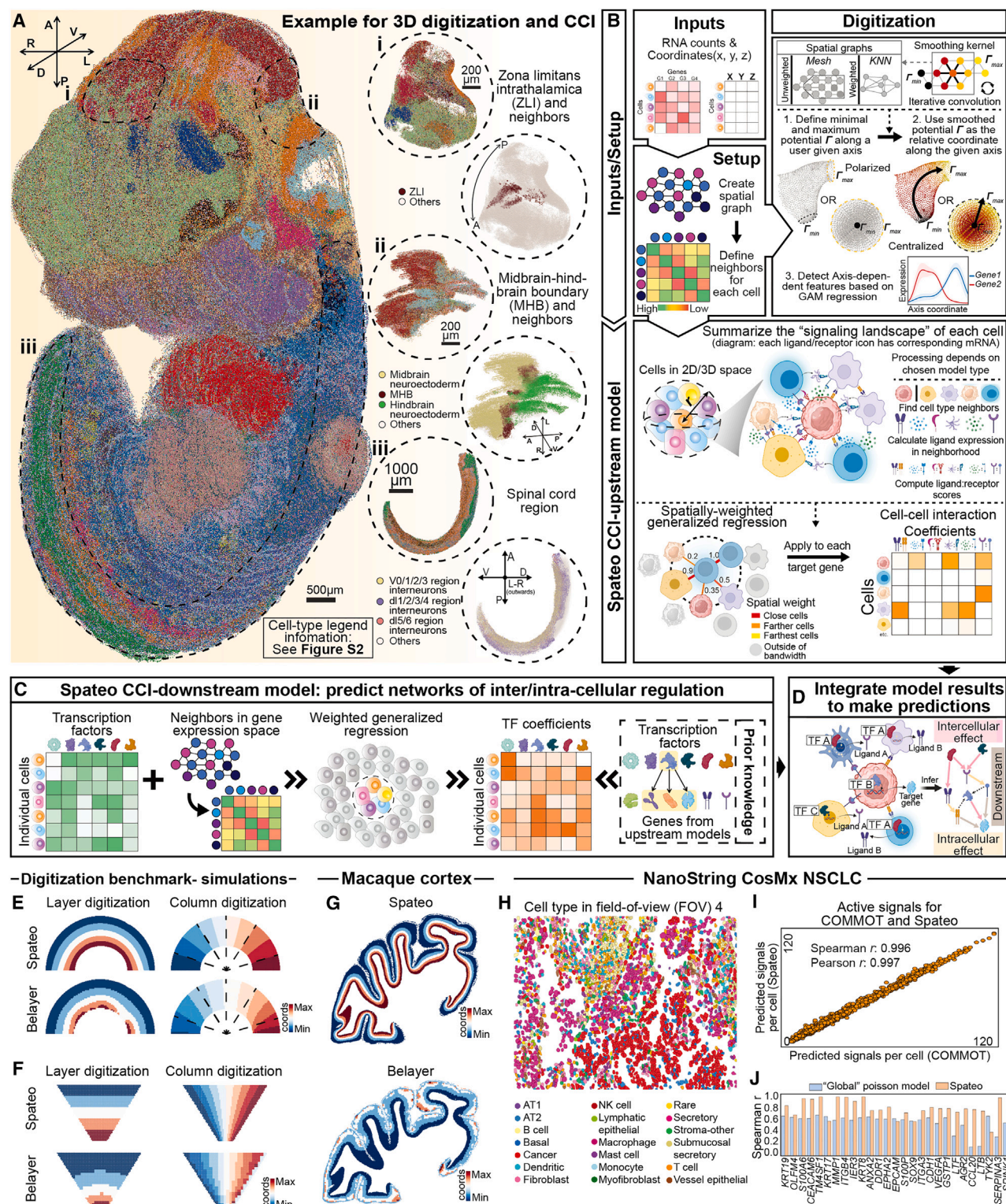
See also Figure S4.

**Cell**
Article

**Figure 4. A multi-scale, 3D-aware digitization and cell-cell interaction modeling framework to dissect molecular landscapes of the 3D mouse hologram**

(A) 3D reconstructed E11.5 stage embryo, colored by cell type and with the outlined zona limitans intrathalamica (i), midbrain-hindbrain boundary (ii), and spinal cord (iii) regions. Scale bars for the whole embryo, ZLI region, and spinal cord denote distances of 500 μm, 200 μm, and 1000 μm, respectively.

(B) Schematic of the digitization and cell-cell interaction (CCI) models (more details in STAR Methods).

*(legend continued on next page)*

enriched either at the nucleus centroid or near the karyotheca in the cytoplasm (Figures S5C and S5D).[54] In a MERFISH mouse brain dataset (Figure S5E),[55] our digitization successfully recapitulated cortical layers to model the laminar enrichment of neurons, achieving higher accuracy than Belayer (Figures S5F–S5I). These analyses demonstrated that Spateo's digitization facilitates the definition of arbitrary spatial axes across multiple scales, thereby enabling the detailed spatial dissection of molecular and signaling heterogeneity.

Our spatially weighted CCI models outperformed similar methods in predicting target gene expression (see STAR Methods for benchmark details) on a non-small cell lung cancer (NSCLC) sample profiled with CosMx,[44] assessed by the Spearman or Pearson correlation between the observed and model-predicted gene expression ($R^2$ consistently >0.8) (Figures 4H–4J). Furthermore, Spateo reported more consistent results, compared with global models, for same-family ligand and receptors, using *WNT* family interactions for analysis (Figures S5L and S5M). Consistent, robust performance was observed across multiple FOVs within the same dataset (Figure S5N) and also across a variety of datasets collected using different methods (Figures S5J–S5N and S5O–S5R).

Taken together, Spateo delivers a general framework to reveal gene expression polarity across multiple scales of biology from cell to tissue to organ, and it explains this variation in terms of the effects of region-specific intercellular and intracellular cell signaling.

## Systematic characterization of functional biological circuits underlying central nervous system development

During organogenesis, key 3D embryonic structures in the central nervous system (CNS), such as the ZLI, MHB, and spinal cord, characterized by their unique molecular diversity and cellular interactions. Specifically, the ZLI organizer, a critical brain organizing center, influences neighboring cells to adopt specific developmental paths, shaping the thalamic (p2) and pre-thalamic (p3) regions.[56] Little is known about the molecular patterns and intercellular interactions that manifest around organizers,[57] but new 3D ST methods start to allow for comprehensive analysis of the whole transcriptome, enabling the exploration of molecular features of the ZLI. Recent 3D studies have characterized the ZLI region from E8.5–E9.5[21]; however,

comparatively little is known about the signaling landscape of ZLI region at later stages. The creation of a comprehensive 3D cell atlas of the E11.5 mouse embryo and the introduction of 3D digitization and CCI modeling techniques allow us to explore the complex intercellular or intracellular biological circuits within the ZLI and other tissues at breadth and depth.

To characterize the ZLI organizer, we first extracted two subsets containing the ZLI, a ZLI flanking region, and the "diencephalic ring"[58] (Figure 5A; Tables S3 and S4). We then digitized these regions to respectively define "dorsal-ventral" (D-V) and "rostral-caudal" (R-C) axes (Figures 5B and 5C). Importantly, the ZLI does not extend to the roof plate[59]; indeed, we found the expression of the ZLI marker *Shh*[60] to be exclusive of that of *Fgf8* in the roof plate of the diencephalic ring (Figure 5B). From CCI modeling, we additionally found an effect of *Fgf8* on *Sufu* (Figure 5B, see "roof plate region") and *Gli3* (not shown), which together constitute a *Shh*-repressor complex[61] and thus provide a potential mechanism controlling the extent of dorsal extension of the ZLI. Next, we investigated the ZLI flanking region (Figure 5C) to probe how the ZLI functions as the boundary and signaling center of the p2 and p3 regions. We identified three major spatially variable gene modules along the R-C axis (Figure 5D; STAR Methods), which we hypothesized correspond to the prethalamus (p3), thalamus (p2), and ZLI. To characterize spatial distribution of genes, we mapped them to the p2, p3, and ZLI domains by visualizing their enrichments along the R-C axis (Figure 5E). We found several morphogens of the *Wnt* family to be enriched in the p2 region, except for *Wnt7b* that was instead enriched in the prethalamus (Figure 5E).

To reveal the intercellular signaling landscape involving *Wnt* morphogens and other ligands, we fit CCI models to predict cell-specific ligand effects (STAR Methods) on a panel of target genes selected among those found to exhibit R-C variability. We visualized the enrichment of predicted interaction effects along the R-C axis as well (Figure 5E). Noticing the effect of many *Bmp*-family factors on the stemness-promoting *Id1* in the p2 domain, we fit TF-gene models to infer TF regulators of both the ligands and of the target genes downstream of those ligands' cognate receptors to characterize the factors driving stemness in the E11.5 P2 region. Among others, we predicted *Bmps* to be regulated by *Sox2* and *Id1* by *Smad4*, the *Bmp* transducer[62] (Figure 5F), demonstrating Spateo's ability to map the

---

(C) Downstream model that uses the expression of transcription factors (TFs), a weighted neighbor graph constructed from gene expression, prior knowledge networks, and Spateo's spatial-aware regression framework to predict gene expression as a function of the expression of corresponding regulating ligand-receptor pairs.

(D) Schematic describing how the downstream model (C) can be used to extend insights derived from the ligand or L:R-based models (B, bottom) to infer both intercellular (L:R) interactions and intracellular (TF-target) regulations.

(E) Layer (left) and column (right) digitization by Spateo (upper) and Belayer[25] (lower) on a simulated half circle.

(F) Same as in (E) but for the simulated trapezoid.

(G) Layer digitization by Spateo (left) and Belayer (right) on a Stereo-seq section of the macaque cortex.

(H) Cell segmentation plot of all cells, colored by cell type, in field of view (FOV) 4 from a non-small cell lung cancer (NSCLC) sample profiled with CosMx.[44]

(I) Scatterplot comparing the number of ligand:receptor interactions across cells predicted by COMMOT[45] with those predicted by Spateo in the model preprocessing stage. Each point is a cell; the x axis represents the number of nonzero L:R interactions from the COMMOT array, and the y axis represents that from the Spateo array.

(J) Bar plot of Spearman correlations between observed and model-predicted gene expression, comparing Spateo's spatially weighted model with a non-weighted "global" Poisson generalized linear model that uses the COMMOT L:R array to predict expression. See STAR Methods section COMMOT benchmark for details.
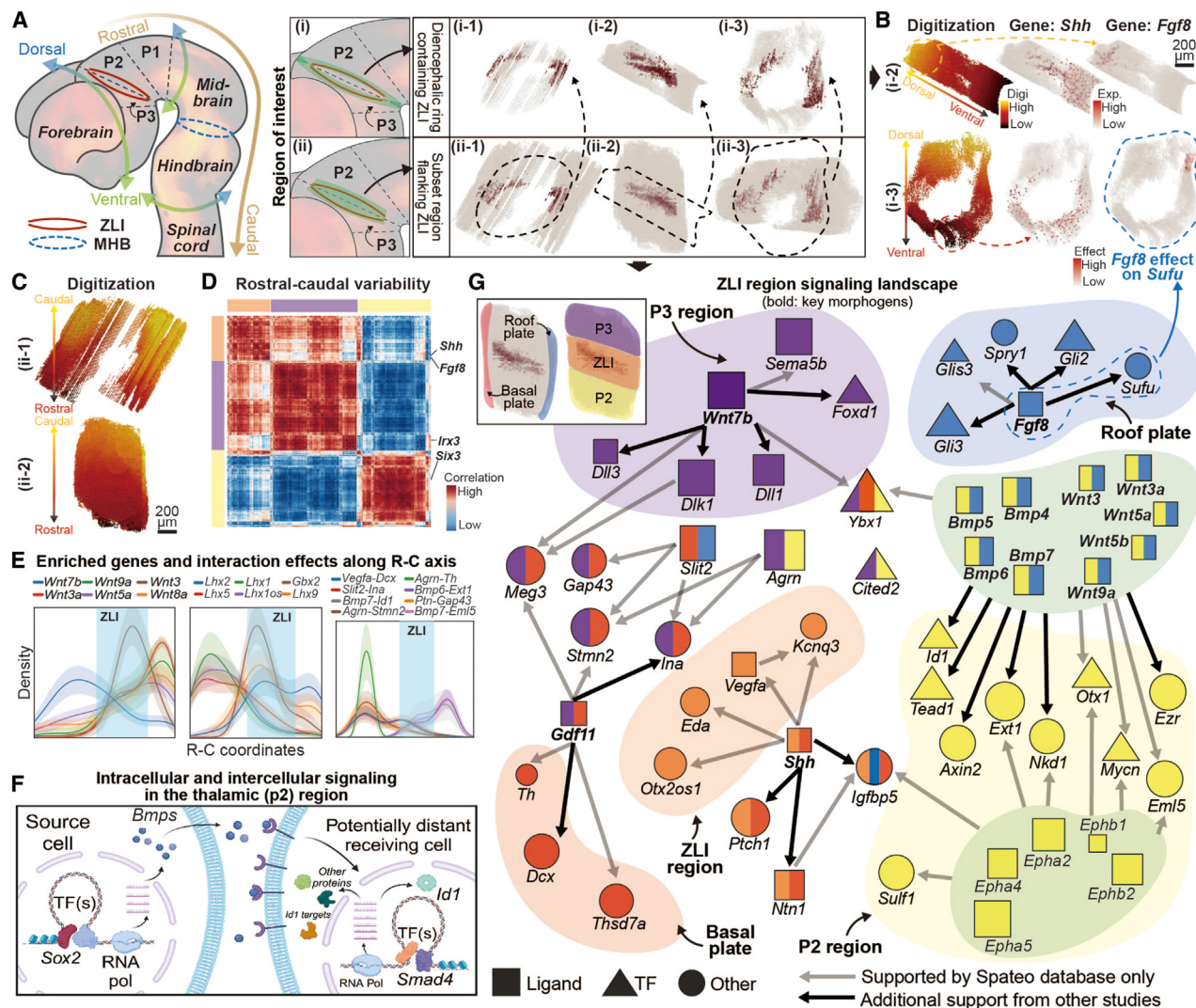
See also Figure S5.

**Figure 5. Spateo identifies networks of intercellular signaling and intracellular regulations in the developing brain proximal to the zona limitans intrathalamica**

(A) Schematic of the upper neural tube at the E11.5 stage, with zona intrathalamica limitans (ZLI) (red) and midbrain-hindbrain boundary (MHB) (light blue) indicated (left). The diencephalic ring (i) and ZLI flanking region (ii) analyzed are highlighted, with cells of the ZLI in maroon (middle). Different views (i-1 to i-3 and ii-1 to ii-3) and relationships (indicated by the dashed encapsulated lines) between these two regions are shown on the right.

(B) Spatial scatterplots of the diencephalic ring from two angles, colored by dorsal-ventral digitization values (left), expression of *Shh* (middle), and expression of *Fgf8* (top right) or effects of *Fgf8* on *Sufu* (bottom right). Scale bar denotes a distance of 200 μm.

(C) Spatial scatterplots of the ZLI flanking region, colored by rostral-caudal digitization values. Scale bar denotes a distance of 200 μm.

(D) Cluster map of genes that are highly variable along the R-C axis in the ZLI subset.

(E) (Left to right) Expression density plots for selected *Wnt* family ligands along the R-C axis, the expression of selected *Lhx* factors along the R-C axis, and the magnitude of selected cell-cell interaction effects along the R-C axis. The ZLI region is highlighted by a light blue rectangle.

(F) Schematic of intercellular or intracellular interactions involving *Bmp7*, *Sox2*, and *Id1*, and *Smad4*.

(G) The signaling landscape of ZLI region. Left: identification of regions in the ZLI subset. Purple: p2 (thalamic), orange: ZLI, yellow: p3 (prethalamic), blue: roof plate, and red: basal plate regions. Right: schematic showing cell-cell interaction effects (arrows) that connect ligands (triangles) to target genes (can be TFs: squares, or others: circles). Green regions (around ephrins and *Bmps/Wnts*) indicate group ligands that similarly affect groups of genes. Genes are colored by their region(s) of enrichment (with split colors if enriched in more than one region) and sized according to average expression. Black arrows indicate interactions supported by other studies, while gray arrows newly predicted interactions. Corr., correlation; Digi, digitization values; Exp., expression.

See also Figure S6.

intercellular/intracellular molecular signaling network of a specific spatial domain. Next, we characterized the precise localization of signaling effects for additional ligands, including *Gdf11*, ephrins, and *Shh*. We mapped each interaction effect to the p2, ZLI, and p3 regions and to the basal plate and roof plate domains along the D-V axis when applicable (D-V axis not shown)
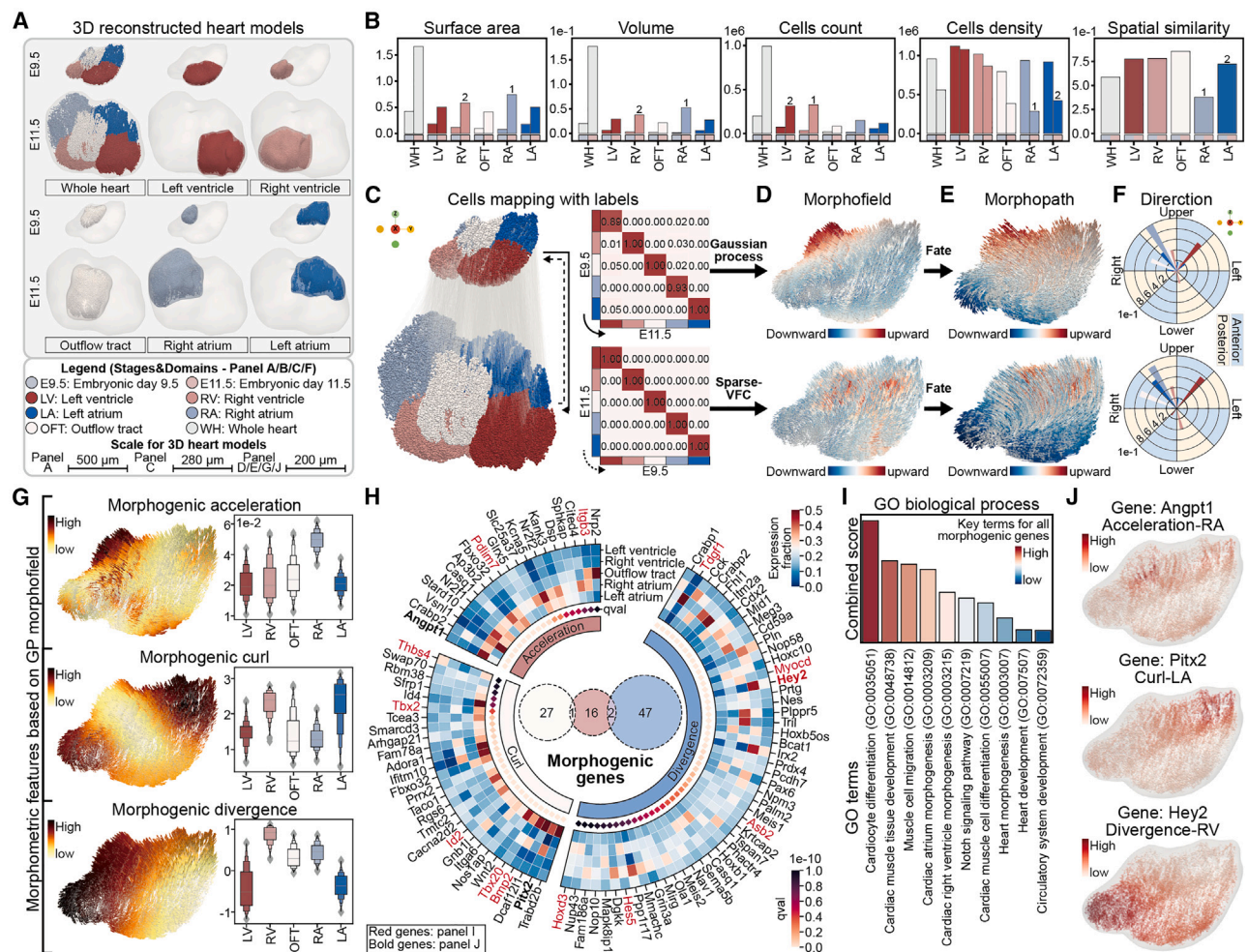
**Figure 6. Spateo characterizes morphometric and molecular dynamics involved in the asymmetrical heart organogenesis**

(A) 3D reconstructed models of the whole heart and each major anatomical structure, namely, left ventricle (LV), right ventricle (RV), outflow tract (OFT), right atrium (RA), and left atrium (LA) from E9.5 (top row) or E11.5 (bottom row).

(B) Bar plot of different 3D morphometric features of the whole heart (WH) or each major structure between E9.5 (left bar for each group) and E11.5 (right bar). The top two domains with the most dramatic changes of each morphometric feature are annotated.

(C) 3D cell alignment map and cell-type transition matrix between E9.5 and E11.5 hearts. The solid line with the arrow indicates the mapping from E9.5 to E11.5 cells, while the dash line with the arrow indicates the mapping from E11.5 back to E9.5. Gray lines between heart structures connect every E9.5 cell to the most likely target cells at E11.5 heart. The transition matrix on the right indicates the transition probability from one cell type to another. Each row of the matrix sums to be 1.

(D) 3D quiver plots represent the cell migration field. Cells are colored by the value of the z axis component of the migration velocity.

(E) 3D streamline plot of the cell migration paths, predicted from the migration vector field.

(F) A 2D bar chart on the polar axis illustrates the average migration directions (different octants, defined by left-right, upper-lower axis, and anterior and posterior direction) and magnitude (length of the bar) of cells of the five major structures (indicated by the color) of the heart. From (D—F), we can see that both Gaussian process or GP and SparseVFC gave similar results, while GP-based vector fields tend to be smoother.

(G) Differential geometry analysis of the morphometric vector field reveals the asymmetrical differentiation of the heart. From top to bottom: on the left, cell migration streamline plots with streamlines colored with the migration acceleration, curl, and divergence; on the right, the boxplots of the corresponding differential geometry quantities across five major structures.

(H) Circos plot of top significant genes (morphogenic genes) that are highly correlated with morphometric curl, acceleration, and divergence. The middle Venn diagram reveals the gene set size and overlapping gene numbers related to morphogenic curl (white), acceleration (dark pink), and divergences (blue) categories. The outside loop includes heatmaps of gene expression for all morphogenic genes, specific to each morphometric property (shown as a separate heatmap). The rows of the heatmap are annotated with major heart structures, while the columns are annotated with gene names. Genes are ordered by q values, increasing in clockwise fashion for each morphogenic category. The color of each box on the heatmap and the dots near the heatmap are used to indicate the gene expression level and q value, respectively. Bold gene names are among the gene sets defined by the GO pathways shown in (l), while red-colored gene names are known morphogenesis genes visualized in (J).

*(legend continued on next page)*

(Figure 5G). Notably, we observed regulation of prethalamus/P3-specific ligands (e.g., *Sema5b*, the *Notch*-like *Dlk1, Dll2*, and *Dll3*) and TFs (e.g., *Foxd1*) by *Wnt7b*, the aforementioned activity of *Fgf8* in the roof plate, and effects of *Gdf11* on early neuronal markers (e.g., *Dcx* and *Thsd7a*) in the basal plate (Figure 5G). We also noted thalamus/P2-specific upregulation by *Wnts* of the negative feedback regulators *Axin2* and *Nkd1*. Furthermore, we observed localized effects of the *Wnt* and *Bmp* ligands on pluripotency maintenance genes, such as *Id1, Tead1, Mycn, Otx1, Cited2*, and *Ybx1*[63–69] (Figure 5G). Many of these observations are further supported by past perturbational studies[62,70–73] (Figure 5G). These results paint a comprehensive picture of the intercellular and intracellular signaling landscape of the ZLI region, spanning several spatial domains (p2, ZLI, p3, roof plate, and basal plate domains) at E11.5.

We further studied the molecular landscape of the MHB and spinal cord in 3D. In the MHB region, we characterized the region specificity of select ligands and target genes, such as the *Fgf*-family factors in the boundary neuroectoderm (Figures S6A and S6B; Table S5). We additionally investigated effects of developmentally critical ligands *Ptn*[74] and *Cdh2*[74,75] on a variety of downstream targets (Figures S6C and S6D). We identified effects of *Ptn* on genes involved in synaptic plasticity (e.g., *Gap43*), axon growth (e.g., *Rtn1*), and others (Figure S6C). We also revealed region-exclusive effects of *Cdh2* on *Tox* and *Abcc4* (Figure S6D). These predicted effects are consistent with previous reports,[76,77] further validating Spateo's ability in identifying biologically meaningful effects. For the spinal cord region, we first integrated cross-sections along the medial-lateral axis into the middle plane, and then we digitized this integrated spinal cord representation to obtain a D-V axis (Figure S6F; Table S6). We observed members of the *Lhx* family (Figure S6G) and other TFs (e.g., *Dbx1, Gbx2*) (Figures S6H and S6I), all important for spinal development,[78–80] to be differentially distributed along the D-V axis. We additionally found *Slit2* to be spatially enriched at the dorsal end (Figure S6J). As *LIM* domain factors (such as the *Lhx* family) help to control *Slit2* expression in the cranium,[81] we investigated whether the *Lhx* factors might similarly affect *Slit2* in the spinal cord. From TF-gene modeling, many *Lhx* genes and other homeobox factors were among the top predicted regulators for *Slit2* (Figure S6K). We identified potential downstream targets of *Slit2* signaling, revealing potential targets involved in adhesion and cytoskeletal dynamics, consistent with its role in axon guidance[82,83] (Figure S6L). In all, these results demonstrate applications of 3D digitization to identify region-specific gene expression patterns and of CCI modeling to predict the subtle interplay between transcriptional regulation and CCIs in the MHB region and spinal cord at E11.5.

## Morphometric vector field predicts cardiac migration paths and characterizes molecular pathways underlying the asymmetrical organogenesis of cardiac chambers

The formation of an embryo and constituent organs is characterized by a tightly controlled morphogenetic process.

Live imaging provides the opportunity to observe morphogenesis over time at high resolution,[84] but it cannot associate complex regulatory programs to such morphometric changes because imaging can only measure a few genes within single cells over time. With our multi-time-point molecular holograms, we can not only compare reconstructed 3D organs over time to reveal distinct modes of cell migration, but we can also predict how each individual cell migrates over time and can ultimately connect macroscopic morphological changes with microscopic molecular expression dynamics in single cells across space.

The heart, one of the first organs to appear during mammalian embryogenesis, is a highly structured organ that is formed through a complex migration process, including heart tube looping, septation, and valve formation.[1] To characterize the cell migration pattern of heart organogenesis and reveal the underlying morphogenetic factors, we first reconstructed 3D models of the heart at E9.5 and E11.5 based on 74 and 64 slices, respectively (Figures 6A and S7A–S7D; Table S7). We also demonstrate that all major structures, namely, left/right ventricle (LV, RV), left/right atrium (LA, RA), and outflow tract (OFT) from both time points can be reconstructed (Figure 6A). We observed high specificity of region-specific genes in each structure for both time points. For example (Figures S7B and S7C), *Cited1*,[85–87] *Cck*,[86] *Tnc*,[87] *Angpt1*,[88] and *Tbx5*[89] are enriched in the LV, RV, OFT, RA, and LA regions at E9.5 and *Hand1*,[85] *Hey2*,[90] *Sema3c*,[91] *Hcn4*,[92] and *Pitx2*[93] are enriched in the corresponding regions at E11.5. Our 3D reconstruction further reveals a significant increase in surface area, volume, and cell number, while the cell density remains relatively constant and structural similarity is high, consistent with previous studies[94] (Figure 6B).

We applied the Gaussian process approach used for 3D alignment of the whole embryo to map cells of the E9.5 heart to those of the E11.5 heart (Figure 6C, left). Major structures (LV, RV, etc.) of E9.5 mapped to the corresponding E11.5 structures with high specificity (Figure 6C, right). Importantly, the Gaussian process can directly return an analytical morphometric vector field of single-cell migration as well, constituting a uniform approach capable of both performing 3D alignment and learning morphometric vector fields. We define such vector fields as "morphometric vector fields" that map cells from an earlier time point to a later time point, revealing cellular migration patterns in physical space (Figures 6D–6F; Video S3) in a manner analogous to our previous RNA velocity vector field approach that instead reveals cell fate transition in gene expression space.[36] To reveal potential underlying regulatory mechanisms of morphogenesis, we leveraged the accompanying transcriptomic data and differential geometry quantities that can be analytically derived from the reconstructed morphometric vector field to reveal cellular migration properties in physical space (STAR Methods). Importantly, for 3D morphometric vector fields, 3D curl (measures the degree of rotation at a given point), acceleration, curvature, torsion (quantifies the degree of twisting of a 3D object), divergence (reveals

---

(I) Bar plot of the combined GSEA score[46] for key terms associated with heart morphogenesis based on GO enrichment analyses of all morphogenic genes.

(J) 3D scatterplot of the interpolated gene expression of example genes (highlighted in H) on cells of the reconstructed E9.5 3D heart. The highest expression region for each category is annotated. A transparent surface mesh is used to reveal the 3D shape of the reconstructed heart.

See also Figures S7, S8, and S9.
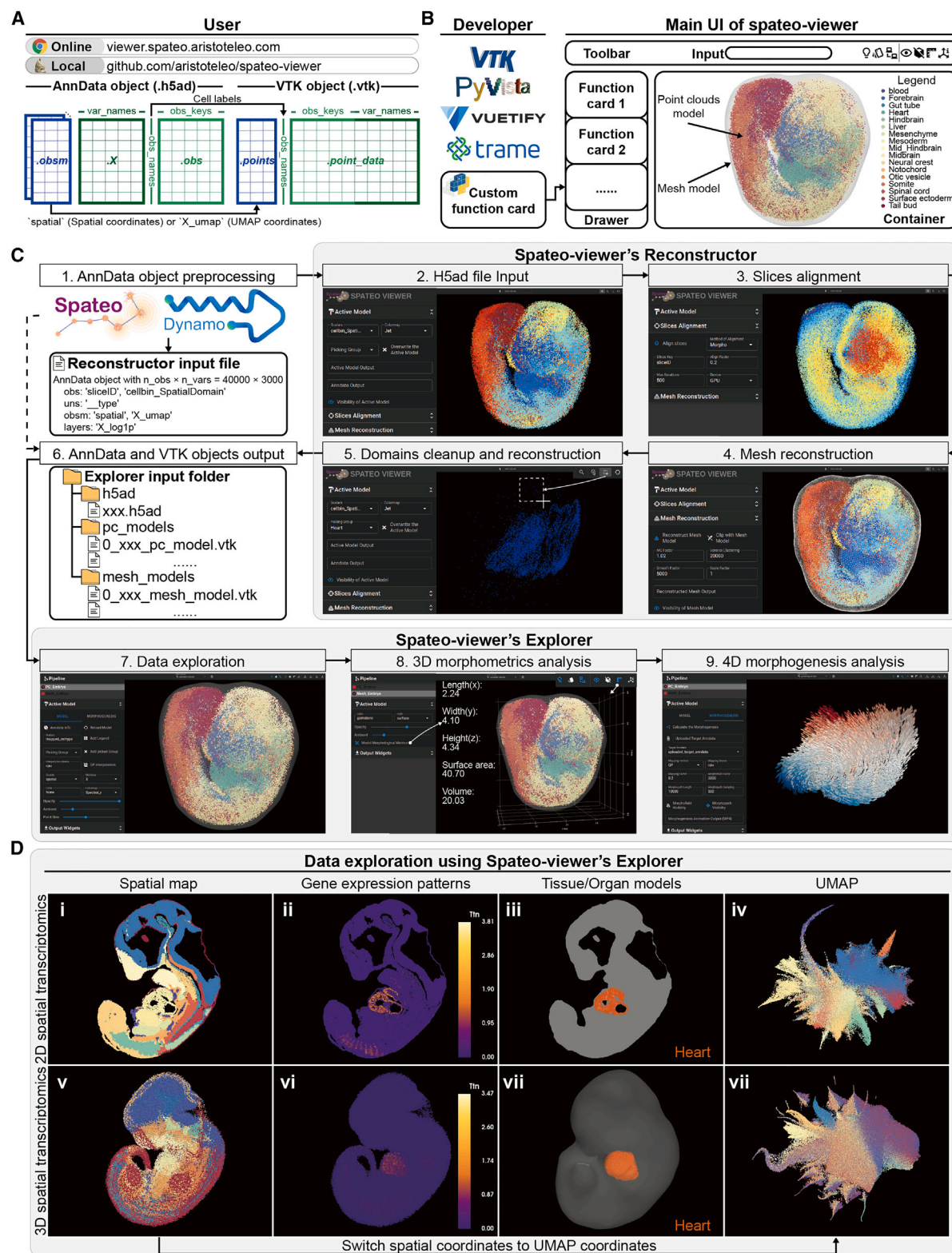
(legend on next page)

whether the tissue is expanding or shrinking), and Jacobian (Figure S7E) (quantifies how migration along one axis influence movement along the other) have real physical meanings, and crucial morphogenetic genes can be identified by finding genes with significant correlations with these morphometric properties. Interestingly, from our differential geometry analysis, we revealed the asymmetrical migration pattern of the heart (Figures 6G and S7E). We showed, for example, that the RA has the highest acceleration, the RV and LA have the largest curl, while LV has the lowest divergence, consistent with the timeline of the formation of these structures (Figure 6G): compared with the LV that originated from the first heart field, LA/RA and RV originating from the second heart field are integrated into the heart at a late stage.[1] Furthermore, their possible progenitors and less mature states contribute to their quick expansion/morphogenesis, related to the high acceleration and curl at this time point (Figure 6G). The LV is much more mature and has a limited differentiation ability that makes it hard to contribute to other structures, characterized by its lowest divergence.[95] We next detected a set of "morphogenic genes" that are highly correlated with morphometric curl, acceleration, and divergence, including known cell migration markers,[96,97] such as *Tbx2/Bmp2*; key genes important for atrioventricular canal (AVC) development[95,98,99]; *Tdgf1* for anterior heart tube development[100]; *Angpt1* for right atrial chambers' morphogenesis[88]; *Pitx2* for the L/R asymmetric formation[101]; and *Hey2* for the ventricular formation[90] (Figures 6H–6J). Gene set enrichment analyses (GSEAs)[46] of this set of "morphogenic genes" further revealed key terms associated with heart morphogenesis, for example, muscle cell migration, cardiac atrium morphogenesis, and cardiac RV morphogenesis (Figure 6I).

Collectively, these analyses reveal that Spateo characterizes molecular pathways involved in asymmetrical heart organogenesis by predicting the morphogenesis paths of cell migration, thus connecting macroscopic morphological dynamics with microscopic expression dynamics.

### Quantifying volumetric dynamics, expression polarity and predicting morphogenesis factors during *Drosophila* germ band retraction

Embryogenesis is characterized by specific "organogenesis modes," dynamic patterns of organ morphogenesis orchestrated by sequential gene regulatory programs. These gene programs also dictate hierarchical cell fate specification and organization into complex 3D units of structure and function.[102] In addition to mice, *Drosophila* is an excellent model system to investigate the kinetics underlying organogenesis modes and to explore the relationship between morphogenesis and gene expression given its fast life cycle and dramatic but highly organized morphological and gene expression changes during embryogenesis. Specifically, from S11 and S13 stages of the *Drosophila* embryo, we found that although the overall embryonic structures across these two time points are rather similar (0.93) (Figures S8A and S8B; Videos S4 and S5), we were able to identify three general major organogenesis modes: (1) organ migration or movement, e.g., CNS, amnioserosa, and muscle; (2) organ fusion/convergence, by which multiple pieces fuse into a mature organ, e.g., midgut; and (3) organ expansion, primarily driven by cell growth, e.g., hindgut and salivary gland (Figure S8B).

We next characterized the gene expression polarity in the whole *Drosophila* embryo, starting with quantifying the backbone of the germband by learning a principal curve that passes through the middle of cells in the germband along the anterior-posterior (A-P) axis[103] (Figure S8C). We then identified genes that are dynamically changing along this principal backbone using regression analyses, identifying head-specific genes such as *Dfd*; thorax-specific genes such as *Scr* and *Antp*; and abdomen-specific genes such as *Ubx*, *Adb-A*, and *Adb-B*[104] (Figures S8D–S8F; Table S8). These Hox gene expression patterns are further confirmed after interpolating the original noisy data, revealing a consistent expression pattern with the Berkeley Drosophila Genome Project (BDGP) *in situ* database[105] (Figure S8G).

In contrast to the analytical morphometric vector field analyses of only the single heart organ in Figure 6 and VideoS6, we modeled main organs within the germband, consisting of CNS, hindgut, midgut, muscle and salivary gland jointly, considering the fact that developing organs of the germband influence one another and are constrained by extrinsic physical and biomechanical factors (Figure S8H). The analytical vector field revealed high curl (Figure S8I) and acceleration (Figure S8J) values at the tail of the germband, indicative of a strong contraction at this region (Table S9). To further quantify the genes and gene programs that may drive the morphogenesis for these organs, we plotted average acceleration and curl values of all the cells with non-zero gene expression for individual genes corresponding to a backbone index as a function of backbone indices (Figure S8K). Interestingly, known morphogenesis genes, such as *peb/hnt*,[106] cad,[107] Abd-B,[108] etc., were identified to have the highest mean backbone values and thus be enriched at the tail of the germ band (Figure S8K). We also detected a repertoire of less characterized genes, such as *CG2930* and *CG31463*. The

**Figure 7. Spateo-viewer, a versatile spatial transcriptomics data exploration web application that allows interactive 3D spatial transcriptomics data exploration, slice alignment, 3D model reconstruction, morphometric traits, and morphogenesis analysis**

(A) Two access modes of Spateo-viewer and required user data objects. Spateo-viewer is deployed online and can be accessed at https://viewer.spateo.aristoteleo.com/ or used locally as a standalone App. The input to Spateo-viewer (http://viewer.spateo.aristoteleo.com/) is an AnnData object, commonly used for storing spatial transcriptomics datasets where the.obsm slot contains spatial, UMAP, or other coordinates; the.X slot contains gene expression values; and the.obs contains cell annotation information. AnnData objects can be then converted into VTK objects to generate the.points and.point_data slots either by Spateo or Spateo-viewer's Reconstructor (see below).

(B) The schematic of Spateo-viewer's GUI interfaces. Shown on the left are developer toolkits that enable Spateo-viewer's functionalities.

(C) Typical workflow and the Reconstructor or Explorer of the Spateo web application.

(D) Spateo's Explorer allows various data exploration modes for both 2D and 3D spatial transcriptomics datasets. PC, point clouds; UMAP, uniform manifold approximation and projection for dimension reduction.

See also Figure S10.

importance of these detected genes during morphogenesis was also reflected by Gene Ontology (GO) analysis,[46] which further reveals the enrichment in pathways associated with the germband extension, embryonic hindgut morphogenesis, and others (Figure S8L). Lastly, we showed that although there is little spatial co-localization between muscle cells and neural cells, midgut and hindgut cells strongly co-colocalize with muscle cells, suggesting a critical role of the muscle cells in the rapid migration of both the hindgut and midgut cells at this stage (Figure S8M). Interestingly, we also found that the expression of Neo, Fkh,[109] and COX8 is strongly correlated with the curl magnitude, confirming their roles in modulating the cell migration of hindgut/midgut, which is critical for the germband retraction and remodeling (Figure S8N). Furthermore, when we focus on the CNS (Video S7) and midgut (Video S8), we also demonstrate that Spateo infers cell migration paths and identifies morphogenesis regulators of each individual region (Figure S9).

Collectively, these results, in combination with the previous section, highlight how Spateo can go beyond descriptive spatial analyses to more dynamical and predictive 3D spatiotemporal modeling with morphometric vector fields, significantly advancing our ability to leverage 3D ST to reveal regulatory mechanisms underlying emergent properties.

### Spateo-viewer, the "Google Earth" of 3D spatial transcriptomics

The size and complexity of emerging 3D spatiotemporally resolved ST datasets pose a huge challenge on how to best explore such datasets. A user-friendly web application analogous to "Google Earth" is necessary to allow smooth, interactive access to 3D single-cell data. We thus developed the Spateo-viewer (http://viewer.spateo.aristoteleo.com/) to enable interactive visualizations of 3D gene expression patterns, intuitive browsing of tissue architecture and 3D organ models, and animation of organ morphogenesis patterns.

Spateo-viewer is a versatile tool for interactive slice alignment, 3D data and model reconstruction, exploration, visualization, and downstream volumetric and morphometric analyses, leveraging powerful 3D libraries from computer vision (CV) and our Spateo and Dynamo packages (Figures 7A and 7B). To connect with tools from the CV ecosystem, Spateo or Spateo-viewer will first convert the *h5ad* data format used by the single-cell genomics field to *vtk* files used by visualization toolkit (VTK),[110] PyVista,[111] Vuetify, and trame. Specifically, the spatial information, uniform manifold approximation and projection (UMAP) or other coordinate information (from.*obsm*), the expression of genes of interests (from.*X* or.*layers*), and cell annotation information (from.*obs*) will be converted into.*points* and.*point_data* (Figure 7A). Once the data conversion is complete, the lightweight *vtk* files can then be conveniently visualized and manipulated in the interactive data browser (Figure 7B).

Spateo-viewer features a highly modularized architecture and streamlined workflow (Figures 7C and 7D). In particular, Spateo-viewer comprises two major applications: the Reconstructor module enables slices alignment and 3D reconstruction (Figure 7C), while the Explorer module facilitates 3D data exploration and volumetric and morphometric analyses (Figure 7D). A typical Spateo-viewer workflow starts with data preprocessing. The

Reconstructor, once provided with preprocessed data, can be used to first align slices to reconstruct the 3D point cloud, which can then be used to create surface meshes, followed by domain cleanup and mesh reconstruction for each individual tissue/organ. The cleaned data can be exported as *anndata* objects to interact with other tools, including Spateo and Dynamo, followed by further analysis in the Explorer. The Explorer allows data exploration and 3D volumetric analysis (e.g., estimations of length/width/height/surface area, cell density, and cell-type distribution) for each organ. If data from multiple time points are provided, 4D morphogenesis analysis can be applied to understand the molecular mechanisms driving the spatiotemporal kinetics of different organs. The data exploration in the Explorer can operate either in the 2D space or 3D space to visualize not only the intricate spatial distribution of different spatial domains, cell types, or specific organs but also the gene expression in either the 2D or 3D physical or reduced expression space.

To sum up, Spateo-viewer presents an interactive, intuitive, and lightweighted online application to allow 3D data manipulation and visualization of the emerging 3D ST. It is also highly modularized and extendable, making it a flexible tool to allow researchers to include new analysis modules (Figure S10). As the 3D ST continue to expand, we believe it will become an indispensable tool for the field.

### DISCUSSION

We are now at the dawn of a new era of spatial biology, with subcellular resolution, large FOV methods such as Stereo-seq theoretically enabling construction of time-resolved embryo-scale datasets in 3D space. To fully study these datasets, a scalable analytical framework that explicitly models gene expression and signaling dynamics in 3D space over time is critical. Here, we present Spateo, a 3D aware foundational framework for advanced spatiotemporal modeling, consisting of four integral phases: it (1) reconstructs 3D embryo and organ models from sequential 2D serial sections with a scalable and accurate algorithm that allows for partial, non-rigid, multi-slice refinement and mesh correction; (2) builds a multi-scale spatial domain digitization framework to study levels of biology ranging from single-cell level to embryo level; (3) dissects L:R interactions and predicts interaction-associated downstream effects on gene expression of several 3D structures including ZLI, MHB organizers, and spinal cord at single-cell resolution with a spatially aware regression model; and (4) learns morphometric vector fields that describe cellular migration patterns and facilitate identification of regulatory mechanisms underlying cell movement and morphogenesis.

These innovative 3D aware spatiotemporal modeling approaches fueled a series of discoveries otherwise proven to be challenging. In the first stage, Spateo's 3D alignment approach, built on Gaussian process and variational inference, enabled us to more accurately and robustly reconstruct the molecular holograms of whole mouse embryos in 3D space across multiple time points, compared with existing state-of-the-art approaches. Importantly, Spateo can be extended to other modalities, e.g., H&E staining images and the use of off-the-shelf image feature extractors and descriptors, such as scale-invariant feature transform or SIFT[112] and DIScrete

# Cell
## Article

Keypoints or DISK,[113] to extract feature points (see Spateo's online tutorials). Furthermore, it can conveniently integrate with other approaches, e.g., Spateo can further improve the initial alignment from alternative tools by multi-slice refinement and mesh correction. In the second stage, our multi-scale digitization framework revealed not only nucleus- and cytoplasm-enriched genes in subcellular resolution but also spatial polarity genes that define the 3D signaling organization of the secondary organizers such as the ZLI and MHB and spinal cord. To investigate the CCIs and transcription regulation underlying the spatial heterogeneity of the gene expression, in stage three, we further developed an integrative framework to uncover and characterize the unique molecular networks that span both intercellular and intracellular interactions, painting an intricate signaling landscape of the ZLI region. Lastly, in the fourth stage, Spateo learned morphometric vector fields that accurately predict the 4D spatiotemporal migration path of asymmetric murine heart organogenesis and the *Drosophila* germband retraction, connecting macroscopic cellular morphogenesis with underlying microscopic molecular pathways. We anticipate that future in-depth molecular and genetic studies will be able to validate these extensive predictions to advance our understanding of embryogenesis and disease.

As spatial technologies continue to mature and percolate through biological laboratories, we foresee an explosion of diverse optimization and application of ST and further development of Spateo. Many single-cell genomics approaches can be translated to spatial genomics approaches, including single-cell multi-omics; RNA metabolic labeling; Perturb-seq and lineage tracing to enable multi-view; and spatiotemporally resolved, lineage-resolved, and perturbation-resolved cell-state dynamics *in situ* and in 3D space. We additionally foresee opportunities to apply Spateo to understand biological systems in many circumstances, e.g., by generating spatially resolved cross-species cell atlases and comparing 3D models of organs between different species to understand the evolutionary emergence of tissue structures, such as the evolution of the four-chamber heart in mammals from the single-chamber heart in invertebrates.[102] By further optimizing the 3D morphometric vector field approach, we expect many unique opportunities to directly link regulatory and functional genes with morphological changes at organ and embryo levels.

Spateo complies with best practice in software engineering, which includes a modularized and extendable infrastructure that allows continuous optimization; future integration with other frameworks from the Aristotle GitHub organization, a foundational software ecosystem for predictive genomics (https://github.com/aristoteleo), currently including Dynamo[36] and Dynast (https://github.com/aristoteleo/dynast-release); and active community contribution to draw from expertise across the field to build a uniform software ecosystem that enables dynamic, quantitative, and predictive analyses of single cells and ST.

## Limitations of the study

First, while the cost of sequencing has dramatically decreased over the past two decades, that of emergent ST still prevents large-scale 3D ST. The rapid development of cost-effective open-sourced technologies, such as seq-scope,[114] openST,[38,114]

novaST,[115] etc., and commercial platforms like singular genomics' G4X, characterized by its large FOVs, will hopefully democratize this technology in the near future, further broadening the impact and applicability of Spateo. Second, although we show that our spatial-aware regression model enables dissection of the intercellular and intracellular interactions at 3D space, models that account for nonlinear gene regulation, such as graph neural networks, may be developed to reveal complicated nonlinear and higher-order L:R interactions and combinatorial gene regulations in a spatially resolved manner. Third, while our current morphometric vector field approach does not explicitly model gene expression dynamics or RNA velocity, they could be unified into a single learning task to learn reaction-diffusion like spatiotemporal models.

**Cell**
Article

## AUTHOR CONTRIBUTIONS

X.Q., Y. Bai., S. Liu., X.X., and J.M. conceived the project. X.Q., Y. Bai., Y. Lu., and D.Y.Z. developed the models and theories. Specifically, Y. Lu. developed the 3D alignment algorithm with guidance from X.Q. and J.M.; D.Y.Z. developed the CCI model with guidance from X.Q.; Y. Bai. and Z.J. developed the digitization algorithm; and X.Q., J.Y., and Y. Lu. developed the algorithms for spatial imputation, backbone analyses, morphometric analyses, and morphometric vector field modeling. X.Q., Y. Bai., D.Y.Z., Y. Lu., J.Y., Z.J., J.M., H.P., and S.W. developed Spateo, implemented the code, and performed the analyses. M.C. conducted the mouse Stereo-seq experiment, and M.W. and Z.T. conducted the *Drosophila* Stereo-seq experiment. S.H. designed the logo for Spateo, under the supervision of S. Liu. and X.Q. X.Q., Y. Bai., D.Y.Z., Y. Lu., J.Y., and Z.J. designed and generated the figures. X.Q., D.Y.Z., Y.L, Y. Bai., and Z.J. wrote the manuscript with feedback from all other authors.

## DECLARATION OF INTERESTS

J.S.W. declares outside interest in 5 AM Venture, Amgen, Chroma Medicine, KSQ Therapeutics, Maze Therapeutics, Tenaya Therapeutics, Tessera Therapeutics, and Third Rock Ventures, all unrelated to this work. J.S.W is also a member of the *Cell* journal advisory board, which is also unrelated to this work.

## STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:
- KEY RESOURCES TABLE
- EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS
- METHOD DETAILS
  - The 3D spatial transcriptomics Stereo-seq dataset of whole embryons at E9.5 and E11.5
  - Alignment of spatial transcriptomics profilings of serial sections to create 3D models at whole embryo level
  - Variational Bayesian Inference
  - Stochastic Variational Inference
  - Guiding Optimization
  - Joint modeling multiple slices
  - Mesh Correction
  - Spatial domain digitalization
  - Spatially-aware modeling of networks of cell-cell interaction or CCI in 3D spatial transcriptomics
  - Spatially-weighted modeling for characterization of spatial specificity of signaling effects
  - Downstream analyses
  - Morphometric vector field and morphometric differential geometry analyses
  - Apply generalized linear models (GLM) to detect differentially expressed genes in different contexts
  - Explore 3D spatial transcriptomics data with Spateo-viewer
  - Supplementary Methods
  - Analysis of Central Nervous System of Stereo-seq mouse embryo data
  - Spateo Alignment Benchmark
- QUANTIFICATION AND STATISTICAL ANALYSIS
- ADDITIONAL RESOURCES

## SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cell.2024.10.011.

## REFERENCES

1. Barresi, M.J.F., and Gilbert, S.F. (2019). Developmental Biology (Sinauer Associates, Incorporated).

2. Packer, J.S., Zhu, Q., Huynh, C., Sivaramakrishnan, P., Preston, E., Dueck, H., Stefanik, D., Tan, K., Trapnell, C., Kim, J., et al. (2019). A lineage-resolved molecular atlas of C. elegans embryogenesis at single-cell resolution. Science *365*, eaax1971. https://doi.org/10.1126/science.aax1971.

3. Cao, J., Packer, J.S., Ramani, V., Cusanovich, D.A., Huynh, C., Daza, R., Qiu, X., Lee, C., Furlan, S.N., Steemers, F.J., et al. (2017). Comprehensive single-cell transcriptional profiling of a multicellular organism. Science *357*, 661–667. https://doi.org/10.1126/science.aam8940.

4. Byrne, J.H. (2019). The Oxford Handbook of Invertebrate Neurobiology (Oxford University Press).

5. Calderon, D., Blecher-Gonen, R., Huang, X., Secchia, S., Kentro, J., Daza, R.M., Martin, B., Dulja, A., Schaub, C., Trapnell, C., et al. (2022). The continuum of Drosophila embryonic development at single-cell resolution. Science *377*, eabn5800. https://doi.org/10.1126/science.abn5800.

6. Cao, J., Spielmann, M., Qiu, X., Huang, X., Ibrahim, D.M., Hill, A.J., Zhang, F., Mundlos, S., Christiansen, L., Steemers, F.J., et al. (2019). The single-cell transcriptional landscape of mammalian organogenesis. Nature *566*, 496–502. https://doi.org/10.1038/s41586-019-0969-x.

7. Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F., et al. (2018). Mapping the mouse cell atlas by microwell-Seq. Cell *173*, 1307. https://doi.org/10.1016/j.cell.2018.05.012.

8. Cao, J., O'Day, D.R., Pliner, H.A., Kingsley, P.D., Deng, M., Daza, R.M., Zager, M.A., Aldinger, K.A., Blecher-Gonen, R., Zhang, F., et al. (2020). A human cell atlas of fetal gene expression. Science *370*, eaba7721. https://doi.org/10.1126/science.aba7721.

9. Regev, A., Teichmann, S.A., Lander, E.S., Amit, I., Benoist, C., Birney, E., Bodenmiller, B., Campbell, P., Carninci, P., Clatworthy, M., et al. (2017). The Human Cell Atlas. Elife *6*, e27041. https://doi.org/10.7554/eLife.27041.

10. Moses, L., and Pachter, L. (2022). Publisher Correction: museum of spatial transcriptomics. Nat. Methods *19*, 628. https://doi.org/10.1038/s41592-022-01494-3.

11. Rao, A., Barkley, D., França, G.S., and Yanai, I. (2021). Exploring tissue architecture using spatial transcriptomics. Nature *596*, 211–220. https://doi.org/10.1038/s41586-021-03634-9.

12. Marx, V. (2021). Method of the Year: spatially resolved transcriptomics. Nat. Methods *18*, 9–14. https://doi.org/10.1038/s41592-020-01033-y.

13. Ståhl, P.L., Salmén, F., Vickovic, S., Lundmark, A., Navarro, J.F., Magnusson, J., Giacomello, S., Asp, M., Westholm, J.O., Huss, M., et al. (2016). Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. Science *353*, 78–82. https://doi.org/10.1126/science.aaf2403.

14. Chen, K.H., Boettiger, A.N., Moffitt, J.R., Wang, S., and Zhuang, X. (2015). RNA imaging. Spatially resolved, highly multiplexed RNA profiling in single cells. Science *348*, aaa6090.

15. Eng, C.L., Lawson, M., Zhu, Q., Dries, R., Koulena, N., Takei, Y., Yun, J., Cronin, C., Karp, C., Yuan, G.-C., et al. (2019). Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH. Nature *568*, 235–239. https://doi.org/10.1038/s41586-019-1049-y.

16. Wang, X., Allen, W.E., Wright, M.A., Sylwestrak, E.L., Samusik, N., Vesuna, S., Evans, K., Liu, C., Ramakrishnan, C., Liu, J., et al. (2018). Three-dimensional intact-tissue sequencing of single-cell transcriptional

states. Science *361*, eaat5691. https://doi.org/10.1126/science.aat5691.

17. Cho, C.-S., Xi, J., Si, Y., Park, S.-R., Hsu, J.-E., Kim, M., Jun, G., Kang, H.M., and Lee, J.H. (2021). Microscopic examination of spatial transcriptome using Seq-Scope. Cell *184*, 3559–3572.e22. https://doi.org/10.1016/j.cell.2021.05.010.

18. Fu, X., Sun, L., Dong, R., Chen, J.Y., Silakit, R., Condon, L.F., Lin, Y., Lin, S., Palmiter, R.D., and Gu, L. (2022). Polony gels enable amplifiable DNA stamping and spatial transcriptomics of chronic pain. Cell *185*, 4621–4633.e17. https://doi.org/10.1016/j.cell.2022.10.021.

19. Chen, A., Liao, S., Cheng, M., Ma, K., Wu, L., Lai, Y., Qiu, X., Yang, J., Xu, J., Hao, S., et al. (2022). Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. Cell *185*, 1777–1792.e21. https://doi.org/10.1016/j.cell.2022.04.003.

20. Xiao, Z., Cui, L., Yuan, Y., He, N., Xie, X., Lin, S., Yang, X., Zhang, X., Shi, P., Wei, Z., et al. (2024). 3D reconstruction of a gastrulating human embryo. Cell *187*, 2855–2874.e19. https://doi.org/10.1016/j.cell.2024.03.041.

21. Sampath Kumar, A., Tian, L., Bolondi, A., Hernández, A.A., Stickels, R., Kretzmer, H., Murray, E., Wittler, L., Walther, M., Barakat, G., et al. (2023). Spatiotemporal transcriptomic maps of whole mouse embryos at the onset of organogenesis. Nat. Genet. *55*, 1176–1185. https://doi.org/10.1038/s41588-023-01435-6.

22. Farah, E.N., Hu, R.K., Kern, C., Zhang, Q., Lu, T.-Y., Ma, Q., Tran, S., Zhang, B., Carlin, D., Monell, A., et al. (2024). Spatially organized cellular communities form the developing human heart. Nature *627*, 854–864. https://doi.org/10.1038/s41586-024-07171-z.

23. Cheng, M., Zheng, H., Fang, Q., Bai, Y., Liu, C., Pan, H., Zhang, Z., Lu, Q., Shi, C., Xia, T., et al. (2024). Three-dimension transcriptomics maps of whole mouse embryo during organogenesis. Preprint at bioRxiv. https://doi.org/10.1101/2024.08.17.608366.

24. Zeira, R., Land, M., Strzalkowski, A., and Raphael, B.J. (2022). Alignment and integration of spatial transcriptomics data. Nat. Methods *19*, 567–575. https://doi.org/10.1038/s41592-022-01459-6.

25. Ma, C., Chitra, U., Zhang, S., and Raphael, B.J. (2022). Belayer: modeling discrete and continuous spatial variation in gene expression from spatially resolved transcriptomics. Cell Syst. *13*, 786–797.e13. https://doi.org/10.1016/j.cels.2022.09.002.

26. Garcia-Alonso, L., Handfield, L.-F., Roberts, K., Nikolakopoulou, K., Fernando, R.C., Gardner, L., Woodhams, B., Arutyunyan, A., Polanski, K., Hoo, R., et al. (2021). Mapping the temporal and spatial dynamics of the human endometrium in vivo and in vitro. Nat. Genet. *53*, 1698–1711. https://doi.org/10.1038/s41588-021-00972-2.

27. Cang, Z., and Nie, Q. (2020). Inferring spatial and signaling relationships between cells from single cell transcriptomic data. Nat. Commun. *11*, 2084. https://doi.org/10.1038/s41467-020-15968-5.

28. Hou, R., Denisenko, E., Ong, H.T., Ramilowski, J.A., and Forrest, A.R.R. (2020). Predicting cell-to-cell communication networks using NATMI. Nat. Commun. *11*, 5011. https://doi.org/10.1038/s41467-020-18873-z.

29. Fischer, D.S., Schaar, A.C., and Theis, F.J. (2023). Modeling intercellular communication in tissues using spatial graphs of cells. Nat. Biotechnol. *41*, 332–336. https://doi.org/10.1038/s41587-022-01467-z.

30. Dries, R., Zhu, Q., Dong, R., Eng, C.L., Li, H., Liu, K., Fu, Y., Zhao, T., Sarkar, A., Bao, F., et al. (2021). Giotto: a toolbox for integrative analysis and visualization of spatial expression data. Genome Biol. *22*, 78. https://doi.org/10.1186/s13059-021-02286-2.

31. Liu, X., Zeira, R., and Raphael, B.J. (2023). Partial alignment of multislice spatially resolved transcriptomics data. Genome Res. *33*, 1124–1132. https://doi.org/10.1101/gr.277670.123.

32. Klein, D., Palla, G., Lange, M., Klein, M., Piran, Z., Gander, M., Meng-Papaxanthos, L., Sterr, M., Bastidas-Ponce, A., Tarquis-Medina, M., et al. (2023). Mapping cells through time and space with moscot. Preprint at bioRxiv. https://doi.org/10.1101/2023.05.11.540374.

33. Xia, C.-R., Cao, Z.-J., Tu, X.-M., and Gao, G. (2023). Spatial-linked alignment tool (SLAT) for aligning heterogenous slices. Nat. Commun. *14*, 7236. https://doi.org/10.1038/s41467-023-43105-5.

34. Clifton, K., Anant, M., Aihara, G., Atta, L., Aimiuwu, O.K., Kebschull, J.M., Miller, M.I., Tward, D., and Fan, J. (2023). STalign: alignment of spatial transcriptomics data using diffeomorphic metric mapping. Nat. Commun. *14*, 8123. https://doi.org/10.1038/s41467-023-43915-7.

35. Xu, H., Wang, S., Fang, M., Luo, S., Chen, C., Wan, S., Wang, R., Tang, M., Xue, T., Li, B., et al. (2023). SPACEL: deep learning-based characterization of spatial transcriptome architectures. Nat. Commun. *14*, 7603. https://doi.org/10.1038/s41467-023-43220-3.

36. Qiu, X., Zhang, Y., Martin-Rufino, J.D., Weng, C., Hosseinzadeh, S., Yang, D., Pogson, A.N., Hein, M.Y., Hoi Joseph Min, K., Wang, L., et al. (2022). Mapping transcriptomic vector fields of single cells. Cell *185*, 690–711.e45. https://doi.org/10.1016/j.cell.2021.12.045.

37. Zhang, M., Pan, X., Jung, W., Halpern, A.R., Eichhorn, S.W., Lei, Z., Cohen, L., Smith, K.A., Tasic, B., Yao, Z., et al. (2023). Molecularly defined and spatially resolved cell atlas of the whole mouse brain. Nature *624*, 343–354. https://doi.org/10.1038/s41586-023-06808-9.

38. Schott, M., León-Periñán, D., Splendiani, E., Strenger, L., Licha, J.R., Pentimalli, T.M., Schallenberg, S., Alles, J., Tagliaferro, S.S., Boltengagen, A., et al. (2023). Open-ST: high-resolution spatial transcriptomics in 3D. Preprint at bioRxiv. https://doi.org/10.1101/2023.12.22.572554.

39. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. Nat. Methods *15*, 1053–1058. https://doi.org/10.1038/s41592-018-0229-2.

40. Chen, T., and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '16 (Association for Computing Machinery), pp. 785–794. https://doi.org/10.1145/2939672.2939785.

41. Qiu, C., Martin, B.K., Welsh, I.C., Daza, R.M., Le, T.-M., Huang, X., Nichols, E.K., Taylor, M.L., Fulton, O., O'Day, D.R., et al. (2024). A single-cell time-lapse of mouse prenatal development from gastrula to birth. Nature *626*, 1084–1093. https://doi.org/10.1038/s41586-024-07069-w.

42. Wang, Q., Ding, S.-L., Li, Y., Royall, J., Feng, D., Lesnar, P., Graddis, N., Naeemi, M., Facer, B., Ho, A., et al. (2020). The Allen Mouse Brain Common coordinate framework: A 3D reference atlas. Cell *181*, 936–953.e20. https://doi.org/10.1016/j.cell.2020.04.007.

43. Chen, A., Sun, Y., Lei, Y., Li, C., Liao, S., Meng, J., Bai, Y., Liu, Z., Liang, Z., Zhu, Z., et al. (2023). Single-cell spatial transcriptome reveals cell-type organization in the macaque cortex. Cell *186*, 3726–3743.e24. https://doi.org/10.1016/j.cell.2023.06.009.

44. He, S., Bhatt, R., Brown, C., Brown, E.A., Buhr, D.L., Chantranuvatana, K., Danaher, P., Dunaway, D., Garrison, R.G., Geiss, G., et al. (2022). High-plex imaging of RNA and proteins at subcellular resolution in fixed tissue by spatial molecular imaging. Nat. Biotechnol. *40*, 1794–1806. https://doi.org/10.1038/s41587-022-01483-z.

45. Cang, Z., Zhao, Y., Almet, A.A., Stabell, A., Ramos, R., Plikus, M.V., Atwood, S.X., and Nie, Q. (2023). Screening cell–cell communication in spatial transcriptomics via collective optimal transport. Nat. Methods *20*, 218–228. https://doi.org/10.1038/s41592-022-01728-4.

46. Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., et al. (2005). Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc. Natl. Acad. Sci. USA *102*, 15545–15550. https://doi.org/10.1073/pnas.0506580102.

47. Rasmussen, C.E. (2004). Gaussian processes in machine learning. In Advanced Lectures on Machine Learning, ML Summer Schools, O. Bousquet, U. von Luxburg, and G. Rätsch, eds. (Tübingen, Germany: Springer), pp. 63–71. Revised Lectures. https://doi.org/10.1007/978-3-540-28650-9_4.

48. Bishop, C.M. (2016). Pattern Recognition and Machine Learning (Springer).

49. Chen, X., Fischer, S., Rue, M.C.P., Zhang, A., Mukherjee, D., Kanold, P.O., Gillis, J., and Zador, A.M. (2024). Whole-cortex in situ sequencing reveals input-dependent area identity. Nature, 1–10. https://doi.org/10.1038/s41586-024-07221-6.

50. Kiecker, C., and Lumsden, A. (2005). Compartments and their boundaries in vertebrate brain development. Nat. Rev. Neurosci. 6, 553–564. https://doi.org/10.1038/nrn1702.

51. Kiecker, C., and Lumsden, A. (2012). The role of organizers in patterning the nervous system. Annu. Rev. Neurosci. 35, 347–367. https://doi.org/10.1146/annurev-neuro-062111-150543.

52. Lai, H.C., Seal, R.P., and Johnson, J.E. (2016). Making sense out of spinal cord somatosensory development. Development 143, 3434–3448. https://doi.org/10.1242/dev.139592.

53. Cadwell, C.R., Bhaduri, A., Mostajo-Radji, M.A., Keefe, M.G., and Nowakowski, T.J. (2019). Development and arealization of the cerebral cortex. Neuron 103, 980–1004. https://doi.org/10.1016/j.neuron.2019.07.009.

54. Mah, C.K., Ahmed, N., Lopez, N.A., Lam, D.C., Pong, A., Monell, A., Kern, C., Han, Y., Prasad, G., Cesnik, A.J., et al. (2024). Bento: a toolkit for subcellular analysis of spatial transcriptomics data. Genome Biol. 25, 82. https://doi.org/10.1186/s13059-024-03217-7.

55. Zhang, M., Eichhorn, S.W., Zingg, B., Yao, Z., Cotter, K., Zeng, H., Dong, H., and Zhuang, X. (2021). Spatially resolved cell atlas of the mouse primary motor cortex by MERFISH. Nature 598, 137–143. https://doi.org/10.1038/s41586-021-03705-x.

56. Martinez-Ferre, A., and Martinez, S. (2012). Molecular regionalization of the diencephalon. Front. Neurosci. 6, 73. https://doi.org/10.3389/fnins.2012.00073.

57. Chiang, C., Litingtung, Y., Lee, E., Young, K.E., Corden, J.L., Westphal, H., and Beachy, P.A. (1996). Cyclopia and defective axial patterning in mice lacking Sonic hedgehog gene function. Nature 383, 407–413. https://doi.org/10.1038/383407a0.

58. Rash, B.G., and Grove, E.A. (2011). Shh and Gli3 regulate formation of the telencephalic-diencephalic junction and suppress an isthmus-like signaling source in the forebrain. Dev. Biol. 359, 242–250. https://doi.org/10.1016/j.ydbio.2011.08.026.

59. Vieira, C., Garda, A.-L., Shimamura, K., and Martinez, S. (2005). Thalamic development induced by Shh in the chick embryo. Dev. Biol. 284, 351–363. https://doi.org/10.1016/j.ydbio.2005.05.031.

60. Zeltser, L.M. (2005). Shh-dependent formation of the ZLI is opposed by signals from the dorsal diencephalon. Development 132, 2023–2033. https://doi.org/10.1242/dev.01783.

61. Pearse, R.V., 2nd, Collier, L.S., Scott, M.P., and Tabin, C.J. (1999). Vertebrate homologs of Drosophila suppressor of fused interact with the gli family of transcriptional regulators. Dev. Biol. 212, 323–336. https://doi.org/10.1006/dbio.1999.9335.

62. Guglielmi, L., Heliot, C., Kumar, S., Alexandrov, Y., Gori, I., Papaleonidopoulou, F., Barrington, C., East, P., Economou, A.D., French, P.M.W., et al. (2021). Smad4 controls signaling robustness and morphogenesis by differentially contributing to the Nodal and BMP pathways. Nat. Commun. 12, 6374. https://doi.org/10.1038/s41467-021-26486-3.

63. Guo, C., Xue, Y., Yang, G., Yin, S., Shi, W., Cheng, Y., Yan, X., Fan, S., Zhang, H., and Zeng, F. (2016). Nanog RNA-binding proteins YBX1 and ILF3 affect pluripotency of embryonic stem cells. Cell Biol. Int. 40, 847–860. https://doi.org/10.1002/cbin.10539.

64. Li, Q., Ramírez-Bergeron, D.L., Dunwoodie, S.L., and Yang, Y.-C. (2012). Cited2 gene controls pluripotency and cardiomyocyte differentiation of murine embryonic stem cells through Oct4 gene. J. Biol. Chem. 287, 29088–29100. https://doi.org/10.1074/jbc.M112.378034.

65. Jankovic, V., Ciarrocchi, A., Boccuni, P., DeBlasio, T., Benezra, R., and Nimer, S.D. (2007). Id1 restrains myeloid commitment, maintaining the self-renewal capacity of hematopoietic stem cells. Proc. Natl. Acad. Sci. USA 104, 1260–1265. https://doi.org/10.1073/pnas.0607894104.

66. Schultz, K.M., Banisadr, G., Lastra, R.O., McGuire, T., Kessler, J.A., Miller, R.J., and McGarry, T.J. (2011). Geminin-deficient neural stem cells exhibit normal cell division and normal neurogenesis. PLoS One 6, e17736. https://doi.org/10.1371/journal.pone.0017736.

67. Tamm, C., Böwer, N., and Annerén, C. (2011). Regulation of mouse embryonic stem cell self-renewal by a Yes-YAP-TEAD2 signaling pathway downstream of LIF. J. Cell Sci. 124, 1136–1144. https://doi.org/10.1242/jcs.075796.

68. Hou, Y., Li, W., Sheng, Y., Li, L., Huang, Y., Zhang, Z., Zhu, T., Peace, D., Quigley, J.G., Wu, W., et al. (2015). The transcription factor Foxm1 is essential for the quiescence and maintenance of hematopoietic stem cells. Nat. Immunol. 16, 810–818. https://doi.org/10.1038/ni.3204.

69. Piunti, A., and Shilatifard, A. (2021). The roles of Polycomb repressive complexes in mammalian development and cancer. Nat. Rev. Mol. Cell Biol. 22, 326. https://doi.org/10.1038/s41580-021-00341-1.

70. Hayward, P., Kalmar, T., and Martinez Arias, A.M. (2008). Wnt/Notch signalling and information processing during development. Development 135, 411. https://doi.org/10.1242/dev.000505.

71. Fu, L., Hu, Y., Song, M., Liu, Z., Zhang, W., Yu, F.-X., Wu, J., Wang, S., Izpisua Belmonte, J.C., Chan, P., et al. (2019). Up-regulation of FOXD1 by YAP alleviates senescence and osteoarthritis. PLOS Biol. 17, e3000201. https://doi.org/10.1371/journal.pbio.3000201.

72. Park, H.W., Kim, Y.C., Yu, B., Moroishi, T., Mo, J.-S., Plouffe, S.W., Meng, Z., Lin, K.C., Yu, F.-X., Alexander, C.M., et al. (2015). Alternative Wnt signaling activates YAP/TAZ. Cell 162, 780–794. https://doi.org/10.1016/j.cell.2015.07.013.

73. Moigneu, C., Abdellaoui, S., Ramos-Brossier, M., Pfaffenseller, B., Wollenhaupt-Aguiar, B., de Azevedo Cardoso, T., Camus, C., Chiche, A., Kuperwasser, N., Azevedo da Silva, R., et al. (2023). Systemic GDF11 attenuates depression-like phenotype in aged mice via stimulation of neuronal autophagy. Nat Aging 3, 213–228. https://doi.org/10.1038/s43587-022-00352-3.

74. Amet, L.E., Lauri, S.E., Hienola, A., Croll, S.D., Lu, Y., Levorse, J.M., Prabhakaran, B., Taira, T., Rauvala, H., and Vogt, T.F. (2001). Enhanced hippocampal long-term potentiation in mice lacking heparin-binding growth-associated molecule. Mol. Cell. Neurosci. 17, 1014–1024. https://doi.org/10.1006/mcne.2001.0998.

75. Radice, G.L., Rayburn, H., Matsunami, H., Knudsen, K.A., Takeichi, M., and Hynes, R.O. (1997). Developmental defects in mouse embryos lacking N-cadherin. Dev. Biol. 181, 64–78. https://doi.org/10.1006/dbio.1996.8443.

76. Li, H., Xu, L., Jiang, W., Qiu, X., Xu, H., Zhu, F., Hu, Y., Liang, S., Cai, C., Qiu, W., et al. (2023). Pleiotrophin ameliorates age-induced adult hippocampal neurogenesis decline and cognitive dysfunction. Cell Rep. 42, 113022. https://doi.org/10.1016/j.celrep.2023.113022.

77. Halperin, D., Stavsky, A., Kadir, R., Drabkin, M., Wormser, O., Yogev, Y., Dolgin, V., Proskorovski-Ohayon, R., Perez, Y., Nudelman, H., et al. (2021). CDH2 mutation affecting N-cadherin function causes attention-deficit hyperactivity disorder in humans and mice. Nat. Commun. 12, 6187. https://doi.org/10.1038/s41467-021-26426-1.

78. Pillai, A., Mansouri, A., Behringer, R., Westphal, H., and Goulding, M. (2007). Lhx1 and Lhx5 maintain the inhibitory-neurotransmitter status of interneurons in the dorsal spinal cord. Development 134, 357–366. https://doi.org/10.1242/dev.02717.

79. Pierani, A., Moran-Rivard, L., Sunshine, M.J., Littman, D.R., Goulding, M., and Jessell, T.M. (2001). Control of interneuron fate in the developing spinal cord by the progenitor homeodomain protein Dbx1. Neuron 29, 367–384. https://doi.org/10.1016/S0896-6273(01)00212-4.

80. Luu, B., Ellisor, D., and Zervas, M. (2011). The lineage contribution and role of Gbx2 in spinal cord development. PLoS One 6, e20940. https://doi.org/10.1371/journal.pone.0020940.

**Cell**
**Article**

**CellPress**
OPEN ACCESS

81. Kim, K.-T., Kim, N., Kim, H.-K., Lee, H., Gruner, H.N., Gergics, P., Park, C., Mastick, G.S., Park, H.-C., and Song, M.-R. (2016). ISL1-based LIM complexes control Slit2 transcription in developing cranial motor neurons. Sci. Rep. 6, 36491. https://doi.org/10.1038/srep36491.

82. Yeh, M.L., Gonda, Y., Mommersteeg, M.T.M., Barber, M., Ypsilanti, A.R., Hanashima, C., Parnavelas, J.G., and Andrews, W.D. (2014). Robo1 modulates proliferation and neurogenesis in the developing neocortex. J. Neurosci. 34, 5717–5731. https://doi.org/10.1523/JNEUROSCI.4256-13.2014.

83. Martinez, D., Zuhdi, N., Reyes, M., Ortega, B., Giovannone, D., Lee, V.M., and de Bellard, M.E. (2018). Screen for Slit/Robo signaling in trunk neural cells reveals new players. Gene Expr. Patterns 28, 22–33. https://doi.org/10.1016/j.gep.2018.01.002.

84. McDole, K., Guignard, L., Amat, F., Berger, A., Malandain, G., Royer, L.A., Turaga, S.C., Branson, K., and Keller, P.J. (2018). In Toto imaging and reconstruction of post-implantation mouse development at the single-cell level. Cell 175, 859–876.e33. https://doi.org/10.1016/j.cell.2018.09.031.

85. McFadden, D.G., Barbosa, A.C., Richardson, J.A., Schneider, M.D., Srivastava, D., and Olson, E.N. (2005). The Hand1 and Hand2 transcription factors regulate expansion of the embryonic cardiac ventricles in a gene dosage-dependent manner. Development 132, 189. https://doi.org/10.1242/dev.01562.

86. Dong, X., Wang, C., Zhang, J., Wang, S., Li, H., Kang, Y., Tian, S., and Fu, L. (2017). Cholecystokinin expression in the development of postinfarction heart failure. Cell. Physiol. Biochem. 43, 2479. https://doi.org/10.1159/000484454.

87. Imanaka-Yoshida, K. (2021). Tenascin-C in heart diseases-the role of inflammation. Int. J. Mol. Sci. 22, 5828. https://doi.org/10.3390/ijms22115828.

88. Kim, K.H., Nakaoka, Y., Augustin, H.G., and Koh, G.Y. (2018). Myocardial angiopoietin-1 controls atrial chamber morphogenesis by spatiotemporal degradation of cardiac jelly. Cell Rep. 23, 2455–2466. https://doi.org/10.1016/j.celrep.2018.04.080.

89. Steimle, J.D., and Moskowitz, I.P. (2017). TBX5: A key regulator of heart development. Curr. Top. Dev. Biol. 122, 195–221. https://doi.org/10.1016/bs.ctdb.2016.08.008.

90. Seya, D., Ihara, D., Shirai, M., Kawamura, T., Watanabe, Y., and Nakagawa, O. (2021). A role of Hey2 transcription factor for right ventricle development through regulation of Tbx2-Mycn pathway during cardiac morphogenesis. Dev. Growth Differ. 63, 82–92. https://doi.org/10.1111/dgd.12707.

91. Kodo, K., Shibata, S., Miyagawa-Tomita, S., Ong, S.-G., Takahashi, H., Kume, T., Okano, H., Matsuoka, R., and Yamagishi, H. (2017). Regulation of Sema3c and the interaction between cardiac neural crest and second heart field during outflow tract development. Sci. Rep. 7, 6771. https://doi.org/10.1038/s41598-017-06964-9.

92. Vedantham, V., Evangelista, M., Huang, Y., and Srivastava, D. (2013). Spatiotemporal regulation of an Hcn4 enhancer defines a role for Mef2c and HDACs in cardiac electrical patterning. Dev. Biol. 373, 149–162. https://doi.org/10.1016/j.ydbio.2012.10.017.

93. Tessari, A., Pietrobon, M., Notte, A., Cifelli, G., Gage, P.J., Schneider, M.D., Lembo, G., and Campione, M. (2008). Myocardial Pitx2 differentially regulates the left atrial identity and ventricular asymmetric remodeling programs. Circ. Res. 102, 813–822. https://doi.org/10.1161/CIRCRESAHA.107.163188.

94. de Boer, B.A., van den Berg, G., de Boer, P.A.J., Moorman, A.F.M., and Ruijter, J.M. (2012). Growth of the developing mouse heart: an interactive qualitative and quantitative 3D atlas. Dev. Biol. 368, 203–213. https://doi.org/10.1016/j.ydbio.2012.05.001.

95. Rivera-Feliciano, J., and Tabin, C.J. (2006). Bmp2 instructs cardiac progenitors to form the heart-valve-inducing field. Dev. Biol. 295, 580–588. https://doi.org/10.1016/j.ydbio.2006.03.043.

96. Chen, W., Liu, X., Li, W., Shen, H., Zeng, Z., Yin, K., Priest, J.R., and Zhou, Z. (2021). Single-cell transcriptomic landscape of cardiac neural crest cell derivatives during development. EMBO Rep. 22, e52389. https://doi.org/10.15252/embr.202152389.

97. Litviňuková, M., Talavera-López, C., Maatz, H., Reichart, D., Worth, C.L., Lindberg, E.L., Kanda, M., Polanski, K., Heinig, M., Lee, M., et al. (2020). Cells of the adult human heart. Nature 588, 466–472. https://doi.org/10.1038/s41586-020-2797-4.

98. Singh, R., Hoogaars, W.M., Barnett, P., Grieskamp, T., Rana, M.S., Buermans, H., Farin, H.F., Petry, M., Heallen, T., Martin, J.F., et al. (2012). Tbx2 and Tbx3 induce atrioventricular myocardial development and endocardial cushion formation. Cell. Mol. Life Sci. 69, 1377–1389. https://doi.org/10.1007/s00018-011-0884-2.

99. Habets, P.E.M.H., Moorman, A.F.M., Clout, D.E.W., van Roon, M.A., Lingbeek, M., van Lohuizen, M., Campione, M., and Christoffels, V.M. (2002). Cooperative action of Tbx2 and Nkx2.5 inhibits ANF expression in the atrioventricular canal: implications for cardiac chamber formation. Genes Dev. 16, 1234–1246. https://doi.org/10.1101/gad.222902.

100. Behrens, A.N. (2013). Nkx2-5 regulates Tdgf1 (Cripto) early during cardiac development. J. Clin. Exp. Cardiol 01. https://doi.org/10.4172/2155-9880.S11-003.

101. Ai, D., Liu, W., Ma, L., Dong, F., Lu, M.-F., Wang, D., Verzi, M.P., Cai, C., Gage, P.J., Evans, S., et al. (2006). Pitx2 regulates cardiac left-right asymmetry by patterning second cardiac lineage-derived myocardium. Dev. Biol. 296, 437–449. https://doi.org/10.1016/j.ydbio.2006.06.009.

102. Olson, E.N. (2006). Gene regulatory networks in the evolution and development of the heart. Science 313, 1922–1927. https://doi.org/10.1126/science.1132292.

103. Qiu, X., Mao, Q., Tang, Y., Wang, L., Chawla, R., Pliner, H.A., and Trapnell, C. (2017). Reversed graph embedding resolves complex single-cell trajectories. Nat. Methods 14, 979–982. https://doi.org/10.1038/nmeth.4402.

104. Lemons, D., and McGinnis, W. (2006). Genomic evolution of Hox gene clusters. Science 313, 1918–1922. https://doi.org/10.1126/science.1132040.

105. Janssen, R., Schomburg, C., Prpic, N.-M., and Budd, G.E. (2022). A comprehensive study of arthropod and onychophoran Fox gene expression patterns. PLoS One 17, e0270790. https://doi.org/10.1371/journal.pone.0270790.

106. Frank, L.H., and Rushlow, C. (1996). A group of genes required for maintenance of the amnioserosa tissue in Drosophila. Development 122, 1343–1352. https://doi.org/10.1242/dev.122.5.1343.

107. Moreno, E., and Morata, G. (1999). Caudal is the Hox gene that specifies the most posterior Drosophile segment. Nature 400, 873–877. https://doi.org/10.1038/23709.

108. Singh, N.P., and Mishra, R.K. (2014). Role of abd-A and Abd-B in development of abdominal epithelia breaks posterior prevalence rule. PLoS Genet. 10, e1004717. https://doi.org/10.1371/journal.pgen.1004717.

109. Johnson, D.M., and Andrew, D.J. (2019). Role of tbc1 in Drosophila embryonic salivary glands. BMC Mol. Cell Biol. 20, 19. https://doi.org/10.1186/s12860-019-0198-z.

110. Schroeder, W., Martin, K., and Lorensen, B. (2006). The visualization toolkit: an object-oriented approach to 3D Graphics (Ingram).

111. Sullivan, C., and Kaszynski, A.A. (2019). PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK). J. Open Source Software 4, 1450. https://doi.org/10.21105/joss.01450.

112. Lowe, D.G. (2004). Distinctive image features from scale-invariant Keypoints. Int. J. Comput. Vis. 60, 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

113. Tyszkiewicz, M., Fua, P., and Trulls, E. (2020). DISK: learning local features with policy gradient. Adv. Neural Inf. Process. Syst. 33, 14254–14265.

114. Kim, Y., Cheng, W., Cho, C.-S., Hwang, Y., Si, Y., Park, A., Schrank, M., Hsu, J.-E., Xi, J., Kim, M., et al. (2024). Seq-scope protocol: repurposing Illumina sequencing flow cells for high-resolution spatial transcriptomics. Preprint at bioRxiv. https://doi.org/10.1101/2024.03.29.587285.

115. Poovathingal, S., Davie, K., Vandepoel, R., Poulvellarie, N., Verfaillie, A., Corthout, N., and Aerts, S. (2024). Nova-ST: Nano-Patterned Ultra-Dense platform for spatial transcriptomics. Preprint at bioRxiv. https://doi.org/10.1101/2024.02.22.581576.

116. Shi, H., He, Y., Zhou, Y., Huang, J., Maher, K., Wang, B., Tang, Z., Luo, S., Tan, P., Wu, M., et al. (2023). Spatial atlas of the mouse central nervous system at molecular resolution. Nature 622, 552–561. https://doi.org/10.1038/s41586-023-06569-5.

117. Mah, C.K., Ahmed, N., Lam, D., Monell, A., Kern, C., Han, Y., Cesnik, A.J., Lundberg, E., Zhu, Q., Carter, H., et al. (2022). Bento: A toolkit for subcellular analysis of spatial transcriptomics data. Preprint at bioRxiv. https://doi.org/10.1101/2022.06.10.495510.

118. Wang, M., Hu, Q., Tu, Z., Kong, L., Yao, J., Xiang, R., Chen, Z., Zhao, Y., Zhou, Y., Yu, T., et al. (2024). A single-cell 3D spatiotemporal multi-omics atlas from Drosophila embryogenesis to metamorphosis. Preprint at bioRxiv. https://doi.org/10.1101/2024.02.06.577903.

119. Wang, M., Hu, Q., Lv, T., Wang, Y., Lan, Q., Xiang, R., Tu, Z., Wei, Y., Han, K., Shi, C., et al. (2022). High-resolution 3D spatiotemporal transcriptomic maps of developing Drosophila embryos and larvae. Dev. Cell 57, 1271–1283.e4. https://doi.org/10.1016/j.devcel.2022.04.006.

120. Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. In Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics Proceedings of Machine Learning Research, D. van Dyk and M. Welling, eds. (PMLR), pp. 567–574.

121. Hoffman, M.D., Blei, D.M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. J. Mach. Learn. Res.

122. Malandain, G., Bardinet, E., Nelissen, K., and Vanduffel, W. (2004). Fusion of autoradiographs with an MR volume using 2-D and 3-D linear transformations. Neuroimage 23, 111–127. https://doi.org/10.1016/j.neuroimage.2004.04.038.

123. Porchetto, R., Stramana, F., Paragios, N., and Ferrante, E. (2017). Rigid slice-to-volume medical image registration through markov random fields. In Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging (Springer International Publishing), pp. 172–185. https://doi.org/10.1007/978-3-319-61188-4_16.

124. Akmal Butt, M., and Maragos, P. (1998). Optimum design of chamfer distance transforms. IEEE Trans. Image Process. 7, 1477–1484. https://doi.org/10.1109/83.718487.

125. Akkiraju, N., Edelsbrunner, H., Facello, M., Fu, P., Mucke, E.P., and Varela, C. (1995). Alpha shapes: definition and software. In Proceedings of the 1st International Computational Geometry Software Workshop.

126. Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. IEEE Trans. Pattern Anal. Mach. Intell. 23, 1222. https://doi.org/10.1109/34.969114.

127. Komodakis, N., and Tziritas, G. (2007). Approximate labeling via graph cuts based on linear programming. IEEE Trans. Pattern Anal. Mach. Intell. 29, 1436–1453. https://doi.org/10.1109/TPAMI.2007.1061.

128. Saad, Y. (2003). Iterative Methods for Sparse Linear Systems, Second Edition (SIAM).

129. Epanechnikov, V.A. (1969). Non-parametric estimation of a multivariate probability density. Theory Probab. Appl. 14, 153–158. https://doi.org/10.1137/1114019.

130. Altman, N.S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. Am. Stat. 46, 175–185. https://doi.org/10.1080/00031305.1992.10475879.

131. Jin, S., Guerrero-Juarez, C.F., Zhang, L., Chang, I., Ramos, R., Kuan, C.-H., Myung, P., Plikus, M.V., and Nie, Q. (2021). Inference and analysis of cell-cell communication using CellChat. Nat. Commun. 12, 1088. https://doi.org/10.1038/s41467-021-21246-9.

132. Kanehisa, M., and Goto, S. (2000). KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res. 28, 27–30. https://doi.org/10.1093/nar/28.1.27.

133. Gillespie, M., Jassal, B., Stephan, R., Milacic, M., Rothfels, K., Senff-Ribeiro, A., Griss, J., Sevilla, C., Matthews, L., Gong, C., et al. (2022). The reactome pathway knowledgebase 2022. Nucleic Acids Res. 50, D687–D692. https://doi.org/10.1093/nar/gkab1028.

134. Türei, D., Korcsmáros, T., and Saez-Rodriguez, J. (2016). OmniPath: guidelines and gateway for literature-curated signaling pathway resources. Nat. Methods 13, 966–967. https://doi.org/10.1038/nmeth.4077.

135. Zepp, J.A., Zacharias, W.J., Frank, D.B., Cavanaugh, C.A., Zhou, S., Morley, M.P., and Morrisey, E.E. (2017). Distinct mesenchymal lineages and niches promote epithelial self-renewal and Myofibrogenesis in the lung. Cell 170, 1134–1148.e10. https://doi.org/10.1016/j.cell.2017.07.034.

136. Browaeys, R., Saelens, W., and Saeys, Y. (2020). NicheNet: modeling intercellular communication by linking ligands to target genes. Nat. Methods 17, 159–162. https://doi.org/10.1038/s41592-019-0667-5.

137. Cusanovich, D.A., Hill, A.J., Aghamirzaie, D., Daza, R.M., Pliner, H.A., Berletch, J.B., Filippova, G.N., Huang, X., Christiansen, L., DeWitt, W.S., et al. (2018). A single-cell atlas of in vivo mammalian chromatin accessibility. Cell 174, 1309–1324.e18. https://doi.org/10.1016/j.cell.2018.06.052.

138. Bruse, N., and van Heeringen, S.J. (2018). GimmeMotifs: an analysis framework for transcription factor motif analysis. Preprint at bioRxiv, 474403. https://doi.org/10.1101/474403.

139. Grün, D., Kester, L., and van Oudenaarden, A. (2014). Validation of noise models for single-cell transcriptomics. Nat. Methods 11, 637–640. https://doi.org/10.1038/nmeth.2930.

140. Eraslan, G., Simon, L.M., Mircea, M., Mueller, N.S., and Theis, F.J. (2019). Single-cell RNA-seq denoising using a deep count autoencoder. Nat. Commun. 10, 390. https://doi.org/10.1038/s41467-018-07931-2.

141. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2018). A general and flexible method for signal extraction from single-cell RNA-seq data. Nat. Commun. 9, 284. https://doi.org/10.1038/s41467-017-02554-5.

142. Cable, D.M., Murray, E., Zou, L.S., Goeva, A., Macosko, E.Z., Chen, F., and Irizarry, R.A. (2022). Robust decomposition of cell type mixtures in spatial transcriptomics. Nat. Biotechnol. 40, 517–526. https://doi.org/10.1038/s41587-021-00830-w.

143. Cable, D.M., Murray, E., Shanmugam, V., Zhang, S., Zou, L.S., Diao, M., Chen, H., Macosko, E.Z., Irizarry, R.A., and Chen, F. (2022). Cell type-specific inference of differential expression in spatial transcriptomics. Nat. Methods 19, 1076–1087. https://doi.org/10.1038/s41592-022-01575-3.

144. Kharchenko, P.V., Silberstein, L., and Scadden, D.T. (2014). Bayesian approach to single-cell differential expression analysis. Nat. Methods 11, 740. https://doi.org/10.1038/nmeth.2967.

145. Hoerl, A.E., and Kennard, R.W. (1970). Ridge regression: biased estimation for nonorthogonal problems. Technometrics 12, 55–67. https://doi.org/10.1080/00401706.1970.10488634.

146. Wrana, J.L., Attisano, L., Cárcamo, J., Zentella, A., Doody, J., Laiho, M., Wang, X.F., and Massagué, J. (1992). TGFβ signals through a heteromeric protein kinase receptor complex. Cell 71, 1003–1014. https://doi.org/10.1016/0092-8674(92)90395-s.

147. Sato, N., and Miyajima, A. (1994). Multimeric cytokine receptors: common versus specific functions. Curr. Opin. Cell Biol. 6, 174–179. https://doi.org/10.1016/0955-0674(94)90133-3.

148. Francis, K., and Palsson, B.O. (1997). Effective intercellular communication distances are determined by the relative time constants for cyto/

**Cell**
Article

*CellPress*
OPEN ACCESS

chemokine secretion and diffusion. Proc. Natl. Acad. Sci. USA *94*, 12258–12262. https://doi.org/10.1073/pnas.94.23.12258.

149. Geerts, A. (2001). History, heterogeneity, developmental biology, and functions of quiescent hepatic stellate cells. Semin. Liver Dis. *21*, 311–336. https://doi.org/10.1055/s-2001-17550.

150. Abbott, N.J., Rönnbäck, L., and Hansson, E. (2006). Astrocyte–endothelial interactions at the blood–brain barrier. Nat. Rev. Neurosci. *7*, 41–53. https://doi.org/10.1038/nrn1824.

151. Vunjak-Novakovic, G., Tandon, N., Godier, A., Maidhof, R., Marsano, A., Martens, T.P., and Radisic, M. (2010). Challenges in cardiac tissue engineering. Tissue Eng. Part B Rev. *16*, 169–187. https://doi.org/10.1089/ten.TEB.2009.0352.

152. Alberts, B.M., Johnson, A., and Lewis, J. (2002). Mol. Biol. Cell.

153. Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. J. R. Stat. Soc. *57*, 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x.

154. Hensman, J., Matthews, A., and Ghahramani, Z. (2014). Scalable Variational Gaussian Process Classification.

155. Gardner, J.R., Pleiss, G., Bindel, D., Weinberger, K.Q., and Wilson, A.G. (2018). GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration.

156. La Manno, G., Soldatov, R., Zeisel, A., Braun, E., Hochgerner, H., Petukhov, V., Lidschreiber, K., Kastriti, M.E., Lönnerberg, P., Furlan, A., et al. (2018). RNA velocity of single cells. Nature *560*, 494–498. https://doi.org/10.1038/s41586-018-0414-6.

157. Ma, J., Zhao, J., Tian, J., Bai, X., and Tu, Z. (2013). Regularized vector field learning with sparse approximation for mismatch removal. Pattern Recognit. *46*, 3519–3532. https://doi.org/10.1016/j.patcog.2013.05.017.

158. Levin, M. (2012). Morphogenetic fields in embryogenesis, regeneration, and cancer: non-local control of complex patterning. Biosystems. *109*, 243–261. https://doi.org/10.1016/j.biosystems.2012.04.005.

159. Gorban, A.N., Kégl, B., Wunsch, D.C., and Zinovyev, A. (2007). Principal Manifolds for Data Visualization and Dimension Reduction (Springer Science & Business Media).

160. Albergante, L., Mirkes, E., Bac, J., Chen, H., Martin, A., Faure, L., Barillot, E., Pinello, L., Gorban, A., and Zinovyev, A. (2020). Robust and scalable learning of complex intrinsic dataset geometry via ElPiGraph. Entropy *22*, 296. https://doi.org/10.3390/e22030296.

161. Qiu, C., Martin, B.K., Welsh, I.C., Daza, R.M., Le, T.-M., Huang, X., Nichols, E.K., Taylor, M.L., Fulton, O., O'Day, D.R., et al. (2023). A single-cell transcriptional timelapse of mouse embryonic development, from gastrula to pup. Preprint at bioRxiv. https://doi.org/10.1101/2023.04.05.535726.

162. Jing, Z., Zhu, Q., Li, L., Xie, Y., Wu, X., Fang, Q., Yang, B., Dai, B., Xu, X., Pan, H., et al. (2024). Spaco: A comprehensive tool for coloring spatial data at single-cell resolution. Patterns (N Y) *5*, 100915. https://doi.org/10.1016/j.patter.2023.100915.

163. Jing, Z., Yang, B., and Bai, Y. (2024). Protocol for enhancing visualization clarity for categorical spatial datasets using Spaco. Star Protoc. *5*, 103062. https://doi.org/10.1016/j.xpro.2024.103062.

164. Tickle, T., Tirosh, I., Georgescu, C., Brown, M., and Haas, B. (2019). inferCNV of the Trinity CTAT Project (Klarman Cell Observatory, Broad Institute of MIT and Harvard).

165. Russell, A.J.C., Weir, J.A., Nadaf, N.M., Shabet, M., Kumar, V., Kambhampati, S., Raichur, R., Marrero, G.J., Liu, S., Balderrama, K.S., et al. (2024). Publisher Correction: slide-tags single-nucleus barcoding for multimodal spatial genomics. Nature *625*, E11. https://doi.org/10.1038/s41586-023-06961-1.

166. Hagemann, A.I.H., and Scholpp, S. (2012). The tale of the three brothers – shh, Wnt, and Fgf during development of the thalamus. Front. Neurosci. *6*, 76. https://doi.org/10.3389/fnins.2012.00076.

167. Gupta, S., and Sen, J. (2016). Roof plate mediated morphogenesis of the forebrain: new players join the game. Dev. Biol. *413*, 145–152. https://doi.org/10.1016/j.ydbio.2016.03.019.

168. Wu, H.-H., Ivkovic, S., Murray, R.C., Jaramillo, S., Lyons, K.M., Johnson, J.E., and Calof, A.L. (2003). Autoregulation of neurogenesis by GDF11. Neuron *37*, 197–207. https://doi.org/10.1016/S0896-6273(02)01172-8.

# STAR★METHODS

## KEY RESOURCES TABLE

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| **Biological samples** | | |
| Adult mouse brain | This paper | N/A |
| Drosophila melanogaster w1118 | Tsinghua Fly Center | N/A |
| **Chemicals, peptides, and recombinant proteins** | | |
| Exonuclease I | NEB | Cat# M0293L |
| Exonuclease I buffer | NEB | Cat# M0293L |
| AMPURE XP beads | Beckman Coulter | Cat# A63881 |
| Agar | Vetec | Cat# V900500 |
| Propionic acid | Aladdin | Cat# P110444 |
| Phosphate acid | LingFeng, Shanghai | N/A |
| Bromophenol blue | Macklin | Cat# B802656 |
| Tissue-Teck OCT | Sakura | Cat# 4583 |
| Pepsin | Sigma | Cat# P7000 |
| 20X Saline-sodium citrate (SSC) | Thermo | Cat# AM9770 |
| Invitrogen™ Qubit ssDNA HS Reagent | Thermo | Cat#Q10212 |
| KAPA HiFi Hotstart Ready Mix | Roche | KK2602 |
| **Critical commercial assays** | | |
| Cornmeal-sucrose-agar Drosophila media | Hopebio | Cat# HB8590 |
| *Drosophila* incubator | Laifu | Cat# PGX-280A-3H |
| Cryostat | Leica | Cat# CM1950 |
| **Deposited data** | | |
| E9.5 mouse whole-embryo Stereo-seq dataset | Cheng et al.[23] | https://doi.org/10.1101/2024.08.17.608366 |
| E11.5 mouse whole-embryo Stereo-seq dataset | Cheng et al.[23] | https://doi.org/10.1101/2024.08.17.608366 |
| Processed data of mouse embryo Stereo-seq | This study | http://spateodata.aristoteleo.com/ |
| Non-small cell lung cancer CosMx data | He et al.[44] | https://doi.org/10.1038/s41587-022-01483-z |
| Mouse cortex MERFISH data | Zhang et al.[55] | https://doi.org/10.1038/s41587-022-01483-z |
| E9.5 mouse whole-embryo scRNA-seq dataset | Qiu et al.[41] | https://doi.org/10.1038/s41586-024-07069-w |
| E11.5 mouse whole-embryo scRNA-seq dataset | Chengxiang et al.[41] | https://doi.org/10.1038/s41586-024-07069-w |
| **Oligonucleotides** | | |
| Stereo-seq-TSO: CTGCTGACGTACTGAGAGGC/rG//rG//iXNA_G/ | This paper | N/A |
| cDNA PCR primer: CTGCTGACGTACTGAGAGGC | This paper | N/A |
| Stereo-seq-library-F: /5phos/CTGCTGACGTACTGAGAGG*C*A | This paper | N/A |
| Stereo-seq-library-R: GAGACGTTCTCGACTCAGCAGA | This paper | N/A |
| Stereo-seq-library-splint-oligo: GTACGTCAGCAGGAGACGTTCTCG | This paper | N/A |
| **Software and algorithms** | | |
| *Spateo (version: 1.1.0)* | This paper | https://github.com/aristoteleo/spateo-release |
| *Spateo-viewer* | This paper | https://github.com/aristoteleo/spateo-viewer?tab=readme-ov-file |
| *Spateo-viewer website* | This paper | http://viewer.spateo.aristoteleo.com/ |
| ***dynamo*** (version: 1.4.1) | This paper | https://github.com/aristoteleo/dynamo-release |
| paste-bio (version: 1.4.0) | Zeira et al.[24] | https://github.com/raphael-group/paste |

*(Continued on next page)*

***Continued***

| REAGENT or RESOURCE | SOURCE | IDENTIFIER |
|---|---|---|
| PASTE2 (version: commit 517d6584) | Liu et al.[31] | https://github.com/raphael-group/paste2 |
| Moscot (version: 0.3.4.dev3+gf71f976) | Klein et al.[32] | https://github.com/theislab/moscot |
| scSLAT (version: 0.2.2) | Xia et al.[33] | https://github.com/gao-lab/SLAT |
| STalign (version: 1.0) | Clifton et al.[34] | https://github.com/JEFworks-Lab/STalign |
| SPACEL (version: 1.1.8) | Xu et al.[35] | https://github.com/QuKunLab/SPACEL |
| opencv-python (version: 4.9.0.80) | OpenCV team | https://opencv.org/ |
| kornia (version: 0.7.2) | Kornia AI organization | https://github.com/kornia/kornia |
| sc3D (version: 1.1.0) | Sampath Kumar et al.[21] | https://github.com/GuignardLab/sc3D |
| thin-plate-spline (version: 1.1.1) | N/A | https://github.com/raphaelreme/tps |
| Belayer | Ma et al.[25] | https://doi.org/10.5281/zenodo.6970061 |
| Commot (version: 0.0.3) | Cang et al.[45] | https://github.com/zcang/COMMOT |
| NCEM (version: 0.1.4) | Fischer et al.[29] | https://github.com/theislab/ncem |

## EXPERIMENTAL MODEL AND STUDY PARTICIPANT DETAILS

We focus on computational modeling in this study. Detailed experimental information for 3D mouse embryo profiling and Drosophila embryo profiling can be found in Cheng et al.[23] and Wang.[118]

## METHOD DETAILS

### The 3D spatial transcriptomics Stereo-seq dataset of whole embryons at E9.5 and E11.5

Detailed experimental procedures for embryonic sectioning, Serial Block-Face Imaging (SBFI), and sequencing can be found in Cheng et al.[23]

### Alignment of spatial transcriptomics profilings of serial sections to create 3D models at whole embryo level

Most existing spatial transcriptomics methods offer 2D spatially resolved transcriptomes but tissues, organs and embryos are 3D entities with unique spatial organizations and functionalities. Leveraging the ultra-high field of view of Stereo-seq or others, we can perform continuous slicing of the same tissue, organ or embryos (such as *Drosophila*[119] or even whole mouse embryos[19]). However, after the slicing and sequencing, the relative coordinates of the cells across sections are lost. In Spateo, we introduce a probabilistic model for aligning spatial transcriptomic slices to create aligned 3D point clouds at the whole-embryo scale, enabling the construction of 3D models and performing various downstream volumetric and morphometric analyses.

#### *Problem definition of ST alignment*

Consider two spatially-resolved transcriptomics samples, such as two consecutive tissue sections from the same embryo or two well-aligned 3D embryos from two different developmental stages (applicable to all the following discussion), $S^I = \{\mathbf{X}^I, \mathbf{Z}^I\}$, where $I \in \{A, B\}$ denotes samples $A$ and $B$ respectively, $\mathbf{X}^I \in \mathbb{R}^{N_I \times D}$ denotes the $D$-dimensional spatial coordinates of $N_I$ spots (or cells, applicable to all the following discussion) in sample $I$ where $D$ can be either 2 or 3, and $\mathbf{Z}^I \in \mathbb{R}^{N_I \times G}$ corresponds to $G$ features, *e.g.*, genes, of the measured readout at those spots. Our goal is to align the two samples such that corresponding spots between samples have similar readout while the spatial distributions of spots are also preserved across samples. In this section, we assume that sample $B$ is the reference and that sample $A$ will be aligned to the coordinate system of sample $B$ by a transformation $\mathcal{T}$. In what follows, we present a Bayesian generative model for aligning ST data that is robust, efficient, and capable of performing flexible partial alignment, non-rigid deformations. This model can additionally be used to jointly align multiple sections all at once via a multi-slice refinement mode to alleviate error accumulation of sequential alignment of many slices, *e.g.*, about 90 slices for our whole 3d embryonic dataset.

#### *Generative process*

In Spateo, we suppose the spots in model or source sample $A$ are model points and spots in reference or target sample $B$ are data points, and sample $B$ is generated from the transformed sample $A$, *i.e.*, $\mathcal{T}(S^A) = \{\mathcal{T}(\mathbf{X}^A), \mathbf{Z}^A\}$. Note that the reverse order of generation works too (although may resulted in different mappings as the hypothesized generation process is directional) and can be achieved by swap sample $A$ and sample $B$. Considering the partial overlapping or outliers resulting from missing tissue regions between samples or scattered spots due to technical issues during tissue sectioning or library preparation that ST experiments may encounter, we assume the following *i.i.d.* generative process:

For a spot in sample $B$, $s_n^B \in S^B$ is selected to be generated as an outlier with probability $1 - \gamma$ or as an inlier generated from sample $A$ with the probability $\gamma$. Next we describe the generation of outliers and inliers respectively.

*Outlier generation.* We suppose $s_n^B$ is generated from an outlier distribution $p_o(\mathbf{x}_n^B, \mathbf{z}_n^B)$. As the pattern of outliers is highly diverse and is independent of the sample $A$, it is difficult to describe them with a specific distribution. A general and reasonable assumption is that they follow a uniform distribution $1/a$, where the output space is a bounded region that covers the spatial coordinates $\mathbf{X}^B$ and $a$ is a constant denoting the volume of the region.

*Inlier generation.* Assume spot $s_n^B$ can be generated by a set of spots from the transformed sample $A$, we select an index $m \in \{1, 2, \cdots, N_A\}$ among all possible spots from sample $A$, indicating $s_n^B$ is generated by $s_m^A$, with a weighted probability $\alpha_m$, where $\sum_{m=1}^{N_A} \alpha_m = 1$. Then, for spatial coordinates generation, we suppose the position of $\mathbf{x}_n^B$ follows a Gaussian centered on $\mathcal{T}(\mathbf{x}_m^A)$ with an isotropic variance $\sigma^2$, *i.e.*, $\mathbf{x}_n^B \sim \mathcal{N}(\mathbf{x}_n^B | \mathcal{T}(\mathbf{x}_m^A), \sigma^2 \mathbf{I}_D)$; for transcriptomic readout generation, we define a generalized form of $\mathbf{z}_n^B \sim p(\mathbf{z}_n^B | \mathbf{z}_m^A)$, which can be described by many distributions, such as Gaussian and negative binomial; for modeling other available information generation processes such as images, labels, and other modalities, *e.g.*, chromatin accessibility or proteomics, we can similarly augment a likelihood to other features. By repeating the above steps $N_B$-times, we can generate sample $B$ from sample $A$.

With the generative process defined above, the likelihood is a combination of a two-component mixture distribution (inlier-outlier selection) and $N_A$-component mixture distribution (generation spot selection). For clarity, we first introduce some notations. Assume $\mathbf{C} = \{c_1, c_2, \cdots, c_{N_B}\} \in \{0, 1\}^{N_B}$ is the vector of indicator variables, where $c_n = 1$ indicates $s_n^B$ is an inlier and comes from inlier distribution, otherwise, an outlier. $\mathbf{E} = \{e_1, e_2, \cdots, e_{N_B}\} \in \{1, 2, \cdots, N_A\}^{N_B}$ is the vector of indicating index variables, where $e_n = m$ means that $s_n^B$ is generated by $s_m^A$. The joint distribution of $(\mathbf{x}_n^B, \mathbf{z}_n^B, c_n, e_n)$ given $(\mathbf{X}^A, \mathbf{Z}^A, \sigma^2, \boldsymbol{\alpha}, \gamma)$ is as follows:

$$p(\mathbf{x}_n^B, \mathbf{z}_n^B, c_n, e_n | \mathbf{X}^A, \mathbf{Z}^A, \mathcal{T}, \sigma^2, \boldsymbol{\alpha}, \gamma) = \left\{ (1-\gamma) p_o(\mathbf{x}_n^B, \mathbf{z}_n^{\mathbf{B}}) \right\}^{(1-c_n)} \left\{ \gamma \prod_{m=1}^{N_A} \left( \alpha_m \mathcal{N}(\mathbf{x}_n^B | \mathcal{T}(\mathbf{x}_m^A), \sigma^2 \mathbf{I}_D) \cdot p_z(\mathbf{z}_n^B | \mathbf{z}_m^A) \right)^{\delta_m(e_n)} \right\}^{c_n}$$

where $\delta_m(e_n)$ is an indicator function, with a value of 1 if $e_n = m$ and 0 otherwise.

### Transformation Model

The transformation model $\mathcal{T}(\mathbf{x})$ from the above equation provides a spatial constraint that acts as a regularization. We decompose the transformation model into rigid and non-rigid transformations as $\mathcal{T}(\mathbf{x}) = \mathcal{R}(\mathbf{x}) + \mathbf{f}(\mathbf{x})$, where $\mathcal{R}$ and $\mathbf{f}$ denote the rigid and non-rigid transformations, respectively.

**Rigid transformation** can roughly align the general coordinate system, which is a simple combination of rotation with translation: $\mathcal{R}(\mathbf{x}) = \mathbf{x}\mathbf{R}^\top + \mathbf{t}$, where $\mathbf{R} \in SO(D)$ denotes a rotation matrix and $\mathbf{t}$ is a translation vector where SO stands for the special orthogonal group.

**Non-rigid transformation** can align localized distortions that may arise from tissue deformation during tissue sectioning and library preparation procedure. We use a Gaussian process model to represent this nonlinear transformation: $f(\mathbf{x}) = \mathcal{GP}(\mu, k)$, where $\mu : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is a mean function and $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^D$ is a semi-positive definite covariance function (also known as a kernel function). In Spateo, the default kernel is squared exponential (SE) kernel with the form $k(\mathbf{x}, \mathbf{x}') = \exp(-\beta \|\mathbf{x} - \mathbf{x}'\|^2)$.

### Prior Distributions

Prior distributions describe beliefs about the model variables before the inference. As described above, we place a GP prior over the latent function to describe non-rigid transformation:

$$p(\mathbf{f} | \mathbf{X}^A) \sim \mathcal{N}(\mathbf{f} | \mu(\mathbf{X}^A), \mathbf{K}_{NN}), \mathbf{f} = f(\mathbf{x}^A)$$

where $\mathbf{K}_{NN}$ is the covariance function evaluated between all the points. Since we don't know the overall offset in advance, we simply set the the mean function to $\mu(\mathbf{X}^A) = 0$. Note that it does not reduce the generalizability as long as the coordinate is correctly transformed or normalized.

The variable $\gamma$ from the above joint distribution controls the probability of occurrence of the outlier and inlier models. We define its prior as a Beta distribution:

$$p(\gamma) = \text{Beta}(\gamma | B_a, B_b)$$

where $B_a$ and $B_b$ are hyperparameters that control the shape of the Beta distribution.

The variable $\boldsymbol{\alpha}$ from the above joint distribution defines the probability that each model spot will generate the target spot. We define its prior as a Dirichlet distribution:

$$p(\boldsymbol{\alpha}) = \text{Dir}(\boldsymbol{\alpha} | \kappa 1_{N_A}),$$

where $\kappa > 0$ is a hyperparameter that controls the shape of the Dirichlet distribution and $1_{N_A}$ is the vector of all 1 of size $N_A$.

For other parameters $(\mathbf{X}^A, \mathbf{Z}^A, \mathcal{R}, \sigma^2)$, we introduce noninformative priors and treat them under the flat likelihood assumption.

### Full joint Distribution

By incorporating the prior distribution into the likelihood distribution, we obtain the full joint distribution as follows:

$$p(\mathcal{S}^A, \mathcal{S}^B, \boldsymbol{\theta}) \propto p(\mathbf{f} | \mathbf{X}^A) p(\gamma) p(\boldsymbol{\alpha}) \prod_{n=1}^{N_B} p(\mathbf{x}_n^B, \mathbf{z}_n^B, c_n, e_n | \mathbf{X}^A, \mathbf{Z}^A, \mathbf{f}, \sigma^2, \boldsymbol{\alpha}, \gamma)$$

## Cell
### Article

where $\theta = (\mathbf{f}, \mathcal{R}, \mathbf{C}, \mathbf{E}, \sigma^2, \boldsymbol{\alpha}, \gamma)$ is the set of latent variables.

### Variational Bayesian Inference

In the previous section, we assumed a generative process between the two samples, however, what we are really interested in is how to get a reasonable estimate of the posterior distribution of the hidden variables $\theta$ in the generative process. These posterior distributions may help biologists gain insights into biological processes in a data-driven manner, *e.g.*, $p(e_n | S^A, S^B)$ describes from which cell/spot in $S^A$ the $s_n^B \in S^B$ is most likely generated; $p(\mathbf{f} | S^A, S^B)$ and $p(\mathcal{R} | S^A, S^B)$ indicate the non-rigid and rigid transformations of $S^B$ mapped to $S^A$, respectively; $p(\boldsymbol{\alpha} | S^A, S^B)$ may reveal the probability of proliferation and apoptosis of the cells in $S^A$, etc. To this end, we develop a variational Bayesian algorithm to estimate these posterior distributions.

### Inference

One approach to solving the problem of getting estimates of the posterior distribution is to compute a reasonable estimate of $\theta$, such as the expectation of $\theta$ over $p(\theta | \mathcal{D})$, or the maximum mode of $p(\theta | \mathcal{D})$, where $\mathcal{D} = (S^A, S^B)$ is the observed data. However, the exact estimation is computational intractable and approximation is needed. Thus we introduce a variational posterior $q(\theta)$ to directly approximate the posterior $p(\theta | \mathcal{D})$ such that the Kullback-Leibler (KL) divergence between $p(\theta | \mathcal{D})$ and $q(\theta)$ is minimized.[48] The minimization is known to be able to equivalently expressed as maximization of the lower bound for the true log marginal likelihood (ELBO) as follows[48]:

$$\mathcal{L}(q) = \int q(\theta) \log\left(\frac{p(\theta, \mathcal{D})}{q(\theta)}\right) d\theta.$$

### Mean-field factorization

If there is no constraint imposed on $q(\theta)$, then the above optimization problem is still intractable since the optimal solution of this optimization problem is $\widehat{q}(\theta) = p(\theta | \mathcal{D})$, which is the exact posterior. Thus, we use the *mean field approximation* to factorize the distribution $q(\theta)$ into three independent groups as follows:

$$q(\theta) = q_1(\mathbf{f}, \gamma, \boldsymbol{\alpha}) q_2(\mathbf{C}, \mathbf{E}) q_3(\sigma^2, \mathcal{R})$$

In addition, we assume $q_3(\sigma^2, \mathcal{R}) = \delta(\sigma^2, \mathcal{R})$ for simplification, where $\delta(\sigma^2, \mathcal{R})$ denotes a Dirac delta function, *i.e.*, $q_3$ is the distribution with a point mass at $\sigma^2$ and $\mathcal{R}$. In the following, for brevity of notation, we use $q_1$, $q_2$, and $q_3$ to denote $q_1(\mathbf{f}, \gamma, \boldsymbol{\alpha})$, $q_2(\mathbf{C}, \mathbf{E})$, and $q_3(\sigma^2, \mathcal{R})$ respectively.

With the assumed independent factorization, maximizing ELBO can then be achieved by optimizing each of the factors independently. For $q_i$, the maximization has the form:

$$\widehat{q}_i = \arg\max_{q_i} \int q_i \log\left(\frac{\exp\left\{\mathbb{E}_{j \neq i}[\log p(\theta, \mathcal{D})]\right\}}{q_i}\right) d\theta_i,$$

where

$$\mathbb{E}_{j \neq i}[\log p(\theta, \mathcal{D})] = \int \log p(\theta, \mathcal{D}) \prod_{j \neq i} q_j d\theta_j$$

denotes the expectation over $q_j$ for $j \neq i$. Note that above equation is the negative KL divergence between $q_j$ and $\exp\left\{\mathbb{E}_{j \neq i}[\log p(\theta, \mathcal{D})]\right\}$ and thus the optimal $q_i$ is obtained:

$$\log \widehat{q}_i = \mathbb{E}_{j \neq i}[\log p(\theta, \mathcal{D})] + const.$$

### Sparse Inducing Variable Approximation

Typically, inference involving Gaussian process model possesses $\mathcal{O}(N^3)$ computational complexity where $N$ is the dataset size. However, with the increasing scale of current ST technology, reaching 3D whole embryo level, such as Stereo-seq, where a single slice contains hundreds of thousands of spots and embryos reconstructed from dozens of those slices contain millions of number of spots, it is difficult to use GP model directly on large-scale data without downsampling. To this end, in Spateo, we introduce inducing variables to drastically reduce the computational complexity of the GP model to $\mathcal{O}(Nm^2)$, where $m$ is the number of inducing variables. To be specific, we introduce $m$ pseudo-inputs $\mathbf{X}_m$ lying in the coordinate space of $\mathbf{X}^A$, and their inducing variables $\mathbf{u}$, where $\mathbf{u}$ is similar to $\mathbf{f}$ having the same GP prior that $p(\mathbf{u}) \sim \mathcal{N}(0, \mathbf{K}_{mm})$. Our goal is to use the inducing variables $\mathbf{u}$ to approximate the $q(\mathbf{f})$. For convenience, the locations of the inducing points, *i.e.*, $\mathbf{X}_m$, are $m$ points randomly selected from the $\mathbf{X}^A$. Following,[120] we suppose $\mathbf{u}$ is a sufficient statistic for $\mathbf{f}$, *i.e.*, for any value $\mathbf{u}$, $p(z | \mathbf{f}, \mathbf{u}) = p(z | \mathbf{u})$ holds. After incorporating inducing variables, the mean-field factorization of the first term becomes

$$q_1(\mathbf{f}, \mathbf{u}, \gamma, \boldsymbol{\alpha}) = p(\mathbf{f} | \mathbf{u}) q_1(\mathbf{u}, \gamma, \boldsymbol{\alpha}),$$

where $p(\mathbf{f} | \mathbf{u}) \sim \mathcal{N}(\mathbf{f} | \mathbf{U}\mathbf{G}^{-1}\mathbf{u}, \Gamma - \mathbf{U}\mathbf{G}^{-1}\mathbf{U}^\top) = \mathcal{N}(\mathbf{f} | \mu, \Gamma_{\mathbf{f}|\mathbf{u}})$ according to the formula of the conditional Gaussian, $\mathbf{U} = \mathbf{K}_{nm}$ is the cross-kernel matrix evaluating the kernel function between the inducing points $\mathbf{X}_m$ and model points position $\mathbf{X}^A$, $\mathbf{G} = \mathbf{K}_{mm}$, and $\Gamma = \mathbf{K}_{nn}$ are the kernel matrix between all inducing points and model points position respectively.

### Maximize Lower Bound

**Updating** $q_1$ involves non-rigid deformation, *i.e.*, finding the optimal posterior GP given the $q_2$, and $q_3$. In addition, the generation probabilities of the outliers and each model point are also updated.

Suppose $q_2(\mathbf{C}, \mathbf{E})$ and $q_3(\sigma^2, \mathcal{R})$ are given and denote $p_{mn} = \mathbb{E}[c_n \delta_m(e_n)]$ and $\widehat{N} = \sum_{m=1}^{N_A} \sum_{n=1}^{N_B} p_{mn}$, the bound becomes

$$\mathcal{L} = \int q_1 q_2 q_3 \log \int \frac{\overbrace{p(\mathbf{f}|\mathbf{u})}\, p(\mathbf{u})p(\gamma)p(\alpha)P(\mathbf{X}^B, \mathbf{Z}^B, \mathbf{C}, \mathbf{E}|\mathbf{X}^B, \mathbf{Z}^B, \mathbf{f}, \sigma^2, \alpha, \gamma)}{\overbrace{p(\mathbf{f}|\mathbf{u})}\, q_1(\mathbf{u}, \gamma, \alpha)q_2 q_3} dq_1 dq_2 dq_3$$

$$= \int q_1 \log p(\mathbf{u}) dq_1 + \int q_1 \log p(\gamma) dq_1 + \int q_1 \log p(\alpha) dq_1 - \int q_1 \log q_1 dq$$

$$+ \log(1-\gamma)\sum_{n=1}^{N_B}(1-c_n) + \log\gamma\sum_{n=1}^{N_B}c_n + \sum_{n=1}^{N_B}\sum_{m=1}^{N_A} p_{mn}\log\alpha_m$$

$$+ \sum_{n=1}^{N_B}\sum_{m=1}^{N_A} p_{mn}\log\mathcal{N}\left(\mathbf{x}_n^B | \mathbf{x}_m^A \mathbf{R}^\top + \mathbf{t} + \mathbf{f}_{(m)}, \sigma^2\right).$$

$$+ \log(1-\gamma)\sum_{n=1}^{N_B}(1-c_n) + \log\gamma\sum_{n=1}^{N_B}c_n + \sum_{n=1}^{N_B}\sum_{n=1}^{N_A} p_{mn}\log\alpha_m$$

$$+ \sum_{n=1}^{N_B}\sum_{n=1}^{N_A} p_{mn}\log\mathcal{N}\left(\mathbf{x}_n^B | \mathbf{x}_m^A \mathbf{R}^\top + \mathbf{t} + \mathbf{f}_{(m)}, \sigma^2\right).$$

We find that $\mathcal{L}$ can be factorized into a sum of terms containing only $(\mathbf{f}, \mathbf{u})$, $\gamma$, and $\alpha$ respectively. Therefore, we can further decompose $q_1$ into the marginal of each factor

$q_1(\mathbf{f}, \mathbf{u}, \gamma, \alpha) = p(\mathbf{f}|\mathbf{u})\phi(\mathbf{u})q_1(\gamma)q_1(\alpha)$. Focusing on the terms involving $\mathbf{u}$, $\gamma$, and $\alpha$ separately, without detailed derivation, we can infer that the optimal posterior distribution of $\widehat{\phi}(\mathbf{u})$, $\widehat{p}_1(\gamma)$, and $\widehat{p}_1(\alpha)$ obeys a Gaussian distribution, Beta distribution, and Dirichlet distribution, respectively. More specifically, we have

$$\widehat{\phi}(\mathbf{u}) \sim \mathcal{N}(\mathbf{u}|\mu_u, \mathbf{A}_u)$$
$$\widehat{q}_1(\gamma) = \text{Beta}(\gamma|\widehat{N} + B_a, N_B - \widehat{N} + B_b)$$
$$\widehat{q}_1(\alpha) = \text{Dir}(\alpha|\kappa 1_{N_A} + \mathbf{K})$$

where

$$\mu_u = \sigma^{-2}\mathbf{G}\Sigma\mathbf{U}^\top\left(\mathbf{P}\mathbf{X}^B - \text{d}(\mathbf{K})\mathbf{R}^A\right)$$
$$\mathbf{A} = \mathbf{G}\Sigma\mathbf{G}$$
$$\Sigma = \left(\mathbf{G} + \sigma^{-2}\mathbf{U}^\top \text{d}(\mathbf{K})\mathbf{R}^A\right)^{-1}$$
$$\mathbf{R}^A = \mathbf{X}^A\mathbf{R}^\top + \mathbf{t}$$

$\mathbf{K} = \mathbf{P}1_{N_B} \in \mathbb{R}^{N_A \times 1}$, $\text{d}(\cdot)$ denotes $\text{diag}(\cdot)$. Note that we can recover $q(\mathbf{f})$ from $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})\phi(\mathbf{u})$ by marginalizing out $\mathbf{u}$, which gives the optimal $\widehat{q}(\mathbf{f}) \sim \mathcal{N}(\mathbf{f}|\mu_f, \mathbf{A}_f)$, where $\mu_f = \mathbf{U}\mathbf{G}^{-1}\mu_f$ and $\mathbf{A}_f = \Gamma_{\mathbf{f}|\mathbf{u}} + \mathbf{U}\Sigma\mathbf{U}^\top$.

With $\widehat{q}_1(\mathbf{f})$, $\widehat{q}_1(\gamma)$, and $\widehat{q}_1(\alpha)$, we are able to compute $\langle \mathcal{N}_{mn}^{\mathbf{X}}\rangle = \exp\left(\mathbb{E}\left[\log\mathcal{N}\left(\mathbf{x}_n^B|\mathbf{r}_m^A + \mathbf{f_m}, \sigma^2\right)\right]\right)$, $\langle\gamma\rangle = \exp\left(\mathbb{E}[\log\gamma]\right)$, $\langle 1-\gamma\rangle = \exp\left(\mathbb{E}[\log(1-\gamma)]\right)$, and $\langle\alpha_m\rangle = \exp\left(\mathbb{E}[\log\alpha_m]\right)$ according to the property of their distributions. We present them without detailed derivation as follows:

$$\langle\gamma\rangle = \exp\{\psi(B_a + \widehat{N}) - \psi(B_a + B_b + N_B)\}$$

$$\langle 1-\gamma\rangle = \exp\{\psi(B_b + N - \widehat{N}) - \psi(B_a + B_b + N_B)\}$$

$$\langle\alpha_m\rangle = \exp\{\psi(\kappa + \mathbf{K}_m) - \psi(\kappa N_A + \widehat{N})\}$$

$$\langle\mathcal{N}_{mn}^{\mathbf{X}}\rangle = \mathcal{N}\left(\mathbf{x}_n^B|\mathbf{r}_m^A + \mathbf{f}_m^A, \sigma^2\right)\exp\left(-\frac{1}{2\sigma^2}\mathbf{A}_{fmm}\right)$$

where $\mathbf{A}_{fmm}$ is the $m$-th element of the diagonal of the matrix $\mathbf{A}_f$ and $\psi()$ is the digamma function which is defined as the logarithmic derivative of the gamma function.

**Updating** $q_2$ involves solving for the optimal posterior generative probability, *i.e.*, the probabilistic relationship of matching points between slices $S^A$ and $S^B$. Suppose $q_1$ and $q_3$ are given, we obtain

$$\log\widehat{q}_2(\mathbf{C}, \mathbf{E}) = \sum_{n=1}^{N_B}\left\{(1-c_n)\log\left(\frac{\langle 1-\gamma\rangle}{a}\right) + c_n\log\langle\gamma\rangle + \sum_{m=1}^{N_A}c_n\delta_m(e_n)\left(\log\langle\alpha_m\rangle + \log\langle\mathcal{N}_{mn}^{\mathbf{X}}\rangle + \log p_z(\mathbf{z}_n^B|\mathbf{z}_m^A)\right)\right\}$$

We see that $\hat{q}_2(\mathbf{C}, \mathbf{E})$ is factorized into $\hat{q}_2(\mathbf{C}, \mathbf{E}) = \prod_{n=1}^{N_B} \hat{q}_2^{(n)}(c_n, e_n)$. Then, taking the exponential of $\log \hat{q}_2^{(n)}(c_n, e_n)$, we have

$$\hat{q}_2^{(n)}(c_n, e_n) \propto \left( \frac{\langle 1 - \gamma \rangle}{a} \right)^{(1-c_n)} \prod_{m=1}^{N_A} \left( \langle \gamma \rangle \langle \alpha_m \rangle \langle \mathcal{N}_{mn}^{\mathbf{X}} \rangle p_z(\mathbf{z}_n^B | \mathbf{z}_{\mathbf{m}}^{\mathbf{A}}) \right)^{c_n \delta_m(e_n)}$$

The normalization constant of $\hat{q}_2^{(n)}(c_n, e_n)$ is obtained by the summation of the right-hand side of the above equation for all pairs of $c_n \in \{0, 1\}$ and $e_n \in \{1, 2, \cdots, N_A\}$ as follows:

$$\langle 1 - \gamma \rangle \Big/ a + \langle \gamma \rangle \sum_{m=1}^{N_A} \langle \alpha_m \rangle \langle \mathcal{N}_{mn}^{\mathbf{X}} \rangle p_z(\mathbf{z}_n^B | \mathbf{z}_m^A).$$

Thus, we obtain the closed-form expression of $\hat{q}_2^{(n)}(c_n, e_n)$:

$$\hat{q}_2^{(n)}(c_n, e_n) = \left( 1 - \mathbf{K}_n' \right)^{1-c_n} \left( \prod_{m=1}^{N_A} p_{mn}^{\delta_m(e_n)} \right)^{c_n},$$

where

$$p_{mn} = \frac{\langle \gamma \rangle \langle \alpha_m \rangle \langle \mathcal{N}_{mn}^{\mathbf{X}} \rangle p_z(\mathbf{z}_n^B | \mathbf{z}_{\mathbf{m}}^{\mathbf{A}})}{\langle 1 - \gamma \rangle \Big/ a + \langle \gamma \rangle \sum_{m=1}^{N_A} \langle \alpha_m \rangle \langle \mathcal{N}_{mn}^{\mathbf{X}} \rangle p_z(\mathbf{z}_n^B | \mathbf{z}_m^A)},$$

$\mathbf{K}' = \mathbf{P}^\top 1_{N_A} \in \mathbb{R}^{N_B \times 1}$. We see that $\hat{q}_2^{(n)}(c_n, e_n)$ is a combination of a Bernoulli distribution and a categorical distribution, and we obtain:

$$\mathbb{E}(c_n \delta_m(e_n)) = p_{mn},$$

$$\mathbb{E}(c_n) = \sum_{m=1}^{N_A} p_{mn} = \mathbf{K}_n'.$$

Updating $q_3$ this term involves the estimation of rigid transformation and noise. Suppose $q_1$ and $q_2$ are given. Since $q_3(\sigma^2, \mathcal{R})$ is a Dirac delta function, which is characterized only by its mode. Therefore, we directly maximize the lower bound without using the general solution of variational Bayesian inference. Taking the expectation of the logarithm of the full joint distribution over $q_1$, $q_2$ and $q_3$, the lower bound of the model evidence can be obtained as follows:

$$\mathcal{L} \sim -\frac{1}{2\sigma^2} \int \mathcal{N}(\mathbf{f}|\mu_{\mathbf{f}}, \mathbf{A}_{\mathbf{f}}) \left( \mathbf{f} - d(\mathbf{K})^{-1} \left( \mathbf{P}\mathbf{X}^B - d(\mathbf{K})\mathbf{R}^A \right) \right)^\top d(\mathbf{K}) \left( \mathbf{f} - d(\mathbf{K})^{-1} \left( \mathbf{P}\mathbf{X}^B - d(\mathbf{K})\mathbf{R}^A \right) \right) - \frac{1}{2} \log \sigma^2 \sum_{n=1}^{N_B} \sum_{m=1}^{N_A} p_{mn}$$

Taking the derivative of $\sigma^2$ and equating it to zero, we obtain a closed-form expression:

$$\hat{\sigma}^2 = \frac{1}{\widetilde{N}} \int \mathcal{N}(\mathbf{f}|\mu_{\mathbf{f}}, \mathbf{A}_{\mathbf{f}}) \left( \mathbf{f} - d(\mathbf{K})^{-1} \left( \mathbf{P}\mathbf{X}^B - d(\mathbf{K})\mathbf{R}^A \right) \right)^\top d(\mathbf{K}) \left( \mathbf{f} - d(\mathbf{K})^{-1} \left( \mathbf{P}\mathbf{X}^B - d(\mathbf{K})\mathbf{R}^A \right) \right)$$

$$= \frac{1}{\widetilde{N}} \left( \left( \mu_{\mathbf{f}} - d(\mathbf{K})^{-1} \left( \mathbf{P}\mathbf{X}^B - d(\mathbf{K})\mathbf{R}^A \right) \right)^\top d(\mathbf{K}) \left( \mu_{\mathbf{f}} - d(\mathbf{K})^{-1} \left( \mathbf{P}\mathbf{X}^B - d(\mathbf{K})\mathbf{R}^A \right) \right) + \text{Tr}(d(\mathbf{K})\mathbf{A}_{\mathbf{f}}) \right)$$

Focus on $\mathcal{R}$, we have

$$\mathcal{L} \sim \sum_{n=1}^{N_B} \sum_{m=1}^{N_A} p_{mn} \left( \mathbf{x}_n^B - \mathbf{x}_m^A \mathbf{R}^\top - \mathbf{t} - \mathbf{f}_m \right)^\top \left( \mathbf{x}_n^B - \mathbf{x}_m^A \mathbf{R}^\top - \mathbf{t} - \mathbf{f}_m \right)$$

Taking partial derivative with respect to $t$ and equate it to zero, we obtain

$$\mathbf{t} = \frac{1}{\widetilde{N}} \left( \mathbf{K}'\mathbf{X}^B - \mathbf{K}\mathbf{f} - \mathbf{K}\mathbf{X}^A \mathbf{R}^\top \right) = \mu_{\mathbf{X}^B} - \mu_{\mathbf{f}} - \mu_{\mathbf{X}^A} \mathbf{R}^\top$$

where $\mu_{\mathbf{X}^B} = \frac{1}{\widetilde{N}} \mathbf{K}'\mathbf{X}^B$, $\mu_{\mathbf{f}} = \frac{1}{\widetilde{N}} \mathbf{K}\mathbf{f}$, $\mu_{\mathbf{X}^A} = \frac{1}{\widetilde{N}} \mathbf{K}\mathbf{X}^A$. Substituting $\mathbf{t}$ back, we obtain

$$\mathcal{L} \sim \sum_{n=1}^{N_B} \sum_{m=1}^{N_A} p_{mn} \left( \overline{\mathbf{x}}_n^B - \overline{\mathbf{x}}_m^A \mathbf{R}^\top - \overline{\mathbf{f}}_m \right)^\top \left( \overline{\mathbf{x}}_n^B - \overline{\mathbf{x}}_m^A \mathbf{R}^\top - \overline{\mathbf{f}}_m \right) = -\frac{1}{\sigma^2} \text{Tr}\left( \left( \overline{\mathbf{f}}^\top d(\mathbf{K}) - \overline{\mathbf{X}}^{B\top} \mathbf{P}^\top \right) \overline{\mathbf{X}}^A \mathbf{R}^\top \right)$$

where $\overline{\mathbf{x}}_n^B = \mathbf{x}_n^B - \mu_{\mathbf{X}^B}$, $\overline{\mathbf{x}}_m^A = \mathbf{x}_m^A - \mu_{\mathbf{X}^A}$, $\overline{\mathbf{f}}_m = \mathbf{f}_m - \mu_{\mathbf{f}}$. Afterwards, we obtain the optimal rotation matrix $\mathbf{R}$ using SVD method:

$$\mathbf{R} = \mathbf{UCV}^\top,$$

where $\mathbf{U}SS\mathbf{V} = \mathrm{svd}(\mathbf{A})$, $\mathbf{C} = \mathrm{diag}(1, \cdots, 1, \det(\mathbf{UV}^\top))$, and $\mathbf{A} = (\overline{\mathbf{f}}^\top d(\mathbf{K}) - \overline{\mathbf{X}}^{B\top}\mathbf{P}^\top)\overline{\mathbf{X}}^A$.

### Stochastic Variational Inference

Although Spateo introduces inducing variables to reduce the computational cost drastically, it is still not easy to scale to embryo-scale data. This is because we still need to traverse the entire data for each coordinate update. To this end, in Spateo, we further utilized stochastic variational inference (SVI)[121] to perform coordinate ascent, which is very similar to SGD that conducts stochastic optimization using noisy yet computationally inexpensive gradient estimates. As performing SVI requires global variables, we use $\mathbf{u}$, $\gamma$, and $\alpha$ to fill this role. With SVI, the time complexity of Spateo further reduced to $\mathcal{O}(bN_A)$ with batch size $b$, making Spateo linear with respect to the number of points.

### Guiding Optimization

We find that the above alignment is somehow sensitive to initialization, especially for the partial overlapping cases, as the search space is large and the optimization problem is highly non-convex. To address this limitation, we introduce a guiding optimization strategy that initializes a coarse alignment and subsequently optimizes it to be more robust.

#### Initialize with Putative Correspondences

We use $K$-nearest neighbor algorithm for the gene expression feature to initialize the putative correspondences on the basis of similarity between gene expression profiles of spots/cells between slices. As gene expression feature is often noisy, the $K$ can be set to a high value, e.g., 5. Also to avoid many-to-one correspondence, we calculate the mutual nearest neighbors for the spots between two slices.

#### Rigid Transformation from Noisy Putative Set

Now suppose we have obtained the putative set $\mathcal{P} = \{\mathbf{c}_i = (\mathbf{x}_i^C, \mathbf{y}_i^C)\}_{i=1}^N$ using the $K$-nearest search, where $\mathbf{x}_i^C$ and $\mathbf{y}_i^C$ are the spatial coordinates in two slices respectively, which includes many outliers. Our goal is to find a coarse rigid transformation $\mathbf{R}^C$ and $\mathbf{t}^C$ from the noisy putative set. We formulate this problem in a *Bayesian* framework:

$$p(\mathbf{c}_i) = (1 - \gamma)\frac{1}{(2\pi\sigma^2)^{D/2}}e^{-\frac{\|\mathbf{y}_i^C - \mathbf{x}_i^C\mathbf{R}^{C\top} - \mathbf{t}^C\|^2}{2\sigma^2}} + \gamma\frac{1}{a}$$

where $\gamma$, $\sigma^2$, $D$, and $a$ have the same meaning in the section "generative process." Then the EM algorithm can be used to derive the optimal MLE solution, which is similar in Dynamo or SparseVFC. **E-step**:

$$p_i^c = \frac{e^{-\frac{\|\mathbf{y}_i^C - \mathbf{x}_i^C\mathbf{R}^{C\top} - \mathbf{t}^C\|^2}{2\sigma^2}}}{e^{-\frac{\|\mathbf{y}_i^C - \mathbf{x}_i^C\mathbf{R}^{C\top} - \mathbf{t}^C\|^2}{2\sigma^2}} + \frac{\gamma\left((2\pi\sigma^2)^{D/2}\right)}{(1-\gamma)a}}$$

**M-step** minimizing the expectation of the complete negative log-likelihood function as

$$\sum_i^N p_i^c\frac{\|\mathbf{y}_i^C - \mathbf{x}_i^C\mathbf{R}^{C\top} - \mathbf{t}^C\|^2}{2\sigma^2} + \frac{D}{2}\log\sigma^2\sum_i^N p_i^c,$$

Thus we can update $\mathbf{R}^C$ and $\mathbf{t}^C$ by the SVD method. First solve the optimal translation:

$$\mathbf{t}^C = \frac{\mathbf{P}^{C\top}(\mathbf{Y}^C - \mathbf{X}^C\mathbf{R}^{C\top})}{\mathbf{P}^{C\top}\mathbf{1}} = \mu_{\mathbf{Y}^c} - \mu_{\mathbf{X}^c}\mathbf{R}^{C\top}$$

And solve the rotation matrix:

$$\mathbf{R}^C = \mathbf{UCV}^\top$$

where $\mathbf{U}SS\mathbf{V} = \mathrm{svd}(\mathbf{A})$, $\mathbf{C} = \mathrm{diag}(1, \ldots, 1, \det(\mathbf{UV}^\top))$, and $\mathbf{A} = \mathbf{Y}^{C\top}\mathrm{diag}\left(\mathbf{P}^C\right)\mathbf{X}^C$.

In the end, we update $\sigma^2$:

$$\sigma^2 = \frac{\sum_i^N p_i^c\|\mathbf{y}_i^C - \mathbf{x}_i^C\mathbf{R}^{C\top} - \mathbf{t}^C\|^2}{D\mathbf{P}^{C\top}\mathbf{1}}$$

After convergence, we get the posterior $\mathbf{P}^C$ and $\mathbf{R}^C$, $\mathbf{t}^C$ as well, which can be utilized for guide optimization.

### Guide Optimization

After the above step, we obtain a consensus set using the results of $\mathbf{P}^C$ as $\mathcal{I} = \{i|p_i^c > 0.5, p_i^c \in \mathbf{P}^C\}$. The consensus set can guide the following alignment step. When we optimize the rigid transformation, we can use the consensus set as a regularization term:

$$\min_{\mathbf{R},\mathbf{t}} \frac{1}{2\sigma^2} \sum_{n=1}^{N_B} \sum_{m=1}^{N_A} p_{mn} \|\mathbf{x}_m^A \mathbf{R}^\top + \mathbf{t} + \mathbf{f}_m - \mathbf{x}_n^B\|^2 + \lambda \sum_{i \in \mathcal{I}} p_i^c \|\mathbf{x}_i^C \mathbf{R}^\top + \mathbf{t} - \mathbf{y}_i^C\|^2$$

where $\lambda$ is a regularization parameter that controls the degree of regularization. We note that as $\sigma^2$ becomes small, the regularization term actually becomes smaller, this makes the guidance weaker. We also note that the regularization term does not affect the non-rigid transformation. Next, we derive the new solutions for the rigid transformation $\mathbf{R}$ and $\mathbf{t}$ in update of q3 from section Maximize Lower Bound, where we update the formulation of t and A:

$$\widehat{\mathbf{t}} = \frac{(1^\top \mathbf{P} \mathbf{X}^B - 1^\top \mathbf{P}^\top \mathbf{f} - 1^\top \mathbf{P}^\top \mathbf{X}^A \mathbf{R}^\top) + 2\lambda\sigma^2 (\mathbf{P}^{C\top} \mathbf{Y}^C - \mathbf{P}^{C\top} \mathbf{X}^C \mathbf{R}^\top)}{\widehat{N} + 2\lambda\sigma^2 \mathbf{P}^{C\top} 1}$$

$$\mathbf{A} = \left[ \mathbf{X}^{A\top} \mathbf{P} 1 \mathbf{t} + \mathbf{X}^{A\top} \mathrm{diag}(\mathbf{P}1)\mathbf{f} - \mathbf{X}^{A\top} \mathbf{P} \mathbf{X}^B + 2\lambda\sigma^2 (\mathbf{X}^{C\top} \mathbf{P}^C \mathbf{t} - \mathbf{X}^{C\top} \mathrm{diag}(\mathbf{P}^C) \mathbf{Y}^C) \right]^\top$$

The above sections introduce the mathematical foundation of the pairwise alignment. In the Spateo package, the pairwise alignment of consecutive slices can be achieved by simply calling the following function:

```
align_slices, pis = st.align.morpho_align(
    models=slices,
    spatial_key="spatial",
    key_added="align_spatial",
    device="0", # or cpu
)
```

### Joint modeling multiple slices

The pairwise align in Spateo described above is robust, efficient, and accurate, capable of reconstructing embryos containing millions of spots in a few minutes, in a sequential-run fashion, i.e., $S^1 \leftarrow S^2 \cdots \leftarrow S^L$. However, the sequential strategy often suffers from the risk of error propagation. If an error occurs in some two-slice alignment, this error will propagate through the entire embryo. Moreover, a global offset may arise if the number of slices is large, due to error accumulation. To this end, Spateo jointly models multiple slices, rather than limiting to pairs of slices, and performs multi-slice refinement to eliminate the error propagation problem. By considering multiple slices, Spateo is able to use more information and thus better handle cases where some part of a slice is missing due to experimental imperfections, as other slices containing this region can provide complementary information.

Consider a set of spatially-resolved transcriptomics samples with order $\mathcal{G} = \{S_1, \cdots, S_L\}$ where $S^i = \{\mathbf{X}^i, \mathbf{Z}^i\}$ and the index $i$ is stored in order. Our goal now becomes to recover a series of transformations $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_L\}$, such that for all $i$, $\mathcal{T}_i(S^i)$ should align neighboring $\mathcal{T}_j(S^j)$ in spatial and phenotypic readout. We assume that a slice is generated from the neighboring slices and consider the neighboring slices to be model points and this selected slice to be data points. The inlier generative process is slightly different from that of pairwise alignment. In inlier generation, we add an extra step of selecting the generation slice: we select an index $k \in \mathcal{N}_i \neq i$, indicating $s_n^i$ is generated from slice $S^k$, with a probability $\omega_k^i$, where $\mathcal{N}_i$ is the neighborhood of slice $i$ and $\sum_{k \in \mathcal{N}_i \neq i} \omega_k^i = 1$.

In addition to the inlier generation, technically the multi-slice refinement is very similar to the pairwise refinement, so we don't provide a detailed derivation.

In the Spateo package, we can perform the multi-slice refinement by the following function:

```
align_slices_global = st.align.morpho_multi_refinement(
    models=align_slices,
    spatial_key="align_spatial",
    key_added="align_spatial_global",
    device="0", # or cpu
    neighbor_size=2,
)
```

## Mesh Correction
### *Motivation and background*
Spateo's pairwise-multi alignment scheme is able to reconstruct a 3D embryo that is coherent and smooth in terms of geometry and gene expression across the 3D space. However, it should be noted that the reconstructed result is sometimes slightly incompatible with the actual structure. In fact, it is due to the so-called "Banana problem" or "Z-shift problem", namely a 3D curved object cannot be exactly reconstructed from cross-sections without any additional information.[122] Fundamentally, this is because we implicitly assume the alignment across slices is perpendicular to slices and it thus doesn't allow alignment to form a curved structure. Note that the problem is more significant when the slices are farther apart, or when the slices exhibit a non-centered structure, e.g., a curved banana or half-brain. This problem can be solved by inserting fiducial markers that are obtained by sticking a straight probe in the embryo before slicing. These fiducial markers can be tracked across each slice, thus allowing recovery of the original geometry structure of the embryo. However, this experimental method also introduces errors if the probe and the cutting plane of the slice are not orthogonal. Furthermore, probe holes may sometimes collapse, making them difficult to trace, and the holes may also destroy part of the tissue of interest in the slice. In Spateo, we design an algorithm to accurately recover the true anatomy structure by incorporating additional shape geometry information without destroying the full embryo.

### *Problem formulation*
Given a set of spatial transcriptomic slices $\mathcal{G} = \{S_1, \cdots, S_L\}$ aligned by the above generative probabilistic model with their corresponding heights $\mathbf{z} = [z_1, z_2, \cdots, z_L]^\top$, recorded during the experiment, and separately a morphological 3D mesh of the entire tissue/organ/embryo $\mathcal{M}$ produced with imaging or other means, we wish to eliminate the z-shift error of each slice caused by the alignment by integrating the morphological information of the mesh. To utilize this information, the first step is to align the mesh $\mathcal{M}$ and the slices set $\mathcal{G}$, i.e., find a similarity transformation $\mathcal{T}$, that transforms the mesh $\mathcal{M}$ into the coordinate space of $\mathcal{G}$ to best align the slices $\mathcal{G}$, which can be formulated as

$$\widehat{\mathcal{T}} = \arg\min{}_{\mathcal{T}} S(\mathcal{T}(\mathcal{M}); \mathcal{G}, \mathbf{z}),$$

where $\mathcal{T}(\mathcal{M})$ transforms the morphological shape $\mathcal{M}$ by the transformation $\mathcal{T}$, and $S$ is a loss function that measures the similarity between the transformed morphological shape $\mathcal{T}(\mathcal{M})$ and the slices set $\mathcal{G}$ with $\mathbf{z}$ heights. After the alignment, it is much easier to calculate the z-shift error for each slice and thus perform z-shift correction, where a series of z-shift translations (translation on the same z-plane) $\mathbf{t} = \{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_L\}$ are estimated for each slice.

### *Method*
*Finding Optimal Transformation as Discrete Labeling Problem.* Our goal is to find the optimal similarity transformation, which can be partitioned into seven parameters: $\theta = (r_x, r_y, r_z, t_x, t_y, t_z, s)$, where the first three parameters $(r_x, r_y, r_z)$ are the rotation angles along the X-Y-Z axes, the next two parameters $(t_x, t_y)$ are the translation in the X-Y plane, the next parameter $t_z$ is the translation in the Z direction, and the last parameter $s$ is the scaling factor. We wish the z-shifts of slices will not affect the estimation of the optimal transformation, and therefore the loss function S should be designed to be translation invariant, which will be introduced in detail later. In such a case, it is meaningless to optimize the X-Y translation parameters $(t_x, t_y)$. We set the $(t_x, t_y)$ to the mean value of all points on slices, i.e.,

$$(t_x, t_y) = \frac{1}{L} \sum_{i=1}^{L} \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}$$

Therefore, the number of parameters to be optimized is reduced to five: $\theta = (r_x, r_y, r_z, t_z, s)$, reducing the search space.
$\theta = (r_x, r_y, r_z, t_z, s)$

Next, we formulate the problem of finding the optimal $\theta$ as a discrete labeling problem using pairwise Markov Random Field (MRF),[123] which discretes the continuous parameters space and evaluates the pairwise energy or loss, and then determines the optimal parameters combination. A benefit of such a strategy is that it is less likely to be trapped in the optimization. Specifically, a fully connected graph of parameters $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is first constructed, where each node $v_i \in \mathcal{V}$ associates a parameter in $\theta$, resulting in $|\mathcal{V}| = 5$ nodes. Since $\mathcal{G}$ is fully connected, the edge set $\mathcal{E} = \{(v_i, v_j)\}$, $\forall i \neq j$. As a discrete labeling problem, each node $v_i$ is assigned a discrete label $l_i$ from the label space $\mathcal{L}$. In our problem, the label space $\mathcal{L}$ is defined as the set of all possible values of each parameter in $\theta$. We will discuss the discrete strategy for each parameter in detail later.

Typically, a discrete labeling problem on a pairwise MRF is associated with a unary potential function $u_i(l_i)$ and a binary potential function $b_{ij}(l_i, l_j)$, that minimizing the following energy function:

$$E(\mathbf{L}; U, B) = \sum_{i \in \mathcal{V}} u_i(l_i) + \sum_{(ij) \in \mathcal{E}} b_{ij}(l_i, l_j),$$

where $\mathbf{L} = \{l_1, \cdots, l_n\}$ is the label vector of all nodes, $U = \{u_i(\cdot)\}$ is the unary potential function associated to $v_i \in \mathcal{V}$, and $B = \{b_{ij}(\cdot, \cdot)\}$ is the binary potential function associated to edge $(v_i, v_j) \in \mathcal{E}$. The unary potential function $u_i(l_i)$ measures the cost of assigning label $l_i$ to node $v_i$, and the binary potential function $b_{ij}(l_i, l_j)$ measures the cost of assigning label $l_i$ to node $v_i$ and label $l_j$ to node $v_j$ simultaneously. Note that these potential functions return scalar values, and the lower the value, the better the label assignment. The optimal label assignment $\mathbf{L}^*$ is the one that minimizes the energy function, i.e., $\mathbf{L}^* = \arg\min_{\mathbf{L}} E(\mathbf{L}; U, B)$.

Discrete strategy of the labeling space directly affects the accuracy and running time of the optimization. We discretize the continuous solution space as the labeling space. For linear parameters $\theta \in (r_x, r_y, r_z, t_z)$, the labeling space is defined as: $\mathcal{L}_\theta = \theta + \{0, \pm\frac{\theta_{max}}{\kappa}, \pm\frac{2\theta_{max}}{\kappa}, \cdots, \pm\frac{\kappa\theta_{max}}{\kappa}\}$, where $\theta_{max}$ is a user-defined parameter that defines the maximum deviation of $\theta$, and $\kappa$ controls the number of labels for each parameter. For the scaling factor $s$, the labeling space is defined as:

$$\mathcal{L}_s = \exp\left\{\log s + \left\{0, \pm\frac{\log s_{max}}{\kappa}, \pm\frac{2\log s_{max}}{\kappa}, \cdots, \pm\frac{\kappa\log s_{max}}{\kappa}\right\}\right\}.$$

The pairwise term is defined as $b_{ij}(l_i, l_j) = S(\mathcal{T}_\theta(\mathcal{M}; l_i, l_j); \mathcal{G}, \mathbf{z})$, where $\mathcal{T}_\theta(\cdot; l_i, l_j)$ represents the transformation with the parameters $\theta$ and the labels $l_i$ and $l_j$ assigned to $v_i$ and $v_j$, respectively, and the similarity function. For the unary term, we assume that there is no prior knowledge about the optimal label assignment, and thus set the unary potential function as $u_i(l_i) = 0$, $\forall i \in \mathcal{V}$. In this case, the energy function $E(\mathbf{L}; U, B)$ is simplified as:

$$E(\mathbf{L}; B) = \sum_{(ij) \in \mathcal{E}} S(\mathcal{T}_\theta(\mathcal{M}; l_i, l_j); \mathcal{G}, \mathbf{z}).$$

*Translation-invariant similarity function.* Similarity function $S$ measures the similarity between the transformed morphological shape $\mathcal{T}(\mathcal{M})$ and the slices $\mathcal{G}$ on $\mathbf{z}$ heights. It can be decomposed as follows:

$$S(\mathcal{T}(\mathcal{M}); \mathcal{G}, \mathbf{z}) = \sum_{i=1}^{L} S_i(\mathcal{T}(\mathcal{M}); S_i, z_i),$$

where $S_i$ is the similarity function between the transformed morphological shape $\mathcal{T}(\mathcal{M})$ and the $i$-th slice $S_i$ on the $i$-th height $z_i$. To simplify the problem, we cut the morphological shape with the plane of $z = z_i$ to get a pseudo-contour and extract the contour of slice (described in detail later), turning $S_i$ into a contour similarity problem.

Measuring the similarity between two contours is much easier, and we can use for example the Chamfer distance[124] as the similarity function, which is defined as

$$CD(\mathcal{C}_1, \mathcal{C}_2) = \frac{1}{|\mathcal{C}_1|} \sum_{\mathbf{x} \in \mathcal{C}_1} \min_{\mathbf{y} \in \mathcal{C}_2} \|\mathbf{x} - \mathbf{y}\|_2^2 + \frac{1}{|\mathcal{C}_2|} \sum_{\mathbf{y} \in \mathcal{C}_2} \min_{\mathbf{x} \in \mathcal{C}_1} \|\mathbf{x} - \mathbf{y}\|_2^2,$$

where $\mathcal{C}_1$ and $\mathcal{C}_2$ denote two de-meaning contours, and $|\mathcal{C}_1|$, $|\mathcal{C}_2|$ are the number of points in $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. The Chamfer distance is symmetric and translation invariant, and it is also easy to calculate. Nevertheless, a well-known limitation for chamfer distance is its sensitivity to outliers and difficulty in handling partial overlap cases, e.g., some slices are only partially observed (measured). To address this issue, we propose to use the robust Iterative Closest Point or ICP algorithm, an outlier robust point cloud registration algorithm, to measure the similarity between two contours.

The calculation of robust ICP is simple, and can be summarized as follows. Given two contours $\mathcal{C}_1$ and $\mathcal{C}_2$, it first finds the closest point $\mathbf{y}_i$ in $\mathcal{C}_2$ for each point $\mathbf{x}_i$ in $\mathcal{C}_1$ within a manually set threshold. Second, it estimates the translation $\mathbf{t}$ that minimizes:

$$\widehat{\mathbf{t}} = \min\arg_{\mathbf{t}} \sum_{i=1}^{|\mathcal{C}_1|} \|\mathbf{x}_i - (\mathbf{y}_i + \mathbf{t})\|_2^2.$$

Then it translates $\mathcal{C}_2$ by $\widehat{\mathbf{t}}$, and repeats the above steps until convergence. In the end, the similarity between $\mathcal{C}_1$ and $\mathcal{C}_2$ is defined as the number of inliers divided by the number of points in $\mathcal{C}_1$, which can be expressed as:

$$S_i(\mathcal{T}(\mathcal{M}); S_i, z_i) = \text{ICP}(C_1, C_2) = \frac{1}{|C_1|} \sum_{i=1}^{|C_1|} \mathbb{I}\left(\|\mathbf{x}_i - (\mathbf{y}_i + \hat{\mathbf{t}})\|_2 < \epsilon\right),$$

where $\epsilon$ is the threshold, and $\mathbb{I}(\cdot)$ is the indicator function.

### *Extracting Contour*
**Contour from mesh**

We use pyvista's contour function directly to get the contour of the mesh $\mathcal{M}$ at $z$ height.

**Contour from slice**

The contour of the slice $S_i$ can be extracted using the $\alpha$-shape algorithm.[125] The $\alpha$-shape algorithm is a generalization of the convex hull algorithm, which can extract the contour of a point set with holes. The $\alpha$-shape algorithm is parameterized by a parameter $\alpha$, which controls the size of the holes. When $\alpha$ is small, the $\alpha$-shape algorithm is equivalent to the convex hull algorithm, and when $\alpha$ is large, the $\alpha$-shape algorithm is equivalent to the minimum spanning tree algorithm. Alternatively, we also employ an image-based method that first converts the slice into a grayscale image, then applies a grid to segment it into manageable regions, followed by using the well-developed OpenCV contour extraction algorithm to delineate the contours of the slice.

### *Discrete Optimization*
The discrete multi-labeling optimization problem discussed above can be solved using the graph-cut techniques, *e.g.*, $\alpha$-expansion[126] and FastPD algorithm.[127]

The process of performing mesh correction in Spateo package is as follows:

```python
# initialize z_shift_correction
mesh_correction = st.align.mesh_correction(
    models=slices,
    spatial_key="align_spatial",
    key_added="align_spatial_z_correct",
    device="0", # or cpu
    z_values=z_values, # put the height of each slices here
)
# extract the contours
mesh_correction.extract_contours(method="alpha", alpha=0.5)
# normalize the scale of the mesh to the slices
_, scale, mean_mesh, mean_contour = mesh_correction.normalize_mesh()
# perform the discrete optimization
init_parameters = np.array([0,0,0,0,1])
best_parameters, _ = mesh_correction.start_discrete_optimization(
    init_parameters=init_parameters,
    max_rotation_angle=5,
    max_translation_scale=0.05,
    max_scaling=1.05,
    labelNum=8,
    nIters=10,
)
align_slices = mesh_correction.z_shift_correction(
    models=slices,
)
```

## Cell
### Article

**CellPress**
OPEN ACCESS

We list the major parameters for Spateo pairwise alignment, multi-slice refinement, and mesh correction, as well as the corresponding explanations and default settings in the following table:

**Major parameters of 3D alignment algorithm, their explanation and the default values.**

| Parameters | Explanation | Default |
|---|---|---|
| $\beta$ | Length-scale of the squared exponential or SE kernel. Larger value means less correlation between points and more flexible non-rigid deformation, and *vice versa*. | 1e-2 |
| $B_a$ | Hyperparameter that controls the shape of the Beta distribution for the prior of $\gamma$. Larger value means more outliers are expected, and *vice versa*. | 1 |
| $B_b$ | Hyperparameter that controls the shape of the Beta distribution for the prior of $\gamma$. Larger value means less outliers are expected, and *vice versa*. | 1 |
| $\kappa$ | Hyperparameter that controls the shape of the Dirichlet distribution. A larger $i$-th value means that cell $i$ in sample A has a greater likelihood to generate more cells, and *vice versa*. | 1 |
| $m$ | Number of sparse inducing points used for Nyström approximation for the kernel. Larger means better approximation ability, but increased computation, and *vice versa*. | 15 |
| $b$ | Size of the mini-batch of SVI. | size(sampleB) / 10 |
| $K$ | The number of top $K$ nearest neighbors in guiding optimization. | 10 |
| $\lambda$ | Weight for guiding optimization | 1 |
| Multi-max-iter | Maximum number of iterations of the optimization | 200 |
| $\mathcal{N}$ | Number of neighbors considered in multi-slice refinement. | 4 |
| multi-max-iter | Maximum number of iterations for the multi-slice refinement. | 3 |
| $k$ | Number of labels for each transformation parameter | 15 |
| $\epsilon$ | Inlier-outlier threshold for the ICP in the mesh correction. | 0.1 * spatial scale |
| mesh-max-iter | Maximum number of iterations for the mesh correction. | 10 |

### Create surface polygon mesh, volumetric mesh, and voxel models of a whole embryo, or individual organs from 3D point clouds with PyVista

In addition to the scalable, efficient, and powerful 3D alignment algorithm, another key innovation of Spateo is its ability to perform various downstream 3D modeling and morphometric analyses with the 3D aligned points clouds. In particular, we leverage the PolyData data structure from PyVista to represent the 3D point cloud, where each cell is annotated with the tissue identity, using the following function:

```
pc_model, _ = st.tdr.construct_pc(
    adata=adata,
    spatial_key="align_spatial",
    groupby="annotation",
    key_added="annotation",
)
```

With the point cloud data, we can build upon several established algorithms to create surface polygon meshes for the entire embryo or different organs, as shown below:

```
mesh_model, _, _ = st.tdr.construct_surface(
    pc=pc_model,
    key_added="annotation",
    label="mesh_model",
    cs_method="marching_cube",
    cs_args={"mc_scale_factor": 1.},
    smooth=5000,
    scale_factor=1.0,
)
```

The methods for constructing surface meshes include:

'pyvista': Generate a 3D tetrahedral mesh based on pyvista.
'alpha_shape': Computes a triangle mesh based on the alpha shape algorithm.
'ball_pivoting': Computes a triangle mesh based on the Ball Pivoting algorithm.
'poisson': Computes a triangle mesh based on the Screened Poisson Reconstruction.
'marching_cube': Computes a triangle mesh based on the marching cube algorithm.

We next set the diameter of each cell's 3D geometry as that from the segmentation to obtain 3D volumetric meshes of each cell. In Spateo, three geometries, including cube, sphere, and ellipsoid, are supported:

```
cells_model = st.tdr.construct_cells(
    pc=pc_model,
    cell_size=pc_model.point_data["cell_radius"],
    geometry="sphere",
    xyz_scale=(1, 1, 1),
)
```

We can also voxelize the closed surface mesh into 3D voxels:

```
voxel_model, _ = st.tdr.voxelize_mesh(
    mesh=mesh_model,
    key_added="annotation",
    label="voxel_model",
    smooth=500,
)
```

The implementation of building point cloud, surface mesh, and 3D cell geometry or voxel models in Spateo is a highly flexible strategy that can be generally applied to a single organ or the entire embryo.

**Spatial domain digitalization**
In this section we will first describe digitization in a 2D space and then generalize it to 3D space. Spatial domain digitalization describes the process of constructing a spatial coordinate reference system in accordance with any arbitrary axis, enabling

# Cell
**Article**

identification of genes with graded or periodic distributions along the directions defined by this coordinate system. Mathematically, for an arbitrarily-shaped spatial domain, a digitalization result is determined by the shape of domain boundaries, boundary-derived isolines (analogous to equipotential lines in physics or cartography) and perpendicular streamlines that define regions within the reconstructed coordinates system. In detail, digitalization of a spatial domain $\Omega$ is adapted by mapping the potential field in physics, *i.e.*, a scalar field such as the electrical field, given boundary conditions. To be specific, a continuous $\Omega$ is digitized according to the gradient of a scalar variable $\nabla\psi$ (e.g. spatial layer/column values), which depends inversely on the distance of a given bucket r (a bin or a cell) to the target boundary $\Gamma$ (as $\partial\Omega$). Such a scheme can be modeled by Poisson's Equation:

$$\nabla^2\psi(\mathbf{r}) = f(\mathbf{r}) \text{ for } r \in \Omega,$$

where f is the known function of r in the domain $\Omega$. Importantly, the potential field $\psi$ can only be solved when boundary conditions (BCs) are defined. The conditions are either specified by the bucket-wise values (called Dirichlet or class I BC) under certain known function $g_1$,

$$\psi(r) = g_1(r) \text{ for } r \in \Gamma_D,$$

or by the normal derivative of the solution on the boundary (called Neumann or class II BC)

$$\frac{\partial}{\partial\widehat{\mathbf{n}}}\psi(r) = g_2(r) \text{ for } r \in \Gamma_N.$$

These BCs are crucial to determine $\psi$ via iteration methods in numerical analyses, especially when the geometries of boundaries are complex and irregular.

In this study, we first digitalize the spatial domain with a steady-state field (and then infer the effect of the true internal sources by highly variable genes along the potential gradients that violate the null hypothesis), thus it simplifies Poisson's equation to its homogenous case (f = 0): Laplace's equation. In the three-dimensional (3D) space, Laplace's equation is written as:

$$\nabla^2\psi = \frac{\partial^2}{\partial x^2}\psi + \frac{\partial^2}{\partial y^2}\psi + \frac{\partial^2}{\partial z^2}\psi = 0.$$

For visualization purposes, we first use the basic two-dimensional form to demonstrate the numerical solution with the Jacobi method under the supplied boundary condition:

$$\frac{\partial^2}{\partial x^2}\psi + \frac{\partial^2}{\partial y^2}\psi = 0,$$

and further extend it to the 3D space.

Numerically, the equation is discretized as

$$\frac{(\Psi_{i+1j} - \Psi_{ij}) - (\Psi_{ij} - \Psi_{i-1j})}{\triangle x^2} + \frac{(\Psi_{ij+1} - \Psi_{ij}) - (\Psi_{ij} - \Psi_{ij-1})}{\triangle y^2} = 0.$$

When we choose $\Delta x = \Delta y$ for homogeneity in both $x,y$ dimensions, the equation can be written as

$$\Psi_{ij} = \frac{1}{4}(\Psi_{i-1j} + \Psi_{i+1j} + \Psi_{ij-1} + \Psi_{ij+1}).$$

Now, the discrete form of Laplace's equation shows that the potential of the central grid point is the equal-weighted average of four neighborhoods.

Based on the Jacobi method,[128] for all interior grids at the (k+1)-th iteration, we have:

$$\Psi_{ij}^{k+1} = \frac{1}{4}\left(\Psi_{i-1j}^k + \Psi_{i-1j}^k + \Psi_{ij}^k + \Psi_{ij}^k\right).$$

And for an enclosed domain, any neighbor of an interior grid would not fall out of the boundary, no matter how irregular the domain shape is. The iteration process ends either with convergence when the normalized L2 loss

$$Loss = \frac{\sqrt{\sum_{interior}\left(\Psi_{ij}^{k+1} - \Psi_{ij}^k\right)^2}}{\sqrt{\sum_{interior}\left(\Psi_{ij}^k\right)^2}}$$

between two iterations is below 1e-5 by default, or when iterations surpass the maximum number of iterations (default 100000).

We then solve the real-world boundary problem. For any shape of a bounded domain, the boundary $\Gamma$ can be arbitrarily divided into four connected boundaries with specified breakpoints. With two sides, $\Gamma_{D_2}$ and $\Gamma_{D_4}$, respectively hold the fixed potentials as the Dirichlet BCs, the equipotential lines (the dotted lines with arrows) are naturally perpendicular to the Neumann BCs $\Gamma_{N_1}$ and $\Gamma_{N_3}$. Note

that the normal derivative for an irregular boundary can not satisfy the numerical estimation, we project the original Neumann boundaries to two custom line segments, thus approximate the Neumann BC to Dirichlet BC with uniform distribution along these segments (or original boundaries, with slight loss of precision) for downstream analysis.

It is derived as

$$\frac{\partial}{\partial \hat{\mathbf{n}}} \psi(r) = 0 \;\Rightarrow\; \psi(r) \perp \hat{\mathbf{n}} \text{ for } r \in \Gamma_N \;\Rightarrow\; \psi(r) = \text{uniform}(c_1, c_2) \text{ with } \nabla\psi.$$

And

$$\psi(r) = c_1 \text{ for } r \in \Gamma_{D_2}, \; \psi(r) = c_2 \text{ for } r \in \Gamma_{D_4}.$$

For numerical convenience, the grids that lay outside the potential field are set to value zero and $c_1$ and $c_2$ are symmetric about zero. By default, we set $c_1 = -1$ and $c_2 = +1$,

The final task becomes to solve $\psi$ with the equation:

$$\frac{\partial^2}{\partial x^2} \psi + \frac{\partial^2}{\partial y^2} \psi = 0,$$

with the boundary conditions

$$\psi_{\Gamma_1} = \text{uniform}(1, -1), \; \psi_{\Gamma_2} = -1, \; \psi_{\Gamma_3} = \text{uniform}(-1, 1), \; \psi_{\Gamma_4} = 1.$$

To hold these Dirichlet BCs during the iterative updating, the values of the boundary grids should be re-initialized during each iteration. This process is crucial to insulate the influence of any exterior grid when only the four closest neighbor grids are applied to calculate the value of the target grid in the Jacobi algorithm.

Once the solution of $\psi$ is obtained (numerically, every grid has its potential value $\Psi_{i,j}$), the domain is theoretically digitalized and can be partitioned into meshes of any equal-area size. To dissect the "layers", the equipotential lines are generated by ligation of the grids with the same value from boundary $\Gamma_{N_1}$ to $\Gamma_{N_3}$ and evenly spaced with a constant interval. For the "columns", we propose two approaches to handle the boundaries $\Gamma_D$, with varying degrees of complexity. The first approach is the "one-step" approach.

When the boundaries are relatively regular or smooth, the concept of calculating streamlines that are perpendicular to the equipotential lines is workable. It is equivalent to calculating the tangent vector field,

$$\tan \theta_{i,j} = \frac{\Delta y_{i,j}}{\Delta x_{i,j}},$$

for each grid, and it is numerically solved by the Runge Kutta fourth order method (RK4) for approximating the solution of ordinary differential equations. After the tangent field is constructed, the streamlines are computed by starting at any point of $\Gamma_{D_2}$, perpendicularly crossing each equipotential line under the direction of the tangent field, and finally reaches the boundary $\Gamma_{D_4}$. However, in practice, the RK4 method decreases its precision along with increment of the boundary irregularity.

The second approach is the "two-round" approach. In the first round of the digitalization, only equipotential lines are recorded as the result of domain layering. And the method then can be interpreted as rotating the domain 90 degrees, and repeating the whole process but orthogonally columning the domain in the second round. This approach is a rough but effective approximation that can be adapted in most real-world cases without any extra parameter settings.

In the Spateo package we can use st.dd.digitize to digitize the spatial domain of interests into different layers or columns:

```python
# Digitize the area of interest
st.dd.digitize(
    adata=adata_bin30,
    ctrs=contours,
    ctr_idx=0,
    pnt_xy=pnt_xy,
    pnt_xY=pnt_xY,
    pnt_Xy=pnt_Xy,
    pnt_XY=pnt_XY,
    spatial_key="spatial_bin30"
)
```

**Cell**
Article

For an enclosed 3D domain, the Jacobi method still works, where the three-dimensional form of potential for each central point becomes:

$$\Psi_{i,j,k} = \frac{1}{6}\left(\Psi_{i-1,j,k} + \Psi_{i+1,j,k} + \Psi_{i,j-1,k} + \Psi_{i,j+1,k} + \Psi_{i,j,k-1} + \Psi_{i,j,k+1}\right).$$

However, it could be difficult to define the boundary surface in 3D space, and the rate of convergence usually slows down when comparing the solution in two-dimensional space.

The potential equation can be further generalized with a customized adjacency matrix to adapt for non-orthogonal coordinates:

$$\Psi_v = \sum_{u=0}^{n-1} A_{u,v}\Psi_u,$$

where $\Psi_v$ is the potential for cell $v$, and $A$ is the adjacency matrix indicating the neighbor relations between $n$ cells. $A$ can be either unweighted grid, such as 3D mesh, or weighted network like k-nearest neighbor graph. Then, the potential for the $(k+1)$-th iteration can be written as:

$$\Psi^{k+1} = A\Psi^k.$$

In this case, the initial potential $\Psi^0$ should be tailored with the potential of boundary cells defined as $c_1 = -1$ (the lower bound) and $c_2 = +1$ (the upper bound) by default. The solution of $\psi$ can be obtained using the same iterative solver and L2 loss function as described previously.

```
# Digitize a generalized point cloud
digitized_potentials = st.dd.digitize_general(
    pc=point_cloud,
    adj_mtx=adjacency_mtx,
    boundary_lower=point_list_1,
    boundary_upper=point_list_2,
)
```

### Spatially-aware modeling of networks of cell-cell interaction or CCI in 3D spatial transcriptomics

In the following, we will explain the algorithmic details related to the cell-cell interaction introduced in the schematic of Figures 4C and 5B. In the schematic, the inputs to CCI include coordinates to construct the spatial graph (x, y, z) and an RNA counts matrix. The goal for CCI models is to predict the expression of target gene(s) as a function of signaling information (which is derived from ligand and/or receptor expression). In short, using 3D positions of each cell (Figure 4B, lower panel, upper left), the "signaling landscape" is defined in terms of which cells it may be communicating with and the molecules involved (Figure 4B, lower panel, upper right). Processing depends on the chosen model type (details in section "classes of cell-cell interaction model"). The total number of neighbors of each cell type can be counted, resulting in a matrix of shape (number of cells x number of cell type labels), or total expression of ligands can be queried for all neighboring cells, resulting in a matrix of shape (number of cells x number of ligands) or (number of cells x number of L:R interactions) if total ligand expression in the neighborhood is convolved with receptor expression in the cell (bottom right box, labeled **"L:R model"**). For each target gene, a spatially-weighted generalized regression model (details in section "spatially-weighted modeling for characterization of spatial specificity of signaling effects") is fit (Figure 4B, lower panel, lower left), incorporating spatial distance. The result is an array of cells x coefficients (Figure 4B, lower panel, lower right, labeled "Coefficients"). Furthermore, the downstream model (Figure 4C) can further extend the upstream model in Figure 4B to infer both intercellular (L:R) interactions and intracellular (TF-target) regulations. This downstream model is also a weighted generalized regression model (details in "weighted modeling for downstream (transcription factor-gene) model"), using gene expression rather than spatial coordinates to derive the weights. The weighting in upstream and downstream models helps to account for heterogeneity within the sample.

#### Problem definition

Our goal is to make predictions about inter- and intracellular molecular relationships involved in the process of cell-cell interaction (*i.e.* the downstream expression induced by the activity of a ligand, or the regulatory effect of a transcription factor (TF) on a particular ligand-encoding gene). From spatial transcriptomics data, we use the spatial information for each cell to construct a spatially-weighted graph defining the most likely cells to interact with any given cell. We use prior knowledge networks to simplify the complex task of considering possible effectors from amongst the thousands of ligands, TFs, etc. Using this knowledge, we set up a regression task, where the predictor variables are the possible modulatory genes (ligands, receptors, or TFs, etc.) and the

**Cell**
Article

response variable(s) to be described is the chosen target gene(s), which can be any gene. We introduce a spatially-weighted inferential framework to perform the regression, capable of assigning cell-specific coefficients to provide insight into where a particular interaction may be occurring and how other genes might cooperate with it. We describe each of these processes in sequence below.

### Infrastructure

*Spatial and expression weights.*

We represent the spatial and expression neighborhood of each cell via a weighted undirected graph $G(V,E)$, where each vertex $v \in V$ represents a cell and each other vertex in its neighborhood is connected to $v$ by an edge with weight dependent on the distance between the two vertices. These distances are computed using the spatial coordinates of each cell in 2D slices or 3D aligned slices, or using gene expression to compute distance in gene expression space (see below). Programmatically, this is encoded as a weighted adjacency matrix. The neighborhood of each cell is defined using either the fixed-radius nearest neighbor or k-nearest neighbor approach. This is implemented by the bandwidth parameter $bw$ in terms of $m$ or $k$, where $m$ represents either the radial distance bound from each cell within which neighboring cells are given nonzero spatial weight (given by boolean parameter "fixed") or a value that is computed individually for each cell such that the nearest $k$ neighbors are given nonzero spatial weight. For each cell $i$, the distance vector $d$ of shape $(n,)$ in either gene expression space (given by the principal components, or as an indicator array of identical shape to the expression array where expressed genes are represented as ones and non-expressed genes as zeros, etc.) or physical space is computed by finding the distance between cell $i$ and all $n$ cells $j$:

$$d_{i,j} = \sqrt{\sum_{c=1}^{N} (coord_{c,i} - coord_{c,j})^2},$$

where $N$ is the dimension of the genes when computing distances in gene expression space, or either two or three depending on the dimensions of the sample when computing distances in physical space. $c$ is the axis index (*i.e.* $x, y, z$). Distance can be conditioned on variables "ct" and "cov", where "ct" is a vector of shape $(n,)$ with the numerically encoded cell type identity of each cell, and "cov" is similarly a binary vector of shape $(n,)$ representing any arbitrary distinction (*e.g.* "drug" vs. "control"). If provided, cells not of the same cell type or not matching the categorical characteristic of a given cell have spatial weights set to zero. Each weight decays as a function of distance, determined by a combination of user-defined bandwidth and kernel parameters. Six kernel options are available:

"Triangular":

$$w_{ij} = \begin{cases} 1 - \left(\dfrac{d_{ij}}{bw}\right), & \text{if } d_{ij} < bw, \\ 0, & \text{otherwise.} \end{cases}$$

"Uniform":

$$w_{ij} = \begin{cases} 0.5, & \text{if } d_{ij} < bw, \\ 0, & \text{otherwise.} \end{cases}$$

"Quadratic"[129]:

$$w_{ij} = \begin{cases} \dfrac{3}{4}\left[1 - \left(\dfrac{d_{ij}}{bw}\right)^2\right], & \text{if } d_{ij} < bw, \\ 0, & \text{otherwise.} \end{cases}$$

"Bisquare"[130]:

$$w_{ij} = \begin{cases} \left[1 - \left(\dfrac{d_{ij}}{bw}\right)^2\right]^2, & \text{if } d_{ij} < bw, \\ 0, & \text{otherwise.} \end{cases}$$

"Gaussian":

$$w_{ij} = \begin{cases} e^{-0.5 \times \left(\frac{d_{ij}}{bw}\right)^2}, & \text{if } d_{ij} < bw, \\ 0, & \text{otherwise.} \end{cases}$$

"Exponential":

$$w_{ij} = \begin{cases} e^{-\left(\frac{d_{ij}}{bw}\right)}, & \text{if } d_{ij} < bw, \\ 0, & \text{otherwise.} \end{cases}$$

where $d_{i,j}$ is the distance between cell $i$ and cell $j$, and $bw$ is the bandwidth parameter. By default, a bisquare kernel is used. In modeling, the process of spatial weight computation is used to construct the predictor matrix when incorporating e.g., ligand expression information from the neighboring cells. In addition, it can be used to select the neighboring cells considered in the regression process for each cell.

Spatial/gene expression weight matrix construction can be performed manually with commands similar to the below example:

```python
from functools import partial
from multiprocessing import Pool

get_wi_partial = partial(
            st.tl.get_wi,
            n_samples=10000,
            coords=coords,
            fixed_bw=True,
            exclude_self=True,
            kernel="bisquare",
            bw=10,
            threshold=0.01,
            sparse_array=True,
            normalize_weights=True,
        )

with Pool() as pool:
    weights = pool.map(get_wi_partial, range(n_samples))
w = scipy.sparse.vstack(weights)
```

where "n_samples" is the total number of samples in the dataset, "coords" is an array containing the coordinates of each sample, "exclude_self" indicates whether to include or ignore the sample itself in computation, "bw" is either a radial distance bound from each cell or a non-fixed value representing $k$ nearest neighbors (for "fixed_bw" is True or False, respectively, as explained above), "threshold" is a weight below which spatial weights will be set to 0, "sparse_array" returns the resulting matrix as a sparse matrix, and "normalize_weights" scales each weight such that the sum of the weights from each cell to all its neighbor cells is one.

*Structure of prior knowledge files of intercellular and intracellular interactions.* For many of the model classes of cell-cell interactions to be described, information about ligand-receptor (L:R) pairs, links between receptors and their downstream transcription factors (receptor-TF), and transcription factor (TF)-target (TF-target) binding pairs is required, either directly or to be used to create the predictor array. Default databases are included in Spateo's "database" subdirectory of the "tools" main directory of the package. However these can be separately provided by supplying any file path that contains the relevant files in .csv format and with the correct naming convention. At a minimum, the ligand-receptor file must have three columns, labeled "from" (containing ligands), "to" (containing their paired receptors) and "type". The "type" column indicates whether the interaction is mediated by ligands that are membrane-bound (with the specific label being "Cell-Cell Contact"), secreted ("Secreted Signaling"), or components of the extracellular matrix ("ECM-Receptor"). The receptor-TF file must have two columns, labeled "receptor" and "tf", with self-explanatory contents. The TF-target database file must contain an array $X \in \mathbb{R}^{N \times M}$, where $N$ is the number of target genes (such that each row corresponds to a gene) and $M$ is the number of transcription factors (such that each column corresponds to a transcription factor). The default L:R database was originally compiled for CellChat,[131–135] information for the default receptor-TF database was compiled from a combination of the Kyoto Encyclopedia of Genes and Genomes (KEGG)[132], REACTOME[133] and OmniPath.[134] A "signaling importance score" was calculated and included in the default database; although not used explicitly in modeling, it can be used to filter the database. This score is computed using Personalized PageRank (PPR)[135], with the receptor as the seed node, similar to the process employed by NicheNet.[136] The default mouse TF-target database is assembled from a mouse scATAC-seq atlas, spanning over 100,000 cells from 13 tissues.[137] The default human TF-target database was compiled by querying the promoter regions of each gene for motifs from the gimmemotifs[138] vertebrate database, version 5.

*Data imputation.* To account for the sparsity of gene expression data, we developed a protocol for interpolation of gene expression, informed by biological prior knowledge (*i.e.* TF-target relationships) and expression patterns in the local neighborhood of each cell. Overall, this operation can be described by

$$X_{\text{new}} = W_{\text{final}} \cdot X,$$

where $W_{\text{final}}$ is initially the spatial weights matrix, computed as described in the "Spatial weights" section. This initially serves to restrict the smoothing operation to use data from only the neighboring cells. We additionally condition smoothing to use data from only cells of the same type, which are more likely to share molecular features (marker gene profiles, chromatin accessibility states, etc.). We compute the Hadamard product between the weights matrix and the "cell type mask":

$$W_{ct} = W \circ \text{ct},$$

Where each row of the ct binary array corresponds to a cell, and columns are used to specify cell type identity:

$$\text{ct}[i,j] = \begin{cases} 1, \text{if cell type i = cell type j,} \\ 0, \text{otherwise.} \end{cases}$$

Furthermore, we condition smoothing to use data from only cells that have a similar transcription factor expression profile. From gene expression array $X$, we extract subset $X_{TF}$ using the information from our database (see "structure of prior knowledge files of intercellular and intracellular interactions"). For each cell $i$ and TF $j$, we define whether the TF is expressed,

$$B[i,j] = \begin{cases} 1, \text{if } X_{\text{TF}}[i,j] > 0, \\ 0, \text{otherwise.} \end{cases}$$

We compute the Jaccard similarity between each pair of cells $i$ and $k$:

$$J[i,k] = \frac{|B[i,:] \cap B[k,:]|}{|B[i,:] \cup B[k,:]|},$$

# Cell
## Article

**✦ CellPress**
OPEN ACCESS

where the intersection $|B[i,:] \cap B[k,:]|$ can be calculated as the dot product between the TF expression vectors for the two cells, and the union as the total sum of both vectors minus the intersection. This quantity thus represents the fraction of TFs that are shared by two cells from the set that are expressed by at least one. The full formula for the Jaccard similarity can be represented as:

$$J[i,k] = \frac{\sum_j B[i,j] \cdot B[k,j]}{\left(\sum_j B[i,j]\right) + \left(\sum_j B[k,j]\right) - \sum_j B[i,j] \cdot B[k,j]}.$$

We define another binary array to use as a mask by checking if the Jaccard index is higher than the median of all nonzero Jaccard similarity ($P_{median}^+(J)$):

$$J_m[i,j] = \begin{cases} 1, \text{if } J[i,j] \geq P_{median}^+(J), \\ 0, \text{otherwise}. \end{cases}$$

The final weights can be defined as:

$$W_{final} = W_{ct} \circ J_m.$$

*Stratified subsampling to select cells for modeling.* As an individual regression is performed for each calibration location, the calibration process is computationally intensive and time-intensive. To mitigate this for larger datasets, we developed an optional spatial subsampling strategy to reduce the total number of calculations by performing them on a subset of cells of the whole sample. We select cells uniformly across the sample using the spatial coordinates. We then approximate the interaction effect for non-sampled cells based on those predicted for their selected neighbors, taking advantage of the fact that cell-cell interaction occurs over local neighborhoods. We first partition the dataset into spatial clusters using $k$-means clustering on the spatial coordinates. We set an arbitrary number of clusters, equal to 1% of the number of cells (such that 100 cells on average are sorted into each spatial cluster). For the resulting spatial clusters

$$C = \{c_1, c_2, \ldots, c_{n_{clust}}\},$$

we compute the expression density of each target gene:

$$density(c) = \frac{count\_nonzero(y \in c)}{length(c)}.$$

Within each spatial cluster, we randomly subsample cells, but apply weighting based on the size of the spatial cluster and the computed expression density:

$$n_{sample\_nonzeros} = int\left(\lceil (length(stratum)/2) \times density \rceil\right),$$

where *stratum* indicates the set of cells in the given spatial cluster. We sample equal or more cells without target expression to those with target expression. To map non-sampled points to sampled points, we compute the pairwise Euclidean distance between all non-sampled cells and sampled cells. After model fitting, for each non-sampled point, we assign the coefficients predicted for its nearest sampled neighbor. The high number of spatial clusters in the sampling procedure prevents non-sampled points from being far from sampled points such that they are not exposed to similar signaling environments.

### Classes of cell-cell interaction model

Spateo's cell-cell interaction modeling framework consists of three classes of models, the ligand model, ligand:receptor (L:R) model, and the niche model. All these models reveal the intercellular interactions while the downstream model can be used to further reveal intracellular interactions. These models all aim to quantify the effect of cell-cell interaction on gene expression in the receiving cell but differ in the way how each model class is defined. Specifically, they differ in the composition of the predictor matrix: the ligand:receptor (L:R) model convolves receptor expression levels with an aggregative measure of expression of the ligand in the local neighborhood. The ligand model is similar, but considers only expression of the ligand in the local neighborhood; however, this is conditional on the presence of at least one cognate receptor expressed in the receiving cell, or on the presence of transcription factors associated with said receptor(s). The niche model instead creates a binary matrix where each feature is a cell type, and ones indicate presence of a cell of that type in the local neighborhood, aiming to describe gene expression in a cell as a function of the presence of a putatively interacting cell type. Post-training, downstream models can also be employed to elucidate the impact of transcription

factor expression on the expression of ligands that were utilized by the upstream models or on the expression of target genes predicted by the upstream models, thereby constructing an integrated network of intercellular communication.

### Generalized linear models

Multiple potential sources of technical variation lead to overdispersion and high dropout rates in single-cell data, producing characteristic right-skewed count distributions and breaking the normality assumption of the raw or log-transformed counts that result in inferential underpowering of general linear models. In accordance, a generalized model is used instead that fits data at the level of reads or UMIs, under the assumption that gene expression follows a negative binomial,[139] zero-inflated negative binomial,[39,140,141] or Poisson distribution.[142–144] Poisson distributions are more computationally efficient in comparison, so a Poisson distributional assumption was made in the main and supplemental analyses of our mouse embryo dataset, which involve large quantities of cells. Methodological details will likewise be provided through the lens of this assumption. For each gene of interest, $Y_{i,j}$, the observed RNA counts for cell $i$ and gene $j$, is modeled. Poisson sampling is assumed such that

$$Y_{i,j}|\lambda_{i,j} \sim \text{Poisson}(\lambda_{i,j}),$$

with expected transcript count $\lambda_{i,j}$. In the case of a generalized linear model (GLM), the predicted variable is not a linear function of the predictors, but instead related through a differentiable link function:

$$g(\lambda_{i,j}) = X\beta.$$

The final predicted value therefore comes from application of the inverse link function:

$$\lambda_{i,j} = g^{-1}(X\beta).$$

In our case, the resulting $\lambda_{i,j}$ is a mixture of $k$ explanatory variables $x_k \in \mathbb{R}^{k,j}$ for each gene $j$, defined generally by

$$\lambda_{i,j} = \exp\left(\sum_{k=1}^{K} \beta_{k,j}x_k\right),$$

where each predictor vector $x_k$ for variable $k$ is parameterized by linear coefficient $\beta_k, k = \{1,2,...,K\}$. In general, each variable $k$ can take two forms. For Spateo's modeling core, these are:

Indicator variable, as is the case for the **Niche model**. In this case, $x_k$ is always either 0 or 1, representing *e.g.* a classification into a particular cell type among a set of possible cell types (among any other possibility involving categorical labels).

Discrete or continuous variable, as is the case with the ligand-receptor and ligand models. In this case, $x_k$ can take on continuous values representing *e.g.* gene expression. However, it is recommended to discretize this such that raw counts are used for model input, as the recommended distributional assumption is either Poisson or negative binomial. Both the Poisson and negative binomial distributions are discrete, representing a count of a number of events, so to avoid violating this, transcript counts should be used instead of normalized data.

In the context of our Poisson regression, each coefficient $\beta_{k,j}$ can be interpreted as the log fold-change of gene expression per unit change in $x_{k,j}$ for variate $k$ and gene $j$. For model fitting, the Poisson distribution is parameterized such that the probability of observing given transcript count $Y_{i,j}$ is maximized; this can be achieved by maximizing the log-likelihood of a Poisson distribution:

$$l(\mu|\mathbf{y}) = \sum_{i=1}^{n}\{y_i\theta_i - e^{\theta_i} - \log y_i!\},$$

where $\theta_i$ is the linear predictor of the model:

$$\theta_i = \eta_i = \mathbf{x}_i^t\beta,$$

and $\mu$ is the result of applying the inverse link function to the linear predictor:

$$\mu = \exp(x_i\beta).$$

The optimization process involves choosing parameters $\beta$ such that the likelihood is maximized, a process which can be represented as follows:

$$\text{argmax}_\beta l(\mu|\mathbf{y}).$$

# Cell
## Article

As the relationship between the estimation and the parameters is nonlinear, the roots/optima of the equation can be found using the Newton-Raphson method, which updates $\beta$ iteratively until convergence using the derivative of the log-likelihood scoring function, included here to illustrate the concept of iterative updates to $\beta$:

$$\beta^{k+1} = \beta^k + (\mathbf{X}^T W X)^{-1} S(\beta^k),$$

where $\mathbf{X}^T W X$ is the negative of the derivative of the scoring function. This is simplified by instead using the equivalency of Newton-Raphson to iteratively reweighted least squares (IWLS); from simplifying the Newton-Raphson equation:

$$\beta^{k+1} = (\mathbf{X}^T W X)^{-1} \mathbf{X}^T W z, \text{ where } \mathbf{z} = \left[ \mathbf{X}\beta^k + \frac{\partial \eta}{\partial \mu}(\mathbf{Y} - \mu) \right]$$

As shown here, $z$ represents pseudo data initially generated from the observed data and the estimates of the model parameters. For Poisson data:

$$\frac{\partial \eta}{\partial \mu} = \exp(-\eta).$$

Recall from the above equations that $\eta$ is an alias for the linear predictor. $\mathbf{X}$ is an $n$ by $k$ matrix of predictor variables, and $\mathbf{Y}$ is an $n$ by $1$ vector response variable. $\mathbf{W}$ is an $n$ by $n$ matrix, defined as:

$$\mathbf{W} = A\frac{\partial \mu}{\partial \eta}.$$

For a Poisson GLM, $A$ is the identity matrix.

As the number of molecules is typically high and the nature of the distributional assumption contributes to explosive increases in estimated values for unit increases in the predictor, estimated parameters are constrained to have smaller values by incorporating an additional ridge regression cost function[145]:

$$J(\beta) = -\left[ \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right],$$

where $\lambda$ is the regularization parameter, which can be provided during model instantiation.

### Predictor matrix setup

For all models, the shared required inputs are the gene expression matrix $Y \in \mathbb{R}^{N \times G}$, where $N$ is the number of cells and $G$ the number of genes, and the array of spatial coordinates $C \in \mathbb{R}^{N \times M}$, where $N$ is the number of cells and $M$ the spatial dimensionality (either two or three dimensions) of the data. Expression vectors for all ligands, receptors, transcription factors, and target genes are taken from the gene expression matrix. Given values for the number of neighbors to consider for membrane-bound and secreted L:R interactions, spatial weights matrices are automatically computed from the spatial coordinates. In model instantiation, each of these is automatically extracted from a given path to an AnnData object-containing file (and given a few guiding inputs *e.g.*, the name of the key to the AnnData field containing cell type information). Models can be defined as follows:

```python
import os
import spateo as st

adata_path = "./adata.h5ad"
# This is given in the form of an arbitrarily-named file- the base name of
the .csv file will be used to save outputs- here, this is "cci_example"
output_path = "./outputs/cci_example.csv"
# Paths to .txt files where each line is a ligand/receptor/target gene to
predict
ligand_path = "./inputs/cci_ligands.txt"
receptor_path = "./inputs/cci_receptors.txt"
target_path = "./inputs/cci_targets.txt"
cci_dir_path = "./database"

# Model type (model types described below)
mod_type = "lr"
species = "human"
group_key = "celltypes"
coords_key = "spatial"
distance_membrane_bound = cci_lower_bound
n_neighbors_membrane_bound = 6
distance_secreted = cci_upper_bound
n_neighbors_secreted = 70
minbw = cci_lower_bound * 1.5
maxbw = cci_upper_bound * 1.5

if not os.path.exists(os.path.dirname(output_path)):
    os.makedirs(os.path.dirname(output_path))

parser, args_list = st.tl.define_spateo_argparse(
    adata_path=adata_path,
    custom_lig_path=ligand_path,
    custom_rec_path=receptor_path,
    targets_path=target_path,
    cci_dir=cci_dir_path,
    mod_type=mod_type,
    species=species,
    group_key=group_key,
    coords_key=coords_key,
    distance_membrane_bound=distance_membrane_bound,
    n_neighbors_membrane_bound=n_neighbors_membrane_bound,
    distance_secreted=distance_secreted,
    n_neighbors_secreted=n_neighbors_secreted,
    minbw=minbw,
    maxbw=maxbw,
    output_path=output_path,
)

swr_model = st.tl.MuSIC(parser, args_list)
swr_model._set_up_model()
swr_model.fit()
swr_model.predict_and_save()
```

### Ligand:receptor (L:R) model

The LR model is designed to quantify the effect of cell-cell interaction via convolving receptor expression levels with the average local neighborhood's ligand expression. Similar to the ligand model, three key components are used to construct the predictor matrix for this model: a list of receptors (or the path to a text file containing one receptor per line) and the other two are the same as the Ligand model.

Some proteins, for example transforming growth factor-$\beta$ receptors and certain cytokine receptors, require multi-subunit assembly for function.[146,147] For receptors for which this is true, the geometric mean is used to combine the expression vectors of the individual components.

$$r_{complex} = \left( \prod_{i=1}^{n} x_i \right)^{\frac{1}{n}}.$$

The expression of all receptors (and complexes) $r$ comprise the receptor array $R$.

For each cell, the amount of ligand in the cell's spatial neighborhood is quantified by the summation of the ligand's expression across all neighbor cells, modified by the spatial weights (as specified in the "Spatial and expression weights" subsection of the "Infrastructure" section). For cell $i$ and ligand $j$,

$$L_{ij} = \begin{cases} \sum_{k=1}^{n} w_{ik,secreted} \times X_{kj}, \text{ if ligand not membrane-bound,} \\ \sum_{k=1}^{n} w_{ik,membrane-bound} \times X_{kj}, \text{ otherwise.} \end{cases}$$

where $w_{ik}$ is the weight between cell $i$ and cell $k$, and $X_{kj}$ is the number of counts for ligand $j$ in cell $k$. Similarly, we can also calculate the L:R paired array. We start from two arrays where each column corresponds to one of $j$ ligands or one of $l$ receptors or receptor complexes, respectively. Each row of both arrays is one cell. The L:R paired array can be constructed by first finding the set of possible L:R pairs, referencing the provided L:R database and then performing an elementwise multiplication for each relevant L:R pair:

$$LR_i = a_i \cdot b_i,$$

where $a_i$ or $b_i$ is the $i^{th}$ element in the column of $L$ or $R$ corresponding to the ligand or receptor of interest in cell $i$.

The resulting feature space is often high-dimensional, with possibly multiple ligands paired to a given receptor. As a given receptor will activate similar downstream processes for multiple ligands, we combine ligands that can bind a given receptor as a biologically-informed feature selection. The combined column is created by taking the arithmetic mean of the columns for each individual ligand. The final predictor matrix is log-transformed and then normalized by min-max scaling, to alleviate differences in scale between features.

### Ligand model

The ligand model is designed to quantify the effect of cell-cell interaction by considering the ligand expression landscape in the spatial neighborhood. One advantage of the ligand model over its L:R counterpart is its comparatively less combinatorial complexity. Furthermore, although it may be unable to evaluate how a predicted signal is being received and interpreted to result in upregulation of expression from the model outputs due to the missing receptor information, the incorporation of ligands is biologically significant, and, in contrast to L:R model, allows the inclusion of ligands for which the ligand-receptor database is incomplete. Two components (in addition to the AnnData object containing spatial coordinates and the expression matrix) are used to construct the predictor matrix for this model: a list of ligands (or the path to a text file containing one ligand per line), and the path to the directory containing database information. This directory must contain, at minimum, a file titled "lr_db_{mouse or human}.csv", a file titled "{mouse or human}_receptor_TF_db.csv", and a file titled "{mouse or human}_GRN.csv" depending on the species the data was sourced from. The predictor array $L_{ij}$, is constructed as previously described in the "L:R model" section. Information from the ligand-receptor and receptor-TF databases is used to modify the $L$ matrix post-construction by conditioning on the expression of these receptors or their downstream transcription factors, even though the cognate receptors are not directly involved. For each ligand $j$ and cell $i$, $L_{ij}$ is set to 0 if this cell does not express any corresponding receptors or transcription factors (if the cell does express corresponding receptors or transcription factors, $L_{ij}$ is left as is). The final predictor matrix is log-transformed and then normalized by min-max scaling, to alleviate differences in scale between features.

### Niche model

The niche model formulates gene expression in a cell as a function of the spatial composition of putatively interacting cell types in the neighborhood. While this model does not make an appearance in the text, it is also implemented as part of the modeling framework. Beyond ligand-receptor interactions, the umbrella term "cell-cell interactions" encompasses a wide variety of mechanisms that leverage diverse molecules, such as ions, metabolites, and even direct genetic transfer that is common between prokaryotic cells. For model systems where these mechanisms are thought to be particularly important, or which were collected with particularly low-plex fluorescence-based spatial transcriptomic assays (*e.g.* MERFISH, STARmap, Xenium, CosMx, etc.) that include few ligands and receptors, it may be more appropriate to describe in terms of the enrichment of cell types in the microenvironment, as a first step

towards future molecular-level elucidation. For this model, to construct the predictor matrix, the name of the key in the AnnData object's "obs" slot containing cell type labels is needed. Cell type information initially takes the form $CT \in \mathbb{R}^{N \times 1}$, where $N$ is the number of cells. This is a categorical variable that can take on $n$ distinct values, corresponding to each distinct cell type label. This is converted to a one-hot "cell type identity" array, such that each column (each cell) contains all zeros with the exception of the row that corresponds to its cell type.

$$O(CT) = \begin{bmatrix} \delta(CT_1, c_1) & \delta(CT_1, c_2) & \cdots & \delta(CT_1, c_n) \\ \delta(CT_2, c_1) & \delta(CT_2, c_2) & \cdots & \delta(CT_2, c_n) \\ \vdots & \vdots & \ddots & \vdots \\ \delta(CT_n, c_1) & \delta(CT_n, c_2) & \cdots & \delta(CT_n, c_n) \end{bmatrix},$$

where $c_1, c_2, \cdots, c_n$ are the unique cell type categories and

$$\delta(x, y) = \begin{cases} 1, \text{if } x = y, \\ 0, \text{otherwise.} \end{cases}$$

This facilitates a spatial lag operation to quantify the number of cells of a particular type present in the neighborhood of a given cell, modulated by the spatial weights:

$$X_{CT} = \sum_{k=1}^{n} w_{ik,secreted} \times O(CT)_{kj},$$

where $w_{ik}$ is the weight between cell $i$ and cell $k$, and $O(CT)_{kj}$ is the cell type identity indicator for cell type $j$ for cell $k$, either a 1 or 0 as constructed. Weights for secreted ligands are used for this operation, as cells secreting these ligands are the most distant cells that are predicted to be influential via L:R signaling.

### Downstream (transcription factor-gene) model

This model is designed to operate downstream of the central cell-cell interaction models. These models aim to either identify TFs that are potentially responsible for induction of ligand gene expression, or which cooperate with the ligand's cognate receptor to induce expression of downstream target genes. The predictor matrix for this model is derived from transcription factor expression. Unlike ligands, TF-target relationships do not depend on the positions of the cells. Instead of spatial coordinates, this model uses distance in "gene expression space" to derive the weights used in its inferential process (although not necessary for predictor matrix set up, these measurements are used in the modeling process- see section "spatially-weighted modeling for characterization of spatial specificity of signaling effects"). To do so, we binarize gene expression to signify expressed or not expressed, and for each pair of cells, we perform a comparison of the similarity of binarized gene expression profiles with the Jaccard index:

$$J_{ij} = \frac{|g_i \cap g_j|}{|g_i \cup g_j|},$$

where $g$ is a binary vector denoting expression (or absence of expression) of each target gene in cell $i$ or cell $j$.

### Feature selection using biological priors

For ligand and L:R models, the constructed predictor matrix is reflective of the "signaling landscape" (in terms of ligands and optionally receptors). However, not all of these may be truly influential for a particular target. To mitigate the possibility of an unrelated or less confident ligand/interaction being used to predict the expression of a particular target, a copy of the full array is modified at runtime in a target-specific manner, subsetting only to the pertinent ligand/L:R features. To model the regulations from the receptors to downstream targets, we recursively select the ligands from the targets. Specifically, for a particular target gene, the TF-target database is queried to return all transcription factors predicted to be able to regulate that target, defined as the set of "primary TFs". In addition, transcription factors that can regulate expression of any of these primary TFs are included. From this complete set of TFs, the receptor-TF database is queried to return receptors in the same signaling pathways as these TFs, and finally the L:R database is queried to return ligands that can bind these receptors. Features incorporating this final subset of ligands are retained for the modeling of ligands-target interactions.

### Mathematical notation for each model class.

Given the mathematical details from "Generalized linear models", a particular gene's expression $Y_{i,j}$ for cell $i$ and target gene $j$ can thus be modeled as

- Ligand model

$$Y_{i,j} \sim \text{Poisson}\left\{ \exp\left[ \sum_{g=0}^{f} \beta_{j(i,g)} \left( 1_{(i,L \cdot R, TF)} \sum_{h=1, h \neq i}^{N} \omega_{i,h} L_{h,g} \right) \right] \right\},$$

where $\beta_{j(i,g)}$ is the coefficient corresponding to the $g^{th}$ ligand, $\omega_{i,h}$ is the spatial weight between cells $i$ and $h$, $L_{h,g}$ is the expression of the ligand in the $g^{th}$ L:R pair in cell $h$, and $1_{(i,L \cdot R, TF)}$ is the value at cell $i$ of the indicator array formed by conditioning ligand

expression of the $g^{th}$ ligand on relevant receptor/transcription factor expression. For cell $i$ and ligand $g$, the element $i_g$ is 1 only if at least one of the ligand's cognate receptors and/or if at least one of the transcription factors downstream of these receptors is also expressed.

- L:R model

$$Y_{i,j} \sim \text{Poisson}\left\{ \exp\left[ \sum_{g=0}^{f} \beta_{j(i,g)} \left( R_{i,g}^T \sum_{h=1,h\neq i}^{N} \omega_{i,h} L_{h,g} \right) \right] \right\},$$

where $\beta_{j(i,g)}$ is the $g^{th}$ variate for the $i^{th}$ cell in the coefficients array for target gene $j$, here corresponding to the $g^{th}$ ligand-receptor pair, and $R_{i,g}$ is the expression of the receptor in the $g^{th}$ L:R pair for the $i^{th}$.

- Niche model

$$Y_{i,j} \sim \text{Poisson}\left\{ \exp\left[ \sum_{g=0}^{C} \beta_{j(i,g)} \left( \sum_{h=1,h\neq i}^{N} \omega_{i,h} c_{h,g} \right) \right] \right\},$$

where $\beta_{j(i,g)}$ is the coefficient corresponding to the $g^{th}$ cell type category, and $c_{h,g}$ is used to denote the cell type of the $h^{th}$ neighboring cell with $C$ possible categorical classifications. This value will be 1 if the $h^{th}$ neighboring cell is of cell type $g$.

## Spatially-weighted modeling for characterization of spatial specificity of signaling effects

In contrast to traditional global models that assume uniformity in the data-generating process, a spatially weighted Poisson-form generalized linear model takes the following form for each response variable:

$$\lambda_i = \exp\left( \sum_{j=1}^{k} \beta_{i,j} x_{ij} \right),$$

where $\lambda_i$ is the response variable value for cell $i$, $x_{ij}$ is the $j^{th}$ predictor variable for cell $i$ and $\beta_{i,j}$ is the $j^{th}$ parameter estimate for cell $i$. From the "generalized linear models" section, the general form for calibration of this model (bearing in mind the adjustment being made for spatial variability) is

$$\beta_i = \left( X^T W_i X \right)^{-1} X^T W_i z, \text{ where } \mathbf{z} = \left[ \mathbf{X}\beta_i + \frac{\partial \eta}{\partial \mu}(\mathbf{Y} - \mu) \right], \frac{\partial \eta}{\partial \mu} = \exp(-\eta)$$

Throughout the fitting process, $z$ represents pseudo data initially generated from the observed data and the estimates of the model parameters. In this case, $\beta_i$ is a $k$ by $1$ vector of parameter estimates for cell $i$, and $W_i$ is an $n$ by $n$ diagonal spatial weighting matrix specific to cell $i$, which is calculated based on a specified kernel function and bandwidth as described in "Spatial and expression weights". Smaller bandwidth values allow for greater degree of spatial variability, however may have larger standard errors due to being estimated using fewer data points. The optimal bandwidth is found by examining the corrected Akaike Information Criterion (cAIC) score, which for a Poisson GLM is given by

$$\text{cAIC} = -2 \times \sum_{i=1}^{n} \left\{ y_i \theta_i - e^{\theta_i} - \log y_i! \right\} + 2 \times k + \frac{2 \times k \times (k+1)}{n - \text{tr}(S) - 1},$$

where $n$ is the number of cells, $k$ is the number of features, the term within the summation is the log-likelihood of the model, and $\text{tr}(S)$ is the trace of the hat matrix $S$, which relates the observed values to the predicted values by the relationship $\hat{y} = Sy$. The sum of the diagonal elements of this matrix serves as an estimate of the effective number of model parameters.

Each row of the hat matrix is computed from the product of the pseudoinverse of the predictor matrix and the final IWLS weights, such that row $i$ can be represented as

$$s_i = x_i \left( X^T W_i X \right)^{-1} X^T W_{IWLS,i},$$

where $\left( X^T W_i X \right)^{-1} X^T$ is the pseudoinverse, $W_i$ is the row of the spatial weights matrix corresponding to cell $i$ and $W_{IWLS}$ (see "generalized linear models") is given by

$$W = A\frac{\partial \mu}{\partial \eta}.$$

A golden search heuristic is used to search for the optimal bandwidth, associated with the lowest cAIC score. This involves iterative regressions, at each iteration choosing a new pair of intermediate "upper limit" and "lower limit" bandwidths within the range specified by the initial specific minimum bandwidth and maximum bandwidth. These bandwidth updates are given by

$$bw_l = bw_l + \left(1 - \frac{\sqrt{5}-1}{2}\right)(|bw_u - bw_l|),$$

$$bw_u = bw_u - \left(1 - \frac{\sqrt{5}-1}{2}\right)(|bw_u - bw_l|),$$

where $bw_l$ is the lower bandwidth tested for a given iteration and $bw_u$ is the higher bandwidth tested for a given iteration. The fractional term is the inverse of the golden ratio.

We suggest the lower bound be no less than that used for membrane-bound ligands, and the upper bound be smaller than or equal to that used for secreted ligands. Spateo also includes functionality to find these bandwidth values given a target number of neighbors to include. This functionality operates by computing the pairwise distance array. Starting from the initially given bandwidth, the average number of neighbors for all cells within a radius given by the bandwidth is computed. The bandwidth is adjusted based on whether the result is higher than or lower than the desired number.

```
lower_bound_bandwidth = st.tl.find_bw_for_n_neighbors(
    adata,
    coords_key="spatial",
    n_anchors=2000,
    target_n_neighbors=20,
    initial_bw=20,
    exclude_self=True
)
```

where "n_anchors" can be used to select $n$ exemplar points (to avoid iterating over every cell in the sample if the sample is large), "target_n_neighbors" is the desired number of neighbors that on average over all exemplar cells can be found within the bounds defined by a given arbitrary bandwidth.

While the exact distance scales cannot be known, we use what is known about ligand-receptor interaction as guiding principles to infer appropriate distance parameters. For the purposes of the analyses here, "target_n_neighbors" 27 or 250 is used for calculations relevant for membrane-bound signaling or secreted signaling respectively. For the membrane-bound-signaling, we assume that each cell will interact only with the cells in its immediate vicinity by these molecular mechanisms. For secreted signaling, "target_n_neighbors" 250 is used. The entire domain of communication for a single cell is approximately 250 microns across,[148] the average diameter of a cell is on the order of 10 microns, and the spacing between two "neighboring" cells is on the order of 20–40 microns,[149–151] with 30 used as an intermediate value. To elaborate, from the provided measurement for the distance limit of secreted signaling (250 microns), we approximate this to be ~25 times the diameter of a typical eukaryotic cell.[152] Assuming 30 micrometers between cells as an average value, the effective diameter of each cell increases by 30 micrometers. The ratio of the volume enclosed by the signaling limit,

$$V_{signal} = \frac{4}{3}\pi\left(\frac{250}{2}\right)^3$$

to that of the effective volume for each cell,

$$V_{cell} = \frac{4}{3}\pi\left(\frac{40}{2}\right)^3$$

is 244. For membrane-bound signaling, we assume cells will only interact with their immediate neighbors, but lack a specific value for the length scale. In this case, we use cells as a unit of measurement and assume the length scale of signaling is three cells in any direction (the sending cell and its immediate neighbors on either side), forming a sphere with diameter three times that of the cells within. This results in 27 predicted potential interacting partners for any given cell. For two-dimensional data, similar calculations (using the area of a circle rather than the volume of a sphere) result in 70 neighbors for secreted signaling and 9 for membrane-bound signaling.

### Standard errors of the local parameter estimates

To calculate the standard error of our maximum likelihood estimation, we compute the Fisher information matrix, which measures the amount of information that a random variable $Y$ (*i.e.* the observed data) carries about an unknown parameter $\beta$ (*i.e.* the model coefficients) upon which the probability of $Y$ is thought to depend. This matrix $i$ can be estimated as follows:

$$I(\widehat{\beta}) = \boldsymbol{X}^T \boldsymbol{W_{IWLS}} \boldsymbol{X},$$

where $\boldsymbol{X}$ is the predictor matrix. The covariance matrix of the estimated coefficients, hereby denoted $\boldsymbol{C_i}$, is given by the inverse Fisher matrix. Additionally, let the dispersion of the errors from the model be represented as

$$\widehat{\sigma}^2 = \frac{\sum (y_i - \widehat{y}_i)^2}{n - \mathrm{tr}(\boldsymbol{S})},$$

where $n$ is the total number of observations, $y_i$ is the observed value of the dependent variable and $\widehat{y}_i$ is the predicted value for cell $i$, and $\mathrm{tr}(\boldsymbol{S})$ is the trace of the hat matrix $\boldsymbol{S}$. The sum of the diagonal elements of this matrix serves as an estimate of the effective number of model parameters. From the diagonal elements of the covariance matrix and the dispersion values, we compute the standard errors using:

$$SE(\widehat{\beta}) = \sqrt{\mathrm{diag}(C_i)\widehat{\sigma}^2}.$$

Following completion of the inferential process, the local standard errors for all parameter estimates are:

$$SE(\widehat{\boldsymbol{\beta}}) = [SE(\widehat{\boldsymbol{\beta}}_1), SE(\widehat{\boldsymbol{\beta}}_2), ..., SE(\widehat{\boldsymbol{\beta}}_n)]^T_{n \times k}.$$

### Statistical testing of the local parameter estimates

For each local parameter estimate $\widehat{\beta}_i$ and each standard error $SE(\widehat{\beta}_i)$, we use the Wald test to test the null hypothesis that the local parameter estimate is equal to zero. The Wald statistic can be computed as

$$W_i = \frac{(\widehat{\boldsymbol{\beta}}_i - \boldsymbol{\beta}_0)^2}{SE(\widehat{\boldsymbol{\beta}}_i)^2},$$

where $\beta_0$ is the value under the null hypothesis (zero here). Due to the large size of single-cell spatial data, we can assume that the Wald statistic follows approximately a standard normal distribution, and use the cumulative distribution function (CDF) of the standard normal ($\Phi$) to compute the p-value:

$$p_{\mathrm{vals}} = 2 \times (1 - \Phi(|\mathrm{wald\_statistic}|)).$$

To assign "significant" labels, $q$ values are calculated across all features to control the false discovery rate (FDR) using the Benjamini-Hochberg procedure[153] using a default FDR of 0.05.

### Downstream analyses

After having fit one of the CCI models (**L:R, ligand, niche**), additional interpretive analyses can be performed by initializing a downstream model (overview provided in "downstream (transcription factor-gene) model" section of "predictor matrix setup") such that we can integrate both intercellular and intracellular interactions:

```python
from st.tl.CCI_effects_modeling import define_spateo_argparse
from st.tl.CCI_effects_modeling.MuSIC_downstream import MuSIC_Interpreter

parser, args_list = define_spateo_argparse(
    adata_path="./adata.h5ad",
    custom_lig_path="./ligands.txt",
    custom_rec_path="./receptors.txt",
    targets_path="./target_genes.txt",
    cci_dir="./tools/database",
    mod_type="lr",
    species="human",
    group_key="cell_type",
    coords_key="coords",
    distance_membrane_bound=10.0,
    distance_secreted=100.0,
    minbw=10.0,
    maxbw=100.0,
    output_path="./example_run/results.csv",
)

downstream_model = MuSIC_Interpreter(parser, args_list)
```

As the arguments are shared with the main model, if this is part of the workflow, it is recommended to define both in a coding environment.

*Predictor matrix setup for downstream (transcription factor-gene) model*

The predictor matrix for this model is derived from transcription factor expression. This is only accessible from running the setup and inference member functions of an instance of the downstream model class.

In this example, placeholder values are given to function arguments, but on a case-by-case basis these will be identical to what is used to define the "upstream" CCI model. From here, models can be initiated to use the relevant member functions:

```
downstream_model.CCI_deg_detection_setup(
    group_key="cell_type",
    use_ligands=False,
    use_receptors=False,
    use_targets=True
)


downstream_model.CCI_deg_detection(
    group_key="cell_type",
    cci_dir_path="./tools/database",
    use_ligands=False,
    use_receptors=False,
    use_targets=True,
    use_dim_reduction=False,
    distr="poisson"
)
```

Mathematical notation for downstream (transcription factor-gene) model
A particular gene's expression $Y_{i,j}$ for cell $i$ and target gene $j$ can thus be modeled as

$$Y_{i,j} = \text{Poisson}\left[\exp\left(\sum_{g=0}^{f} \beta_{j(i,g)} TF_{i,g}\right)\right],$$

where $\beta_{j(i,g)}$ corresponds to the $g^{th}$ transcription factor, and $TF_{i,g}$ to the expression of the $g^{th}$ transcription factor in the $i^{th}$ cell.
***Weighted modeling for downstream (transcription factor-gene) model***
Downstream models also employ a weighted framework (see "spatially-weighted modeling for characterization of spatial specificity of signaling effects"). However, for the downstream models, these weights ($W_i$ from "spatially-weighted modeling for characterization of spatial specificity of signaling effects") are based on similarity in gene expression rather than spatial distance. Similar to the "transcription factor-gene" model, we calculate the Jaccard index. For the minimum bandwidth, by default we choose the Jaccard score such that on average, 2% of cells in the sample are included in the threshold defined by this bandwidth. For the maximum bandwidth, by default we choose the Jaccard score such that on average, 5% of cells in the sample are included in the threshold defined by this bandwidth.

```
lower_bound_bandwidth = st.tl.find_bw_for_n_neighbors(
    adata,
    coords_key="spatial",
    n_anchors=2000,
    target_n_neighbors=int(0.02 * adata.n_obs),
    initial_bw=0.5,
    exclude_self=True
)
```

where "n_anchors" can be used to select *n* exemplar points (to avoid iterating over every cell in the sample if the sample is large), and "target_n_neighbors" is the desired number of neighbors that on average over all exemplar cells can be found within the bounds defined by a given arbitrary bandwidth.

### Computation of gene expression (and CCI effect) distribution heatmaps

This function visualizes the relative enrichment of gene expression at various points along a defined spatial axis. It is recommended to use the key from the AnnData ".obsm" field created by Spateo's digitization to provide the position coordinates, although any position key can be used. Henceforth, the position coordinates are referred to as "pos". In the case the "pos" contains Euclidean coordinates, it is recommended to round the relevant dimension (*i.e.* x-, y-, or z-) to the nearest multiple of ten (this may differ depending on the scale of the data), to create "bins" for use in computing spatial enrichment. For each gene, the mean expression across the entire tissue slice/3D reconstructed structure is used as the baseline value, denoted $\overline{X}$. The natural log fold change for gene *g* over the baseline is calculated for all cells; these values are z-scored and for each value of "pos", the average z-score is taken over all cells that share that value:

$$\overline{Z_{LFC_{j,p}}} = \frac{1}{N_p} \sum_{i=1}^{N_p} Z_{LFC_{j,i}},$$

where $LFC_{j,i}$ is the log-fold change for gene *j* and the $i^{th}$ cell within position grouping p and $Z_{LFC}$ is the z-score of this value. For gene *j* and region *p*, the final values of this computation are returned from a smoothing operation:

$$S(Z_{LFC_{j,p}}) = \text{RollingMean}\left(\frac{1}{N_p} \sum_{i=1}^{N_p} \frac{\ln\left(1 + \frac{X_{g,p}}{\overline{X_j}}\right) - \mu}{\sigma}, \text{window} = 3\right),$$

where $S$ is the smoothed $\overline{Z_{LFC_{j,p}}}$ for gene *j* at position *p*, $\mu$ and $\sigma$ are the mean and variance of the log-fold change across all genes, respectively. A smoothing operation is used for the purpose of more continuity in the visualization, applied over the immediately preceding and succeeding positions. For visualization, features are ordered based on how early along the relative position the highest z-scores occur.

```
downstream_model.gene_expression_heatmap(
    use_target_genes=True,
    position_key="spatial",
    coord_column=0,
    cmap="bwr",
    title="Gene expression hotspots along axis",
    fontsize=14,
    save_show_or_return="show"
)
```

Since dataframes containing the expression of ligands, target genes, etc. are automatically created in preprocessing for fitting the upstream model, arguments "use_target_genes", "use_ligands", or "use_receptors" can be used to conveniently extract a relevant set of genes to visualize (otherwise, the gene subset to visualize must be manually provided using argument "genes"). Here, "coord_column" is used when "position_key" refers to a field in AnnData ".obsm", to select which coordinate column to use. Otherwise, it is ignored.

The spatially-weighted nature of the modeling framework also enables this analysis for inferred cell-cell interaction effects.

```
st.tdr.morphofield_gp(
    adata=adataA,
    spatial_key="align_spatial",
    vf_key="VecFld_morpho",
    inplace=True,
)
```

Argument "effect_threshold" is a lower bound used to filter the coefficients array before mean and fold-change computation, and "interaction_subset" is used to manually subset interaction-target combinations (otherwise, all combinations are considered).
*Build 3D continuous expression model.* To reveal the continuous gene expression pattern across 3D space, we developed a deep learning framework as well as a Gaussian process regression method to map the spatial coordinates to the gene expression. The neural network contains three consecutive fully-connected layers and uses common activation functions such as ReLU or Leaky ReLU. The network is trained with the mean squared error (MSE) loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \sum_{g=1}^{G} \left( y_{ig} - \mathbf{f}_\theta(x_i, y_i, z_i)_g \right)^2,$$

where $N$ is the number of the cells in the training set, $G$ is the gene numbers, $y_{ig}$ is the gene expression for gene $g$ and cell $i$, $\mathbf{f}_\theta$ is the neural network with the corresponding parameters $\theta$, and $x_i, y_i, z_i$ are the 3D coordinate of the ith cell. By default, we train the network with the Adam optimizer with the learning rate of 1e-4, a weight decay of 2.5e-5, a batch size of 2000, and 1000 iterations.

For the Gaussian process regression, we implemented the stochastic variational GP regression[154] using the gpytorch[155] package, which leverages the backpropagation of pytorch and GPU cuda acceleration ability. We use the Gaussian likelihood for the normalized expression and squared exponential kernel for the normalized spatial coordinates. The variational evidence lower bound is chosen as the loss for optimizing the variational GP model. By default, the GP model is trained with the Adam optimizer with the learning rate of 1e-2, a batch size of 1024, inducing points number of 512, and 50 training epochs.

After training, we can use either the trained neural network or the GP model to perform arbitrary slicing or prediction to reveal gene expression gradients and patterns within the 3D space.

### Morphometric vector field and morphometric differential geometry analyses

The concept of vector field, a *vector-valued function* that assigns a vector for any point in a space, was originally introduced in physics and used to model the speed and direction of moving fluid or the strength and direction of magnetic or gravitational forces through the physical space. A vector field is often visualized as quiver plots that assigns vectors on a grid of points in a $n$-dimensional space or a streamline plot. The introduction of RNA velocity,[156] defined as the time-derivative of the spliced RNA ($\frac{ds}{dt} = \beta u - \gamma s$), resulted in low dimensional representation of local cell fate predictions, visualized with grid quiver plots. Although such visualizations have often been treated as the "vector field", the development of ***dynamo***[36] firstly enabled the reconstruction of vector fields in functional form.

In Spateo, we further introduce the notion of "morphometric vector field", which reveals the cell migration during development. It is worth noting that once we have an analytical vector field of cell migration, the differential geometry quantities that we can calculate will have direct physical meanings as the morphometric acceleration, divergence, and curl are directly related with morphological changes, such as expansion, contraction, and twisting, and can further reveal the underlying biological processes. In the following, we show that the morphometric vector field as well as the differential geometry quantities can be directly derived from the posterior Gaussian process. Moreover, Spateo is compatible with other OT-based or manually specified cell mapping to compute morphometric vector field and differential geometry quantities based on SparseVFC[157] and dynamo.[36]

*Morphometric Vector Field and Differential Geometry from Gaussian Process*

In this study, we suppose that the morphogenesis of an organ follows a smooth, differentiable vector field or a morphometric vector field in 3D space, that assigns any cell's current physical position ***x*** with a migration velocity vector ***v***, as previously conceptualized.[158] In this study, we show that the same ST alignment method introduced in section "Alignment of spatial transcriptomics profilings of serial sections to create 3D models at whole embryo level" can be also applied to 3D spatio-temporal data directly, where the resulting non-rigid deformation field corresponds to the morphometric vector field which represents the morphological dynamics during the embryo development. The morphometric vector field $\mathbf{V}_*$ at any point in the 3D space $\boldsymbol{x}_*$ represented by the predictive Gaussian process posterior has the following form:

$$q(\mathbf{V}_*) = \mathcal{N}\left( \mathbf{V}_* \middle| \mathbf{U}_* \mathbf{G}^{-1} \mu_u, \mathbf{U}_* \mathbf{G} \mathbf{U}_*^\mathsf{T} \right),$$

where $\mu_u = \sigma^{-2}\mathbf{G}\Sigma\mathbf{U}^\top(\mathbf{P}\mathbf{X}^B - \mathrm{d}(\mathbf{K}\mathbf{R}^A))$ and $\Sigma = (\mathbf{G} + \sigma^{-2}\mathbf{U}^\top\mathrm{d}(\mathbf{K})\mathbf{R}^A)^{-1}$. Denote $\mathbf{C} = \mathbf{G}^{-1}\mu_u$, the exception of $\mathbf{V}_*$ is

$$\mathbb{E}_{q(\mathbf{V}_*)}[\mathbf{V}_*] = \mathbf{U}_*\mathbf{C} = \sum_{i=1}^{k}\Gamma(\boldsymbol{x}_*,\tilde{\boldsymbol{x}}_i)\mathbf{c}_i,$$

where $\Gamma$ is the kernel function, $\tilde{\boldsymbol{x}}_i$ are the inducing points. We can see that the morphometric vector field $\mathbf{V}_*$, or the posterior GP, is a linear combination of the kernels, with the derivative

$$\frac{\partial\Gamma(\boldsymbol{x},\tilde{\boldsymbol{x}})}{\partial\boldsymbol{x}} = -2w\exp\left(-w(\boldsymbol{x}-\tilde{\boldsymbol{x}})^2\right)(\boldsymbol{x}-\tilde{\boldsymbol{x}}) = -2w\Gamma(\boldsymbol{x},\tilde{\boldsymbol{x}})(\boldsymbol{x}-\tilde{\boldsymbol{x}}).$$

From this, we can obtain the Jacobian of the morphometric vector field function:

$$\boldsymbol{J} = \frac{\partial\boldsymbol{f}(\boldsymbol{x})}{\partial\boldsymbol{x}} = -2w\sum_{j=1}^{m}\Gamma\left(\boldsymbol{x},\boldsymbol{x}_j'\right)\mathbf{c}_j\left(\boldsymbol{x}-\boldsymbol{x}_j'\right)^\top,$$

and further obtain Divergence, Curl, Acceleration, and Curvature, etc, see section below and dynamo[36]. To reconstruct the morphometric vector field from Gaussian process, we use the *st.tdr.morphofield_gp* function as shown below:

```
st.tdr.morphofield_sparsevfc(
    adata=adataA,
    spatial_key="align_spatial",
    V_key="V_mapping",
    key_added="VecFld_morpho",
    lambda_=0.02,
    inplace=True,
)
```

### Morphometric Vector Field and Differential Geometry from Cell Mapping

As stated above, Spateo can also learn a morphometric vector field using SparseVFC based on discrete cell mapping by OT-based methods or based on manual specification. In this paper, we selected this option to study Drosophila embryo morphogenesis, as detailed below.

*Preprocessing for morphometric vector field reconstruction.* A series of preprocessing steps are required for obtaining pairs of current 3D spatial coordinates and migration velocity vectors, required for morphometric vector field reconstruction of organ morphogenesis based on sparseVFC.[157] Specifically, in this study, after we reconstructed 3D models of whole Drosophila embryos, we set the embryo from the first time point as the reference by transforming the coordinates of the embryo such that the centroid of the embryo is located at origin, where the A-P axis and D-V axis correspond to the *x* and *y* dimension in the coordinate system respectively. To place embryo models from later time points to the same coordinate system of the initial reference embryo model, we align 3D embryos across time points. To overcome the computational burden of aligning whole embryos, we downsample 2,000 cells from each embryo and use these cells to perform 3D alignment with PASTE. Next, we rotate and translate the embryos from later time points based on GPA (generalized weighted Procrustes analysis) to place embryos from later time points to the same coordinate system. After aligning and transforming the coordinates of embryos, we are ready to calculate pairs of current 3D spatial coordinates and migration velocity vectors from the aligned embryo across time. We focus on analyzing germ band, consisting of CNS, epidermis, hindgut, midgut, muscle and salivary gland, or individual organs instead of the entire embryo, which is more practical given the complexity of the whole-embryo migration pattern and the imperfect data quality. In particular, for individual organs, we focus on midgut and CNS from E7-9h and E9-10h. We first align all the cells of midgut or CNS between two time points, again using PASTE, but this time reduce the weight for the spatial preservation by setting a small $\alpha$, $\alpha = 10^{-0}$ to allow significant cell migration and give more weights on identifying highly similar cells across time points. We next iterate through the optimal transport matrix $\Pi$ and assign each cell from stage 11 to the mostly like cell in stage 13. Because the cells from two time points are aligned in the coordinate system, we can take the coordinates $X_t$ from the early time point and the difference between the future time point to the current time point as velocity vectors or $V_t = X_{t+1} - X_t$. The pairs of $X_t, V_t$ for all cells at a prior time point can then be used to learn the vector field.

The *st.tdr.cell_directions* function from Spateo enables us to learn the mapping from cells from the early time point to the later time point, as shown below:

```
st.tdr.morphofield_velocity(adata=adataA, vf_key="VecFld_morpho")
st.tdr.morphofield_acceleration(adata=adataA, vf_key="VecFld_morpho")
st.tdr.morphofield_curvature(adata=adataA, vf_key="VecFld_morpho")
st.tdr.morphofield_curl(adata=adataA, vf_key="VecFld_morpho")
st.tdr.morphofield_torsion(adata=adataA, vf_key="VecFld_morpho")
st.tdr.morphofield_divergence(adata=adataA, vf_key="VecFld_morpho")
st.tdr.morphofield_jacobian(adata=adataA, vf_key="VecFld_morpho")
```

**Reconstruct a morphometric vector field with the sparseVFC algorithm.** In order to learn morphometric vector fields from the discrete cell mapping obtained from OT-based alignment approaches, we consider a set of pairs of 3D physical coordinates of cell $x \in \mathcal{X} \subset \mathbb{R}^3$ and migration velocities computed from cell mapping $v \in \mathcal{V} \subset \mathbb{R}^3$, $i.e.$ $\{x_i, v_i \in \mathcal{X} \times \mathcal{V}\}_{i=1}^n$, where $n$ is the number of cells from a prior time point. Although the migration velocity vector based on cell mapping such as the optimal transport alignment is noisy and discrete, as stated previously we suppose that the morphogenesis of an organ follows a smooth, differentiable vector field in 3D space that assigns each cell's current physical position $x$ with a migration velocity vector $v$, as previously conceptualized.[158] In this study, to reconstruct morphometric vector fields from discrete cell mappings, we will apply sparseVFC[157] to reconstruct a morphometric vector field of organ morphogenesis to robustly learn a vector-valued function $f$, which outputs an migration velocity vector $v$ given any physical coordinate $x$ of the cell, based on the observed noisy and discrete pairs of data $\{x_i, v_i \in \mathcal{X} \times \mathcal{V}\}_{i=1}^n$.

The final loss function of vector field learning with sparseVFC is as following:

$$\Phi(f) = \frac{1}{2\sigma^2} \sum_{i=1}^{n} p_i \|v_i - f(x_i)\|^2 + \frac{\lambda}{2} \|f\|_{\mathcal{H}}^2,$$

where $\sigma^2$ is a parameter accounts for inlier noise, $p_i$ is a weight deciding the importance of the $i$-th data point in the loss function, $\lambda$ is the regularization coefficient, $\mathcal{H}$ indicates the sparse reproducing kernel Hilbert space, and the second term corresponds to a vector-valued $L_2$ regularization term. The vector field function can be evaluated at any point in $\mathcal{X}$, as a summation of Gaussian kernels centered on the so-called "control points":

$$f(x) = \sum_{j=1}^{m} \Gamma(x, \tilde{x}_j) c_j,$$

where $m$ is the number of control points and $\tilde{x}$ is the coordinate of the control point. $c$'s are coefficient vectors in $\mathbb{R}^3$. And the Gaussian kernel is defined as[157]:

$$\Gamma(x, \tilde{x}) = \exp\left(-w(x - \tilde{x})^2\right).$$

Overall, the SparseVFC algorithm[157] consists of an **E-step** and an **M-step** to allow modeling of noise velocity (inliers) from the data. See more details at Qiu et al.[36] and Ma et al.[157]

To learn the morphometric vector field in Spateo, we will use the *st.tdr.morphofield_sparsevfc* function as shown below:

```
backbone, backbone_length, _ = st.tdr.construct_backbone(
    model=pc_model,
    nodes_key="nodes",
    rd_method="PrinCurve", # ElPiGraph, SimplePPT, DDRTree
    num_nodes=50,
)
```

**Cell**
Article

$$J = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & \dfrac{\partial f_1}{\partial x_3} \\[2ex] \dfrac{\partial f_2}{\partial x_1} & \dfrac{\partial f_2}{\partial x_2} & \dfrac{\partial f_2}{\partial x_3} \\[2ex] \dfrac{\partial f_3}{\partial x_1} & \dfrac{\partial f_3}{\partial x_2} & \dfrac{\partial f_3}{\partial x_3} \end{bmatrix}.$$

### *Differential geometry analyses of morphometric vector fields*

**Jacobian, divergence, acceleration, curvature curl, torsion.** Once the analytical morphometric vector field is reconstructed, we can move beyond morphometric velocity to calculate higher-order differentials, including morphometric Jacobian, divergence, acceleration, curvature, curl, torsion, etc. We start with introducing *Jacobian*, a 3×3 matrix:

A Jacobian element $\partial f_i / \partial x_j$ will tell you whether the migration velocity of one dimension $i$ will be affected by another dimension $j$. The sum of the diagonal of the Jacobian is *divergence*:

$$\nabla \cdot f = \sum_{i=1}^{3} \frac{\partial f_i}{\partial x_i} = \mathrm{Tr} J.$$

Divergence can be used to reveal whether the tissue is expanding (positive divergence) or shrinking (negative divergence). *Curl* is a quantity measuring the degree of rotation at a given point in the morphometric vector field and is defined as:

$$\nabla \times f = \begin{bmatrix} \dfrac{\partial f_3}{\partial x_2} - \dfrac{\partial f_2}{\partial x_3} \\[2ex] \dfrac{\partial f_1}{\partial x_3} - \dfrac{\partial f_3}{\partial x_1} \\[2ex] \dfrac{\partial f_2}{\partial x_1} - \dfrac{\partial f_1}{\partial x_2} \end{bmatrix}.$$

The *acceleration* is the time derivative of the velocity and is defined as:

$$a = \frac{dv}{dt} = \frac{d}{dt} f(x(t)) = \sum_{i=1}^{d} \frac{\partial f}{\partial x_i} \frac{\partial x_i}{\partial t} = J v.$$

Similarly, the curvature vector of a curve is defined as the derivative of the unit tangent vector $\left(\frac{d}{dt} \frac{v}{|v|}\right)$, divided by the length of the tangent ($|v|$):

$$\kappa = \frac{1}{|v|} \frac{d}{dt} \frac{v}{|v|} = \frac{J v (v \cdot v) - v (v \cdot J v)}{|v|^4}.$$

Another very interesting differential geometry quantity for 3D vector fields, that has not been discussed in Qui et al.,[36] is torsion, defined as:

$$\tau = \frac{(v \times \alpha) \cdot (J \cdot \alpha)}{\|v \times \alpha\|^2},$$

which can be used to quantify the degree of twisting of a 3D object.

In Spateo, various differential geometry quantities can be calculated as the following:

```
git clone https://github.com/aristoteleo/spateo-viewer.git
cd spateo-viewer
pip install -r requirements.txt

#Run the Explorer application:
python stv_explorer.py --port 1234
#See the ExplorerUsage for more details.

#Run the Reconstructor application:
python stv_reconstructor.py --port 1234
#See the ReconstructorUsage for more details.
```

### Volumetric and morphometric analyses

With the reconstructed 3D voxel model of the organ or embryo, we can categorize morphogenesis modes for each organ, including organ expansion, shrinkage, migration and convergence. We can further calculate a series of morphometric properties, including length, surface area, volume, and cell density:

By comparing each morphometric quantity across different time points, we can reveal the temporal morphometric kinetics at organ or embryo level.

*Organ backbone analysis with principal curve and principal graph.* A principal curve or graph is a $p$-dimensional curve or graph that passes through the middle of a data cloud. Previously, principal curves or graphs have been used to infer the pseudotemporal trajectories of linear, bifurcated, circular or other complex biological processes from single cell dataset, *e.g.* scRNA-seq.[103] Here, we extend their application to reveal the structure of an organ based on the reconstructed 3D models. In Spateo, we incorporated three powerful approaches to learn the principal curve or graph that represents the organ skeleton: **NLPCA**[159] (Nonlinear principal component analyses), two **RGE** (reversed graph embedding) algorithms, including **SimplePPT** (Simple principal tree algorithm)

**NLPCA.** NLPCA is a global learning algorithm, implemented in prinPy (https://github.com/artusoma/prinPy) that we adapted in Spateo, to compute the principal curve via nonlinear principal component analysis. This algorithm starts with making an initial guess of a principal curve and iteratively refine the curve by creating an autoassociative neural network with a "bottle-neck" layer which forces the network to learn the most important features of the data (https://github.com/artusoma/prinPy).

**RGE.** Reversed graph embedding (RGE) is a general and powerful framework of graph learning that was championed in accurately and robustly inferring complex pseudotemporal trajectories from scRNA-seq or scATAC-seq datasets. The key novelty of the RGE is that it simultaneously learns a principal graph of the cell trajectory and often a low dimensional representation of the single cell dataset that can be mapped back to the original high-dimensional space. Various RGE algorithms have been developed, including SimplePPT, DDRTree, and others, each is developed for certain learning tasks.

The first RGE technique proposed is the SimplePPT algorithm which is tailored for learning a tree structure in the original space, or in some lower dimension retrieved by dimensionality reduction methods such as PCA. DDRTree is a novel extension of simplePPT from the RGE family, and is used by Monocle 2[103] as the default RGE technique. Compared to the SimplePPT algorithm, it provides two key features: first, DDRTree explicitly learns the principal graph while simultaneously reducing its dimensionality and learning the trajectory. Second, DDRTree also dramatically reduces the computational cost by clustering cells into different groups while learning the principal graph and performing dimension reduction.

**ElPiGraph.** ElPiGraph is a scalable and robust method for approximation of datasets with complex structures, via approximating complex topologies with principal graph ensembles that can be combined into a consensus principal graph which does not require computing the complete data distance matrix or the data point neighbourhood graph.[160]

```
tps.transform(spatial)
```

Once a principal curve or graph is constructed, similar to pseudotime algorithms that are implemented in Monocle $^2/_3$, we can project each cell in the physical 3D space to the nearest points on the principal curve or graph. Then we can define a root principal point, such as the head point of the principal graph, and calculate the geodesic distance along the curve or graph to define a measure of **pseudo-space**. We use "**pseudo-space**" to identify principal-curve/graph dependent genes similar to pseudotime-dependent genes, as previously implemented in Monocle 2/3.

### Apply generalized linear models (GLM) to detect differentially expressed genes in different contexts

In this study, we used GLM as a general approach to identify genes significantly changing as a function of some continuous variables, such as the digitized layers / columns, the pseudospace defined by A-P axis or the principal curve learned for a particular 3D reconstruct organ, or the differential geometry quantities computed after learning the morphological vector field. In general, the full model of the GLM regression is:

$$\log(expression + 1) \sim ns(variable, 3),$$

where ns represents the natural splice. By default, we use 3-order or cubic splines.

And the reduced model is:

$$\log(expression + 1) \sim 1.$$

A likelihood ratio test is then used to compare these two models and to compute $p$-value. We BH adjust the P-value and define significant genes as genes with q-val < 0.05.

### Explore 3D spatial transcriptomics data with Spateo-viewer

The Spateo-viewer website can be accessed at https://viewer.spateo.aristoteleo.com/. We also provided a tutorial of 22 pages to introduce the usage of spateo-viewer: https://github.com/aristoteleo/spateo-viewer/blob/main/usage/spateo-viewer.pdf. The GitHub repository for Spateo can be found here: https://github.com/aristoteleo/spateo-viewer?tab=readme-ov-file

It is also worth noting that Spateo-viewer can be runned as a standalone tool and the users can first git clone the repo, followed by a few lines of code to start the viewer locally and then open it on the local browser for fast interactive and intuitive data exploration and analyses. Run the tool locally may avoid the internet traffic of Spateo-viewer website. The steps to use Spateo-viewer locally can be found here:

```python
y_max, y_min = np.max(sliceA.obsm[spatial_key][:,1]),
np.min(sliceA.obsm[spatial_key][:,1])
Bottom_threshold = (1-overlap_ratio)*(y_max-y_min) / (2-overlap_ratio) +
y_min
Top_threhsold = y_max - (1-overlap_ratio)*(y_max-y_min) / (2-overlap_ratio)
sliceA_crop = sliceA[sliceA.obsm[spatial_key][:,1] >= Bottom_threshold
,:].copy()
sliceB_crop = sliceB[sliceB.obsm[spatial_key][:,1] < Top_threhsold
,:].copy()
```

### Supplementary Methods

#### Transfer cell label from annotated scRNA-seq cell atlas to Stereo-seq

To annotate this massive 3D spatial transcriptomics dataset, we transferred cell type labels from an existing single cell atlas.[161] We pooled data from all slices to result in a combined dataset for each of the E9.5 and E11.5 embryos. We removed poor quality cells by filtering out cells with fewer than 100 total counts, reaching a final total of 904,017 cells across 90 slices for the E9.5 embryo and 6,986,674 cells across 84 slices for E11.5. To annotate spatial domains, we first used spatially constrained clustering (SCC) to identify spatial domains within each slice. Then the spatial domains across slices are integrated and cleaned up in the 3D space manually (Figure S2A). Past comprehensive atlasing efforts have identified and annotated hundreds of cell types and their signature gene expression profiles from single-cell data. We used an scRNA-seq reference[161], and split the reference data into a training and test set. We next used scVI[39] to jointly project the reference and our Stereo-seq data to a lower-dimensional space and then

iteratively annotated the cell type. First, we assigned one of nine possible major trajectories to each cell (mesoderm, neuroectoderm, central nervous system (CNS) neurons, epithelium, muscles, blood lineages, endothelium, peripheral nervous system (PNS) neurons and PNS glia). We next trained a classifier on the latent representation of the training set with XGBoost[40] and then tested it on the test set before applying it to the Stereo-seq data. We followed a similar procedure to subcluster each major trajectory. By the end of this hierarchical process, we assigned all cells of the E11.5 embryo to one of 103 cell types (Figures S2B and S2C), and all cells of the E9.5 embryo to one of 55 cell types. We assessed the quality of cell type annotation by confirming that region specific cell types are enriched in expected spatial locations (*e.g.* Gut epithelium, midbrain neuroectoderm, Eye field, etc.). For cell types that should be present at high density in particular spatial locations (*e.g.* Gut epithelium, midbrain neuroectoderm, Eye field, etc.), we confirmed consistent localization with expectations. Additionally, the average marker gene profile of the top 50 markers for each mapped cell type in the Stereo-seq data was highly cell-type specific and highly consistent with that of the scRNA-seq reference, with expression of each marker found to be significantly higher expressed in the corresponding cell type compared to in all other cell types (Figure S2B). To allow in-depth analyses of the heart, the heart region was identified, isolated and annotated separately.

### Digitization benchmarking

*Digitization on simulated and the Macaque cortex section datasets.* For the two simulated scenarios (Figures 4E and 4F), we manually drew a half circle and a trapezoid with FIJI (ImageJ) and then converted the selected pixels into acceptable format for Spateo and Belayer.[25] Digitization was then performed using *st.dd.digitize* and the Belayer pipeline, respectively.

For the Macaque cortex section, we obtained the T40 section from an atlas of the macaque cortex,[38,43] reported by Chen et al. We dropped the transcriptomics information and only used the spatial coordinates of cells in the section to reduce memory required for data loading. Digitization was then performed using *st.dd.digitize* and the Belayer pipeline, respectively.

The digitization values (*i.e.*, potentials) were uniformly divided into 5 groups, representing layers or columns for visualization purposes.
*Digitization of the Visium mouse brain section dataset.* We used the sagittal section of the mouse brain 10X Visium dataset,[13] and clustered the spots using the Leiden algorithm with Seurat. The cortex region was then extracted manually according to the clustering results, followed by Spateo's digitization. Using digitization values as the relative coordinate along the A-P axis, we can test for significantly expressed genes along the digitization axis using the generalized additive model implemented in Spateo.
*Digitization of the MERFISH U2-OS cell line dataset.* We used a selected set of FOVs from a U2-OS cell line data.[117] We used the cell segmentation result from the original work to define the nucleus and cytoplasm. Then digitization is performed for the nucleus to define an axis from the nucleus centroid to the nucleus boundary. Similarly we used digitization to define an axis from the nucleus outer membrane to cell membrane for the cytoplasm areas. Using digitization values as the relative distances to nuclear centroid or nuclear boundary, we can test for significantly expressed genes along the digitization axis using the generalized additive model implemented in Spateo.
*Digitization of the MERFISH mouse brain section dataset.* We used selected FOVs from an adult mouse brain atlas,[55] and the native cell type annotations from this dataset. Digitization was performed on a cortex FOV using both *st.dd.digitize* and the Belayer pipeline. Then, the digitization values are used to assign the relative cortical depth of each cell in the section. We evaluate the performance of Spateo's and Belayer's digitization by assessing the laminar distribution of layer specific excitatory neuron subtypes. Cell type visualization was optimized using Spaco.[162,163]

Scripts for the benchmarks above can be found in the data and code availability section.

### Cell-cell interaction analysis benchmarking

*Processing of the MERFISH mouse dataset.* For the following described benchmark, we used selected FOVs from an atlas of the adult mouse brain,[55] and the native cell type annotations from this dataset.

### NCEM benchmark

For benchmarking with NCEM,[29] we selected one or two marker genes for each of the "layers" of the brain: L2/3, L4/5, L5, L5/6 and L6 to serve as target genes. We compared NCEM to Spateo's niche model (see the "niche model" subsection of "predictor matrix setup" under "spatially-aware modeling of networks of CCI in 3D spatial transcriptomics" for details). For NCEM, 150 was used for the distance parameter, which was the same value used as Spateo's bound to define the niche. For each target gene, we compared the observed expression vector for each target gene to the reconstruction from the model output using the coefficient of determination ($R^2$). To generate confidence intervals for this calculation, we performed 1000 iterations of bootstrap resampling, each containing 200 cells, and computed the $R^2$ value for each sample. Using the overall $R^2$ and the confidence interval, we used a Fisher Z-test to determine whether the result for Spateo was significantly higher than the result for NCEM.

*Processing of the CosMx cancer dataset.* For the following described benchmarks, we used a lung cancer sample profiled by Nanostring's CosMx, found from the company's website and reported in an article in *Nature Biotechnology*[44]: https://nanostring.com/products/cosmx-spatial-molecular-imager/ffpe-dataset/. This dataset is pre-annotated, however we used inferCNV[164] to identify and annotate cancer cells (previously largely labeled "Basal").

*COMMOT benchmark.* For benchmarking with COMMOT,[45] we used its ligand:receptor (L:R) mapping to regress on gene expression. From the complete set of ligands included in COMMOT's L:R database, we subset to those found to be expressed in at least 5% of cells in the CosMx lung sample. We similarly subset the receptors in this way. We subset L:R pairs to those in which either of the ligand or receptor in the pair could be found in one of our filtered subsets. We constructed separate L:R arrays for secreted signaling and membrane-bound signaling. We used 300 as the distance unit threshold for secreted signals and for extracellular matrix-mediated signals. This is also the upper bandwidth for Spateo modeling (see "spatially-weighted modeling for characterization of spatial specificity of signaling effects" for description of the bandwidth parameter), computed by finding the average distance for 70 nearest neighbors (from a similar calculation as is detailed in "spatially-weighted modeling for characterization of spatial specificity of signaling effects", but for a 2D scenario rather than 3D). For membrane-bound signaling, we used 100 as the distance threshold, also the lower bandwidth for Spateo modeling. To identify the target genes to model, we computed Moran's $I$ for each gene of the CosMx lung sample, filtering to those with statistically significant Moran's $I > 0.25$. For each target, we fit a spatially-weighted Poisson GLM and Poisson generalized linear model (GLM) using the COMMOT L:R array and compared the observed expression vector for each target gene to the reconstruction from the model output using the Pearson correlation and Spearman correlation. Comparing these correlations between observations and model predictions across the two models established a sense of what the spatial weighting added to the general predictive ability.

Instead of optimal transport, Spateo uses spatial weights to aggregate ligand expression in the local neighborhood of each cell (See more details in the "predictor matrix setup" section) to model ligand-receptor interactions. To evaluate the similarity between COMMOT's and Spateo's L:R mapping, we filtered both arrays to L:R pairs shared by both, and computed a Jaccard index. More specifically, we created two vectors, one for the COMMOT array and one for Spateo's array. For each cell, we checked for zero or nonzero in each of the COMMOT and Spateo arrays, adding a 0 or 1 to the appropriate vector, respectively. We calculated the intersection-over-union for these vectors. We additionally counted the total number of nonzero L:R features for each cell and computed correlation coefficients.

To evaluate model coefficients, we compared the relative magnitude for "similar" L:R interactions and the quantitative similarity from the L:R mapping array. We define similar L:R interactions on the basis of biological similarity, for example, belonging to the same family of receptors, *e.g.* integrins, Frizzled receptors, etc. For the similarity comparison, we computed pairwise Pearson correlations between all pairs of columns from the L:R mapping array. Biologically similar L:R interactions with highly correlated (>0.7) mappings were expected to return similarly-sized coefficients.

*CellOracle benchmark.* We compared the ability of CellOracle's gene regulatory network (GRN) inference models to Spateo's downstream transcription factor (TF)-target model. To identify the target genes to model, we computed Moran's $I$ for each gene of the CosMx lung sample, filtering to those with statistically significant Moran's $I > 0.25$ (these targets are shared with the COMMOT benchmark). Although we cannot measure whether a positive predicted coefficient is correct without external ground truth data that cannot currently be measured with spatial transcriptomics, we can evaluate this prerequisite: to constitute a true regulatory relationship, genes encoding both TF and target must be present in the same cell. For a given cell, if a given TF was truly to regulate a given target gene, a positive coefficient should not be assigned unless there is sufficient evidence of their coexistence (in this case, by the presence of TF and target transcripts). Thus, we use co-expression of TF-target pairs in cells with nonzero correlation coefficients as "pseudo-true positives". Similarly, for cells with zero coefficients, absence of co-expression of TF and target is a "pseudo-true negative". For cells with nonzero correlation coefficients, absence of co-expression is a "pseudo-false positive". For cells with zero coefficients, co-expression is a "pseudo-false negative". We note, however, in this specific case, co-expression can occur without a regulatory relationship. We assume that since Spateo's model filters to TFs with prior evidence of binding to particular targets (from scATAC-seq or footprinting,[137,138] see "structure of prior knowledge files of intercellular and intracellular interactions"), for each target, generally all TFs remaining are capable of regulating target expression and that each pseudo-false negative is valid. From these metrics, we calculated true positive rate/TPR, true negative rate/TNR, false positive rate/FPR, false negative rate/FNR, and precision for both the CellOracle and Spateo models.

*Processing of the OpenST mouse head dataset and Slide-Tags human tonsil dataset.* For the COMMOT benchmark, we also used two additional datasets and the native cell type annotations from each: a section of the head of a mouse embryo at E13 measured using OpenST,[38] and a slice of the human tonsil profiled using Slide-Tags.[165] For the mouse data, we used 200 as a distance unit threshold for secreted signaling, and for the tonsil data we used 156.

### Analysis of Central Nervous System of Stereo-seq mouse embryo data
#### Digitization of Zona Limitans Intrathalamica (ZLI) and vicinity
For the ZLI analysis, we defined two subsets, the dicephalic ring or the ZLI flanking region, each shown in Figure 6A. To identify the ZLI, we visualized the expression pattern of the marker *Shh* in this region, and manually assigned a ZLI label to all cells expressing *Shh* within the dorsal-ventral band extending from the floor plate.

# Cell
## Article

**CellPress**
OPEN ACCESS

We considered the ring structure formed by the region of the neural tube the ZLI is part of, namely the diencephalic ring. To extract this ring, according to the known localization of ZLI, we first aimed to extract the neural tube cells near the diencephalon. We gathered the cells within a maximum of 50 distance units from the inner surface as the diencephalic neural tube as shown in the top part of Figure 6A. Then we applied digitization on the extracted neural tube, and delineated the diencephalic ring using a cutoff where digitized potential is between 40 and 65. After the acquisition of this diencephalic ring, we performed digitization along the dorsal ventral axis (Figure 6B). In the digitization procedure, the initial potential field was manually defined by selecting a bunch of floor plate cells as ventral polar and some roof plate cells as dorsal polar. We solved the potential field on the spatial neighbor network of the diencephalic ring cells, and used the smoothed potential as the relative D-V coordinate for further analysis.

To focus on this structure and its immediate surroundings or the ZLI flanking region, we selected cells separated from the cells of the ZLI by less than 100 distance units along the x-axis, 50 units along the z-axis or 25 units along the y-axis (arbitrary cutoffs, with x cutoff being highest because of its alignment along the embryo's anterior-posterior (A-P) axis). For digitization along the rostral caudal axis, we first aimed to identify the inner surface of the neural tube to facilitate the definition of the R-C axis. To reduce computational cost, we manually applied a cutoff on spatial coordinates, using the "z_correction" field in the obsm slot, with x greater than 1300 and y between (300, 750), to roughly extract a subset near diencephalon. Then we constructed the point cloud model of the subset using *construct_pc* of *spateo.tdr* module, followed by surface mesh reconstruction using *construct_surface*. The mesh surface was then clipped by intersecting with a manually defined ellipsoid (created using *ParametricEllipsoid* of *pyvista* package) to delineate the inner surface of the neural tube. Then we extracted the surface mesh grid network using *mesh_model.faces* and *mesh_model.points* from *pyvista*. We manually defined the initial potential field, where a bunch of cells on the rostral polar were initialized with minimal potential 0 and the caudal polar with maximal potential 100. Then the potential field was solved iteratively using the heat equation. Each cell in the ZLI subset was assigned with the smoothed potential of the nearest vertex on the mesh grid, as shown in Figure 6C. This potential was used as the relative coordinate of the cell on the rostral caudal axis of the diencephalic tube for further analysis.

### Digitization of the spinal cord

The spinal cord cells were first extracted based on the transferred annotation. Subsequently, outliers were manually filtered using 3D point clouds where the 3D coordinates of all cells underwent clustering via DBSCAN (*sklearn*) with parameters set to eps=70 and min_samples=10000. Dorsoventral digitization shown in Figure S6F was performed on the 2D representation of these spinal cord cells, which was obtained by only using the X and Y coordinate, along the medial lateral axis. As for Figure S6I, we performed a transverse projection to construct the pseudo cross plane. To expedite computation, a down-sampled subset of 60,000 cells (out of 683,920 cells) was utilized to construct the principal curve of the spinal cord using *spateo.tdr.widgets.changes.Principal_Curve*. The points representing spinal cord cells were then mapped to the principal curve using KDTree (*scipy*) to obtain new Z coordinates, defined by the principal curve. Similarly, for the notochord, the annotated cells (52078 cells, no downsampling required) were used to construct its principal curve using the same method. Subsequently, the point of interest along the principal curve of the spinal cord will be connected to the corresponding point of the principal curve of the notochord with the same Z coordinate to derive the vector A, indicating the direction along the dorsoventral axis on the transverse plane. For each cell in the spinal cord with the same Z coordinate, a vector B was created by connecting its original point to the corresponding point on the principal curve of the spinal cord. By utilizing vector A and vector B, the radians of cells and the dorsoventral axis were calculated via cross product operations. Leveraging the radians and the length of each cell away from the corresponding point on the principal curve of the spinal cord, all spinal cord cells could be projected onto an overall transverse plane.

### Identification of axis variable genes and gene clusters

After digitization, we considered the gene expression patterns along those axes. We first grouped the cells into 100 bins according to their digitized coordinates, and constructed a vector of 100 elements for each gene by calculating its mean expression in each bin. We then identified genes whose expression is significantly related to digitized axis coordinates, using the generalized additive model implemented in the *dynamo.tl.glm_degs* function of the *dynamo* package. We kept the genes with a p-value less than 0.01 as R-C or D-V variable genes. Then, we grouped those axis variable genes into gene clusters using hierarchical clustering. To avoid interference from sparsely expressed genes, the gene vectors were smoothed using a Gaussian kernel of 3 and scaled using min-max normalization before clustering.

### Cell-cell interaction modeling and parameters for the ZLI and vicinity

To choose target genes for cell-cell interaction (CCI) modeling of the ZLI region subset (Figure 6A), we subset the list of all genes found to be significantly variable along the R-C axis (see "identification of axis variable genes and gene clusters" for details of how this was done) (Table S3). We took the set of these genes that were also identified as cell type markers. To do so, we computed the mean expression level for each gene in each cell type and compared it to mean expression of all other cells of all other types, keeping genes with fold change mean expression over others >2.5. To choose ligands, receptors and transcription factors (for downstream models) to include in modeling, we filtered all ligands, receptors, and transcription factors in our database (see "structure of prior knowledge files of intercellular and intracellular interactions" section of the main method details) to those expressed in at least 1000 cells. For each target gene, we fit Poisson ligand models (see "ligand model" subsection of "spatially-aware modeling of cell-cell interaction in 3D spatial transcriptomics"), with a distance of 6.6 units used for computations related to membrane-bound signaling and 16.5 units for computations related to secreted signaling, determined by the methodology detailed in the section "Spatially-weighted modeling for characterization of spatial specificity of signaling effects". For the minimum and maximum

bandwidths for spatially-weighted regression, we used 1.5 times the membrane-bound and secreted distances, or 10.0 and 25.0 units, respectively. For weighted regression for intracellular models (see "weighted modeling for downstream (transcription factor-gene) model"), we used default settings. Notably, this sets the lower bandwidth (for intracellular models, a fixed number of nearest neighbor cells in gene expression space) to be 0.2% of the total number of cells (about 300 cells for the ZLI region) and the upper bandwidth to be 0.5% of the total number of cells (about 700 cells for the ZLI region).

For the diencephalic ring (Figure 6C), genes found to be significantly variable along the D-V axis were used as targets (Table S4), with the same parameters as described for the ZLI otherwise.

### Construction of intercellular & intracellular signaling network

To create the combined intercellular & intracellular network (Figures 6F and 6G), we first selected morphogens of interest in the ZLI region based on their previously established roles in neurogenesis. These included *Wnt* family ligands,[166] *Bmp* family ligands,[167] *Shh*,[166] *Fgf8*[166] and *Gdf11*.[168] We also included ephrins due to their pronounced expression in the region. For each target gene of the ZLI region CCI model (see "cell-cell interaction modeling and parameters for the ZLI and vicinity"), we computed the average predicted effect size for each of our selected ligands over all cells expressing the target gene. For each ligand, we ranked the top target genes based on average predicted effect size. For display in the network, we selected genes based on specificity for the ligand (*i.e.* the gene ranked among the top targets only for that ligand or only for ligands with similar spatial patterns) and/or biological significance. For example, *Dcx* and *Thsd7a* are markers of neurons undergoing migration and dendritic growth.

### Cell-cell interaction modeling and parameters for the MHB and vicinity

To choose target genes for cell-cell interaction modeling of the MHB region subset (Figure S3A), we selected genes found to be significantly variable along the R-C axis (see "identification of axis variable genes and gene clusters" for details of how this was done) (Table S5). For selection of ligands, receptors and transcription factors, we followed the same procedure as for the ZLI. Similarly for the ZLI, for each target gene, we fit Poisson ligand models with a distance of 6.5 units used for computations related to membrane-bound signaling and 16.0 units for computations related to secreted signaling. The minimum and maximum bandwidths (see "spatially-weighted modeling for characterization of spatial specificity of signaling effects") were 9.74 and 24.0 units, respectively, again obtained by multiplying the membrane-bound and secreted signaling distances by 1.5. For intracellular models, the lower and upper bandwidths were set to 250 and 600 cells, respectively (see "weighted modeling for downstream (transcription factor-gene) model").

### Cell-cell interaction modeling and parameters for the spinal cord

To choose target genes, ligands, receptors and set model parameters, we followed a similar procedure for the analyses of ZLI and MHB. For target genes, we selected genes found to be significantly variable along the D-V axis (Table S6).

## Spateo Alignment Benchmark

### Benchmark Datasets

**MERFISH dataset from mouse hemibrain.**[37] The MERFISH dataset includes 129 slices at 100 -μm intervals with 9.3 million cells and measures about 1,100 genes. These 9.3 million cells delineate all 11 major brain regions: olfactory areas, isocortex (CTX), hippocampal formation, cortical subplate, striatum, pallidum, thalamus, hypothalamus, midbrain, hindbrain and cerebellum. The dataset further registered the cell atlas to Allen mouse brain common coordinate frame v3 (CCF v3) using both DAPI images and the cell type based landmarks. Therefore in the evaluation process, we use the CCF coordinate as the ground-truth and evaluate the performance of various alignment algorithms.

**STARMAP Plus dataset from mouse central nervous system.**[116] The STARMap Plus dataset contains nineteen 10-μm-thick CNS tissue slices harvested from three mice, where sixteen slices are from the coronal brain (Well) and three slices are from the sagittal brain (Sagittal). After data processing with ClusterMap workflow, the dataset includes 250 million RNA reads and 1.1 million cells. According to the STARMap Plus protocol, each slice has a three dimensional volume. This enables us to generate pseudo-slices with different spatial distributions in the same coordinate space.

**BAR-Seq dataset from mouse forebrain hemisphere.**[49] The BAR-Seq dataset builds a brain-wide high-resolution map containing 1.2 million cells, which are covered by 40 hemibrain coronal slices. 107 cell type marker genes' expression is interrogated by BAR-seq. More importantly, they have registered BAR-seq data to the Allen mouse brain common coordinate frame v3 (CCF v3) by manual manipulation. Thus it can be used to directly evaluate the performance of various alignment algorithms.

**OpenST dataset of the human metastatic lymph node.**[38] This OpenST dataset generates 10 μm sections spanning a tissue depth of 350 μm. After the experiment, they obtained about 1 million cells across 19 sections, with median capture of genes (313-624) and UMIs (438-1, 008) per segmented cell.

**Stereo-seq from macaque cortex.**[38,43] The Stereo-seq dataset includes a total of 119 sections at 500-μm spacing covering the entire macaque cortex with more than 30 million cells. Each slice contains 226, 310 segmented cortical cells on average with 458 genes and 819 UMIs per segmented cell. We choose this dataset to test the scalability of different methods thanks to the large scale of this Stereo-seq dataset, see the scalability test **section below**.

### Benchmark Tasks

We mainly focus our testing on the following 7 tasks: (1) Non-rigid alignment; (2) Partial overlapping alignment; (3) multi-slice refinement; (4) Mesh correction; (5) Robustness test; (6) Scalability test; (7) Real 3D ST reconstruction from consecutive slices. These test

**CellPress**
OPEN ACCESS

tasks cover the existing challenges of ST alignment and 3D reconstruction. In addition, we also include parameter studies to investigate the sensitivity of the method to the parameters and the setting intervals.

***Non-rigid alignment***

*Simulated data generation.*  We first obtain 4 pseudo-slices for each STARMap Plus slice, denoted as slice A, B, C, and D respectively. Then the TPS algorithm from thin-plate-spline package (1.1.0) followed by a random rotation and translation are applied to the spatial coordinate of each pseudo-slice to obtain distorted slice $A^*, B^*, C^*$, and $D^*$ respectively. Note that we only run the simulation once and save the results for each method, in other words, the input of each method is the same without randomization. TPS transformation requires a set of corresponding source and target points. Specifically, we first generate $N \times N$ grids based on the X-Y coordinates of the ST data, and we set the center of these grids as the source points. In this paper, we set the grid number N to 2. The target points are obtained by adding random Gaussian noise to the source points. By controlling the variance of the Gaussian noise, we can thus control the TPS distortion level. The variance of the Gaussian noise is set to $(H_{grid} + W_{grid}) \cdot distort\_level/100$, where $H_{grid}$ and $W_{grid}$ are the height and width of the grid respectively, the distort level is set by the user. Note that we add the four corners of the ST data to the source and target points as well, without adding Gaussian noise to the target points, in order to prevent the TPS from producing extremely large distortions. After constructing the source and target points, a TPS transformation is fitted by

```
[int(i)+1 for i in np.exp(np.log(100) + np.arange(19)*np.log(10) / 5)] +
[500000]
```

and consequently performs distortion by

```
_, pi = st.tdr.cell_directions(
    adataA=adataA,
    adataB=adataB,
    layer="log1p_X",
    spatial_key="align_spatial",
    key_added="mapping",
    alpha=0.001,
    numItermax=200,
    numItermaxEmd=100000,
    inplace=True,
)
```

where alpha is the regularization parameter and is set to 0.1 for the benchmark, spatial is the x-y coordinate of the ST data.

*Metrics.*  When we align Slice A* to the coordinate system of Slice B, we wish to recover the shape of Slice B, which is similar to Slice A. Thus, ideally, we want $\mathcal{T}$(Slice A*|Slice B) to be as similar as possible to Slice A, where the one-to-one correspondence between Slice A* and Slice A is known. Thus we can directly measure the pairwise mean average error (pairwise MAE) between $\mathcal{T}$(Slice A*|Slice B) and Slice A to indicate the performance of non-rigid alignment, i.e.,

$$\text{pairwise MAE} = \frac{1}{N} \sum_{s_i^A \in S^A} \left\| \mathbf{x}_i^A - \mathcal{T}(\mathbf{x}_i^A) \right\|_2,$$

where $\mathcal{T}$ is the transformation provided by different alignment algorithms. Lower pairwise MAE indicates better capability to handle non-rigid deformation and vice versa.

In the benchmark, we first fix the distort level to 5 and evaluate the performance of each method on all Sagittal and Well slices, as shown in the barplots in the Figure S4C iii). Next, in the Figure S4C iv), we plot the lineplot where the distort level is changed from 0 to 10 on well #10 slice with 10 repeated runs.

### Partial overlapping alignment

*Simulated data generation.* We first split the STARMap Plus slice into 16 pseudo-slices. Next, we simply crop slices based only on the x or y axis with a hard threshold to generate partially overlapping slices, where the overlap ratio is fully controlled by the threshold, as shown in Figure S4A iii). Taking the y-axis as an example, given the Slice A, Slice B and overlap ratio, we can obtain the cropped slices as follows:

```
tps = ThinPlateSpline(alpha=alpha)
tps.fit(src_points, dst_points)
```

After cropping, we then apply a random rotation and translation on the cropped slice A, denoted as sliceA_crop*.

*Metrics.* When we align sliceA_crop* to the coordinate system of sliceB_crop, the $\mathcal{T}$(sliceA_crop*|sliceB_crop) should also align sliceA as sliceA shares the same coordinate system of sliceB_crop. As the one-to-one correspondence between $\mathcal{T}$(sliceA_crop*|sliceB_crop) and sliceA is given, we measure the pairwise MAE described above between them.

In this benchmark, we vary the overlap ratio from 0.3 to 1 with an interval 0.1. To avoid having high similarity between slice pairs, we choose slices with an interval of 8 to form slice pairs. The performance of each method on Sagittal #2 and Well #8 is reported in Figure S4D **iii**).

### Multi-slice refinement

*Simulated data generation.* For multi-slice refinement benchmark in Figure S4E, we first split the STARMap Plus slice into 10 pseudo-slices and employ manual guided strategy to crop slices, *i.e.*, manual interactive cropping, which is a function integrated in the Spateo: spateo.tdr.interactive_rectangle_clip. This cropping strategy generates more complex and difficult situations, *e.g.*, an extremely low overlap ratio between neighboring slices. It is almost impossible to obtain a satisfactory 3D reconstruction result if only neighboring slices are considered, e.g, sequential alignment.

*Metrics.* multi-slice refinement focuses on 3D reconstruction of multiple ST slices, so we should evaluate the entire 3D reconstruction result rather than pairwise alignment. Specifically, we first find the optimal rigid transformation that best aligns the reconstructed 3D structure with the ground truth in the x-y axis using Procrustes Analysis:

$$\widehat{\mathbf{R}}, \widehat{\mathbf{t}} = \arg\min_{\mathbf{R}, \mathbf{t}} \sum_{k=1}^{K} \sum_{n=1}^{N^k} \|\mathbf{x}_n^k \mathbf{R}^\top + \mathbf{t} - \mathcal{T}_k(\mathbf{x}_n^{k*})\|^2,$$

where $K$ is the slice number, $N^k$ is the cells/spots number in slice $k$, $\mathcal{T}_k$ is the transformation for slice $k$ by the alignment algorithm. After that, we can calculate the global MAE, which allows us to assess the global consistency of the 3D reconstruction across all slices:

$$\text{global MAE} = \frac{1}{\sum_{k=1}^{K} N^k} \sum_{k=1}^{K} \sum_{n=1}^{N^k} \|\mathbf{x}_n^k \widehat{\mathbf{R}}^\top + \widehat{\mathbf{t}} - \mathcal{T}_k(\mathbf{x}_n^{k*})\|_2.$$

In this benchmark, we combine the multi-slice refinement with each method, *i.e.*, PASTE, PASTE2, Moscot, SLAT, SPACEL, as well as Spateo (pairwise mode) and evaluate the performance on all Sagittal and Well slices. The visualization as well as the statistical results are shown in the left and right of Figure S4D, respectively.

### Mesh correction

*Data preparation.* As previously mentioned, MERFISH and BARSeq dataset are registered to the Allen CCF v3 by manual manipulation, providing both a mesh and the ground-truth. Therefore, in the mesh correction benchmark, we select the MERFISH and BARSeq dataset for benchmarking. For each slice, we add random rotation and translation on the ground-truth spatial coordinate.

*Metrics.* Because the mesh correction step is also manipulating the entire 3D reconstruction results, we use the same metric described in multi-slice refinement, *i.e.*, global MAE.

In this benchmark, we combine the mesh correction with each method, *i.e.*, PASTE, PASTE2, Moscot, SLAT, SPACEL, as well as Spateo (after multi-slice refinement), and evaluate the performance on the MERFISH and BARSeq dataset. The visualization of the 3D reconstruction of MERFISH dataset before and after mesh correction based on Spateo's results is demonstrated in Figure 4D left. The statistical results are reported in Figure 4D right.

## Cell
**Article**

### Robustness test

*Data preparation.* Our Stereo-seq from mouse embryos contains more than 80 slices at each time point and has a large number of spots, so it is well suited for robustness tests with downsampling with respect to the number of slices or the number of spots. For the slices downsampling, we sample at a constant interval, *e.g.*, preserve interval 5 means the sampled slices were spaced 5 slices apart. For the spots downsampling, we randomly downsample a fraction of single cells.

*Metrics.* We test robustness by comparing the results of Spateo's reconstruction after downsampling to those reconstructed on the full mouse embryo data. For the slices and spots downsampling, we directly measure the MAE and Pearson correlation between the full data and the downsampled one.

### Scalability test

*Data preparation.* The Stereo-Seq dataset from macaque cortex contains more than 200k cells per slice, which is a very large scale of data. We choose slice #85 and #86 for the benchmark, which contain 528, 264 and 492, 701 cells respectively. We randomly downsampled the cells in the slices to generate subset datasets with a range of cell numbers as

```
downstream_model.effect_distribution_heatmap(
    position_key="spatial",
    interaction_subset=interaction_subset,
    coord_column=0,
    effect_threshold=0.05,
    cmap="bwr",
    title="Signaling effect hotspots along axis",
    save_show_or_return="show",
    fontsize=10,
)
```

*Metrics.* We record the time and CPU/GPU memory usage for each downsampling configuration for each method. For those methods that use GPU acceleration, we record the peak GPU memory usage, and for those that use CPU, we record the peak CPU memory usage.

### 3D reconstruction of real ST data from consecutive slices

*Data preparation:.* We use the MERFISH dataset from mouse hemibrain, BAR-Seq dataset from mouse forebrain hemisphere, OpenST dataset from human metastatic lymph node, Stereo-seq from macaque cortex, as well as our 3D spatial transcriptomics Stereo-seq profiling from mouse embryos at E9.5 and E11.5 for this benchmark. To introduce perturbation, we apply random rotation and translation to each slice of each dataset.

*Metrics.* For those datasets with ground-truth, *i.e.*, MERFISH and BARSeq datasets have already been registered to Allen CCF v3, we use pairwise MAE as the metric. For those datasets without ground-truth, *i.e.*, OpenST, Stereo-seq, and our mouse embryos datasets, only non-referenced metrics can be used. Previous methods mainly focus on measuring the consistency of the mapping labels, while ignoring the spatial consistency of the mapping. To this end, we design a new metric called Contextual Label Consistency (CLC) score that considers both label and spatial consistency of the mapping, which is defined as

$$CLC(M) = \frac{1}{N} \sum_{i=1}^{N} \left( I(l_1(i) = l_2(i')) \cdot \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} d(i',j') \right),$$

where $M$ denotes the mapping matrix, $\mathcal{N}_i$ is the spatial neighborhood of spot/cell i, which is calculated by K-nearest neighbors. Considering the effect of downsampling on neighborhood scales, K is set to 2.5% of the average cell number of the slice pair. The prime on an index indicates the most probable corresponding spot/cell in the second sample, *e.g.*, for spot/cell I in the first sample, $i' = \arg \max_i M_{ii'}$ and $d(i',j') = 1$ if $j' \in \mathcal{N}_i'$ otherwise 0. The results are reported in Figure 3A–3E.

### Parameters Studies

*Data preparation.* We use the STARMap Plus Sagittal #3 slice and generate two pseudo-slices for the parameters studies. Subsequently, ratio cropping was applied to two pseudo slices with overlap ratio range from 0.5 to 1. Random rotation and translation are added.

Metrics: We follow the metric used in partial overlapping alignment benchmarks and use the pairwise MAE.

In this benchmark, we investigated the impact of parameters $\lambda$, $K$, $B_a$, $B_b$, and $\kappa$ on performance in different scenarios, *i.e.*, with different overlap ratios. We also include other methods for comparison.

### Benchmark Methods

Spateo is compared with PASTE, PASTE2, Moscot, SLAT, STAlign, and SPACEL, which are recently published ST alignment methods. We run these methods in Python (3.8.16) by their official release Python package:

| Method | PASTE | PASTE2 | Moscot | SLAT | STAlign | SPACEL | Spateo |
|---|---|---|---|---|---|---|---|
| Package | paste-bio 1.4.0 | commit 517d6584 | moscot, 0.3.4.dev3+gf71f976 | scSLAT, 0.2.2 | STalign, 1.0 | SPACEL, 1.1.8 | spateo-release, 1.1.0 |

For GPU hardware acceleration, we use a NVIDIA A100 with 40 GB memory and CUDA 12.5 driver. PASTE, SLAT, STAlign, and Spateo are equipped with torch 2.0.0, and Moscot is equipped with jax 0.4.13 and ott-jax 0.4.4. For PASTE2, the authors have not yet provided a GPU version so it is run on CPU. For SPACEL, the module that performs alignment is called Scube and doesn't support the GPU acceleration. For each method, we used the default parameters and settings in their papers or packages. Specifically, the trade-off parameter $\alpha$ is set to 0.1 for PASTE and PASTE2 and 0.5 for Moscot. Note that due to memory and runtime limitations, to run the OT-based methods as well as SPACEL, we have to downsample the number of cells/spots in each slice in order to successfully run these methods. Specifically, 10k samples for PASTE, 5k samples for PASTE2, 20k samples for Moscot and SLAT and SPACEL are used. PASTE2, as an improved version of PASTE, claims to be able to handle partial overlaps. However, PASTE2 requires the overlap ratio as the input, which is not available in advance in real scenarios. To this end, PASTE2 conducts a brute force search, trying a series of overlapping ratios and choosing the setting that works best according to some metrics. We found this pipeline to be very time-consuming, making PASTE2 to spend hours processing to only align a pair of slices with a few thousands cells/spots. Therefore, we fix the overlap ratio of PASTE2 to 0.99 when there is no partial overlap involved in the benchmark, such as nonrigid alignment benchmark and runtime and scalability benchmark. When performing the partial overlap benchmark, its brute force search strategy is enabled. We denote the rigid and non-rigid modes of Moscot as "Moscot-R" and "Moscot-N" respectively. SLAT requires a rough pre-alignment via expert knowledge or ICP algorithm. However, manual manipulation using expert knowledge disobeys the original purpose of automatically aligning ST and also introduces bias. For a fair comparison, we consider the optimal rigid transformation derived from the ground-truth one-to-one correspondence (if available) as expert knowledge and use it as the pre-alignment, denoted as "SLAT-GT". When no GT is available, we use the Iterative Closest Point or ICP algorithm as the pre-alignment, denoted as "SLAT". Lastly, in the nonrigid benchmark, we additionally include the optimal rigid transformation denoted as "Optimal R" that derived from the GT one-to-one correspondence as a reference.

## QUANTIFICATION AND STATISTICAL ANALYSIS

All statistical analyses were performed using Python scripts, with specific test employed as indicated.

## ADDITIONAL RESOURCES

We developed Spateo-viewer, the "Google Earth" browser for 3D spatial transcriptomics. Spateo-viewer is deployed at http://viewer.spateo.aristoteleo.com/.

# Supplemental figures



**A** — The emergence of multiple forms of three-dimensional spatial transcriptome data

**B** — Sample data mainly used in spateo development

**C** — Basic statistics of 3D ST datasets

**D** Comparison for Mouse E9.5-Stere-seq v. scRNA-seq reference

**E** Comparison for Mouse E11.5-Stere-seq v. scRNA-seq reference

**F** Cell groups | Marker genes
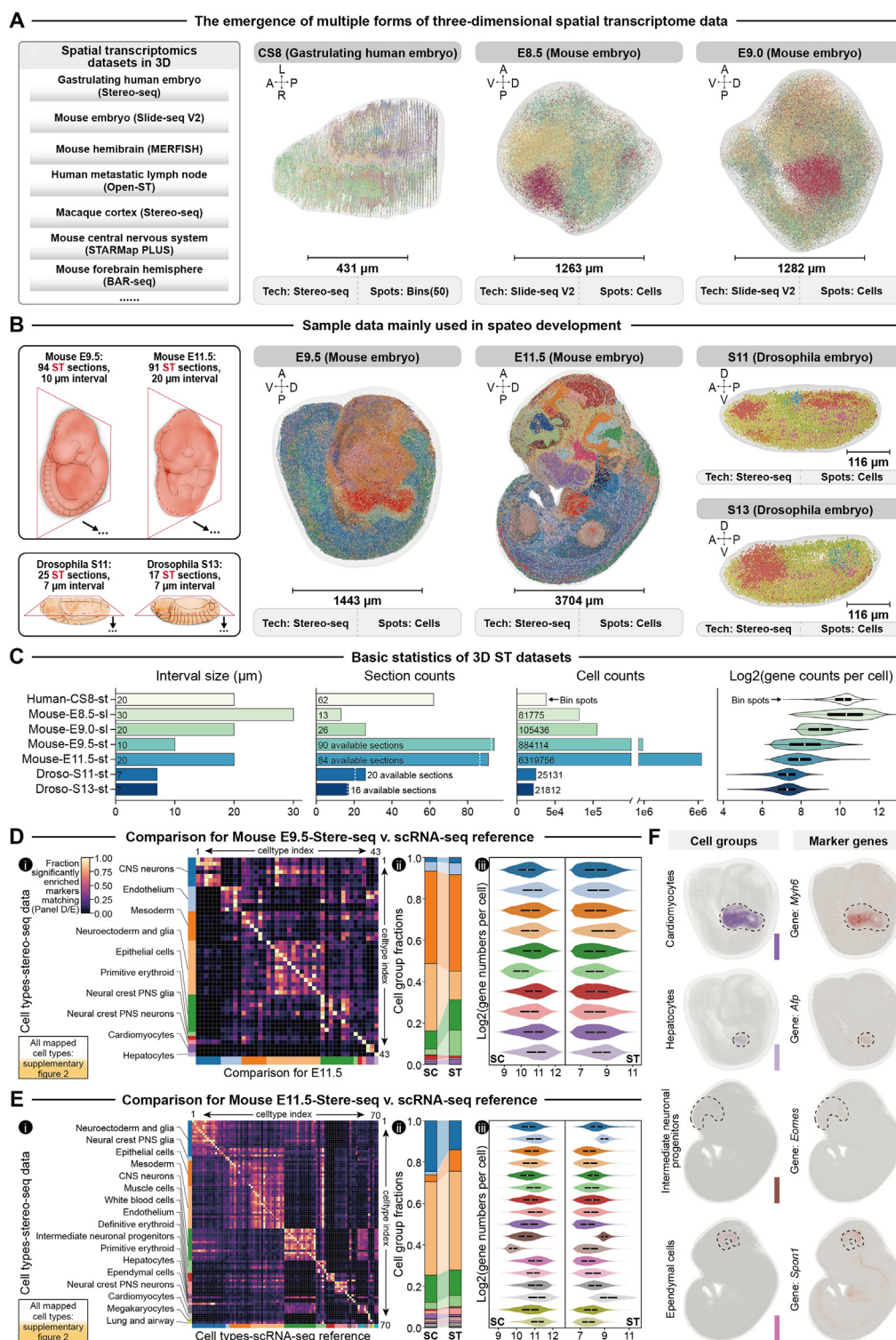
**Figure S1. Basic statistics of the whole mouse and *Drosophila* embryo 3D spatial transcriptomics cell atlases used in this study and their comparison with those from Zhenyu et al. and Kumar et al., related to Figure 1**

(A) Summary and visualizations of existing 3D spatial transcriptomics platforms and datasets. Left column: the existing 3D spatial transcriptomics platforms and datasets used in this study. Three columns on the right: from left to right, 3D visualizations of human embryo of gastrulation profiled by Stereo-seq (CS8) and mouse embryos by Slide-seq V2 (E8.5, E9.0), reconstructed with Spateo (see 3D reconstruction results from the original studies in Figure S3). Color of spots or cells indicates different cell types and uses the same colormap used by Zhenyu et al.[20] or Kumar et al.[21]

(B) Brief description of the whole mouse and *Drosophila* embryo used in this study. Left: schematic describing the embryonic sectioning of both the mouse and *Drosophila* embryo at E9.5 and E11.5, and S11 and S13 stages, respectively. For full details of the mouse embryo dataset, see Cheng et al..[23] Reconstructed 3D time-resolved cell atlas of mouse (middle) and *Drosophila* embryos (right).

(C) Bar plots of the tissue section sampling interval, tissue section number, cell number, and number of genes detected per cell (in $\log_2$ space) for each time point/dataset, highlighting the significantly improved scale from the data used in this study. The Kumar et al. study[21] also includes a partial profiling of E9.5 mouse embryo, which is not shown here.

(D) Comparison for mouse E9.5 whole-embryo Stereo-seq datasets with the scRNA-seq reference. (i) Heatmap of the fraction of matched markers between the scRNA-seq time-lapse reference from Qiu et al.[41,49] and the whole 3D mouse embryo dataset from E9.5 used in this study. (ii) Comparison of the fraction of 10 major cell types between scRNA-seq reference and the Stereo-seq dataset used in this study. (iii) The violin plot of the distribution of the average gene number from cells of each major cell type between the scRNA-seq reference and the Stereo-seq dataset used in this study.

(E) Same as in (D) but for E11.5 time point.

(F) 3D visualization of the cardiomyocytes, hepatocytes in E9.5 embryo dataset, intermediate neuronal progenitors and ependymal cells in E11.5 dataset (left), and the corresponding marker gene expressions (right).

**A** ——— All identified cell types in mouse embryo at E9.5 ———

**CNS neurons**
- Spinal cord motor neurons
- Glutamatergic neurons
- Spinal cord dorsal progenitors
- Neural progenitor cells
- Neuroectoderm-derived
- Cranial motor neurons
- Spinal cord ventral progenitors

**Neural crest PNS glia**
- Neural crest (PNS glia)

**Neural crest PNS neurons**
- Neural crest (PNS neurons)

**Epithelial cells**
- Epithelial precursors
- Placode and neural crest-derived
- Ectoderm-derived
- Intermediate mesoderm
- Amniotic ectoderm
- Anterior intermediate mesoderm
- Olfactory epithelial cells
- Otic epithelial cells
- Gut

**Primitive erythroid**
- Primitive erythroid cells

**Mesoderm**
- Lateral plate mesoderm
- Sclerotome
- Cardiopharyngeal mesoderm
- Cranial mesoderm
- Dermomyotome
- Head and facial mesenchyme
- Renal mesenchyme
- Mesodermal progenitors
- Limb progenitors and lateral plate mesoderm
- Proepicardium
- Gut mesenchyme
- Hepatic mesenchyme

**Neuroectoderm and glia**
- NMPs and spinal cord progenitors
- Spinal cord neuroectoderm
- Hindbrain neuroectoderm
- Anterior roof plate
- Midbrain-hindbrain boundary
- Posterior roof plate
- Midbrain neuroectoderm
- Anterior floor plate
- Telencephalon neuroectoderm
- Diencephalon neuroectoderm
- Hypothalamus (Sim1+) neuroectoderm
- Hypothalamus neuroectoderm
- Dorsal telencephalon neuroectoderm

**Cardiomyocytes**
- Second heart field-derived cardiomyocytes
- LV/atrioventricular canal/common atrium cardiomyocytes

**Endothelium**
- Endothelium
- Vessel endothelial cells
- Arterial endothelial cells
- Hematoendothelial progenitors
- Endocardial cells
- Liver sinusoidal endothelial cell precursors

**Hepatocytes**
- Hepatocytes

**B** ——— All identified cell types in mouse embryo at E11.5 ———

**CNS neurons**
- Neural progenitors
- Spinal cord dorsal progenitors
- Glutamatergic neurons
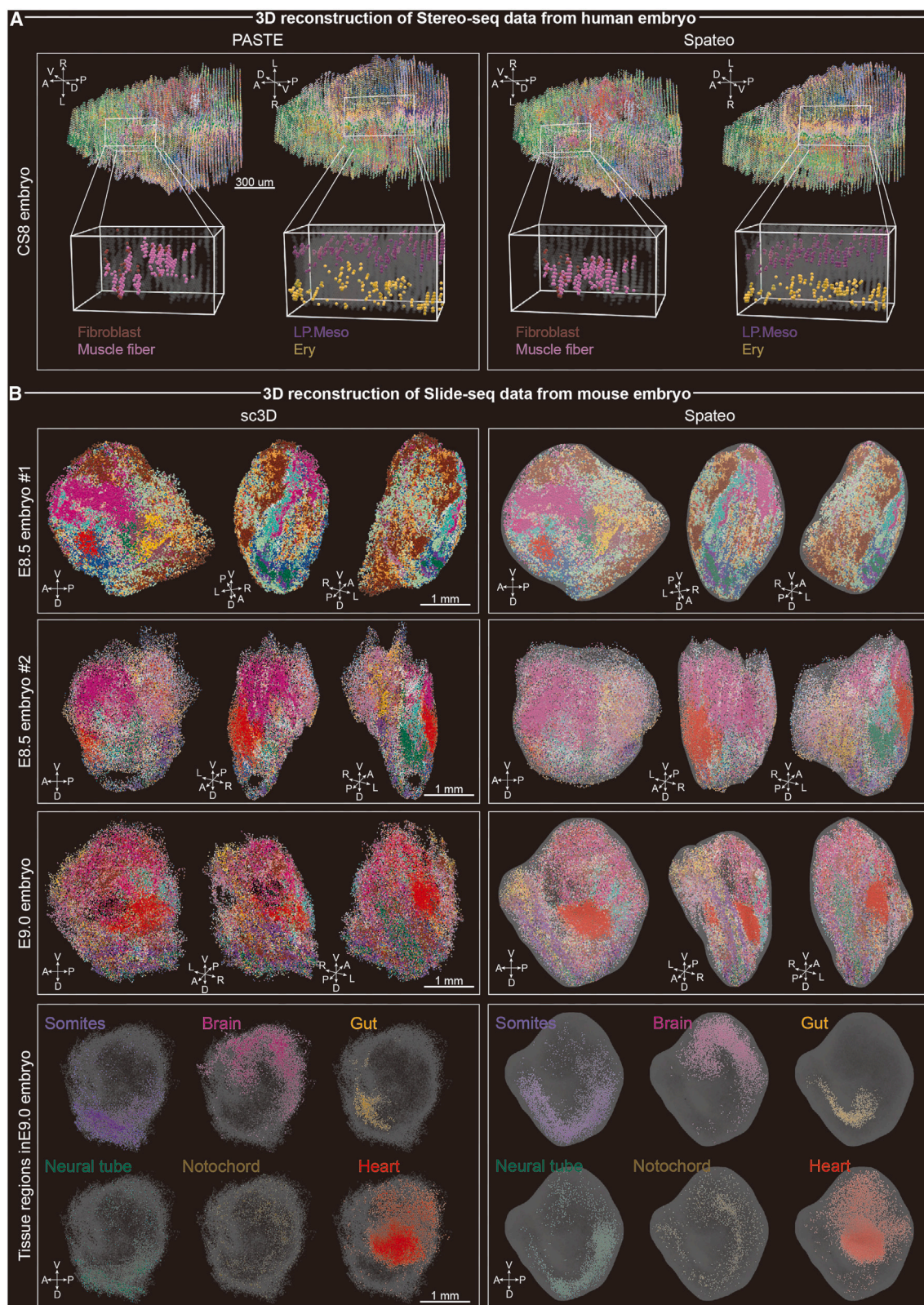- Spinal cord motor neurons
- Cajal-Retzius cells
- GABAergic interneurons
- GABAergic neurons
- Spinal V0/1 interneurons
- Cranial motor neurons
- Spinal cord ventral progenitors
- Spinal V2/3 interneurons
- Zona limitans intrathalamica

**Ependymal cells**
- Choroid plexus epithelial precursors

**Intermediate neuronal progenitors**
- Intermediate neuronal progenitors

**Definitive erythroid**
- Definitive early erythroblasts

**Epithelial cells**
- Epithelial precursors
- Metanephric mesenchyme
- Renal mesenchyme
- Ureteric bud
- Branchial arch epithelium
- Ectoderm-derived
- Placode and neural crest-derived
- Olfactory epithelial cells
- Gut
- Otic epithelial cells
- Pituitary and Pineal gland progenitors
- Apical ectodermal ridge

**Primitive erythroid**
- Primitive erythroid cells

**Neuroectoderm and glia**
- Posterior roof plate
- Midbrain neuroectoderm
- Diencephalon neuroectoderm
- Floorplate and p3 domain
- Anterior roof plate
- Anterior floor plate
- Hindbrain neuroectoderm
- Hypothalamus (Sim1+) neuroectoderm
- Dorsal telencephalon neuroectoderm
- Eye field
- Telencephalon neuroectoderm
- Neuroectoderm-derived
- NMPs and spinal cord progenitors
- Astrocytes
- Spinal cord neuroectoderm
- Hypothalamus neuroectoderm
- Midbrain-hindbrain boundary

**Cardiomyocytes**
- Atrial cardiomyocytes
- Ventricular cardiomyocytes
- Outflow tract cardiomyocytes

**Endothelium**
- CNS capillary endothelial cells
- Endothelium
- Arterial endothelial cells
- Lymphatic vessel endothelial cells
- Liver sinusoidal endothelial cells
- Endocardial cells

**Hepatocytes**
- Hepatocytes

**White blood cells**
- Border-associated macrophages
- Hematopoietic progenitors
- Kupffer cells
- Microglia

**Mesoderm**
- Dermatome
- Cardiac fibroblasts
- Mesodermal progenitors
- Lung mesenchyme
- Gut mesenchyme
- Chondrocytes
- Facial mesenchyme
- Cranial mesoderm
- Mesoderm-derived
- Hepatic mesenchyme
- Limb progenitors and lateral plate mesenchyme
- Early chondrocytes
- Lateral plate and intermediate mesoderm
- Fibroblasts
- Dermomyotome
- Sclerotome
- Proepicardium
- Cardiac mesoderm

**Lung and airway**
- Lung progenitor cells

**Megakaryocytes**
- Megakaryocytes

**Muscle cells**
- Myoblasts
- Muscle progenitor cells
- Somitic muscle progenitors
- Myofibroblasts
- Skeletal muscle progenitors
- Myotubes

**Neural crest PNS glia**
- Myelinating Schwann cells
- Neural crest (PNS glia)

**Neural crest PNS neurons**
- Enteric neurons
- Neural crest (PNS neurons)
- Otic sensory neurons
- Sympathetic neurons

*(legend on next page)*

**Figure S2. Spatial distributions and relative fractions of different cell types in the 3D whole mouse embryo datasets, related to Figure 1**

(A) 3D spatial distribution of all identified cell types at E9.5, grouped into five major cell clusters; relevant groups are organized into a subpanel. Fraction of each cell type within each group is shown as a bar plot.

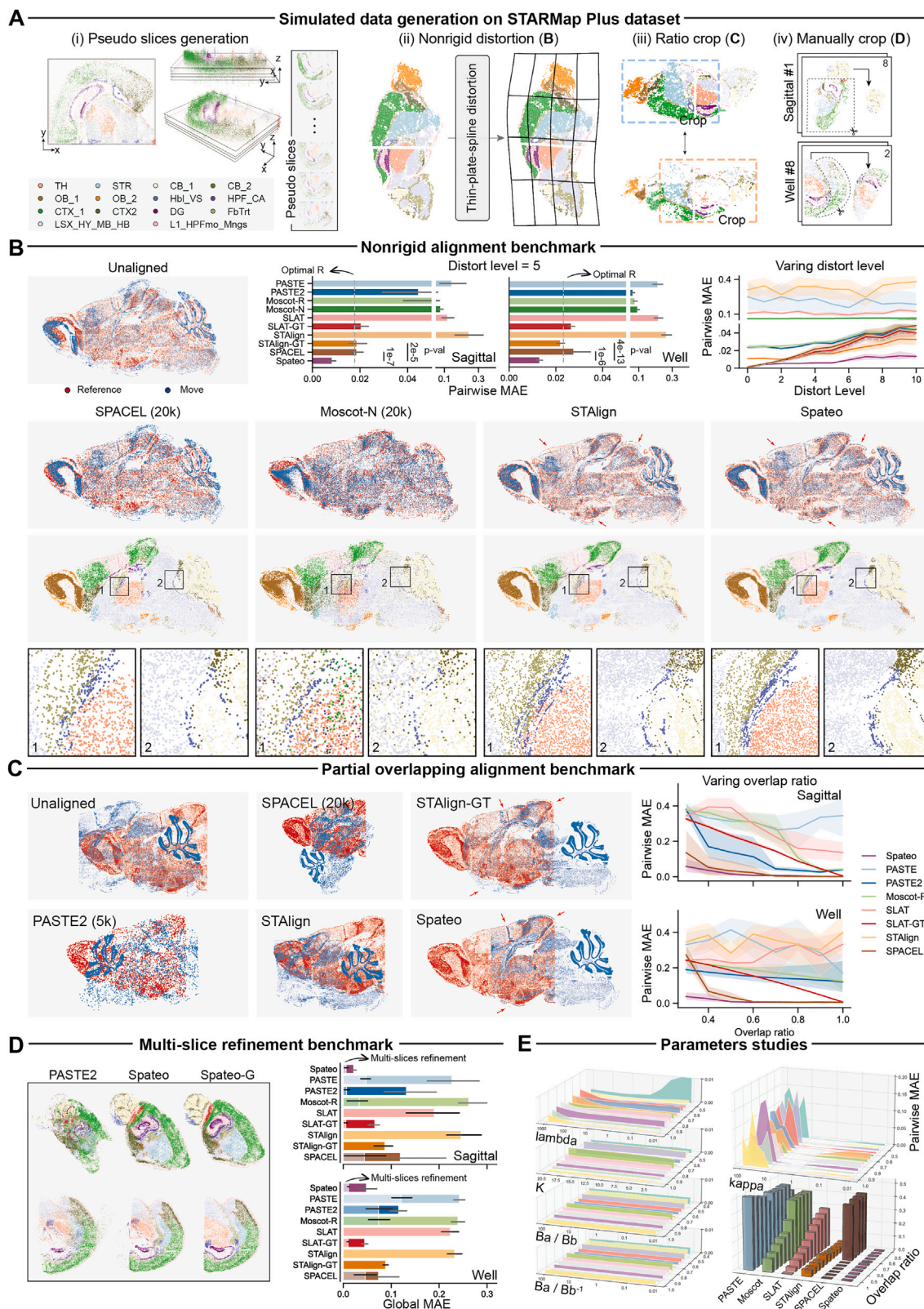(B) Same as in (A) but for E11.5 stage and grouped into six major cell clusters.

**A** 3D reconstruction of Stereo-seq data from human embryo

PASTE

Spateo

CS8 embryo

300 um

Fibroblast
Muscle fiber

LP.Meso
Ery

Fibroblast
Muscle fiber

LP.Meso
Ery

**B** 3D reconstruction of Slide-seq data from mouse embryo

sc3D

Spateo

E8.5 embryo #1

1 mm

E8.5 embryo #2

1 mm

E9.0 embryo

1 mm

Tissue regions in E9.0 embryo

Somites    Brain    Gut

Neural tube    Notochord    Heart

1 mm

Somites    Brain    Gut

Neural tube    Notochord    Heart

*(legend on next page)*

**Figure S3. Comparison of the 3D reconstruction of mouse and human embryos from the original studies and Spateo, related to Figure 2**

(A) 3D reconstruction of human gastrulating embryo Stereo-seq data with Spateo and PASTE used in the original study.[20] Two selected zoom-in regions are used to show more consistent spatial cell-type distribution results for Spateo's approach. The left column shows the results from the PASTE approach used in the original study and the right column from the Spateo approach.

(B) 3D reconstruction of Slide-seq data of mouse embryos with Spateo and sc3D.[21] The 3D Slide-seq spatial transcriptome data of mouse embryos are used to reconstruct the E8.5 embryo #1 (the first row), E8.5 embryo #2 (the second row), and E9.0 embryo (the third row). Several tissue structures are also presented in the fourth row. The left column shows the results from the sc3D approach,[21] reported in the original study, and the right column from the Spateo approach.

**A** — Simulated data generation on STARMap Plus dataset —

(i) Pseudo slices generation    (ii) Nonrigid distortion (**B**)    (iii) Ratio crop (**C**)    (iv) Manually crop (**D**)

**B** — Nonrigid alignment benchmark —

**C** — Partial overlapping alignment benchmark —

**D** — Multi-slice refinement benchmark —    **E** — Parameters studies —

**Figure S4. Extensive benchmarks on simulated and real data demonstrate Spateo's superiority and robustness in 3D alignment of spatial transcriptomics, related to Figure 3**

(A) Simulated data generation process based on STARMap Plus[116] dataset. (i) Each STARMap Plus slice in this dataset has a 3D volume, thus allowing us to obtain several pseudo-slices with ground-truth aligned coordinates by cutting along the z axis. The left and middle subpanels show the data and plane of pseudo-slices under different views. The right subpanel visualizes the artificially created pseudo-slices. (ii) The simulated non-rigid distortion is performed by the thin-plate-spline algorithm. (iii) Ratio cropping strategy to generate partially overlapped slices. For each alignment pair, we crop the slices horizontally or vertically such that the overlapping ratio between slice pairs of alignment will be fully controlled. In this example, the overlap ratio is set to be 50%. (iv) Manual cropping strategy. We manually crop the slices to generate more difficult scenarios for multi-slice refinement benchmark (STAR Methods).

(B) Non-rigid alignment benchmark results. The visual examples of non-rigid alignment on sagittal #3 pseudo-slices are shown in the left bottom. For each group of the result, we visualize the overlay of two aligned slices processed by different methods. In the second row, the red points indicate reference slice and blue points indicate move slice, where we can evaluate the alignment quality of the slice boundary. In the third row, we plot the spatial distribution of cell-type clusters, where we can assess the alignment quality within the slices. The fourth row presents two zoomed-in regions for all methods from the third row. Statistical benchmark in the upper right with pairwise mean absolute error or pairwise MAE, quantifying the absolute alignment error, when the distort level is fixed to 5, where 95% confidence interval is included for each line (STAR Methods). The bar plots in left and middle report the results on the sagittal dataset and well dataset, respectively, where 95% confidence interval lines are included for each bar. Note that the gray dotted line indicates the "Optimal R" (STAR Methods) calculated based on ground-truth correspondences, using the rigid transformation.

(C) Partial alignment benchmark results. The visual example of partial alignment on sagittal #3 pseudo-slices are presented on the left. Statistical benchmark with pairwise mean absolute error or pairwise MAE is shown on the right, on sagittal and well datasets as we change the overlapping ratio, where 95% confidence interval is included for each line.

(D) Multi-slice refinement benchmark results. Left, the visual examples of multi-slice alignment results. Right, statistical benchmark on sagittal and well datasets. Note that the bar plot on the right includes both the result with multi-slice refinement and that without. Error bars (95% confidence interval) are included for both results, while the error bar for the one with multi-slice refinement is moved upward to avoid overlapping with the one without.

(E) Robustness evaluation of Spateo under different sample overlapping ratio and parameter choices. Ba/Bb means Ba and Bb are set to the same value. Ba/Bb$^{-1}$ means Ba and Bb$^{-1}$ are set to the same value. Other methods' performances are also reported in the right bottom. Compared with alternative tools, Spateo outperforms other methods while it is also considerably more robust to different parameters setting and different overlap ratios.
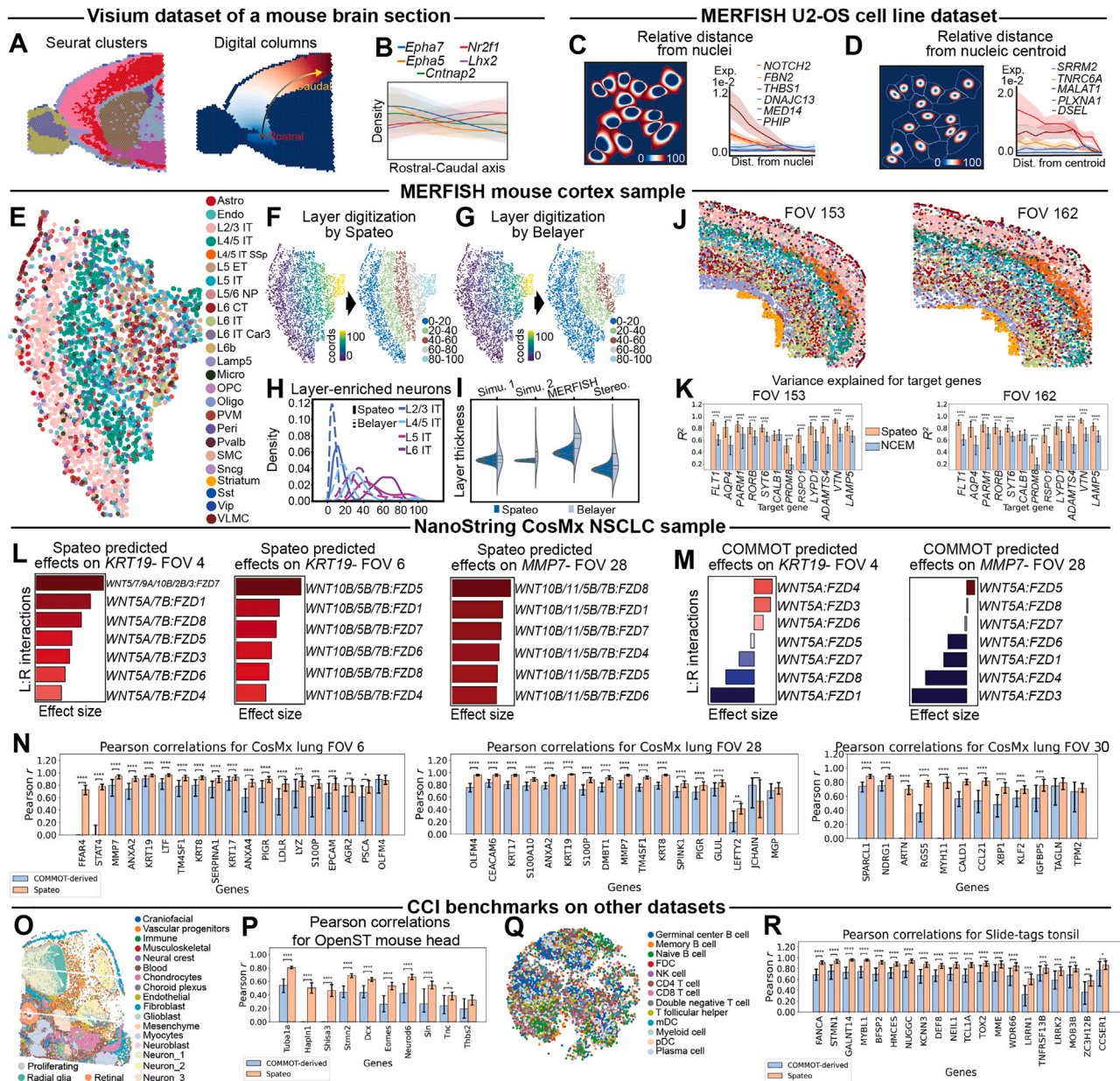
**Figure S5. Assessment and validation of Spateo digitization and cell-cell interaction modeling across various biological systems, related to Figure 4**

(A) Identified spatial domains of a Visium dataset of a mouse brain section[13] (left), with rostral-caudal digitization of the extracted cortex area shown in the right.

(B) The density plot of five genes along the R-C distribution, detected to have either high-to-low or low-to-high expression gradient along the rostral-caudal axis. The transparent shade indicates the 95% confidence interval, as is the case for all other relevant figure panels throughout this paper.

(C) Digitization of the cytoplasm area from a MERFISH U2-OS cell line dataset[54] and the detected top 3 genes, *NOTCH2*, *FBN2*, and *THBS1* (with red, orange, and burgundy lines), which exhibit gradient expression (enrichment) from the nuclear boundary to cell membrane. Three genes with blue lines are randomly selected as negative controls.

(D) Same as (C), but for digitization of the nuclear area. Red, orange, and burgundy lines are those genes—*SRRM2*, *TNRC6A*, and *MALAT1*—that exhibit gradient expression from the center of the nucleus to the boundary.

(E) Spatial scatterplot of cells from the MERFISH dataset of a mouse cortex sample,[55] colored by cell type.

(F) Layer digitization by Spateo. The continuous layer assignment shown on the left with the corresponding evenly grouped assignment shown on the right.

(G) Same as (I) but using Belayer.

(H) Density plot of different laminar-enriched cell types along the digitized axis. Line color indicates neuron subtypes (L2/3 IT: layer 2/S3 intratelencephalic neurons, etc.), while line type indicates digitization methods (solid: Spateo digitization; dashed: Belayer digitization).

(I) Violin plot of layer thickness (distances between perpendicular point pairs between layer boundaries) of digitized layers by Spateo and Belayer in four different cases, including simulated half circle (Simu. 1, refers to Figure 4E), simulated trapezoid (Simu. 2, refers to Figure 4F), the MERFISH mouse cortex sample (MERFISH, refers to Figure S5E), the Stereo-seq macaque cortex sample (Stereo, refers to Figure4G). Dash lines indicate quantiles of the distribution.

(J) Additional FOVs from the MERFISH mouse cortex sample,[55] colored by cell type with optimized palette.[162]

(K) Bar plot comparing Spateo and NCEM model performance on select genes (denoted along the x axis) for FOVs 153 (left) and 162 (right), assessed by variance explained ($R^2$). 95% confidence interval line is included for each bar, and significance is indicated above each pair of bars. *adjusted $p < 0.05$, **adjusted $p < 0.005$, ***adjusted $p < 5e^{-4}$, ****adjusted $p < 5e^{-5}$.

(L) Bar plots of the magnitude of Spateo predicted *Wnt* effects on target genes in the CosMx NSCLC dataset[44] (see Figures 4H–4J; from left to right, *KRT19* in FOV 4, *KRT19* in FOV 6, and *MMP7* in FOV 28). Y axis: L:R interactions, x axis: magnitude of predicted effects. See STAR Methods section COMMOT benchmark for details.

(M) Similar to (L), but for the coefficients from the "COMMOT" model referenced in Figure 4I.

(N) Bar plots of Spearman correlations between observed and model-predicted gene expression, comparing Spateo's spatially weighted model with a non-weighted "global" Poisson generalized linear model that uses the COMMOT L:R array to predict expression in different FOVs of the CosMx NSCLC dataset (from left to right, FOV 6, 28, and 30). See STAR Methods section COMMOT benchmark for details. 95% confidence interval lines are included for each bar, and significance is indicated above each pair of bars. *adjusted $p < 0.05$, **adjusted $p < 0.005$, ***adjusted $p < 5e^{-4}$, ****adjusted $p < 5e^{-5}$.

(O) Spatial scatterplot of cells from the OpenST[38] mouse head sample, colored by cell type.

(P) Pearson correlation comparison bar plot, similar to those described in (N), for select target genes in the mouse head. 95% confidence interval is included for each bar, and significance is indicated above each pair of bars. *adjusted $p < 0.05$, **adjusted $p < 0.005$, ***adjusted $p < 5e^{-4}$, ****adjusted $p < 5e^{-5}$.

(Q) Spatial scatterplot of cells from the Slide-Tags human tonsil dataset,[165] colored by cell type.

(R) Pearson correlation comparison bar plot, similar to those described in (M), for select target genes in the human tonsil. 95% confidence interval is included for each bar, and significance is indicated above each pair of bars. *adjusted $p < 0.05$, **adjusted $p < 0.005$, ***adjusted $p < 5e^{-4}$, ****adjusted $p < 5e^{-5}$.
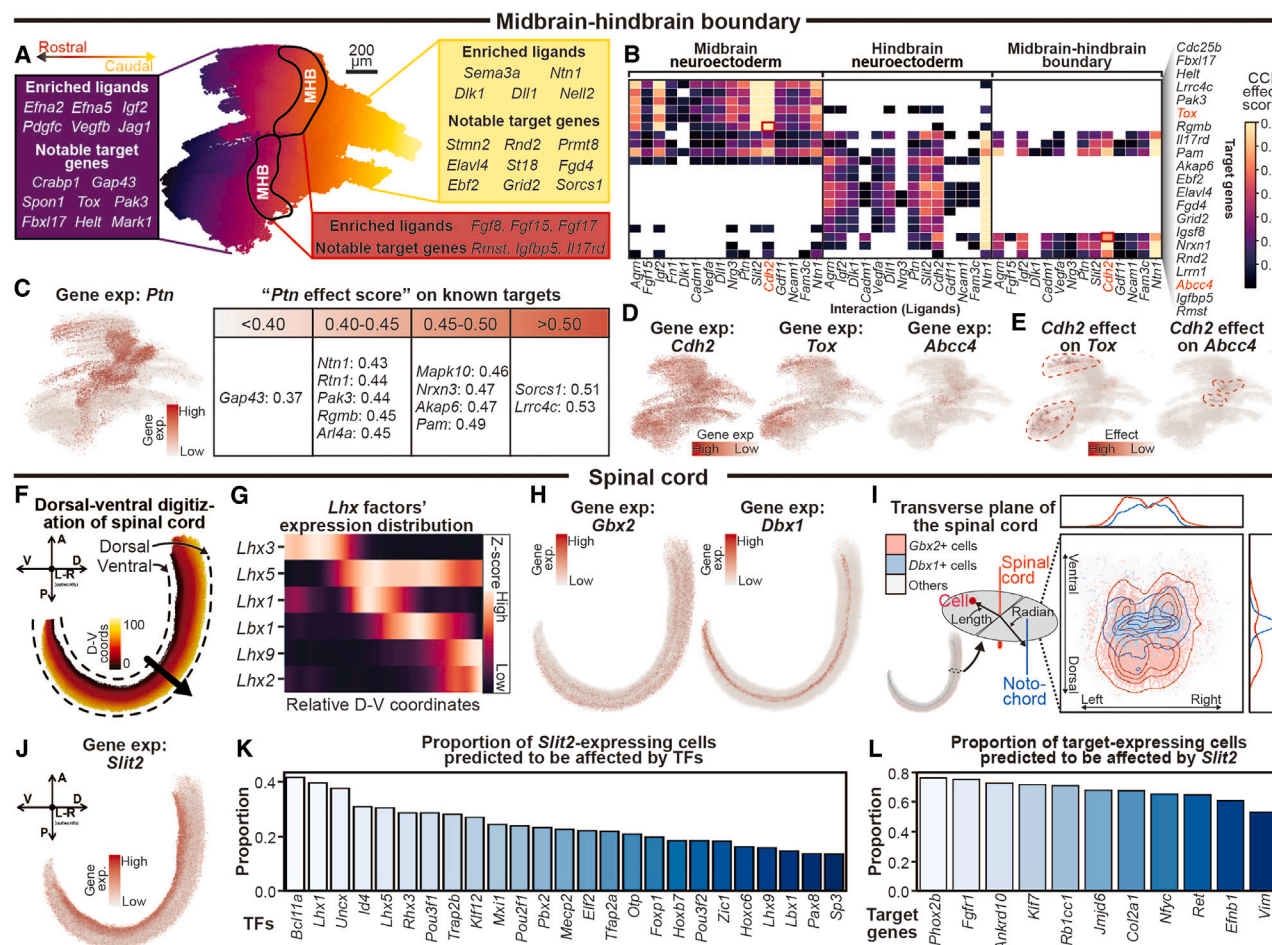
**Figure S6. Spateo characterizes spatially variable gene expression and cell-cell interactions in the midbrain-hindbrain boundary and spinal cord, related to Figure 5**

(A) Diagram of the MHB region, identifying enriched ligands and target genes affected by many of these ligands in the midbrain (region I), hindbrain (region II), and the boundary region (region III), based on the region specificity of the interaction (see B).

(B) Ligand-target gene combinations-effect scores in the midbrain, hindbrain, and MHB. Score derived from the proportion of target-expressing cells predicted to be affected by a particular ligand.

(C) Effect score for selected known *Ptn* target genes, computed similarly as in (B).

(D) Spatial scatterplots of the MHB subset, colored by (left to right) expression of *Cdh2*, *Tox*, and *Abcc4*.

(E) Same as in the (D) but for the predicted effect of *Cdh2* on *Tox* (left) and that of *Cdh2* on *Abcc4* (right).

(F) The digitized spinal cord, colored by the dorsal-ventral (D-V) digitization values.

(G) Heatmap of *Lhx* factors' expression distribution along the D-V axis of the integrated 2D representation of the spinal cord (see STAR Methods).

(H) Spatial scatterplots of the spinal cord 2D projection, colored by expression of *Gbx2* (left) and *Dbx1* (right).

(I) *Gbx2*+ and *Dbx1*+ cells on the projected transverse plane of the spinal cord. Left: schematic demonstration of transverse plane integration (see STAR Methods). The red line and the blue line indicate the principal curve of the spinal cord and notochord, respectively. The red dot illustrated a cell in the spinal cord. While the arrows point to the cell vector representing the distance offset from the spinal cord principal curve, the curve indicates the radian between the cell vector and the reference vector from the spinal cord and notochord, defining the relative angle of the cell. Right: spatial scatterplot of the spinal cord 2D projection, colored by expression of *Gbx2*+ cells or *Dbx1*+ cells. The density plot on two sides corresponds to the density of *Gbx2*+ and *Dbx1*+ cells along either the left-right or dorsal-ventral axes.

(J) Spatial scatterplots of the spinal cord 2D projection, colored by expression of *Slit2*.

(K) Bar plot showing the proportion of spinal cord interneurons expressing *Slit2* and predicted to be affected by TFs (x axis).

(L) Similar to (K), but the y axis represents the proportions of cells with expression of specified target genes indicated from the x axis that are predicted to be affected by *Slit2* in spinal cord interneurons.
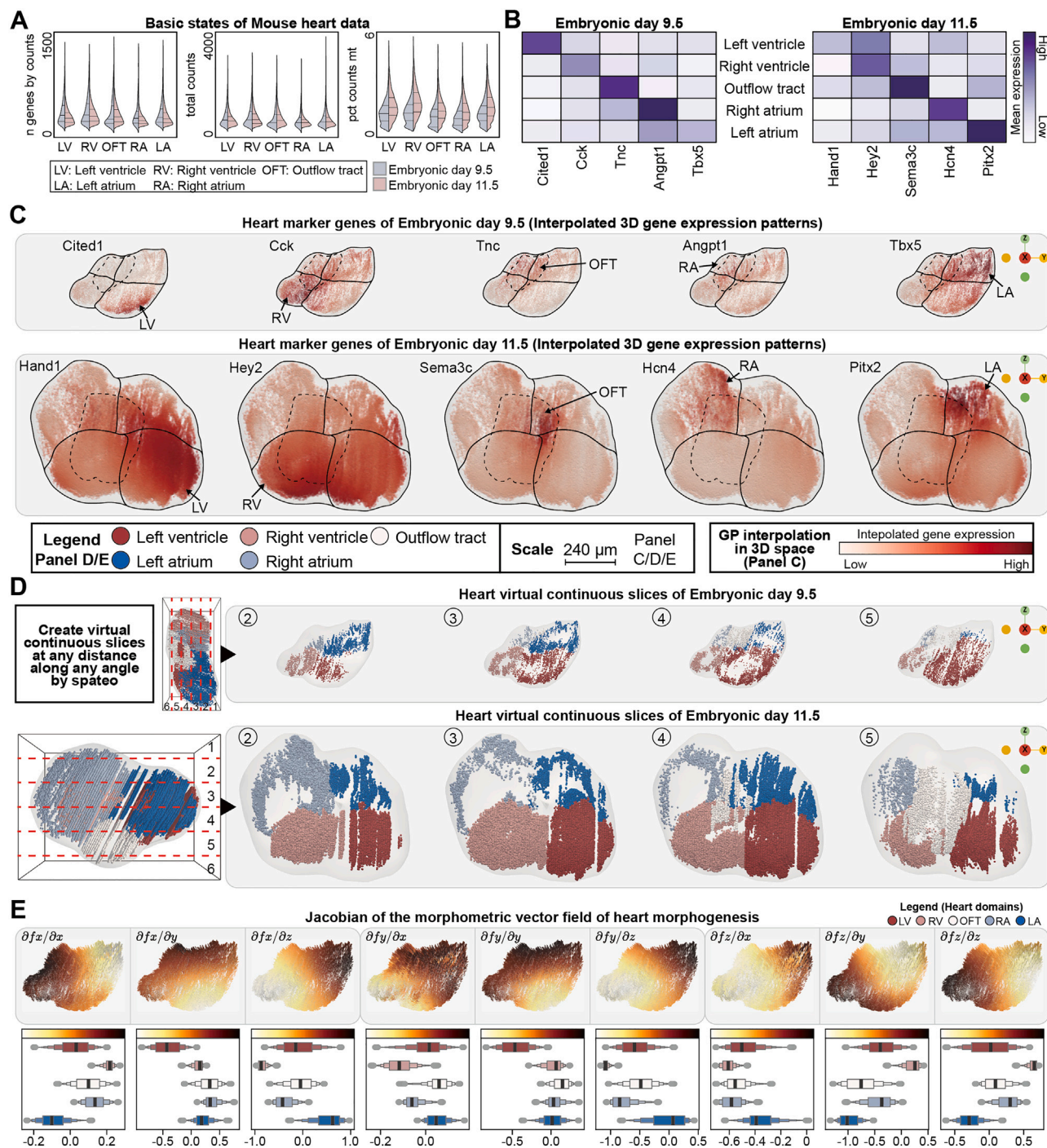
**Figure S7. Spateo characterizes the region-specific gene expression, 3D structure, and morphometric traits during heart organogenesis, related to Figure 6**

(A) Violin plot of the gene numbers, UMI counts, and percentages of mitochondria counts in five major structures of murine heart at E9.5 and E11.5, subsetted from the 3D whole-embryo Stereo-seq dataset.

(B) Heatmap of heart structure-specific markers.

(C) 3D scatterplots of interpolated heart structure-specific markers. Outlines of the heart structures are shown.

(D) 3D visualization of virtual continuous slices of six evenly divided heart sections along the x axis. Only cells from each evenly divided section are shown. Sections 1 and 6 are not visualized given their low cell numbers. Top: E9.5 heart; bottom: E11.5 heart.

(E) Jacobian analysis of the morphometric vector field of the heart. Top: cell migration streamline plots with streamlines colored with the values of six possible Jacobian elements; bottom: the boxplots of the corresponding morphometric Jacobian quantities across five major structures.
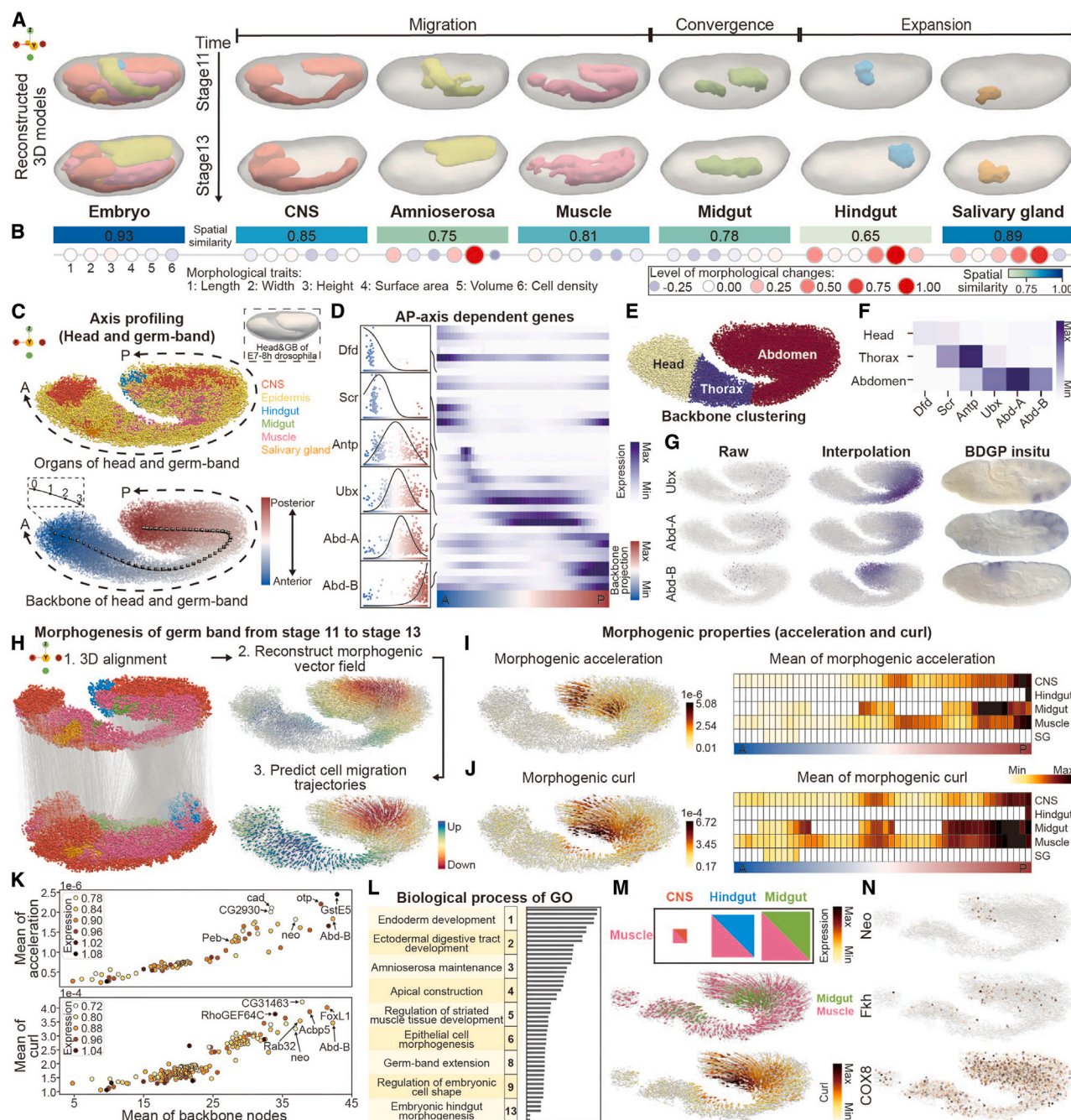
**Figure S8. Spateo enables modeling and prediction of 3D morphometric properties, backbone-dependent expression patterns and morphogenesis regulators, related to Figure 6**

(A) Spateo reveals distinct morphogenesis modes. Left: the reconstructed 3D models of *Drosophila* at the S11 or S13 stage. Right: Spateo facilitates the revelation of migration (CNS, amnioserosa, muscle), convergence (midgut), and expansion (hindgut, salivary gland) modes of *Drosophila* organs.

(B) 3D spatial morphometric similarities and changes of the embryo or across different organs. The top boxplot reveals the spatial similarity between the corresponding 3D structures across two time points. The bottom dot plots characterize the level of changes of different morphometric traits (length, width, height, surface, and volume) of the embryo or different organs at stage S11, compared with stage S13.

(C) Spateo quantifies the backbone of the germband. Top: the spatial distribution of cells from different organs, indicated by different color, along the anterior-posterior or A-P axis. Bottom: the principal curve (backbone) identified for the germband.

(D) Heatmap of significant A-P-axis-dependent genes. Example gene expression and the corresponding fitting curves along the A-P are shown on the left.

(E) Visualization of the head, thorax, and abdomen domains of germband identified by spatially constrained clustering.

(F) Average gene expression heatmap of the head, thorax, and abdomen-specific marker genes.

*(legend continued on next page)*

(G) Spateo reveals consistent gene expression patterns of Ubx, Abd-4, and Abd-B, compared with BDGP *in situ* database after interpolating the sparse raw gene expression.

(H) The workflow to generate the predicted cell migration trajectories, using the germband as an example.

(I) The quiver plot and heatmap of the morphometric acceleration for the germband. Left: the quiver plots of the cells from the germ band, colored by the acceleration calculated from the inferred morphometric vector field. The acceleration is a vector, and we use its length to color cells. Right: the heatmap of the average morphometric acceleration value of the germband along its anterior (A) to posterior (P) axis.

(J) Same as in (H) but for curl.

(K) The gene-wise scatterplot of the mean morphometric acceleration as a function of the mean index of the corresponding principal points of the cells with positive gene expression for each gene along the A-P axis of the embryo.

(L) Top enriched Gene Ontology pathways of the genes with significant acceleration value along the A-P axis.

(M) From top to bottom: the spatial organ interaction plot; the quiver plot of the midgut and muscle cells, colored by cell types; the quiver plot of midgut and muscle cells, colored by curl values.

(N) Scatterplots of the expression of example top genes that significantly change as a function of the curl magnitude across cells.
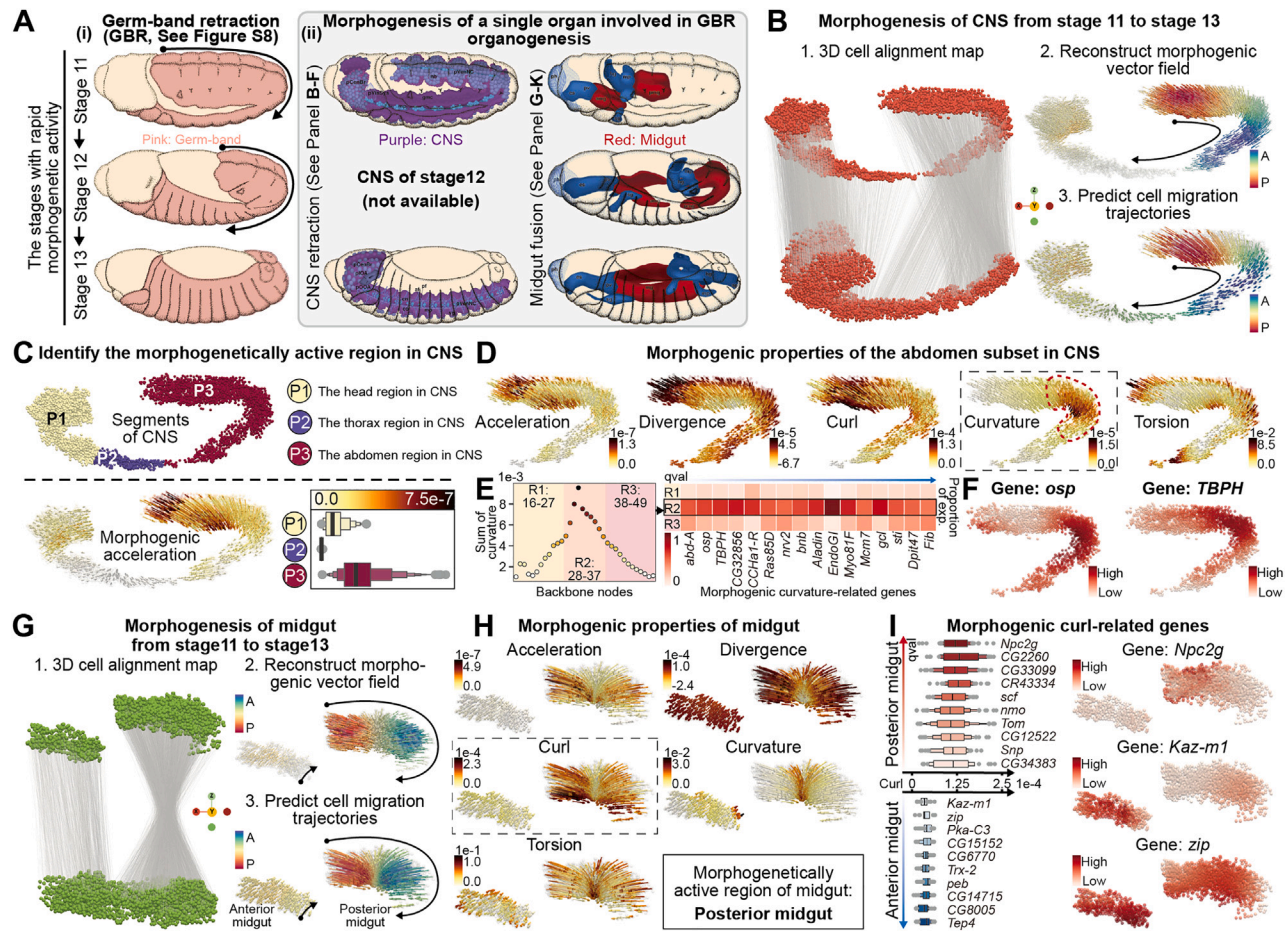
**Figure S9. Spateo infers cell migration paths and identifies morphogenesis regulators of *Drosophila* CNS and midgut, related to Figure 6**

(A) Schematic of *Drosophila* germband retraction (i) and the constituent central nervous system or CNS and midgut (ii) from embryonic stages 11 to 13.

(B) Spateo maps single cells of CNS from stage 12 embryo to that of stage 13 embryo, reconstructs the morphometric vector field, and predicts cell migration trajectories.

(C) Spateo reveals spatial domains of active cell migration. Top: schematic of the P1-P3 CNS segments; bottom left: the quiver plots of the cells from the CNS, colored by the acceleration calculated from the inferred morphometric vector field; bottom right: boxplot of the distribution of the migration acceleration magnitude of the P1, P2, and P3 regions.

(D) Same as in (C) bottom left but from left to right, for acceleration, divergence, curl, curvature, and torsion of the morphometric vector field from a subset of the CNS with significant acceleration shown in (C) bottom left.

(E) Spateo identifies a CNS region with the highest curvature and potential morphogenic genes with spatially enriched gene expression. Left: the scatterplot of the mean morphometric curvature across cells as a function of the index of the principal points from the cells along the A-P axis of the embryo. Backbone is grouped into three regions according to the curvature values, with R2 having the highest overall curvature value. Right: heatmap of potential morphogenic genes show a significant enrichment in the CNS regions 2 or R2. Genes are ordered by q value, incrementally from left to right.

(F) Gene expression scatterplots of known CNS migration factors of osp and TBPH, showing enriched gene expression near the CNS corner.

(G) The same as (B), but for midgut.

(H) The same as (D), but for the midgut. Posterior midgut has the strongest cell migration pattern.

(I) Spateo identifies morphogenic factors that are significantly correlated with curl, for both the posterior and anterior midgut. Left: boxplot of the gene expression distribution of morphogenetic factors for the posterior (top) or anterior (bottom) midgut. Genes are ordered by q value, incrementally from the middle to either side. Right: 3D scatterplot of the gene expression of example morphogenic factors for either the posterior or anterior midgut.
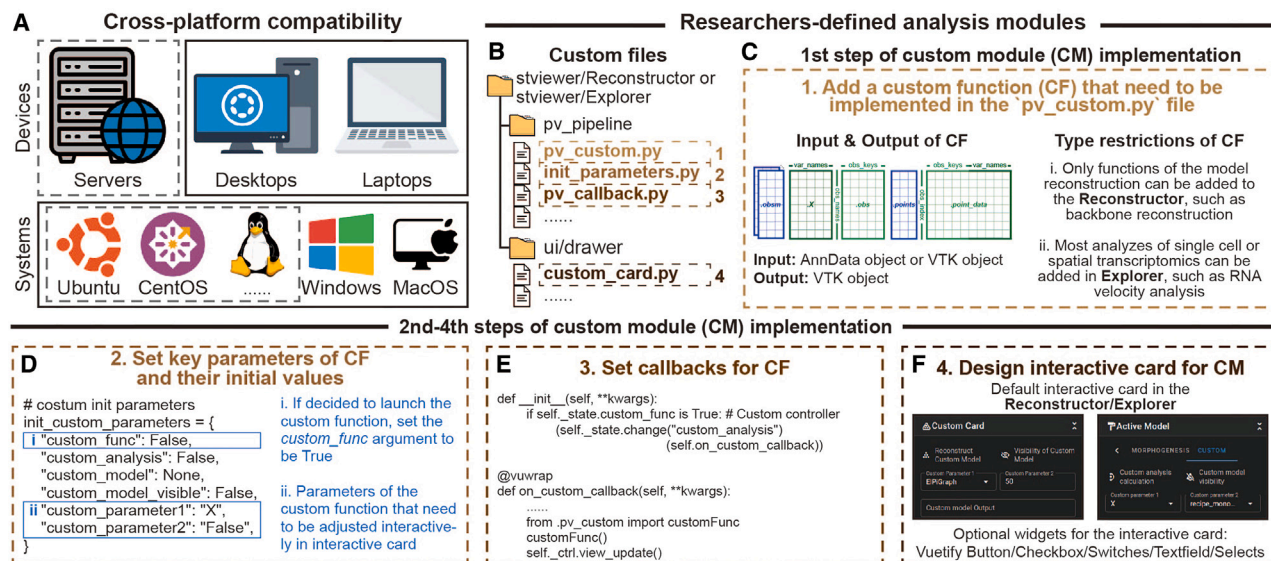
**Figure S10. Spateo-viewer is a flexible and customizable framework, compatible with many different platforms, related to Figure 7**

(A) Spateo-viewer is a cross-platform compatible web application, with support for various operating systems and devices.

(B) File folder structure of Spateo-viewer detailing the location and organization of custom files for implementing new functionalities, applicable to both the Reconstructor and Explorer viewers. Four files are necessary for the custom function implementation, each corresponds to a specific step.

(C) Outline of the first step involved in implementing custom functionalities in Spateo-viewer: adding a custom function to "pv_custom.py."

(D) Same as in (C), but for the second step: setting key parameters with their initial values.

(E) Same as in (C), but for the third step: setting callbacks for the custom function. Detailed code structures are shown.

(F) Same as in (C), but for the last step: designing the interactive card for the custom module.