Deconstructing Complexity: A Computational Topology Approach to Trajectory Inference in the Human Thymus with *tviblindi*

Jan Stuchly[1*#], David Novak[1,5,6*], Nadezda Brdickova[1*], Petra Hadlova[1], Ahmad Iksi[2], Daniela Kuzilkova[1], Michael Svaton[1], Pablo Engel[3], Herve Luche[2], Ana E. Sousa[4], Afonso R. M. Almeida[4], Tomas Kalina[1#]

*Contributed equally

[#]Corresponding author, jan.stuchly@lfmotol.cuni.cz (J.S.); tomas.kalina@lfmotol.cuni.cz (T.K.)

Affiliations:
1 - Childhood Leukaemia Investigation Prague (CLIP), Department of Paediatric Haematology and Oncology, Second Faculty of Medicine, Charles University and University Hospital Motol, Prague, Czech Republic.

2 – Centre d'Immunophénomique - CIPHE (PHENOMIN), Aix Marseille Université (UMS3367), Inserm (US012), CNRS (UAR3367), Marseille, France.

3 - Department of Biomedical Sciences, Medical School, University of Barcelona, Barcelona, Spain.

4 - Instituto de Medicina Molecular João Lobo Antunes, Faculdade de Medicina, Universidade de Lisboa, Portugal.

5 - Data Mining and Modeling for Biomedicine, VIB Center for Inflammation Research, Ghent, Belgium.

6 - Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Ghent, Belgium.

Author Contributions
Conceptualization, T.K., N.B., J.S.; Methodology, T.K., D.N., N.B., P.H., J.S., A.E.S., A.R.M.A; Software, D.N., A.I., J.S.; Formal Analysis, D.N., J.S.; Investigation, N.B., P.H., D.K., M.S., H.L.; Resources, P.E.; Writing – Original Draft, T.K., N.B., P.H., J.S.; Writing – Review & Editing, D.N., D.K., A.E.S., A.R.M.A.; Supervision, T.K., J.S.

Abstract

Interpreting and understanding complex, organ-level mass cytometry datasets represents a formidable interdisciplinary challenge. This study aims to identify, describe, and interpret potential developmental trajectories of thymocytes and mature T cells. We developed *tviblindi*, a trajectory inference algorithm that integrates several autonomous modules - pseudotime inference, random walk simulations, real-time topological classification using persistence homology, and autoencoder-based 2D visualization using the *vaevictis* algorithm. This integration facilitates an interactive exploration of developmental trajectories. The utility and proficiency of *tviblindi* are demonstrated through comprehensive analysis of the thymic and peripheral T-cell compartment. Our approach not only uncovers and elucidates the canonical CD4 and CD8 development but also offers insights into various checkpoints such as TCRβ selection and positive/negative selection, elucidating the crossroads between further development and apoptosis. Finally, we identify and thoroughly characterize thymic regulatory T cells, tracing their development from the negative selection stage to mature thymic regulatory T cells with an extensive proliferation history and an immunophenotype of activated and recirculating cells. *tviblindi* is a versatile and generic approach suitable for any single-cell dataset, equipping biologists with an effective tool for interpreting complex data.

Introduction

Exploring the development of mature, fully functional cells from their progenitors has been the focus of researchers for many decades. While single-cell techniques such as flow cytometry were fundamental to the description of developmental stages, recent advances in single-cell methods measuring large sets of markers (mass cytometry, single-cell RNA-seq) have enabled truly comprehensive descriptions of entire tissues and organs. In parallel, computational methods that capture and describe possible dynamics in the data, as well as topological relationships among cell populations, were developed. After the seminal Wanderlust methodological study[1], trajectory inference (TI) and pseudotime data analysis became an area of intensive research, producing gradually more and more sophisticated tools for a growing number of general topologies[2–7].

However, certain limitations prevent widespread adoption of these tools in analytical workflows of real-world data. First, the computational complexity of existing TI methods is often prohibitive when analyzing large datasets from current single-cell RNA methods (especially when the data comprise cell atlases of entire organisms, embryos, and human organs) or from flow and mass cytometry with datasets of millions of cells[3,8–11]. Second, TI methods often use multiple steps of dimensionality reduction and/or clustering, inadvertently introducing bias. The choice of hyperparameters also fixes the a priori resolution in a way that is difficult to predict. Third, the current TI methods

work well on artificial datasets but lack a straightforward approach to control the effect of noise (technical artifacts or unexpected events) in multiscale topologies of real-world data. Fourth, perhaps the major obstacle is that existing TI tools do not offer the necessary interaction with the analytical process. We believe that such an interaction, which helps the researcher understand the particularities of a sample, is crucial for exploratory analyses of unknown data.

Motivated to develop a generic TI solution useful to a biologist investigating a real-world dataset, we developed *tviblindi. tviblindi* is a modular TI method designed to tackle large datasets, significant noise, technical artifacts and unequal distribution of cells along the time axis. As a proof of concept, we investigate αβ T-cell development in human thymus and the transition of mature T cells to peripheral blood.

Human T cells develop in the thymus, which is seeded by bone marrow-derived $CD45^{pos}CD34^{hi}CD7^{pos}$ progenitors[12,13]. Large $CD34^{hi}CD1a^{neg}$ immature thymocytes develop into small $CD34^{dim}CD1a^{pos}$ immature thymocytes[13], becoming gradually restricted to the T-cell lineage[14]. After losing the dim CD44 expression, right before gaining CD1a, they become committed to a T-cell fate[15].

At this stage, early human $CD34^{pos}$ thymocytes progress through three stages[16]. (1) $CD34^{pos}CD38^{neg}CD1a^{neg}$, (2) $CD34^{pos}CD38^{pos}CD1a^{neg}$ and (3) $CD34^{pos}CD38^{pos}CD1a^{pos}$. The first TCRβ D-J rearrangements are detected in the $CD34^{pos}CD38^{pos}CD1a^{neg}$ population. In-frame TCRβ are mostly selected at the transition from $CD34^{pos}CD38^{pos}CD1a^{pos}$ to the next stage: the $CD4^{pos}$ immature single positive (ISP) population. Concordantly, pre-TCRα (pTα) expression peaks in the $CD34^{pos}CD38^{pos}CD1a^{pos}$ and ISP stages, after which it declines. TCRα rearrangements are initiated when thymocytes progress from $CD34^{pos}CD38^{pos}CD1a^{pos}$ toward the ISP stage, and continue until the $CD3^{pos}$ double positive (DP) stage[16].

The complete TCRαβ receptors are checked for their binding to MHC-self peptide complexes on thymic epithelial cells and dendritic cells. Failure to engage the MHC-self peptide leads to death by neglect[17]. Efficient TCR binding leads to positive selection and further development along the CD4/CD8 axis. Positively selected DP cells express CD69 on their surface and start to downregulate the CD8 co-receptor. If they are selected via the MHC-II peptide complex, the TCR signal persists. When the duration of this signal exceeds a certain time limit, the transcription factor ThPOK is induced and the cells become CD4 cells. If selected on the MHC-I peptide complex, the signal diminishes, the transcription factor RUNX3 is induced, expression of the CD8 co-receptor is reactivated and CD4 eliminated, leading to the CD8 cell fate[18]. On the other hand, strong TCR binding leads to negative selection and apoptosis or to the so-called agonist selection and development of regulatory T cells (Tregs)[19]. During these processes the cells acquire the chemokine receptor CCR7 and move from the thymic cortex to the medulla, where tissue-

specific self-antigens are ectopically expressed on the surface of medullary epithelial cells[20,21].

Currently proposed models postulate that multiple signaling events are likely necessary[19] including TCR and co-stimulatory[22] signals as well as cytokines[23,24] for the regulatory T-cell fate. The upregulation of the high-affinity IL-2 receptor CD25 and its signaling are known to stabilize FOXP3 expression[25,26]. Importantly, JAK/STAT signaling triggered by γ chain cytokines IL-7 and IL-15 enhance FOXP3^{pos} Treg development by proliferation induction[27] and by induction of BCL-2 and its anti-apoptotic effect[28]. While the TCR repertoire of conventional T cells and Tregs are largely distinct and non-overlapping[29], the exact branching point of these two main lineages is still a matter of intensive research[30,31] also employing human data based on single-cell transcriptomics[32].
When investigated using single-cell methods, the process of T-cell development translates into high-dimensional single-cell data with complex topology which need to be interrogated computationally.

We developed *tviblindi,* a new tool which breaks new ground in several aspects:
(1) It offers a highly scalable, linear complexity framework that works at a single-cell level.
(2) Analysis is performed in the original high-dimensional space, avoiding artifacts of dimensionality reduction.
(3) The framework is adapted to discover features taking into consideration their varying scales (along the time axis or in different cell types).
(4) Crucially, our method allows the user to interact with the analytical process and to set the appropriate level of resolution in real time.

In this paper, we first describe our new computational approach to exploration of developmental trajectories. Then we present an analysis of real-world datasets of T-cell development to illustrate the power of our analytical method and to describe the development of human T regulatory cells in detail. Last, we describe novel discrete stages in the development of Tregs and cell surface markers used in their study.

Results

Computational method for the TI and interrogation - *tviblindi*

The objective of *tviblindi* is to offer a flexible framework for TI and topological data analysis (TDA) interrogation of single-cell data. *tviblindi* facilitates efficient interpretation of data as well as sensitive discovery of minor trajectories. These two competing goals are resolved by interactive grouping of random walks, which are used as probes in the high-dimensional space, into trajectories, allowing for real-time adjustment of trajectory-specific resolution. This interactive approach provides an exhaustive description of the data, while allowing the researcher to introduce expert knowledge into the analytical process and to gradually gain insight into a particular dataset.

tviblindi was designed as an integrated framework implementing several independent modules to facilitate the use of diverse approaches to pseudotime estimation and data visualization, without the need to rerun the analysis. *tviblindi* allows to compute pseudotime which is resilient to unequal distributions of cells (e.g., accumulation of double positive T cells during T-cell development or in case of the presence of a developmental block). It also keeps the resolution on the single-cell level, while processing data in a reasonable time frame. Random walks, directed by the pseudotime, are then simulated, thus creating a set of probes in the high-dimensional space. To assemble random walks into meaningful trajectories, *tviblindi* employs witness complex triangulation[33] to capture the topology of the original data without the need for dimensionality reduction. It implements a representation of homology classes, which accounts for incomplete coverage of the high-dimensional points by the triangulation, navigating around the potential pitfall of this computational approach.

Contrary to other approaches, *tviblindi* considers each random walk a separate entity, which allows for probing major as well as minor trajectories while keeping track of the feature dispersion along the time axis. This approach enables sensitive detection of irregularities, such as hubs and discontinuities, in the topology of the underlying graph. While pseudotime captures local geometry of the data, persistent homology[34] aims to capture significant non-local features such as sparse regions (holes), which persist over a range of scales. In simplified terms, the persistence of a sparse region may be understood as the difference between the density of points at its boundary and in its interior.

Most persistent features can be extracted and used to capture the differences between random walks, inducing a multiscale clustering[35]. We generalized and extended this idea to high-dimensional data. We took advantage of the natural hierarchical structure of persistent homology classes to achieve real-time, on-demand aggregation of random walks into trajectories respecting both local geometry and global topology of the point cloud.

Introducing tviblindi on an artificial data set

We showcase our methodology using an artificial dataset created with the *dyntoy* package (for further details see Supplementary note, subsection 1.1 *dyntoy* dataset). Figure 1A shows a schematic representation of the data, capturing the basic topology with sparse regions labeled α and β.

First, the data are represented as an undirected k-nearest neighbor graph (k-NNG; Figure 1B). The cell of origin is defined by the user (either directly, or a population of origin is specified and the point closest to the centroid of this population is then taken) and used to estimate pseudotemporal ordering (pseudotime) directed away from the origin (Figure 1B). Edges of the k-NNG are oriented with respect to the pseudotime forming a directed acyclic graph (DAG).

Next, a large number (typically thousands) of random walks is simulated on the DAG. These walks are finite and their final vertices are candidates to become developmental endpoints (Figure 1C). The user can select one or more potential endpoints for further investigation (Figure 1D labeled x, y, z).

A persistence diagram, which captures the prominence of sparse regions within the point cloud, then allows the user to select significant sparse regions (Figure 1E), and random walks are organized into trajectories by means of hierarchical clustering. This clustering respects the global geometry and classifies trajectories based on how they navigate around the sparse regions (Figure 1F). In the presented artificial dataset, *tviblindi* correctly identified three putative endpoints (x, y, z) and organized simulated random walks into three distinct trajectories for each endpoint (a, b and c, for end x; d, e and f for end y and g, h and i for end z).

Our interactive framework allows the user to inspect the evolution of specific markers (Figure 1G), track key points in their development (e.g., branching point p) and focus on cells at such key points (Figure 1H, I). For a quick overview, a “connectome” summarizing the basic structure of the point cloud and of simulated random walks can be plotted (Figure 1J). We describe each *tviblindi* module below. Detailed algorithmic descriptions of particular modules can be found in the Supplementary note, sections 2-4.

Visualization

Visualization of high-dimensional data is essential for the initial overview and for the informed interactions with the data during downstream analysis. However, any dimensionality reduction introduces simplification, which can lead to misinterpretation. In *tviblindi* framework, the dimensionality reduction step is independent of all other modules (which work directly in the original high-dimensional space), and is used solely for a visual representation of the dataset.

While dimensionality reduction has been extensively and successfully used for single-cell data (e.g., UMAP[36,37] and viSNE[38,39]), interpreting developmental trajectories requires not only the local, but also the global relationships of cells to be captured (often distorted in the lower-dimensional embedding by current dimensionality reduction algorithms). For this purpose, we created *vaevictis*, a variational autoencoder model based on ideas in *scvis*[40] and *ivis*[41] fitted with naive importance sampling. This technique creates a continuous representation of the data rather than isolated islands of populations (Supplementary note, section 2 Dimensionality reduction, Figure 1B, C). Due to the naive importance sampling step, numerically large compact populations are not overrepresented. This way, we obtain a clearer picture of the developmental dynamics in the data (Supplementary note, section 2 Dimensionality reduction, Figure 1D, E). Moreover, once calculated, it can be

applied to new datasets with the same measured parameters (protein markers or detected transcripts), facilitating integration of new results.

Pseudotime and random walks simulation

Correct pseudotemporal ordering is an essential and computationally intensive part of the TI algorithms. Here we use a particular formulation (and modification) of the idea of *expected hitting time*[42]. We calculate the expected number of random steps necessary to reach any cell in a dataset from the cell of origin. To leverage the sparsity of the k-NNG and to calculate the hitting times on a single-cell level, we use the following formulation: given an undirected k-NNG $G = (V, E, p)$, with vertices V , edges E , weights $p: E \rightarrow [0,1]$ representing the probability of transition of each edge respectively, and an origin v_0, the expected hitting time of a cell is a weighted average of hitting times of its neighbors increased by 1 $\tau(y) = \sum_{x:\{x,y\}\in E} \quad p_{xy}(\tau(x) + 1), y \neq v_0$ and $\tau(v_o) = 0$. This formula translates into a sparse linear system, which can be solved efficiently by numerical methods (such as conjugate gradients) to any precision even for graphs with millions of vertices. Consequently, we are able to calculate the pseudotime directly in the original space at a single-cell level. Furthermore, we obtain a measure of success: the relative error reported by the numerical solver.

A key application of *tviblindi* is the comparison of trajectories between normally developing tissues and those with developmental abnormalities, manifesting a block and/or pathological proliferation resulting in an overabundant intermediate population. In such cases, the hitting time calculation would assign the highest pseudotime to this intermediate population, which would hamper the topology of simulated random walks (Supplementary note, section 3, Pseudotime estimation & random walks simulation). To mitigate this effect, the formula above can be modified in the following way: suppose that apart from $G = (V, E, p)$, we have a sparse matrix D, recording mutual distances for all vertices in the graph connected by an edge. Then the definition of the pseudotime as the *expected hitting distance is*

$\tau(y) = \sum_{x:\{x,y\}\in E} \quad p_{xy}\big(\tau(x) + D_{xy}\big), y \neq v_0$ and $\tau(v_o) = 0$.

In other words, the expected distance to travel before reaching vertex y. Both methods are also very robust with respect to the choice of the number of neighbors in the k-NNG (Supplementary note, section 8 Performance evaluation).

Once pseudotime is computed, the edges in the graph $G = (V, E, p)$ get oriented (Figure 1B) and a large number of random walks is simulated on the DAG (Figure 1F). Importantly, all random walks on the graph are finite. The set of terminal vertices of these walks represents estimates of terminal biological fates (for further details see Supplementary note, section 3 Pseudotime estimation & random walks stimulation).

Real-time interactive topological classification

The most distinctive feature of *tviblindi* is the interactive aggregation of simulated random walks into trajectories with respect to the global topology of the point cloud. This procedure is based on the idea of multi-scale topological classification using persistent homology[35]. Random walks are considered distinct entities, and they can be aggregated into trajectories respecting the structure of the point cloud.

Walks are clustered based on how they circumnavigate significant sparse regions identified using persistent homology. The appropriate level of significance can be adjusted by the user by the selection of significant sparse regions on a persistence diagram (Figure 1E). Classification by the means of persistent homology has a natural hierarchical structure induced by filtration [34,35] (see Supplementary note, section 4 Topological clustering of random walks for details), which translates into a dendrogram of random walks (Figure 1F).

Applying classification by persistent homology to a large high-dimensional dataset with considerable technical noise requires several improvements of previously published work[35]. First, as noted above, the selection of significant sparse regions is interactive, which allows the user to choose an appropriate level of detail for each dataset or trajectory. Second, we use witness complex triangulation[33] to adapt for high dimensionality of single-cell data (for which the original methodology was intractable). Third, since the witness complex triangulation does not guarantee filling the space completely, we introduce the detection and representation of sparse regions with infinite persistence. This last modification also allows us to classify random walks into trajectories based on their different endpoints and consequently to study developmental branching. (This is impossible with the original formulation.) See Supplementary Videos and Supplementary note for a presentation of the *tviblindi* graphical user interface (GUI).

Connectome – a fully automated pipeline

For a quick overview of a given dataset, we have implemented a “connectome” functionality inspired by PAGA[5] (Figure 1J). This is a fully automated pipeline to create a directed graph, in which the data is clustered (by default using Louvain community detection[43]). Random walks are contracted to the identified clusters, providing a lower-resolution estimation of developmental trajectories. The clusters are plotted as pie charts of the represented cell populations. The graph layout employs the same 2D representation(s) as *tviblindi* GUI to facilitate the interpretation. The clusters containing the cell of origin and endpoints are automatically detected and denoted O and T respectively. At the same time the orientations and widths of edges reflect the direction and number of underlying random walks.

While this fully automated pipeline performs very well under various conditions we tested (Supplementary note, section 5 Connectome, and below), it suffers from some of the limitations of other fully automated, end-to-end methods. The

resolution is fixed at a cell-population level in advance by the choice of clustering. Also, there is no direct way to interactively choose a resolution at the level of random walks, nor to discover and exclude clear artifacts. Therefore, we only encourage using this functionality as a means to get an overview of any given single-cell dataset with suggested trajectories, before the full interactive *tviblindi* analysis.

tviblindi exhaustively dissects human T-cell development

We applied *tviblindi* to the 34-parameter mass cytometry data obtained in our human T-cell development study (Supplementary Table Mass panel 1). The data contained barcoded human thymocytes and human peripheral blood mononuclear cells (PBMC). Dead cells and cells not belonging to the αβ T cell developmental lineage were excluded, yielding a total of 1,182,802 stained cells for analysis (Supplementary Figure 1). When the cells were visualized using a CD4xCD8 dot plot, we observed the expected patterns of double negative (DN), double positive (DP), CD4 single positive (SP) and CD8 SP populations for a cryopreserved thymus (Figure 2A). We then used the default *vaevictis* dimensionality reduction to reveal the basic structure of the T-cell compartment. Thymic and peripheral blood T cells were projected into distinct areas of the *vaevictis* embedding with a minor overlap (Figure 2B). Localization of the CD8, CD4 and Annexin V positive cells can be viewed on the plot using a heatmap color scale for expression levels of these markers. This provides an interpretable image of the T-cell compartment (Figure 2C).

Next, we gated the population of origin as $CD34^{hi}$ $CD1a^{neg}$ DN cells and analyzed the data with *tviblindi*. We simulated 7500 random walks and discovered eleven groups of endpoints labeled #1 - #11 (Figure 3A). Five of them (#1 to #5) were located in the region of peripheral CD4 T cells (where #5 corresponded to mature naive CD4 T cells transiting to peripheral blood). Endpoint #6 found among the thymic CD4 SP T cells is further described in the section *Non-conventional end interpretation.* Three groups of endpoints were located in the peripheral CD8 T cell compartment (where #7 corresponded to cells of CD8 TEMRA phenotype, #8 to CD8 central memory T cells and #9 to mature naive CD8 T cells transiting to the periphery). Two remaining endpoints (#10 and #11) belonged to the apoptotic thymocytes (compare to Figure 2B and 2C).

The connectome shows a basic overview of dynamics in the data on a cluster level (Figure 3B). The subsequent interactive analysis of simulated random walks gives biological meaning to the observed connections and allows for the detailed interpretation of data on a single-cell level.

First, we chose endpoint #1 representing peripheral CD4 effector memory T cells for further analysis. We applied persistent homology to select the significant sparse regions in the data and used these to cluster individual simulated random walks (Figure 3C). The topology of the point cloud representing human T-cell development is more complex than that of the illustrative artificial dataset shown in Figure 1 and does not offer a clear cutoff

for the choice of significant sparse regions. Interactive selection allows the user to vary the resolution and to investigate specific sparse regions in the data iteratively.

Clustering the random walks resulted in a dendrogram, where leaves with abundant walks represent the dominant developmental trajectories. Nonetheless even the less abundant groups of walks can be selected for in-depth investigation (Figure 3D).

Leaf I. contains the largest number (452) of random walks (Figure 3D and 3E). Of note, leaf I. faithfully represents a larger supercluster of random walks II. (Figure 3D and 3E). For the leaf I. trajectory, we displayed the intensity of expression of key markers along the calculated pseudotime (Figure 3F top). We then mapped the key developmental stages onto the *vaevictis* plot (Figure 3F bottom).

The CD4 ISP stage was found where CD4 and CD1a expression begins, but prior to CD8, intracellular TCRβ and CD3 expression. This is followed by a $CD3^{neg}$ DP stage, where intracellular TCRβ increases. Eventually, the DP cells become $CD3^{pos}$, reflecting the successful expression of the TCRαβ receptor. Next, they lose CD8 and later they lose the cortical thymocyte marker CD1a. Finally, the CD4 SP thymocytes gain CD45RA and are ready to egress from the thymus as naive CD4 SP cells. In the peripheral blood they lose the CD45RA during maturation only to gain it at their terminal stage (Figure 3F). Similarly, if endpoint #7 is chosen, the canonical trajectory leading to peripheral CD8 cells can be visualized and described (Supplementary Figure 2).

This knowledge of the topology of key points of canonical development helped us to interpret additional developmental endpoints and the trajectories leading to them.

Variants of trajectories including selection processes

We then focused on alternative trajectories leading to peripheral CD4 cells corresponding to endpoint #1. Leaf IV., comprising 103 walks (Figure 4A) exhibits a clear artifact connecting highly dividing progenitor cells with doublets, which could invalidate a fully automated analysis (Supplementary Figure 3).

For further in-depth investigation we selected the two remaining larger groups of random walks represented in the dendrogram by leaves highlighted as III. and V. (Figure 4A). After the $CD3^{neg}$ DP stage, these trajectories diverted from the canonical course described above. The cells passed through a region with an overall decrease in phosphorylated tyrosine expression and an increase in Annexin V expression, indicating apoptosis (Figure 4B and 4C). Endpoints #10 and #11 were situated in this apoptotic region (see Figure 3A for their precise location). We therefore interpreted the first part of these trajectories as leading to apoptosis upon failure of DP thymocytes to pass either positive or negative selection checkpoints in the thymic cortex (Figure 4B and 4C).

Counterintuitively, after going through the apoptotic stage (indicated by a green hexagon in Figure 4E), the trajectories reconnected back to the canonical trajectory (indicated by a gray hexagon in Figure 4E). The reason for this was a (valid) connection between CD4 SP cells and apoptotic cells, reflecting negative selection in the thymic medulla. However, this connection went in the opposite direction from the biologically meaningful course of events. One reason for this was the similar immunophenotype of all late apoptotic cells regardless of the stage at which they entered the apoptotic pathway (with characteristic expression profile of the apoptotic cells being negative for most markers, and positive for Annexin V). At the same time, these apoptotic cells are positioned halfway between the cell of origin and CD4 SP cells. This leads to a reversal of the pseudotime gradient, making the false impression that cells may pass through an apoptotic phase into a later developmental stage.

One of the most important questions in exploratory data analysis is whether the chosen level of detail is sufficient to exploit the information present in the data, or whether resolution needs to be increased. The interactive design of *tviblindi* allowed us to change the level of resolution on the persistence diagram and to investigate the random walks heading to apoptosis in more detail. After increasing the resolution, we observed that leaf V random walks split into two clear trajectories (Va and Vb, Figure 4D). The Va trajectories corresponded to the aforementioned failure to pass the positive/negative selection checkpoint. The Vb trajectories diverted from the canonical trajectory even earlier (Figure 4E), before the DP stage, and headed directly toward apoptosis.

Contrary to Va where the cells expressed intracellular TCRβ prior to Annexin V, the Vb cells failed to express TCRβ before entering the apoptotic trajectory (Figure 4F). We interpret this as failure to pass the β selection checkpoint due to unsuccessful TCRβ chain rearrangement. The later appearance of TCR β, after expression of Annexin V (Figure 4F), is caused by mixing with cells entering apoptosis at later developmental stages.
Similarly, if we focus on CD8 T cells, the same checkpoints can be discovered and interrogated (Supplementary Figure 4).

In summary, we show that *tviblindi* organized individual cells expressing various levels of 34 markers into trajectories leading to the expected ends of αβ T-cell development and discovered the corresponding checkpoints. Biological interpretation was enabled by plotting the expression of key markers along the developmental pseudotime. We were able to describe the canonical development of conventional CD4 T cells and CD8 T cells from their progenitors in the thymus to effectors in peripheral blood. We also found trajectories leading to apoptosis, when thymocytes failed to pass TCRβ selection or positive selection or when they were negatively selected (See Supplementary Video 2).

Non-conventional end interpretation

The last analyzed trajectory remaining to be interpreted was the one leading to endpoint #6 (Figure 3A, Figure 5A). The corresponding population is hereafter referred to as End#6. End#6 was found in the thymic region (compare Figure 2B and Figure 3A for precise location in the *vaevictis* plot) and contained CD4 SP cells as well as a smaller fraction of cells which fall within the DP gate (Supplementary Figure 5A). The pseudotime line plot of CD8 expression shows that the End#6 cells gain dim expression of CD8 at the very end of the trajectory (Supplementary Figure 5B, Figure 5B). The possibility that $CD4^{pos}CD8^{dim}$ cells from the End#6 population are doublets consisting of a developing DP cell and a more mature CD4 SP T cell, as previously suggested[44], was ruled out by comparing the DNA content of End#6 SP and DP populations with the DNA content of dividing $CD34^{pos}$ precursor cells (Supplementary Figure 5C).

The trajectory leading to End#6 diverged from the conventional CD4 trajectory at the branching point of negative selection (Figure 5C, compare with Figure 4E) and continued through a distinct intermediate subset before reaching End#6. The End#6 cells have a $CD25^{hi}$, $CD127^{pos}$ phenotype, which distinguishes them from the preceding stage reminiscent of T regulatory (Treg) cells ($CD25^{pos}CD127^{neg}$), which could represent the immature Treg stage (Figure 5B and 5D). This observation led us to investigate the expression of Treg-specific markers in End#6 cells using a modified antibody panel (Supplementary Table Mass panel 2). After we gated the relevant populations and visualized them in the *vaevictis* plot (Supplementary Figure 6), we observed that FOXP3 (transcription factor of Treg cells) and Helios (transcription factor shown to be upregulated upon negative selection in mice[19,45,46]) is expressed by the $CD25^{pos}CD127^{neg}$ immature Treg cells as well as by the $CD25^{hi}CD127^{pos}$ End#6 cells (Figure 5E). Therefore, we hypothesized that End#6 cells constitute a specific Treg population (Figure 5D and 5E).

Some published reports describe the divergence of Treg cell development at the DP stage, with FOXP3 expression already detectable at the DP stage[27,44,47–49]. Therefore, we wanted to ensure that positioning of the $CD8^{dim}$ cells at the end of the trajectory is not an artifact of the TI. We decided to sort the populations of interest by fluorescence-activated cell sorting (FACS) and estimate their developmental order based on the analysis of the presence of T-cell receptor excision circles (TRECs).

We used a specific panel (Supplementary Table Mass panel 3) to select surface markers distinguishing between End#6 cells, immature Tregs, CD4 SP and DP cells (Supplementary Figure 7) and to design a gating strategy for sorting these populations (Supplementary Figure 8). To confirm that the sorting strategy led to identification of the correct populations, we overlaid the conventionally gated populations over the *vaevictis* plot (Supplementary Figure 9). Based on this analysis, we designed an 11-color panel for FACS sorting (Supplementary Table FACS panel) and sorted the key populations from thymus and from pediatric and adult peripheral blood (Supplementary Figure

10). We measured the presence of TRECs and T cell receptor alpha constant gene region (TCRAC) in the sorted populations to estimate the number of cell divisions undergone since the recombination of the TCRα chain (Figure 6A). The number of cell divisions, as estimated by the calculated TREC/TCRAC, ratio was lower than two in immature $CD3^{pos}$ DP cells and immature Treg (DP as well as CD4 SP) cells. Thymic conventional mature naive CD4 SP did not feature a significant number of divisions. The TREC/TCRAC ratio in conventional naive CD4 SP cells in pediatric peripheral blood corresponded to 1-2 cell divisions, while in the adult peripheral blood it corresponded to 3-4 cell divisions. For $CD45RA^{neg}$ Treg cells sorted from pediatric as well as adult peripheral blood, this was more than 5 cell divisions. For the End#6 $CD4^{pos}CD8^{dim}$ and End#6 CD4 SP cells, the TREC/TCRAC ratio corresponded to more than 5 divisions. These results confirm that $CD4^{pos}CD8^{dim}$ cells at the End#6 of the trajectory leading through immature Treg CD4 SP cells are not the true thymic DP cells. In agreement with our TI algorithm, we confirmed that End#6 cells represent a more developmentally advanced stage.

Further investigation revealed an increase in the expression of several other markers at the end of the trajectory: apart from CD127 and CD8 the cells gained expression of T-bet, CD95, TIGIT and CD152 (CTLA-4) (Figure 6B). In fact, End#6 cells, irrespective of CD8 expression, phenotypically resembled the previously described long-lived thymic regulatory cells, which have been reported to recirculate from the periphery back to the thymus[50–53].
To test the hypothesis that End#6 cells are recirculating Tregs homing into thymus, we measured the expression of key chemokine receptors[51–56] in the populations of interest (Figure 6C). In contrast with the thymic mature naive CD4 SP[54], End#6 cells as well as immature Treg cells lack the S1PR1 (CD363) necessary to egress from the thymus. On the other hand, End#6 cells express IL-18R (CD218), which has been reported to cause upregulation of CCR6 (CD196) upon stimulation with IL-18. CCR6 can then mediate the entry of Treg cells into the thymus[52,55]. End#6 cells expressed more CCR6 compared to CD4 SP naive T cells and immature Treg cells. The last measured chemokine receptor, CCR7 (CD197) has been previously reported to distinguish the newly formed Treg cells, which do express CCR7, from the recirculating Treg cells which do not express CCR7[51]. Consistent with this, while our immature Treg cells express CCR7, the End#6 cells do not (Figure 6C).

Discussion

We have developed *tviblindi*, a tool which allows a biologist working with real-world multidimensional datasets to explore the development of cells from progenitors toward differentiated stages. By integrating several modules (random walks simulation, pseudotime inference, real-time interactive topological classification using persistence homology and two-dimensional visualization using the *vaevictis* algorithm) it presents a feasible algorithmic solution for TI, which significantly extends upon existing TI algorithms.

tviblindi simulates possible random walks from a user-defined point of origin and highlights their endpoints in the dataset. It allows selection and grouping of similar random walks into trajectories based on visual interaction with the data. For each trajectory, marker intensity changes along trajectory pseudotime progression can be displayed. Also, the key points (e.g., branching) can be projected back onto a two-dimensional visual representation of the dataset. *tviblindi* does not anticipate a single endpoint nor a single branching point with two endpoints. Importantly, it can be used for both mass cytometry (millions of cells but only dozens of parameters) as well as single-cell RNA-seq (thousands of parameters) data.

The major contribution is, however, its versatility and the level of user interaction. By building a modular approach and an interactive interface, we offer a unique tool for a biologist to intuitively understand the dataset, visualize the data and to focus the analysis on meaningful TI endpoints. An efficient discovery of the underlying dynamics is made possible by using persistent homology to aggregate random walks into trajectories, project them immediately onto a low-dimensional embedding (created using a custom dimensionality reduction algorithm: *vaevictis*), and visualize marker-vs-pseudotime line plots.

Of note, dimensionality reduction is only used to visualize the data, and is not involved in any computation within the other modules. *vaevictis* reduction is designed to show gradual changes of single-cell phenotypes, suppressing the effect of abundant populations and showing local as well as global relationships of cells. This is in contrast to UMAP[36,37]and t-SNE[38,39], which prioritize clustering of locally similar cells at the expense of global relationships. The 2D visualization of thymus and peripheral blood CyTOF data by *vaevictis* gave a faithful representation of the basic scheme of T-cell development.

When $CD34^{pos}$ progenitors were selected as the population of origin in our dataset (thymocytes and PBMC), the endpoints of trajectories were found within peripheral CD4 and CD8 cells, the apoptotic cells, and the population of $CD4^{pos}CD8^{dim}$ thymic cells which we further explored in this paper and which we refer to as End#6 cells.

The endpoints at the thymus/peripheral blood transition (#5 for CD4 SP and #9 for CD8 SP) reflect the accumulation of cells at the state of mature naive T cells. From this state activated T cell may progress to become memory T cells. Of note, certain biological endpoints might be skipped if the final population is too small (the endpoint cells are not sufficiently represented) or if their phenotyping markers are missing (e.g., Th1-type, Th2-type markers are not included in any of our mass cytometry panels).

Furthermore, in the absence of markers resolving similar but developmentally unrelated cells, a random walk can create a connection between these unrelated cells and skip from one trajectory to another. When these immunophenotypically undistinguishable cells themselves represent

developmental fate of two distinct trajectories, the two trajectories may combine into a single trajectory, skipping the endpoint in the middle. This phenomenon has been observed in the apoptotic region where some random walks leading to cells dying due to the failure of DP cells to pass the positive/negative selection checkpoint continued past the apoptotic stage in a reverse direction towards SP cells undergoing negative selection. Here the algorithm correctly found the trajectories, but expert's input (understanding that apoptosis is a one-way process) was essential for the interpretation of the direction.

While the *vaevictis* embedding plots the cells on a 2D plot in an intuitive manner, some inherent limitation of 2D embeddings to comprehensively capture complex topologies were manifested. An example is the failure to visualize positive/negative selection of CD8 SP T cells, which is clearly detected by *tviblindi.* Note that when focusing on the trajectories of cells undergoing positive selection (V. Trajectories in Figure 4) in more detail, we have identified two groups of trajectories, one of them corresponding to the positive selection (Va. trajectory) and one of them corresponding to an earlier β selection (cells entered the apoptotic pathway before expressing TCR β chain, Vb. trajectory). This highlights the benefit of expert-guided adjustment of the level of detail.

Investigation of the canonical CD4 and CD8 T-cell development reassured us that we can reconstruct the known sequence of maturation steps extensively described by CDMaps[57], connecting the thymic and peripheral T-cell compartment. We were able to dismiss artifacts caused by cell doublets. Putting expression levels of markers on a pseudotime axis showed us the proper sequence and intensity of expression changes. Note that, for trajectory leading to CD4 SP cells, both CD4 and CD8 reach an expression peak at the DP stage, both decrease their intensity afterwards and then CD4 steadily increases towards CD4 SP stage. CD4 expression level in the peripheral SP cells is twice as high as in the thymic SP cells. The key points of intensity changes can be traced back to the *vaevictis* 2D plots and groups of trajectories can be selected for further investigation (expression levels and 2D plot location).

A good example of the power of the presented approach was found when interpreting thymic Treg populations. The trajectory leading towards thymic End#6 cells diverges from the conventional CD4 trajectory at the location of negative selection. It goes through a CD45RA^{neg} immature Treg stage, characterized by the expression of CD25 in the absence of CD127 (conventional Treg immunophenotype), and continues towards the End#6 cells highly expressing CD25 and also expressing CD127. Both the immature Treg cells and the End#6 cells express Treg markers FOXP3 and Helios, confirming their Treg identity. Surprisingly, the End#6 cells gain a dim expression of CD8, T-bet, CTLA-4, TIGIT, while they lose CD69, CD38 and CCR7. However, it has been shown that the ThPOK expression can be reduced in CD4 T cells leading to reacquisition of CD8, and thus mimicking the DP immunophenotype. This

was shown for CD8αα CD4 intraepithelial lymphocytes and CD8αβ CD4 cells in human blood as well as DP cells found in various tissues and under various disease settings. These have been reported to have enhanced cytotoxic functions or suppressive regulatory functions[58,59]. Of note, the CD8 expression level is lower than that of the “bona fide” DP thymocytes and *tviblindi* positioned these $CD4^{pos}CD8^{dim}$ thymocytes at the end of the developmental trajectory.

Therefore, we wanted to make sure that their terminal end position is correct and that they are indeed developmentally more advanced than the conventional immature DP cells and the immature CD4 SP Treg cells. Indeed, the proliferation history of End#6 cells (both $CD8^{neg}$ and $CD8^{dim}$) showed many more cell divisions than any other thymic subset, more than naive T cells in the periphery (by a measured dilution of TREC DNA in the sorted subsets). However, we cannot claim with certainty that these cells follow an uninterrupted developmental path from the precursors to End#6.
Indeed, the surface immunophenotype of End#6 cells corresponds to the previously described population of peripheral Treg cells recirculating back to the thymus[31,53]. Future experimental studies specifically designed to investigate this Treg population should address whether these cells represent a resident subset in the human thymus or if they result from recirculation or represent an additional path of thymic Treg differentiation. These future studies could benefit from a *tviblindi* analysis of their immunophenotyping datasets.

In conclusion, we have built and validated a TI method that is modular and interactive, thus combining the information strength of rich single-cell datasets with the power of expert knowledge for the interpretation of developmental trajectories. It has a generic use for mass cytometry, flow cytometry and single-cell RNA-seq. Importantly, the complete workflow is manageable on a desktop computer in a reasonable computation time (See Supplementary note, subsection 8.3 Running times).

Acknowledgement
T.K. and J.S. were financially supported by project NU23-07-00170 of the Czech Republic Ministry of Health. Institutional support was provided by the project National Institute for Cancer Research (Project No. LX22NPO5102), funded by the European Union–Next Generation EU.
D.N. is a fellow of FWO (Fonds Wetenschappelijk Onderzoek – Vlaanderen), supported by Strategic Basic research grant 1S40423N.

Methods

Sample preparation for mass cytometry

Reagents
Unless otherwise stated, buffers and isotope labeled monoclonal antibodies (moAbs) were purchased from Standard BioTools Inc. (South San Francisco, USA). In-house antibody conjugations were performed using MaxPar labeling kits (Standard BioTools Inc.) according to the manufacturer's instructions. Carrier-protein-free moAbs were purchased from Exbio (Prague, Czech Republic), BioLegend (San Diego, USA), R&D Systems (Minneapolis, USA), eBioscience (San Diego, USA, part of Thermo Fisher Scientific Waltham, USA), Miltenyi Biotec (Bergisch Gladbach, Germany), Cell Signalling Technology (Danvers, USA) (for details see Supplementary Table Mass panels).
Antibody cocktails
The amount of labeled moAbs needed for one test was determined by titration. Two sets of antibody cocktails were prepared and frozen as described previously[60]: 1) the antibody cocktail containing moAbs against surface antigens, which was applied prior to fixation and permeabilization steps, and 2) antibody cocktail containing moAbs against intracellular antigens, which was applied after the permeabilization.

Human cells
Thymocytes were obtained from healthy donor thymi of pediatric patients undergoing heart surgery at the Motol University Hospital, after informed consent was given by the patient guardians, in accordance with the Declaration of Helsinki. Peripheral blood mononuclear cells (PBMC) were obtained by density gradient centrifugation from buffy coats of healthy adult donors, at Institute of Hematology and Blood Transfusion in Prague. The pediatric PBMC were obtained from healthy pediatric patients undergoing pre-surgical examination for non-malignant and otherwise unrelated medical issue after consent was given by patients' guardians. Cells were stored in liquid nitrogen and thawed when needed for the experiment. After thawing, the cells were quickly transferred into 10 ml of pre-warmed RPMI supplemented with 10% FBS and 100µg of Pulmozyme (dornase alpha, Roche, Basel, Switzerland) for 30 min at 37°C. In order to remove dead cells and avoid clumping of the thymocytes, the cells were centrifuged at low speed 108g for 15 minutes. Subsequent steps were performed at 4°C.

Staining
Cells were washed in Annexin binding buffer, ABB (Exbio), and counted, approximately 5million thymocytes and 1million PBMC were used for one experiment.
To minimize technical variability, the individual tissues were barcoded with CD45 Abs conjugated with different isotopes (30min, 4°C). Biotin labeled Annexin V was added together with the barcoding Abs, to distinguish apoptotic

cells with exposed phosphatidylserine residues. After this incubation, the sample volume was risen to 1ml and Cisplatin 198Pt was added to distinguish dead cells (5min, 4°C). Cells were then washed twice in ABB (92g, 15 min, 4°C) and the samples were pooled together, spun (92g, 15 min, 4°C) and processed further as one sample.
The surface Ab cocktail, including the isotope labeled anti-biotin mAb to visualize the annexin, was added to the cell pellet (total volume of surface Abs was approx. 100µl, 30min, 4°C).
Cells were washed twice in ABB (92g, 15 min, 4°C), prefixed with 1ml freshly diluted 1,6% PFA (10min RT) and washed once more in Maxpar® Cell Staining Buffer, CSB (1050g, 5 min, 4°C). Then, the cells were fixed and permeabilized using Maxpar® Nuclear Antigen Staining buffer (30min, 4°C) followed by two washings in Maxpar® Nuclear Antigen Staining Perm (1050g, 5 min, 4°C).
In cases when the FOXP3 antibody was used, the cells stained with the surface Abs and washed in ABB were fixed and permeabilized in FOXP3 fixation/permeabilization buffer (30min, 4°C) followed by two washings in FOXP3 permeabilization buffer (600g, 5 min, 4°C).
The intracellular Ab cocktail was added to the cell pellet, together with anti-TCRb frame antibody labeled with APC (it was not possible for us to get this antibody in the unlabeled carrier-free form allowing in-house isotope conjugation) (the total volume of intracellular Abs was approx. 60µl, 30min, 4°C). The cells were washed twice in CSB (1050g, 5min, 4°C). Finally, an APC directed isotope labeled mAb was added (30min, 4°C), cells were washed twice in CSB (1050g, 5min, 4°C), fixed in 1ml freshly diluted 1.6% formaldehyde (10min, RT) and centrifuged (1050g, 5min, 4°C).

Mass cytometry samples acquisition
After incubation in 125nM 191/193Ir in Maxpar® Fix and Perm Buffer (at least 12h, up to 2 weeks), the cells were washed twice in CSB (1050g, 5min, RT) and once in Maxpar® Cell Acquisition Solution (CAS, 1050g, 5min, RT). Then they were diluted up to 1×10^6 cells/ml in 15% EQ™ Four Element Calibration Beads (Standard BioTools Inc.) in CAS and filtered through a 35 µm nylon mesh cell-strainer cap (BD Biosciences, San Jose, CA). The samples were acquired using Helios (Standard BioTools Inc.), CyTOF software version 6.7.1014. The instrument was tuned for acquisition using Tuning Solution (Standard BioTools Inc.) according to the manufacturer's instructions. The noise reduction (the cell length 7–150, lower convolution threshold 200) was applied during the acquisition. The signal was normalized using a Standard BioTools Inc. algorithm, which is based on the 'Bead Passport' concept.

Analysis of mass cytometry data
After export to listmode data (FCS format), files were further processed using FlowJo (v10, BD Biosciences). First, nucleated cells were gated by metal-tagged DNA intercalator 191/193Ir, and thymus versus peripheral blood was debarcoded by manual gating (Supplementary Figure 1). Next, PBMC were gated on $CD3^{pos}$ lineage negative and TCR γδ negative T cells. The thymus cells were gated on lineage negative, viable (platinum negative) and TCR γδ negative thymocytes. Thus, only TCR αβ T cells and their progenitors in the

thymus were used for further processing by the *tviblindi* algorithm described later. When manual gating was performed prior to *tviblindi*, the FlowJo workspace file was used to save manual gate positions, so that the gated cells could be displayed in the GUI, to help the interpret the *vaevictis* plot.
The *tviblindi* algorithm was programmed in R, C++ and Python. The source code is deposited on GitHub (https://github.com/stuchly/tviblindi).
The listmode data was transformed using *arcsinh* with cofactor 5 for subsequent analysis[61]. After data interrogation by *tviblindi*, an 'enhanced' FCS file was created, containing artificial channels with newly computed variables (2-dimensional embedding coordinates, pseudotime, pathway membership identifier, labels and selected points on trajectories) and a graphical output for figures was created using FlowJo layouts.

FACS sorting
Human thymocytes and PBMC obtained as described above for mass cytometry experiments were stained with appropriate amount of fluorescently labeled monoclonal antibodies, at 1 million cells/50ul for 30 minutes on ice in the dark (see Supplementary Table FACS panel). After incubation, cells were washed in ice-cold PBS and sorted using BD FACS Aria III cell sorter into tubes with 10 ul of ice-cold Platinum Direct cell lysis buffer with Proteinase K (ThermoFisher Scientific). The obtained subsets (see Supplementary Figure 10) were then directly processed according to the manufacturer's instructions to minimize any loss of material.

TREC analysis
Extracted DNA was used for the quantification of T-cell receptor excision circles (TREC) and T cell receptor alpha constant gene region (TCRAC) via previously established real-time PCR assay[62–64]. Estimated number of cell divisions was calculated based on the ratio between TREC and TCRAC copies in these populations.

Figure Legends

Figure 1: Analytical process on the artificial *dyntoy* dataset

Dashed lines mark selections made by the user, panels A-C show a scheme of the whole dataset, panels D-I correspond to the interactive GUI for this dataset: (A) The scheme of a point cloud representing single-cell data, where sparse regions α and β are present. (B) The dataset is represented as a k-NNG and the cell of origin is selected. (C) Pseudotime is calculated as expected hitting distance and the edges of the k-NNG are oriented according to the pseudotime. Candidate endpoints are automatically suggested as vertices without outgoing edges (circled). (D) Potential developmental fates (x, y, z) are shown as gray points on the *vaevictis* plot. Endpoints can be selected for further investigation (in this example, all endpoints are selected, dashed line). (E) Sparse regions are represented on a persistence diagram, which enables the selection of the significant sparse regions (dashed line). The sparse regions γ and δ correspond to essential classes created by the selection of disparate endpoints. This way, random walks are classified using the selected ends and sparse regions. (F) All random walks are shown on the *vaevictis* plot and their classification into trajectories is depicted in the hierarchical clustering dendrogram. (G) By selecting branch (a) leading to the endpoint (x) of this dendrogram, the given trajectory can be investigated in detail, including viewing the average expression of multiple markers along pseudotime. (H) By selecting two endpoints (x and y) but only a single relevant marker, its dispersion and possible branching points can be examined. Trajectories (a and e) from panel F are visualized and the branching region (p) is selected. (I) Multiple trajectories (trajectory i in red and a and e in blue) can be visualized on the 2D plot. Points in the selected branching region (p) are highlighted in green. (J) Connectome representing a basic structure of the data and the simulated random walks. The pie charts indicate the distributions of cell populations in each vertex (cluster). Vertex containing the cell of origin (O) and vertices containing endpoints (T).

Figure 2: Visualization of the basic structure of the T-cell compartment using *vaevictis*

(A) Bivariate CD4 x CD8 dot plot of cells from human thymus (brown) and human peripheral blood (blue) acquired using mass cytometry. The cells were gated as DN, DP, CD4 SP and CD8 SP. (B) *vaevictis* plot of 37-parameter (including barcode) mass cytometry panel measurement showing the positions of human thymus (brown) and human peripheral blood (blue), with the $CD34^{pos}$ progenitors shown in green. (C) Expression of CD8, CD4 and Annexin V shown

using a green-yellow-red color gradient on the *vaevictis* plot. Green color indicates the lowest expression and red color indicates the highest.

Figure 3: Developmental endpoints and detailed analysis of major trajectory leading to endpoint #1 by *tviblindi*

(A) *vaevictis* plot of T-cell development in thymus and peripheral blood. Estimated pseudotime is represented by a yellow-to-red color gradient with $CD34^{pos}$ progenitors as the population of origin (purple dot). Gray dots indicate the discovered developmental endpoints. Endpoint #1 highlighted by a blue rectangle represents mature $CD4^{pos}$ effector memory T cells selected for further exploration. (B) Connectome shows low resolution structure of the data and of simulated random walks. (C) Persistence diagram representing sparse regions detected within the point cloud of measured cells. The orange rectangle marks a user-defined selection of sparse regions. (D) Dendrogram of clustered trajectories shows a subcluster of 452 walks (red rectangle, labeled I) within a larger group of random walks (blue rectangle, labeled II). The number to the left of each leaf indicates the number of random walks in the leaf. (E) *vaevictis* plot displaying the above selected trajectories in corresponding colors. (F) Pseudotime line plot showing the average expression of selected individual markers along the trajectory to endpoint #1 (top). The selected areas of interest (green rectangles) are shown in green on the *vaevictis* plots (below).

Figure 4: *tviblindi* analysis of trajectories leading to apoptosis

(A) Dendrogram identical to Figure 3D with additional trajectories selected for closer investigation (blue rectangles) labeled III (137 random walks) and V (159 random walks). (B) *vaevictis* plot showing the topology of trajectories in leaves III and V. Apoptotic cells are shown in green. (C) Pseudotime line plot, which shows the average expression of selected markers along calculated pseudotime. The green rectangle highlights the region with increased expression of apoptotic marker Annexin V and decreased expression of phosphotyrosine. Events in the selected region are displayed in panel 4B in green (see the arrow). (D) A detailed dendrogram of leaf V from panel 4A. Two distinct trajectories were selected for further analysis (blue rectangle, Va and red rectangle, Vb). (E) *vaevictis* plot showing the topology of trajectories in leaves Va and Vb. The green polygon marks the region of apoptosis, the gray polygon marks the region of more advanced stages of thymic and peripheral CD4 T-cell development. (F) Pseudotime line plot, which depicts the average expression of selected markers along the trajectories Va and Vb. The region

of apoptosis and the region of more advanced stages of CD4 T-cell development are highlighted (green and gray rectangle respectively).

Figure 5: *tviblindi* analysis of the trajectory leading to End#6 population
(A) A trajectory leading to End#6 located in the thymic portion of the *vaevictis* plot (compare to Figure 2B). (B) Pseudotime line plot showing average expression of selected individual markers along developmental pseudotime. (C) *vaevictis* plot showing the trajectory to the conventional naive CD4 SP T cells (orange) and the trajectory to End#6 (blue). Conventional naive CD4 SP T cells are shown in pink, the immature Treg stage in dark blue and End#6 in green. (D) Bivariate plot of CD25 and CD127 expression on gated conventional naive CD4 SP T cell (pink), immature Treg stage (dark blue) and End#6 (green). (E) Bivariate plot of Helios and FOXP3 expression on conventional naive CD4 SP T cell (pink), immature Treg stage (dark blue) and End#6 (green) in a validation experiment.

Figure 6: Detailed analysis of End#6 cells
(A) Visualization of the TREC/TCRAC ratio, depicted as the calculated number of divisions for each population. Symbol code: cells from peripheral blood (blue), cells from thymus (brown), cells from adult donor (circle), cells from pediatric donor (triangle). (B) Pseudotime line plot depicting the expression of TIGIT, CD95, CD152, T-bet, CD69 and CD197 markers by cells from trajectory to End#6. (C) Overlaid histograms showing the expression of chemokine and cytokine receptors CD197, CD363, CD218 and CD196 in the respective populations. Symbol code: End#6 cells from thymus (orange), Tregs from peripheral blood (blue), immature Tregs from thymus (brown) and naive CD4 SP T cell from peripheral blood (purple).

References

1. Bendall, S. C. *et al.* Single-Cell Trajectory Detection Uncovers Progression and Regulatory Coordination in Human B Cell Development. *Cell* **157**, 714–725 (2014).
2. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nature Biotechnology 2019 37:5* **37**, 547–554 (2019).
3. Cao, J. *et al.* The single-cell transcriptional landscape of mammalian organogenesis. *Nature 2019 566:7745* **566**, 496–502 (2019).
4. Chen, H. *et al.* Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. *Nature Communications 2019 10:1* **10**, 1–14 (2019).
5. Wolf, F. A. *et al.* PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol* **20**, 1–9 (2019).
6. Setty, M. *et al.* Characterization of cell fate probabilities in single-cell data with Palantir. *Nature Biotechnology 2019 37:4* **37**, 451–460 (2019).
7. Stassen, S. V., Yip, G. G. K., Wong, K. K. Y., Ho, J. W. K. & Tsia, K. K. Generalized and scalable trajectory inference in single-cell omics data with VIA. *Nature Communications 2021 12:1* **12**, 1–18 (2021).
8. Packer, J. S. *et al.* A lineage-resolved molecular atlas of C. Elegans embryogenesis at single-cell resolution. *Science (1979)* **365**, (2019).
9. Briggs, J. A. *et al.* The dynamics of gene expression in vertebrate embryogenesis at single-cell resolution. *Science (1979)* **360**, (2018).
10. Cao, J. *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science (1979)* **357**, 661–667 (2017).
11. Litviňuková, M. *et al.* Cells of the adult human heart. *Nature 2020 588:7838* **588**, 466–472 (2020).
12. Farley, A. M. *et al.* Dynamics of thymus organogenesis and colonization in early human development. *Development* **140**, 2015–2026 (2013).
13. Haddad, R. *et al.* Dynamics of Thymus-Colonizing Cells during Human Development. *Immunity* **24**, 217–230 (2006).
14. Hosokawa, H. & Rothenberg, E. V. How transcription factors drive choice of the T cell fate. *Nature Reviews Immunology 2020 21:3* **21**, 162–176 (2020).
15. Canté-Barrett, K. *et al.* Loss of CD44dim expression from early progenitor cells marks T-cell lineage commitment in the human thymus. *Front Immunol* **8**, 32 (2017).
16. Dik, W. A. *et al.* New insights on human T cell development by quantitative T cell receptor gene rearrangement studies and gene

expression profiling. *Journal of Experimental Medicine* **201**, 1715–1723 (2005).
17. Surh, C. D. & Sprent, J. T-cell apoptosis detected in situ during positive and negative selection in the thymus. *Nature 1994 372:6501* **372**, 100–103 (1994).
18. Kimura, M. Y. *et al.* Timing and duration of MHC I positive selection signals are adjusted in the thymus to prevent lineage errors. *Nature Immunology 2016 17:12* **17**, 1415–1423 (2016).
19. Klein, L., Robey, E. A. & Hsieh, C. S. Central CD4+ T cell tolerance: deletion versus regulatory T cell differentiation. *Nature Reviews Immunology 2018 19:1* **19**, 7–18 (2018).
20. Derbinski, J., Schulte, A., Kyewski, B. & Klein, L. Promiscuous gene expression in medullary thymic epithelial cells mirrors the peripheral self. *Nature Immunology 2001 2:11* **2**, 1032–1039 (2001).
21. Anderson, M. S. *et al.* Projection of an immunological self shadow within the thymus by the aire protein. *Science (1979)* **298**, 1395–1401 (2002).
22. Hsieh, C. S., Lee, H. M. & Lio, C. W. J. Selection of regulatory T cells in the thymus. *Nature Reviews Immunology 2012 12:3* **12**, 157–167 (2012).
23. Liu, Y. *et al.* A critical function for TGF-β signaling in the development of natural CD4+CD25+Foxp3+ regulatory T cells. *Nature Immunology 2008 9:6* **9**, 632–640 (2008).
24. D'Cruz, L. M. & Klein, L. Development and function of agonist-induced CD25+Foxp3+ regulatory T cells in the absence of interleukin 2 signaling. *Nature Immunology 2005 6:11* **6**, 1152–1159 (2005).
25. Burchill, M. A., Yang, J., Vogtenhuber, C., Blazar, B. R. & Farrar, M. A. IL-2 Receptor β-Dependent STAT5 Activation Is Required for the Development of Foxp3+ Regulatory T Cells. *The Journal of Immunology* **178**, 280–290 (2007).
26. Lio, C. W. J. & Hsieh, C. S. A Two-Step Process for Thymic Regulatory T Cell Development. *Immunity* **28**, 100–111 (2008).
27. Caramalho, I. *et al.* Human regulatory T-cell development is dictated by Interleukin-2 and -15 expressed in a non-overlapping pattern in the thymus. *J Autoimmun* **56**, 98–110 (2015).
28. Vanhanen, R., Tuulasvaara, A., Mattila, J., Pätilä, T. & Arstila, T. P. Common gamma chain cytokines promote regulatory T cell development and survival at the CD4+ CD8+ stage in the human thymus. *Scand J Immunol* **88**, e12681 (2018).
29. Golding, A., Darko, S., Wylie, W. H., Douek, D. C. & Shevach, E. M. Deep sequencing of the TCR-β repertoire of human forkhead box protein 3 (FoxP3)+ and FoxP3– T cells suggests that they are

completely distinct and non-overlapping. *Clin Exp Immunol* **188**, 12–21 (2017).
30. Owen, D. L. *et al.* Thymic regulatory T cells arise via two distinct developmental programs. *Nature Immunology 2019 20:2* **20**, 195–205 (2019).
31. Santamaria, J. C., Borelli, A. & Irla, M. Regulatory T Cell Heterogeneity in the Thymus: Impact on Their Functional Activities. *Front Immunol* **12**, 643153 (2021).
32. Heimli, M. *et al.* Multimodal human thymic profiling reveals trajectories and cellular milieu for T agonist selection. *Front Immunol* **13**, 1092028 (2023).
33. Topological estimation using witness complexes. http://diglib.eg.org/handle/10.2312/SPBG.SPBG04.157-166.
34. Edelsbrunner, H. & Harer, J. *Computational Topology: An Introduction*. *Computational Topology* vol. 69 (Miscellaneous Books, 2010).
35. T. Pokorny, F., Hawasly, M. & Ramamoorthy, S. Multiscale Topological Trajectory Classification with Persistent Homology. (2015) doi:10.15607/RSS.2014.X.054.
36. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology 2018 37:1* **37**, 38–44 (2018).
37. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J Open Source Softw* **3**, 861 (2018).
38. Amir, E. A. D. *et al.* viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat Biotechnol* **31**, 545 (2013).
39. Maaten, L. van der & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9**, 2579–2605 (2008).
40. Ding, J., Condon, A. & Shah, S. P. Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nat Commun* **9**, (2018).
41. Szubert, B., Cole, J. E., Monaco, C. & Drozdov, I. Structure-preserving visualisation of high dimensional single-cell datasets. *Sci Rep* **9**, (2019).
42. Förster, Y. P., Gamberi, L., Tzanis, E., Vivo, P. & Annibale, A. Exact and approximate mean first passage times on trees and other necklace structures: a local equilibrium approach. *J Phys A Math Theor* **55**, 115001 (2022).
43. Blondel, V. D., Guillaume, J. L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* **2008**, (2008).

44. Lee, H. M. & Hsieh, C.-S. Rare Development of Foxp3+ Thymocytes in the CD4+CD8+ Subset. *The Journal of Immunology* **183**, 2261–2266 (2009).
45. Mittelstadt, P. R., Taves, M. D. & Ashwell, J. D. Glucocorticoids Oppose Thymocyte Negative Selection by Inhibiting Helios and Nur77. *The Journal of Immunology* **203**, 2163–2170 (2019).
46. Daley, S. R., Hu, D. Y. & Goodnow, C. C. Helios marks strongly autoreactive CD4+ T cells in two major waves of thymic deletion distinguished by induction of PD-1 or NF-κB. *J Exp Med* **210**, 269–285 (2013).
47. Tuovinen, H., Pekkarinen, P. T., Rossi, L. H., Mattila, I. & Arstila, T. P. The FOXP3+ subset of human CD4+CD8+ thymocytes is immature and subject to intrathymic selection. *Immunol Cell Biol* **86**, 523–529 (2008).
48. Lehtoviita, A., Rossi, L. H., Kekäläinen, E., Sairanen, H. & Arstila, T. P. The CD4+CD8+ and CD4+ Subsets of FOXP3+ Thymocytes Differ in their Response to Growth Factor Deprivation or Stimulation. *Scand J Immunol* **70**, 377–383 (2009).
49. Vanhanen, R., Leskinen, K., Mattila, I. P., Saavalainen, P. & Arstila, T. P. Epigenetic and transcriptional analysis supports human regulatory T cell commitment at the CD4+CD8+ thymocyte stage. *Cell Immunol* **347**, 104026 (2020).
50. Yang, E. J., Zou, T., Leichner, T. M., Zhang, S. L. & Kambayashi, T. Both retention and recirculation contribute to long-lived regulatory T-cell accumulation in the thymus. *Eur J Immunol* **44**, 2712–2720 (2014).
51. Cowan, J. E., McCarthy, N. I. & Anderson, G. CCR7 Controls Thymus Recirculation, but Not Production and Emigration, of Foxp3+ T Cells. *Cell Rep* **14**, 1041–1048 (2016).
52. Peligero-Cruz, C. *et al.* Il18 signaling promotes homing of mature tregs into the thymus. *Elife* **9**, 1–23 (2020).
53. Thiault, N. *et al.* Peripheral regulatory T lymphocytes recirculating to the thymus suppress the development of their precursors. *Nature Immunology 2015 16:6* **16**, 628–634 (2015).
54. Allende, M. L., Dreier, J. L., Mandala, S. & Proia, R. L. Expression of the Sphingosine 1-Phosphate Receptor, S1P1, on T-cells Controls Thymic Emigration. *Journal of Biological Chemistry* **279**, 15396–15401 (2004).
55. Cowan, J. E. *et al.* Aire controls the recirculation of murine Foxp3+ regulatory T-cells back to the thymus. *Eur J Immunol* **48**, 844–854 (2018).

56. Matloubian, M. *et al.* Lymphocyte egress from thymus and peripheral lymphoid organs is dependent on S1P receptor 1. *Nature 2004 427:6972* **427**, 355–360 (2004).
57. Kalina, T. *et al.* CD maps—dynamic profiling of CD1–CD100 surface expression on human leukocyte and lymphocyte subsets. *Front Immunol* **10**, 476681 (2019).
58. Overgaard, N. H., Jung, J.-W., Steptoe, R. J. & Wells, J. W. CD4+/CD8+ double-positive T cells: more than just a developmental stage? *J Leukoc Biol* **97**, 31–38 (2015).
59. Taniuchi, I. CD4 Helper and CD8 Cytotoxic T Cell Differentiation. *https://doi.org/10.1146/annurev-immunol-042617-053411* **36**, 579–601 (2018).
60. Schulz, A. R. *et al.* Stabilizing Antibody Cocktails for Mass Cytometry. *Cytometry A* **95**, 910–916 (2019).
61. Nowicka, M. *et al.* CyTOF workflow: differential discovery in high-throughput high-dimensional cytometry datasets. *F1000Research 2019 6:748* **6**, 748 (2019).
62. Froňková, E. *et al.* The TREC/KREC assay for the diagnosis and monitoring of patients with DiGeorge syndrome. *PLoS One* **9**, (2014).
63. Douek, D. C. *et al.* Assessment of thymic output in adults after haematopoietic stemcell transplantation and prediction of T-cell reconstitution. *The Lancet* **355**, 1875–1881 (2000).
64. Sottini, A. *et al.* Simultaneous quantification of recent thymic T-cell and bone marrow B-cell emigrants in patients with primary immunodeficiency undergone to stem cell transplantation. *Clinical Immunology* **136**, 217–227 (2010).

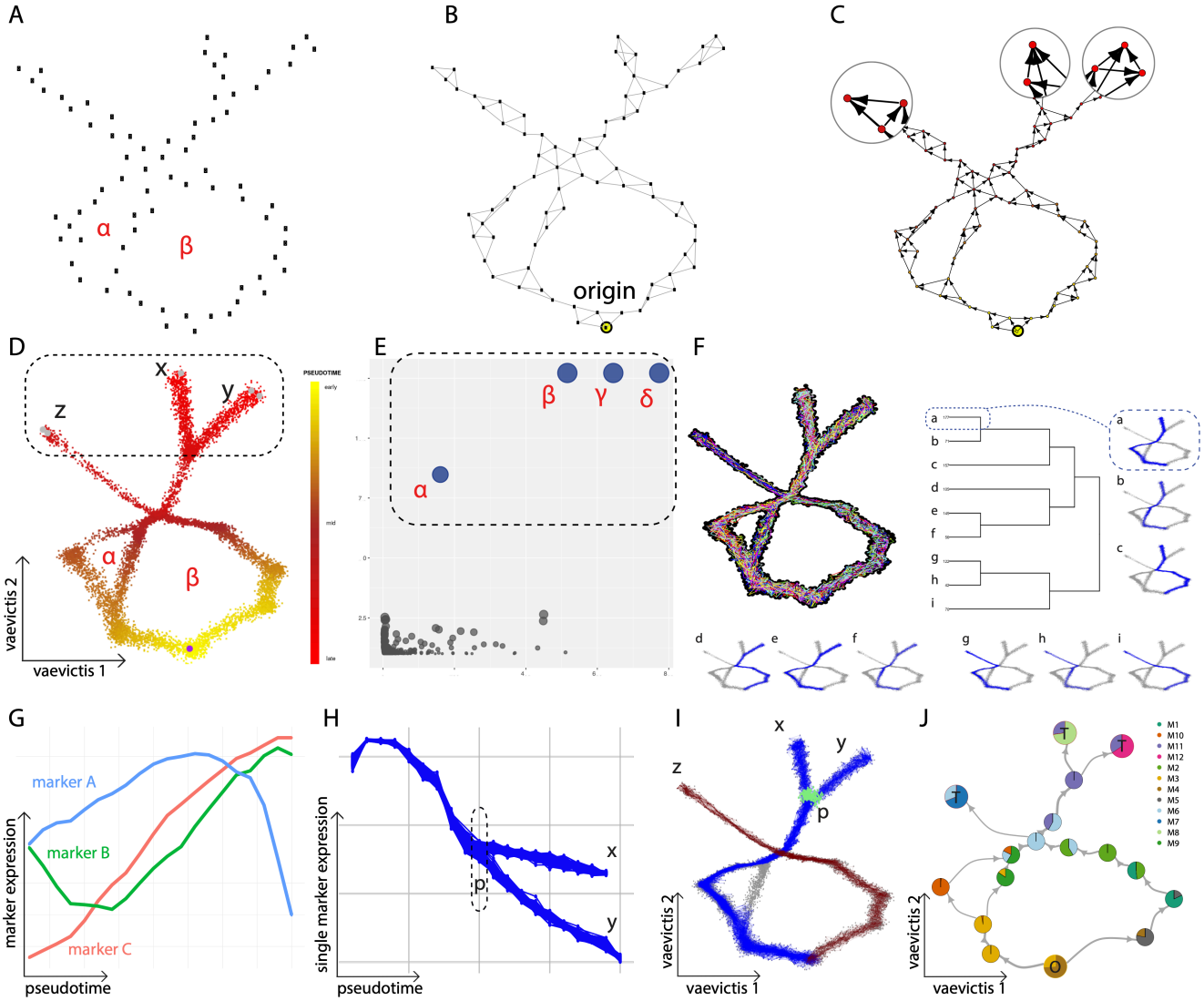


Figure 1: Analytical process on the artificial *dyntoy* dataset Dashed lines mark selections made by the user, panels A-C show a scheme of the whole dataset, panels D-I correspond to the interactive GUI for this dataset: (A) The scheme of a point cloud representing single-cell data, where sparse regions α and β are present. (B) The dataset is represented as a k-NNG and the cell of origin is selected. (C) Pseudotime is calculated as expected hitting distance and the edges of the k-NNG are oriented according to the pseudotime. Candidate endpoints are automatically suggested as vertices without outgoing edges (circled). (D) Potential developmental fates (x, y, z) are shown as gray points on the *vaevictis* plot. Endpoints can be selected for further investigation (in this example, all endpoints are selected, dashed line). (E) Sparse regions are represented on a persistence diagram, which enables the selection of the significant sparse regions (dashed line). The sparse regions γ and δ correspond to essential classes created by the selection of disparate endpoints. This way, random walks are classified using the selected ends and sparse regions. (F) All random walks are shown on the *vaevictis* plot and their classification into trajectories is depicted in the hierarchical clustering dendrogram. (G) By selecting branch (a) leading to the endpoint (x) of this dendrogram, the given trajectory can be investigated in detail, including viewing the average expression of multiple markers along pseudotime. (H) By selecting two endpoints (x and y) but only a single relevant marker, its dispersion and possible branching points can be examined. Trajectories (a and e) from panel F are visualized and the branching region (p) is selected. (I) Multiple trajectories (trajectory i in red and a and e in blue) can be visualized on the 2D plot. Points in the selected branching region (p) are highlighted in green. (J) Connectome representing a basic structure of the data and the simulated random walks. The pie charts indicate the distributions of cell populations in each vertex (cluster). Vertex containing the cell of origin (O) and vertices containing endpoints (T).

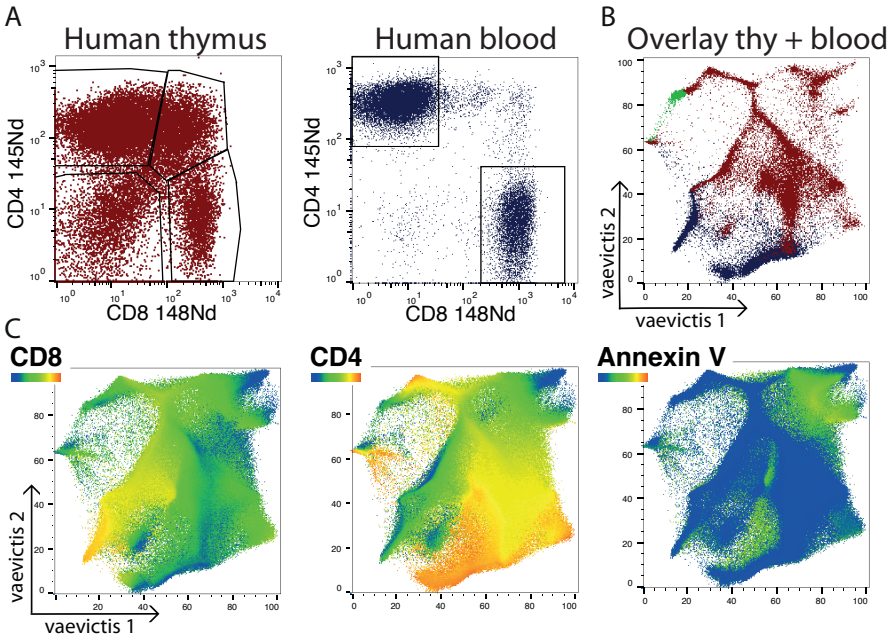


Figure 2: Visualization of the basic structure of the T-cell compartment using *vaevictis* (A) Bivariate CD4 x CD8 dot plot of cells from human thymus (brown) and human peripheral blood (blue) acquired using mass cytometry. The cells were gated as DN, DP, CD4 SP and CD8 SP. (B) *vaevictis* plot of 37-parameter (including barcode) mass cytometry panel measurement showing the positions of human thymus (brown) and human peripheral blood (blue), with the $CD34^{pos}$ progenitors shown in green. (C) Expression of CD8, CD4 and Annexin V shown using a green-yellow-red color gradient on the *vaevictis* plot. Green color indicates the lowest expression and red color indicates the highest.

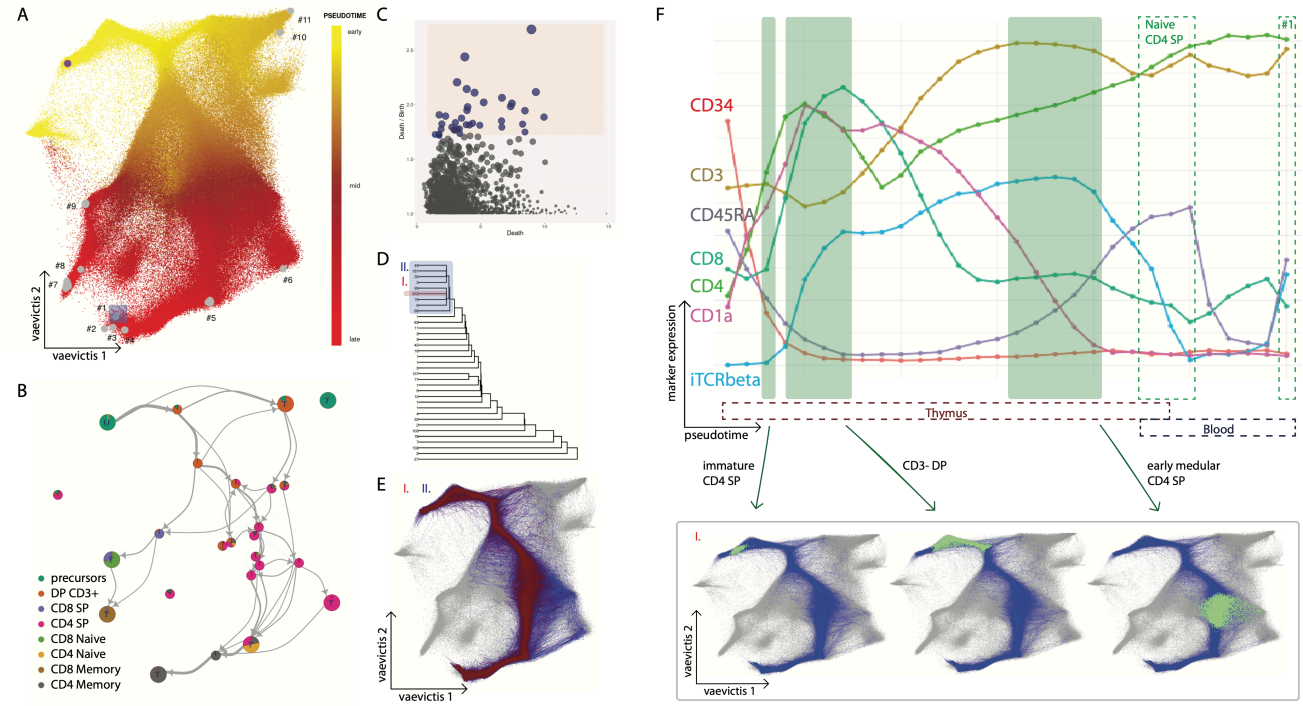


Figure 3: Developmental endpoints and detailed analysis of major trajectory leading to endpoint #1 by *tviblindi*
(A) *vaevictis* plot of T-cell development in thymus and peripheral blood. Estimated pseudotime is represented by a yellow-to-red color gradient with $CD34^{pos}$ progenitors as the population of origin (purple dot). Gray dots indicate the discovered developmental endpoints. Endpoint #1 highlighted by a blue rectangle represents mature $CD4^{pos}$ effector memory T cells selected for further exploration. (B) Connectome shows low resolution structure of the data and of simulated random walks. (C) Persistence diagram representing sparse regions detected within the point cloud of measured cells. The orange rectangle marks a user-defined selection of sparse regions. (D) Dendrogram of clustered trajectories shows a subcluster of 452 walks (red rectangle, labeled I) within a larger group of random walks (blue rectangle, labeled II). The number to the left of each leaf indicates the number of random walks in the leaf. (E) *vaevictis* plot displaying the above selected trajectories in corresponding colors. (F) Pseudotime line plot showing the average expression of selected individual markers along the trajectory to endpoint #1 (top). The selected areas of interest (green rectangles) are shown in green on the *vaevictis* plots (below).

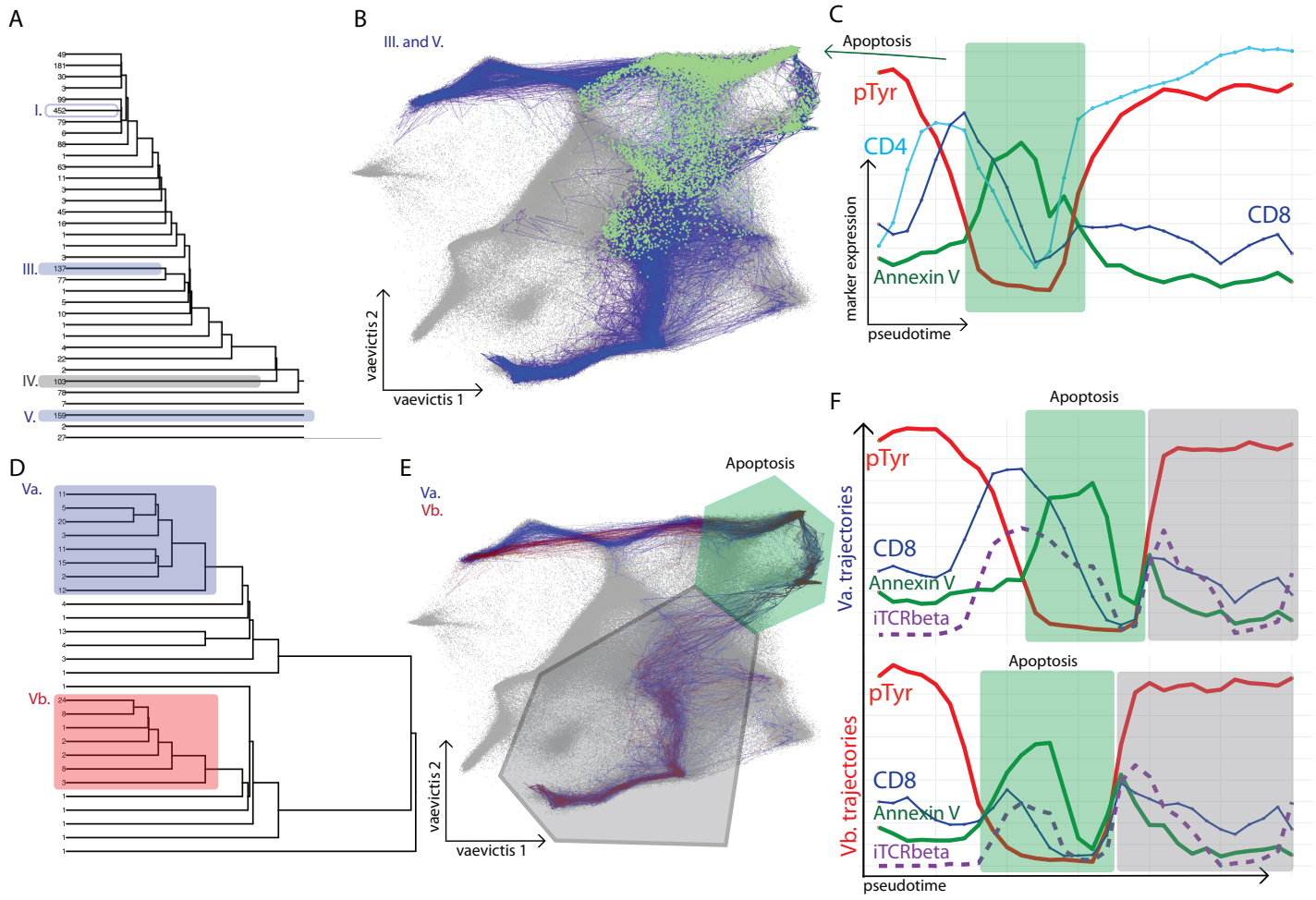


Figure 4: *tviblindi* analysis of trajectories leading to apoptosis (A) Dendrogram identical to Figure 3D with additional trajectories selected for closer investigation (blue rectangles) labeled III (137 random walks) and V (159 random walks). (B) *vaevictis* plot showing the topology of trajectories in leaves III and V. Apoptotic cells are shown in green. (C) Pseudotime line plot, which shows the average expression of selected markers along calculated pseudotime. The green rectangle highlights the region with increased expression of apoptotic marker Annexin V and decreased expression of phosphotyrosine. Events in the selected region are displayed in panel 4B in green (see the arrow). (D) A detailed dendrogram of leaf V from panel 4A. Two distinct trajectories were selected for further analysis (blue rectangle, Va and red rectangle, Vb). (E) *vaevictis* plot showing the topology of trajectories in leaves Va and Vb. The green polygon marks the region of apoptosis, the gray polygon marks the region of more advanced stages of thymic and peripheral CD4 T-cell development. (F) Pseudotime line plot, which depicts the average expression of selected markers along the trajectories Va and Vb. The region of apoptosis and the region of more advanced stages of CD4 T-cell development are highlighted (green and gray rectangle respectively).

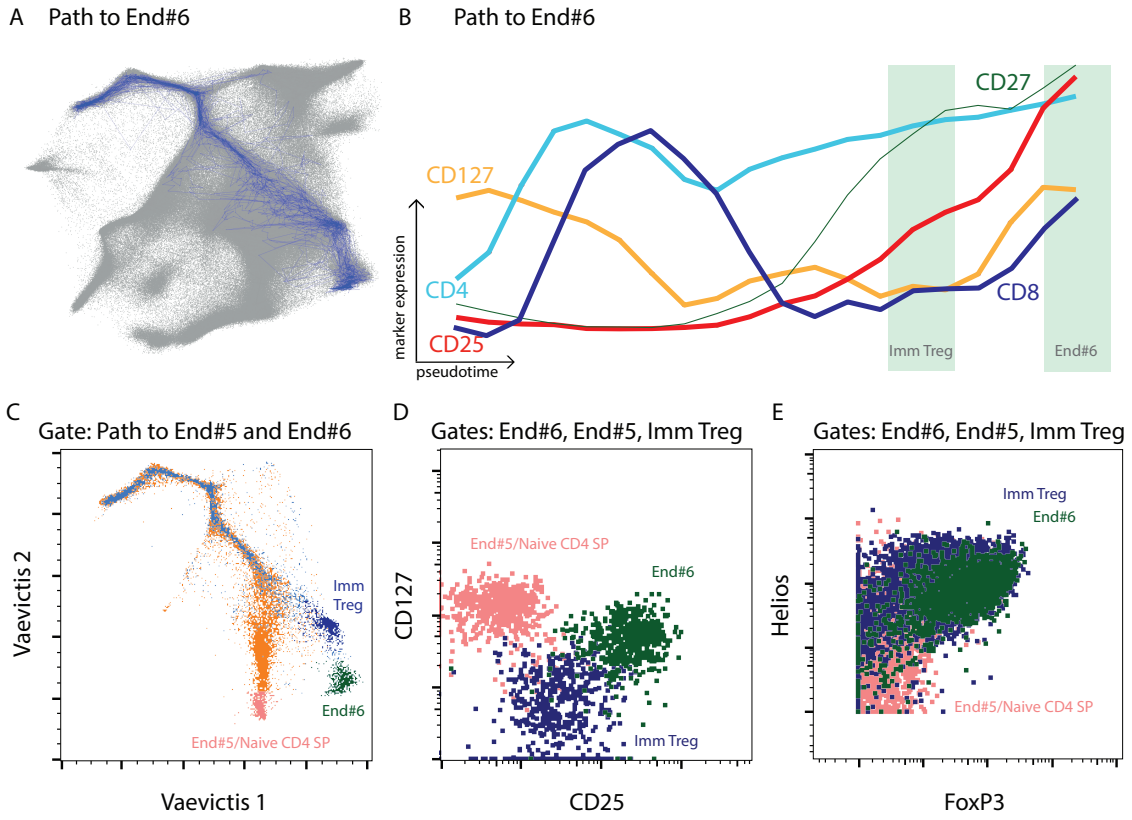


Figure 5: *tviblindi* analysis of the trajectory leading to End#6 population (A) A trajectory leading to End#6 located in the thymic portion of the *vaevictis* plot (compare to Figure 2B). (B) Pseudotime line plot showing average expression of selected individual markers along developmental pseudotime. (C) *vaevictis* plot showing the trajectory to the conventional naive CD4 SP T cells (orange) and the trajectory to End#6 (blue). Conventional naive CD4 SP T cells are shown in pink, the immature Treg stage in dark blue and End#6 in green. (D) Bivariate plot of CD25 and CD127 expression on gated conventional naive CD4 SP T cell (pink), immature Treg stage (dark blue) and End#6 (green). (E) Bivariate plot of Helios and FOXP3 expression on conventional naive CD4 SP T cell (pink), immature Treg stage (dark blue) and End#6 (green) in a validation experiment.

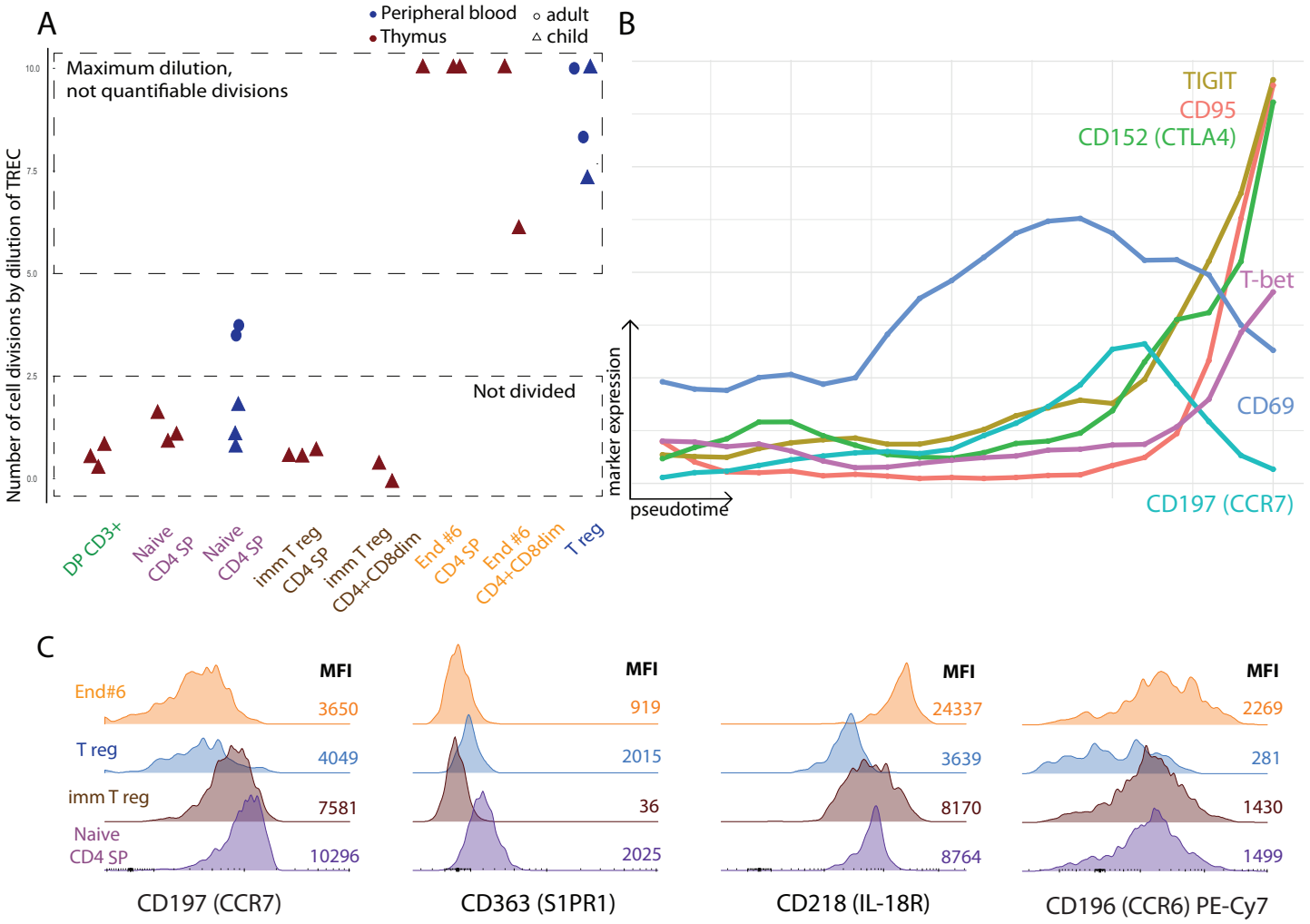
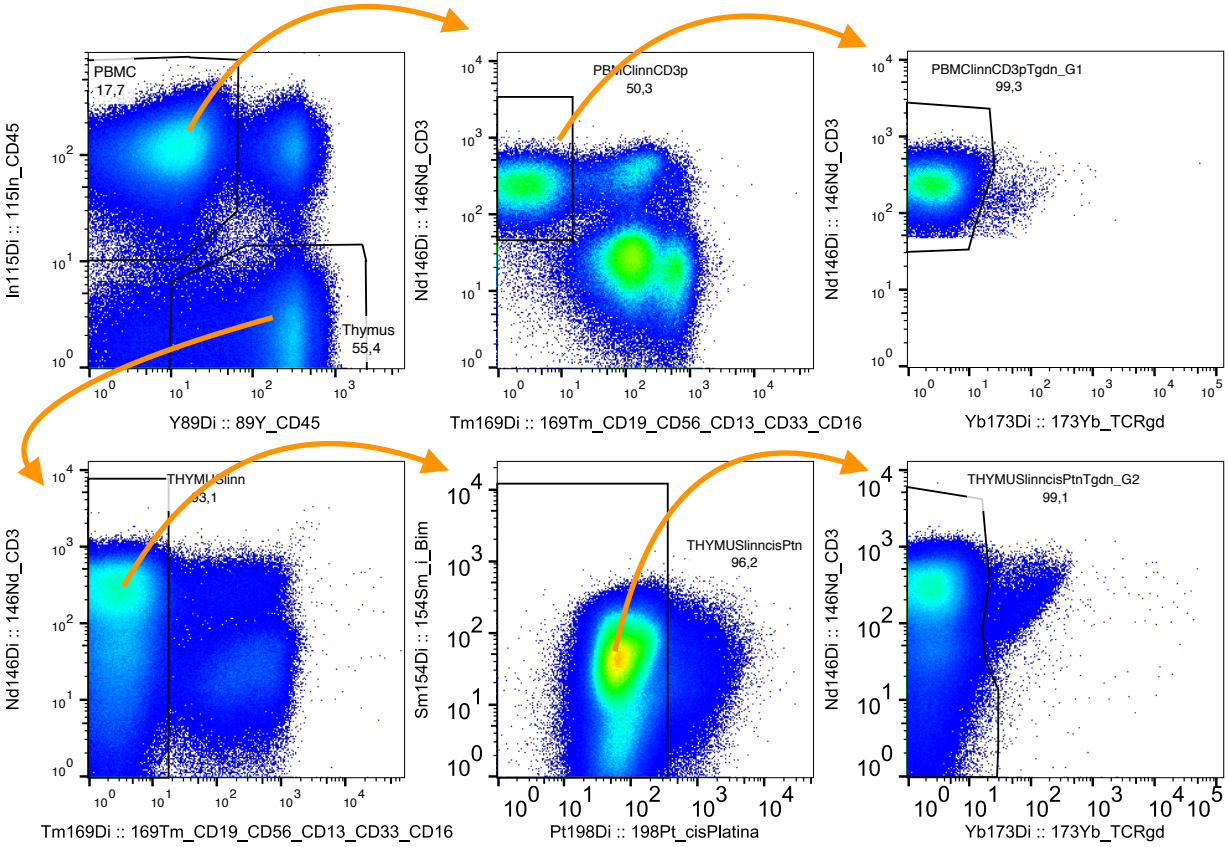
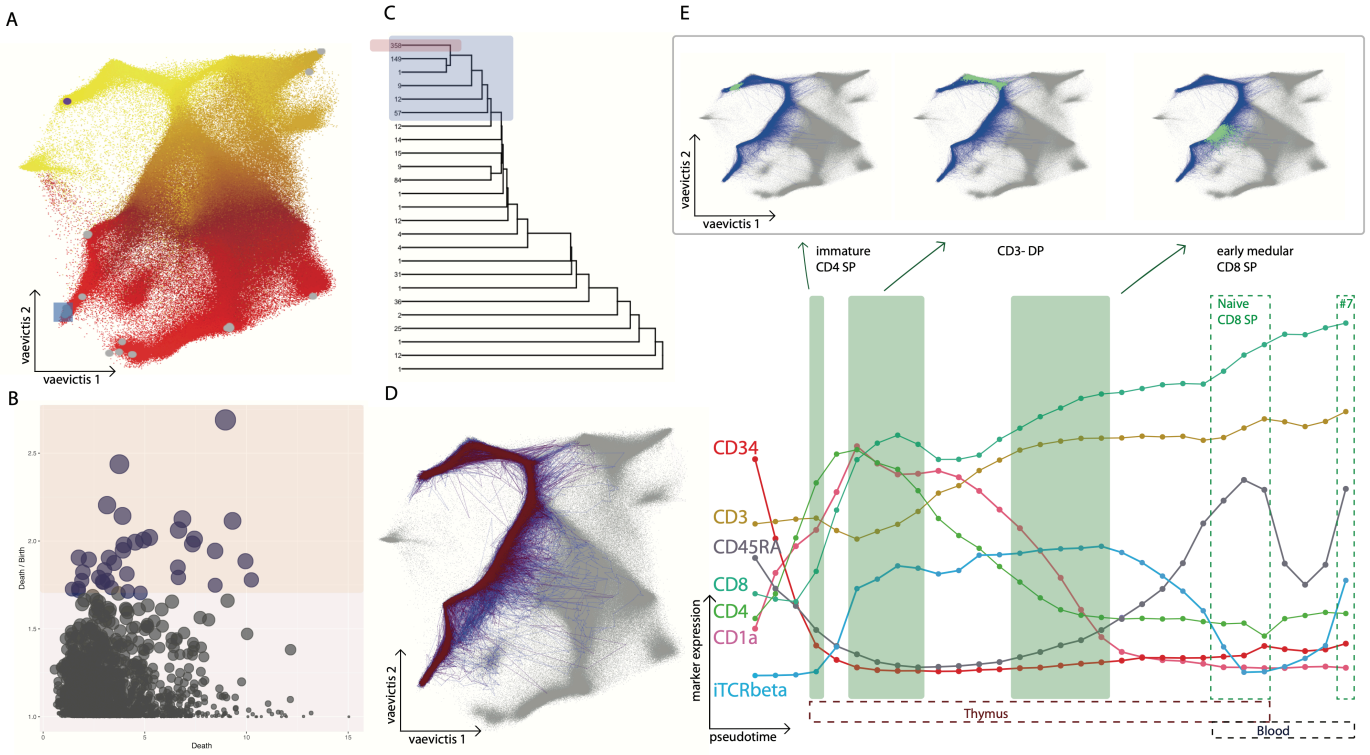


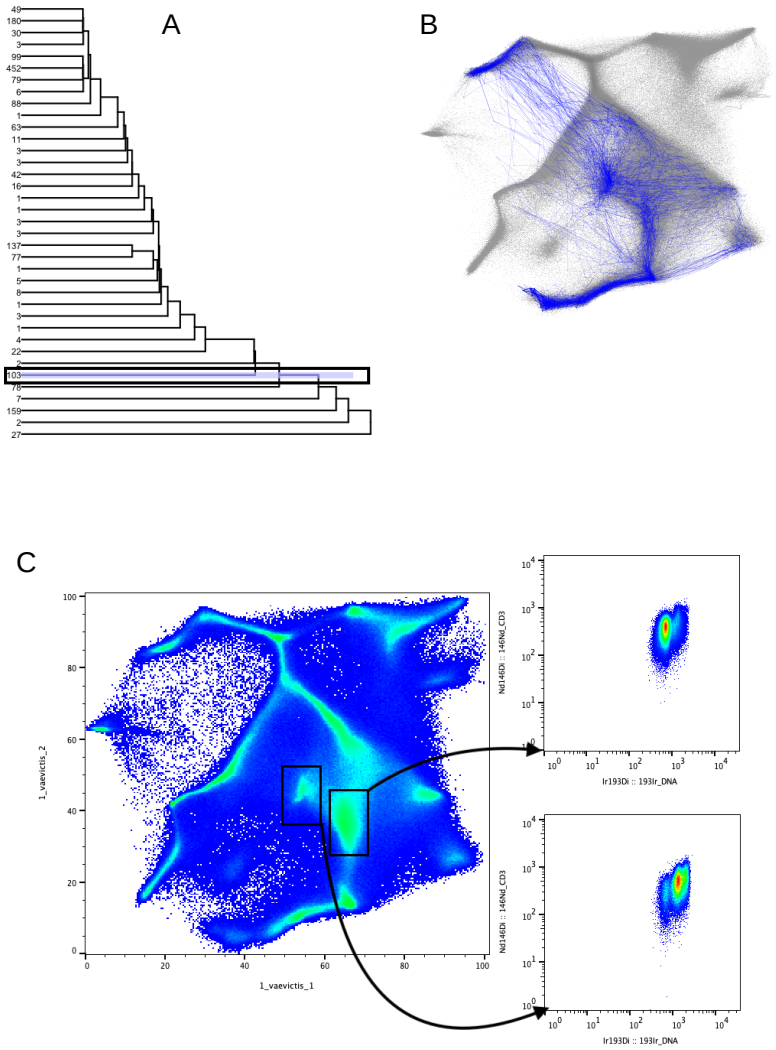
Figure 6: Detailed analysis of End#6 cells (A) Visualization of the TREC/TCRAC ratio, depicted as the calculated number of divisions for each population. Symbol code: cells from peripheral blood (blue), cells from thymus (brown), cells from adult donor (circle), cells from pediatric donor (triangle). (B) Pseudotime line plot depicting the expression of TIGIT, CD95, CD152, T-bet, CD69 and CD197 markers by cells from trajectory to End#6. (C) Overlaid histograms showing the expression of chemokine and cytokine receptors CD197, CD363, CD218 and CD196 in the respective populations. Symbol code: End#6 cells from thymus (orange), Tregs from peripheral blood (blue), immature Tregs from thymus (brown) and naive CD4 SP T cell from peripheral blood (purple).



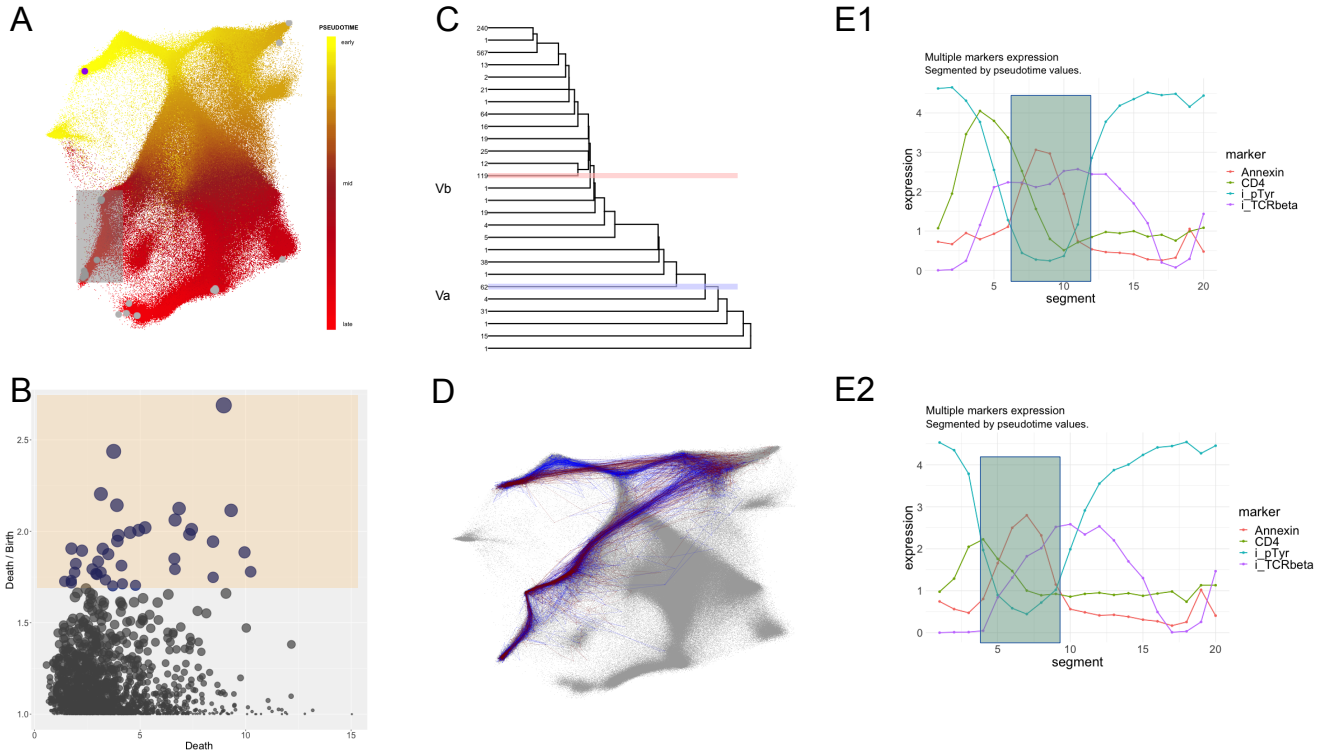
Supplementary Figure 1: Gating strategy for CyTOF data from thymus stained with Mass panel 1 After gating the nucleated cells as positive for metal-tagged DNA intercalator 191/193Ir (not shown), the thymus versus PBMC were debarcoded by gating. Next, PBMC were gated to select CD3 positive cells negative for lineage markers and TCR $\gamma\delta$. Thymocytes were gated to select live cells (platinum negative), negative for lineage markers and TCR $\gamma\delta$. After gating, the cells from thymus and PBMC were merged and used for visualization and for trajectory inference by *tviblindi*.



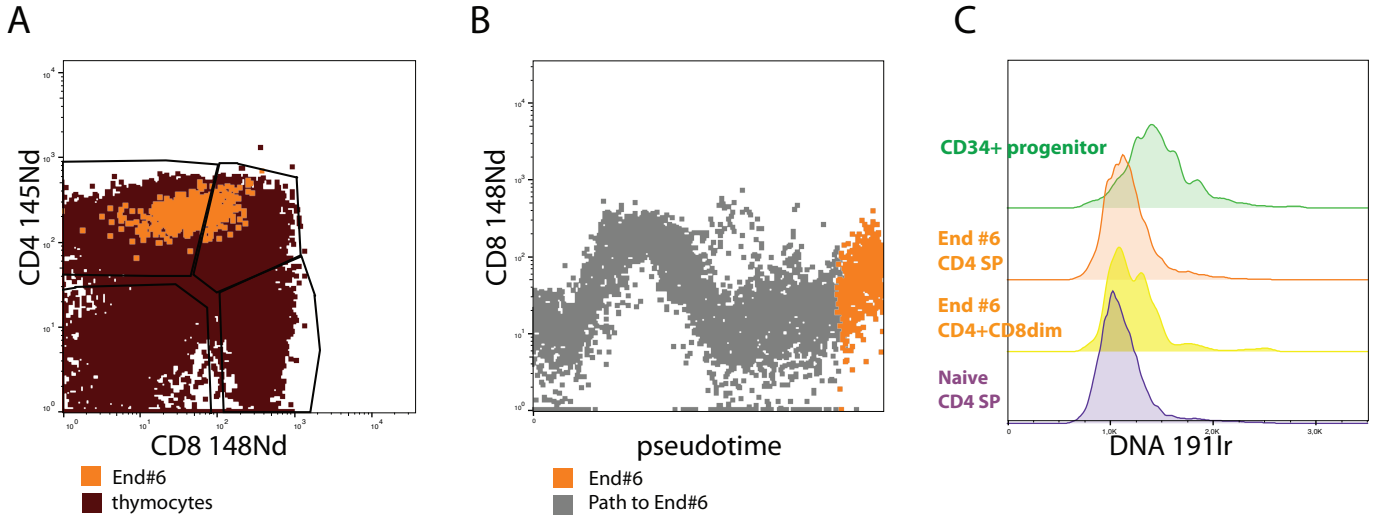
Supplementary Figure 2: *tviblindi* analysis of trajectories leading to CD8 (A) *vaevictis* plot of the T-cell developmental space (thymus and T cells from peripheral blood) where the estimated pseudotime is shown as yellow-to-red color gradient with $CD34^{pos}$ progenitors as the population of origin (purple dot). Discovered endpoints of development are shown (gray dots). Selected endpoint representing the $CD8^{pos}$ effector memory T cell (#7) is highlighted (gray rectangle). (B) Persistent homology graph where sparse regions (holes) in the point cloud of measured cells are selected (orange rectangle). (C) Dendrogram of clustered trajectories shows a larger group of random walks (blue rectangle) and a subcluster of 358 random walks (red rectangle). (D) *vaevictis* plot of the above selected trajectories shows their topology. (E) Pseudotime line plot showing the average expression of selected markers along the developmental pseudotime (bottom), selected areas of interest (green rectangles) and their position shown (green dots in insert) on the *vaevictis* plot of T-cell development.



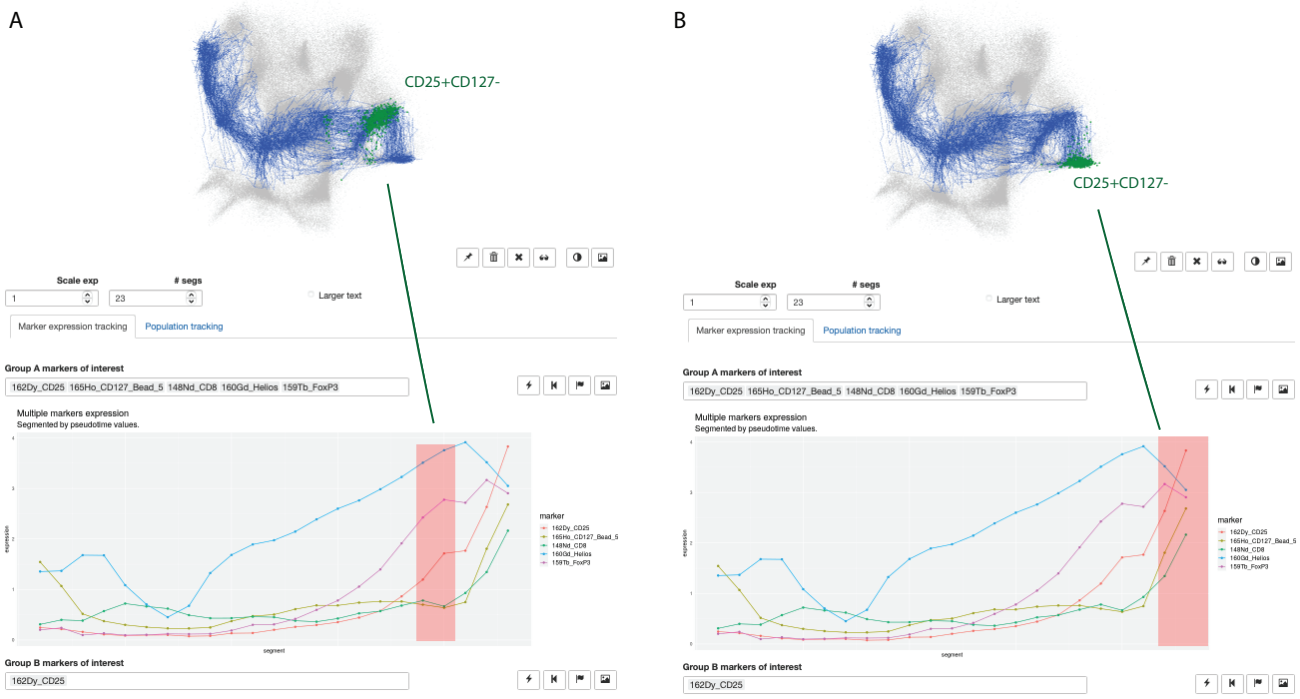
Supplementary Figure 3: Artifact caused by analysis of doublets (A) Dendrogram representing the clustered random walks leading to endpoint #1. A leaf containing 103 random walks was selected for further analysis (blue rectangle). (B) *vaevictis* plot of a trajectory showing a strange connection between immature CD4 SP and the „hub“ population. (C) Hub population has double DNA content (bottom dot plot) compared to mature CD4 SP T cells (top dot plot). The doublets are difficult to exclude from datasets containing high numbers of dividing cells (such as thymocytes).



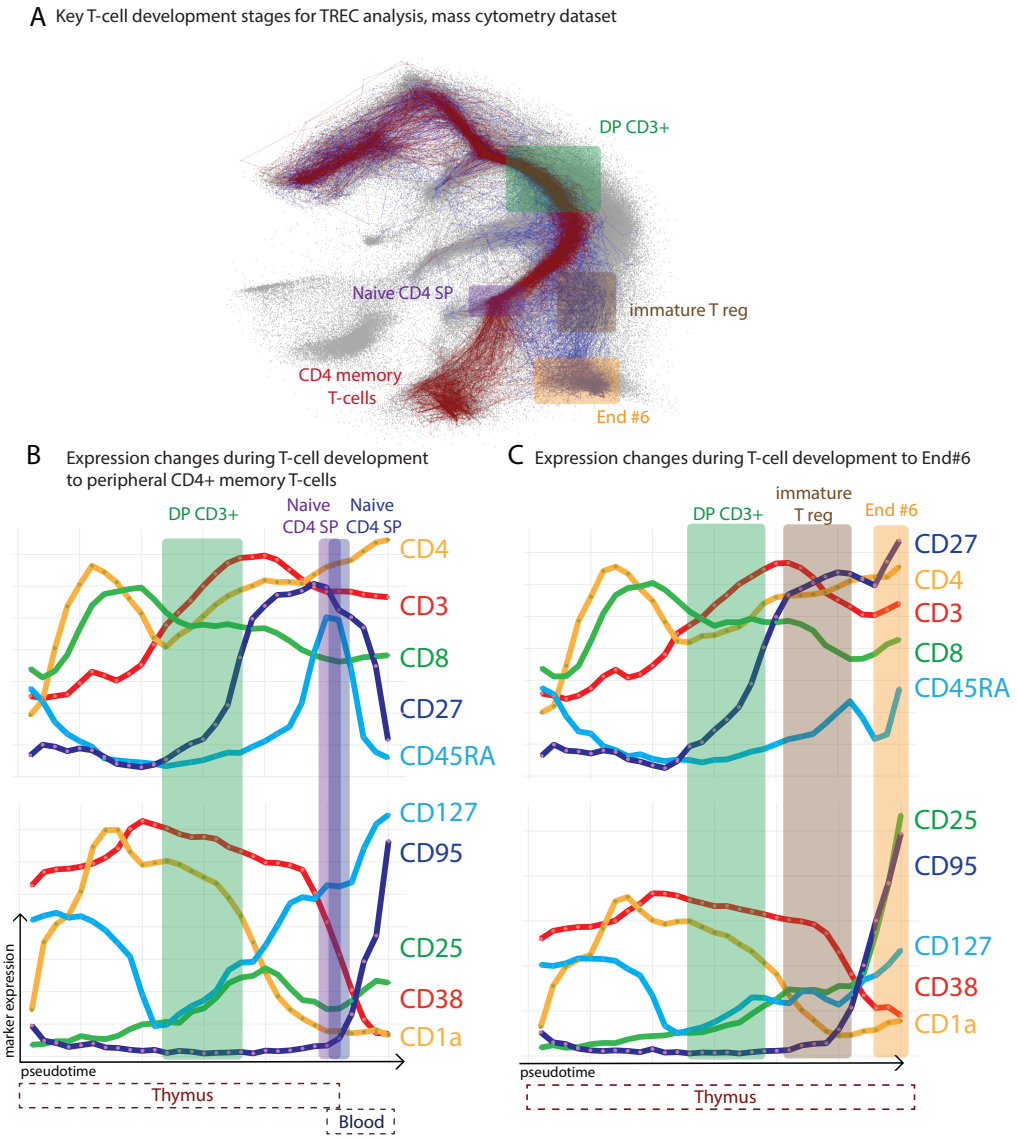
Supplementary Figure 4: Developmental checkpoints on trajectories to CD8 SP T cells (A) The same *vaevictis* plot of T-cell development in thymus and peripheral blood as seen in Figure 4A. Estimated pseudotime is represented by a yellow-to-red color gradient with $CD34^{pos}$ progenitors as the population of origin (purple dot). Gray dots indicate the discovered developmental endpoints. The gray rectangle highlights the endpoints selected for further investigation. All CD8 SP endpoints in peripheral blood (naive and memory CD8 SP) were selected to increase the number of interrogated random walks. (B) Persistent homology diagram where sparse regions in the point cloud of measured cells were selected (orange rectangle). Apart from the canonical trajectories (Supplementary Figure 5), two abundant leaves can be selected Va (blue rectangle) and Vb (red rectangle). (D) *vaevictis* plot of the selected trajectories in respective colors shows their topology. (E) Pseudotime line plots, which depict the average expression of selected markers along the trajectories Va (E1) and Vb (E2). The region of apoptosis is highlighted in green. The first part of the trajectory Va (until the apoptotic region) represents the positive/negative selection of DP cells in thymic cortex. The first part of the trajectory Vb represents β selection. The latter part of the trajectories (after the apoptotic region) represents the negative selection of CD8 SP cells in reverse direction (same situation as in the CD4 compartment described in the main text and Figure 5).



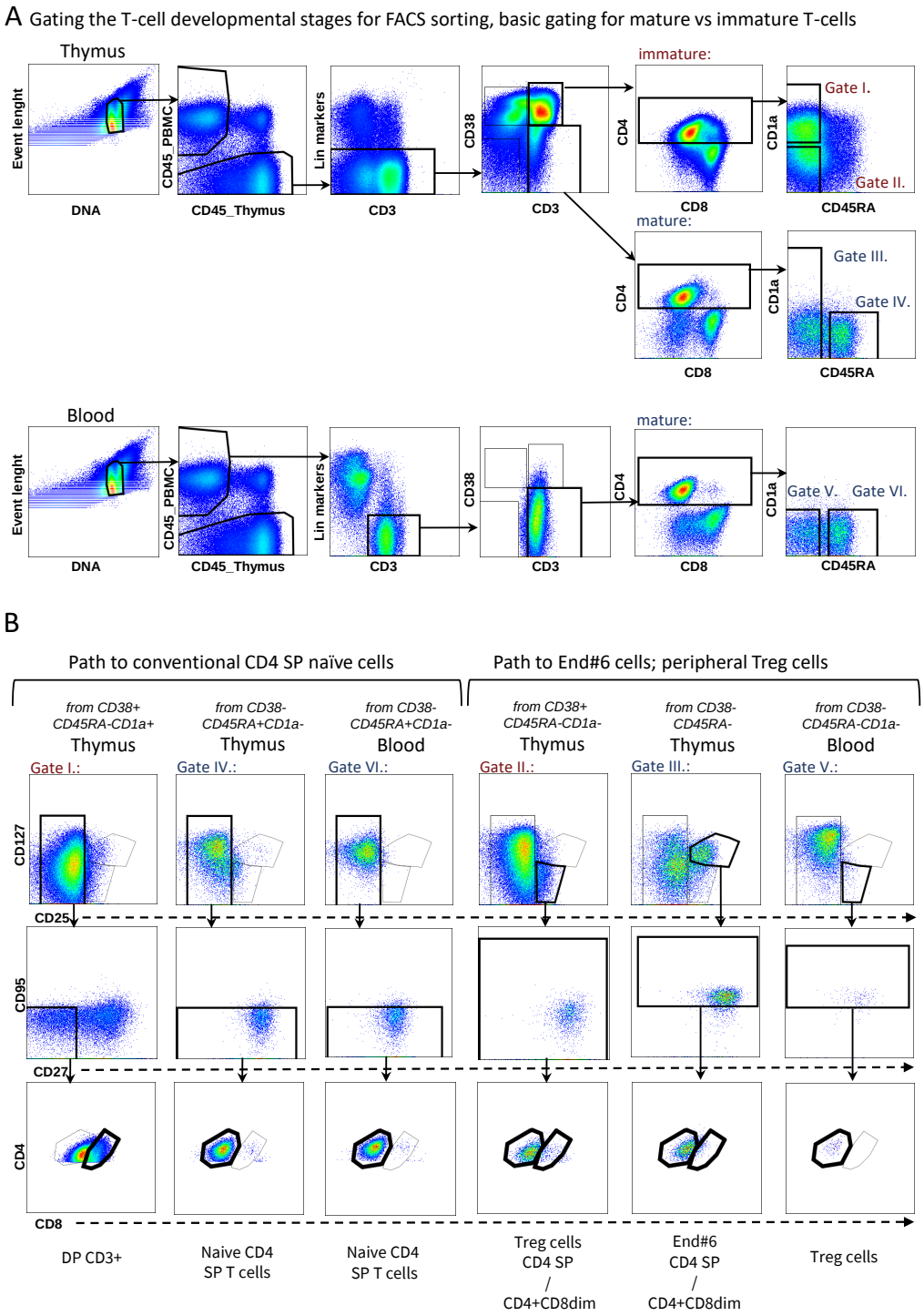
Supplementary Figure 5: Initial characterization of End#6 cells (A) Overlay of End#6 subset (highlighted in orange) dot plot over the conventional gating of thymocytes (brown background). (B) Thymocytes belonging to the trajectory to End#6, showing a late upregulation of CD8 on the pseudotime vs. CD8 dot plot. (C) Histograms of the DNA content in dividing $CD34^{pos}$ progenitors (green), End#6 CD4 SP (orange) and End#6 $CD4^{pos}$ $CD8^{dim}$ (yellow) cells compared to naive CD4 SP thymocytes (purple). Note that while events with more than one copy of DNA are present in $CD34^{pos}$ subset, other subsets do not contain high DNA content.



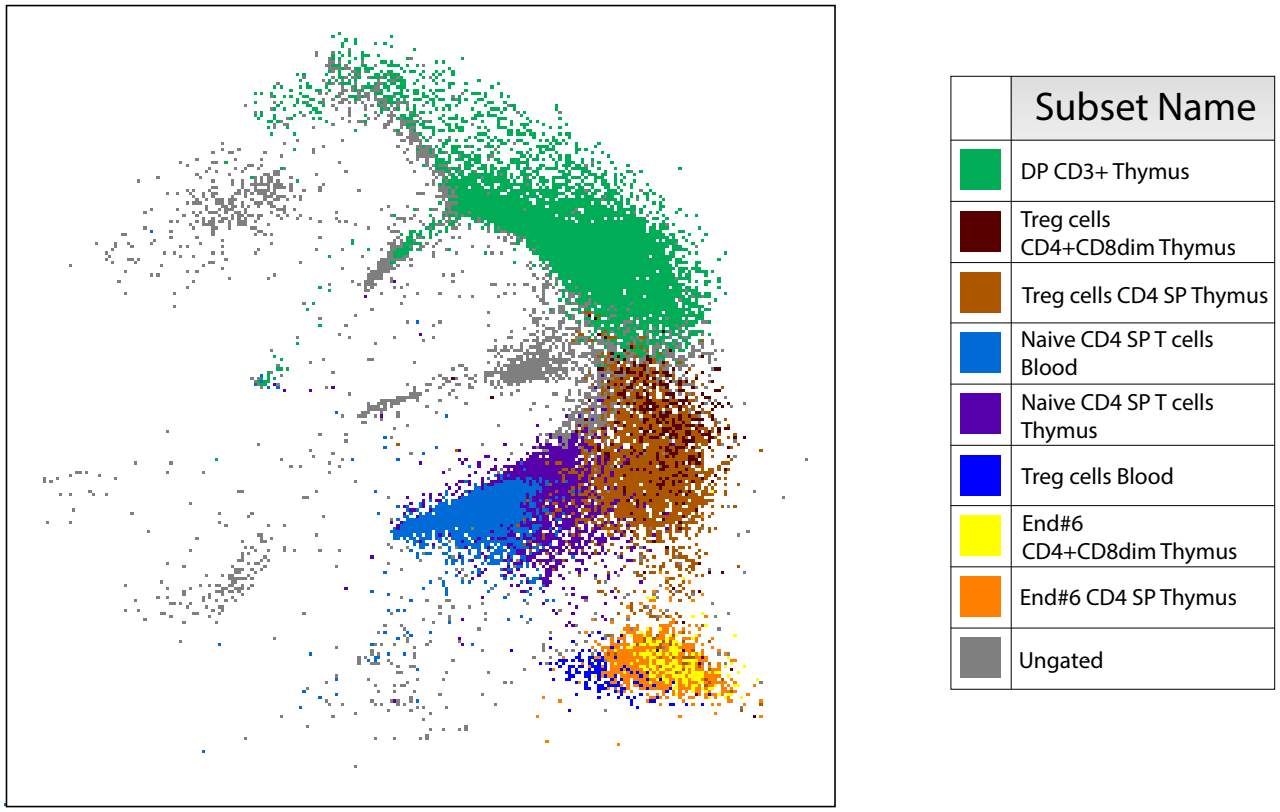
Supplementary Figure 6: *tviblindi* analysis of End#6 cells stained with Mass panel 2 The pseudotime line plots showing the dynamics of the development of CD8, Helios and FOXP3 markers in the trajectory leading to the population corresponding to End#6 in the modified panel (Mass panel 2). As in the original panel, we discovered two clearly defined populations $CD25^{pos}$ $CD127^{neg}$ (A bottom) and $CD25^{pos}$ $CD127^{pos}$ (B bottom), which can be identified based on the expression dynamics of CD25 and CD127 markers. The populations (highlighted in red rectangles) along with random walks are shown on the *vaevictis* plot above. The cell populations highlighted in green (top panels) correspond to the cells selected by the red rectangles.



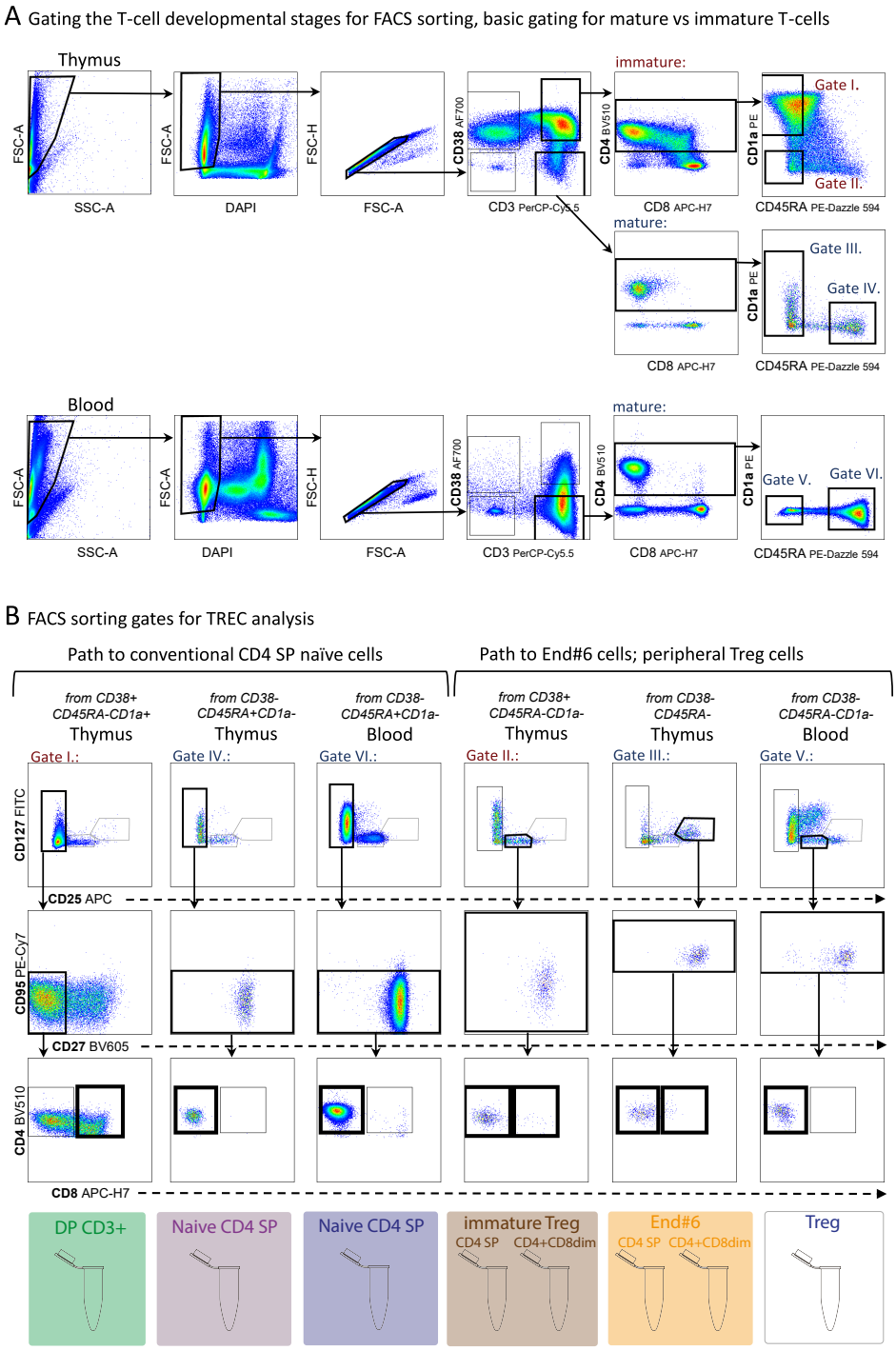
Supplementary Figure 7: *tviblindi* analysis of CyTOF data from thymus stained with Mass panel 3 (A) *vaevictis* plot of thymocytes and PBMC stained with Mass panel 3 (see Supplementary Table). Key T-cell developmental stages selected for further analysis and investigation are highlighted: CD3^{pos} DP cells (green), naive CD4 SP (purple), immature Treg (brown), End#6 cells (orange) and CD4 memory T cells (red). (B) Pseudotime line plot depicting the expression of developmental markers on the trajectory leading to conventional CD4 SP cells in thymus and peripheral blood as labeled below the plot. DP CD3^{pos} T cells are positive for CD38, CD1a and negative for CD27, CD45RA. Naive CD4 SP have low expression of CD38 and are negative for CD1a and positive for CD27 and CD45RA. (C) Pseudotime line plot depicting the expression of developmental markers on the trajectory leading via immature Tregs (brown) to End#6 cells (orange). Immature Treg cells are positive for CD38, negative for CD1a and CD45RA, positive for CD25 and negative for CD127. The End#6 cells are negative for CD38, CD45RA, positive for CD127, positive for CD95 and highly positive for CD25.



Supplementary Figure 8: Conventional gating of populations of interest stained with Mass panel 3 (A) Gating strategy of samples from thymus and peripheral blood stained with Mass panel 3 (see Supplementary Table) which was used to design a sorting panel. The surface markers, which distinguish the populations of interest (labeled as Gate I. to Gate VI.) were used, based on their expression shown in Supplementary Figure 7. Gates containing immature CD38^{pos} cells are labeled in red font, gates containing mature CD38^{neg} cells are labeled in blue font. (B) Each population (Gate I. to VI.) as defined in part A is gated further based on the CD25xCD127 expression, maturity (CD95xCD27) and CD4xCD8 expression. The populations, which were finally designated for sorting are indicated by bold frame of their respective gates.



Supplementary Figure 9: Overlay of conventionally gated populations on *vaevictis* plot *vaevictis* plot of the thymus and PBMC with an overlay of manual gates as described in Supplementary Figure 8 validating the appropriate position of the key cell populations. Each color indicates one cell population, color code to the right.



Supplementary Figure 10: Gating strategy for sorting (A) Common gating scheme of thymocytes and PBMC on FACS Aria sorter (refer to Supplementary Table). (B) Gating strategy of the key stages, which mimicked the CyTOF Mass panel 3 gating shown above in Supplementary Figure 8. Eight T-cell populations were sorted: DP CD3^{pos} (green), naive CD4 SP from thymus (purple), naive CD4 SP from peripheral blood (blue), immature Tregs CD4 SP and CD4^{pos} CD8^{dim} (brown), End#6 cells CD4 SP and CD4^{pos} CD8^{dim} from thymus (orange) and Tregs from peripheral blood (white).

Supplementary Note

Contents

1 Datasets **4**

1.1 *dyntoy* dataset 4

1.2 Human thymus & PBMC 5

2 Dimensionality reduction **5**

3 Pseudotime estimation & random walks simulation **7**

3.1 Theoretical concepts 7

3.2 On numerical aspects of the solution 11

4 Topological clustering of random walks **12**

4.1 Basic notions of simplicial homology 12

4.2 Triangulation of a point cloud & random walks 17

4.2.1 Data denoising 18

4.3 Cycle representation 19

4.4 Topological hierarchical clustering of random walks 23

5 Connectome **24**

6 Graphical user interface (GUI) **24**

7 Running the analysis **31**

8 Performance evaluation **33**

8.1 Sensitivity to hyperparameters 33

8.1.1 Number of nearest neighbors 33

8.1.2 Relaxation parameter for *witness complex* construction 33

8.2 Comparison to state-of-the-art methods 34
8.2.1 *tviblindi* on up-sampled *dyntoy* dataset 35
8.2.2 Monocle 3 36
8.2.3 Stream 40
8.2.4 Palantir 41
8.2.5 VIA 43
8.2.6 PAGA 45
8.3 Running times 48

9 Software acknowledgement **49**

Bibliography **50**

1 Datasets

In this section we give a brief description of datasets used in this supplementary note.

1.1 *dyntoy* dataset

For the purposes of basic demonstrations we created an artificial dataset using the *dyntoy* R package (Cannoodt and Saelens (2022)) as follows:

```
devtools::install_github(‘dynverse/dyntoy’)
library(dyntoy)
n_events <- 10000
n_features <- 3896 #randomly chosen value from typical range
set.seed(12345)
d <- dyntoy::generate_dataset(
  id           = ‘tviblindi_dyntoy_test’,
  model        = ‘connected’,
  num_features = n_features,
  num_cells    = n_events
)
```

Package *dyntoy* simulates single-cell expression data organized into trajectories. It creates a random milestone network (Figure 1a). The *milestones* (named *M#*) can be understood as populations of cells. We used the first 25 principal components for downstream analysis. To further highlight some issues that come up in trajectory inference, we modified this dataset by up-sampling a population labeled *M10*, by iterating over all points assigned to this population and adding random convex combinations of their 30 nearest neighbours. This resulted in a new dataset with $35,795$ points where $26,532$ points lie in

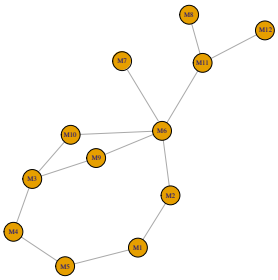
the population *M10*; we will refer to this dataset as the 'up-sampled' *dyntoy* dataset (see Figure 1d,e).

1.2 Human thymus & PBMC

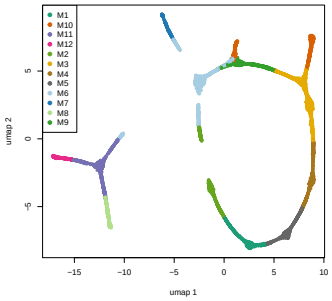
A mass cytometry dataset from human thymus and peripheral mononuclear cells from healthy donors (PBMC) was measured using *Mass panel 1.* (Supplementary Table, Mass panel 1). The analyzed dataset consists of $1,182,802$ events and 34 dimensions (markers).

2 Dimensionality reduction

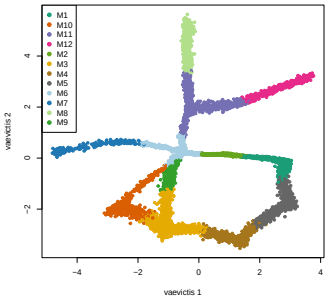
To meet the requirements for data visualization in trajectory inference (TI) problems, i.e. continuity, robustness against the differences in sizes of cell populations and straightforward extrapolation to new datasets, we implemented a variational autoencoder model based on ideas in Szubert et al. (2019); Ding et al. (2018), called *vaevictis*. The objective function is a weighted sum of reconstruction error, variational regularization (divergence from a latent prior distribution) and two regularization terms: the loss function value adapted from t-SNE and triplet loss from *ivis*. For additional speedup (which makes the time complexity virtually independent of dataset size) and improved robustness against differences in population size, the dataset is by default first clustered using k-means clustering and a fixed number of representatives are sampled from each cluster. This method is implemented in a separate GitHub repository: *https://github.com/stuchly/vaevictis*.



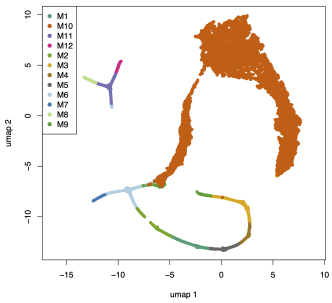
(a) Topology of *dyntoy* data



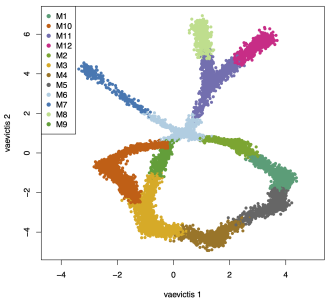
(b) 2D UMAP embedding of *dyntoy* data



(c) 2D *vaevictis* embedding of *dyntoy* data



(d) 2D UMAP embedding of up-sampled *dyntoy* data



(e) 2D *vaevictis* embedding of up-sampled *dyntoy* data

Figure 1: *dyntoy* data structure preservation with dimensionality reduction

Figure 1 illustrates the performance of *vaevictis* in producing a 2-dimensional embedding of the original and the up-sampled *dyntoy* dataset(subsection 1.1), as compared to a corresponding UMAP embedding. While the overall structure of this data (panel a) is preserved by *vaevictis* (panel c), even for data with numerically unbalanced populations (panel d; the population *M10* was enlarged to contain $\sim 74\%$ of all cells, see subsection 1.1), UMAP preserves global structure comparatively worse (panel b and d).

3 Pseudotime estimation & random walks simulation

3.1 Theoretical concepts

Let $G = (V, E)$ be k nearest neigbor graph (k-NNG) (or any suitable connected graph over the data), where V and E stand for vertices (cells) and edges of the graph respectively. We have the corresponding *distance matrix* S

$$S_{ij} = \begin{cases} d(v_i, v_j) & (v_i, v_j) \in E \\ \infty & \text{otherwise} \end{cases} \tag{1}$$

where d is a distance function (we use Euclidean distance). For modeling diffusion on G we also consider the *transition matrix* T

$$T_{ij} = \begin{cases} f(S_{ij}) & (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where f is a suitable kernel function. Our default choice is the Gaussian kernel

$$f(x) = e^{-x^2/\varepsilon},$$

where ε is determined from the distribution of vertices as follows: for an edge $(v_i, v_j) \in E$ $\varepsilon = \text{median}\{d(v_i, v_k)^2, (v_i, v_k) \in E\}$. For convenience we define the *transition probability matrix* P

$$P = D^{-1}T \tag{3}$$

where D is a diagonal matrix of row-sums of T.

We will calculate the *pseudotime* of each vertex of G in the terms of expected length of random walks until a vertex is reached.

Definition 1 (Random walk)**.** *We define a random walk (of possibly infinite length) on $G = (V, E)$ as a series $v^0, v^1, \ldots$ of vertices in G. Each v^{i+1} is chosen from the set $\{v_j : \{v_l, v_j\} \in E, v_l = v^i\}$ with probability $p_{v_j} = P_{lj}$ (P defined in Equation 3).*

If we fix the starting point v_0 of random walks we can define the *pseudotime* as one of the following quantities.

Definition 2 (Pseudotime)**.** *Let $v_0 \in V$ be fixed. We define pseudotime $\tau_{v_0}(v)$ for each $v \in V$ as either* **expected hitting time** *(the expected length of a random walk sequence starting in v_0 before it reaches v) or* **expected hitting distance** *(the expected distance traveled by a random walk starting in v_0 before it reaches v). For both definitions it holds that $\tau_{v_0}(v_0) = 0$.*

The *expected hitting time* is calculated (for $v_j \neq v_0$) as follows (e.g. in Förster et al.

(2022)):

$$\tau_{v_0}(v_j) = \sum_{v_i:\{v_i,v_j\}\in E} P_{ij}(\tau_{v_0}(v_i) + 1) \text{ if } v_j \neq v_0,\ \tau_{v_0}(v_0) = 0$$

The generalization to *expected hitting distance* is straightforward:

$$\tau_{v_0}(v_j) = \sum_{v_i:\{v_i,v_j\}\in E} P_{ij}(\tau_{v_0}(v_i) + S_{ij}) \text{ if } v_j \neq v_0,\ \tau_{v_0}(v_0) = 0$$

Rewriting this in matrix form (with components corresponding to v_0 dropped), we get

$$\boldsymbol{\tau} = P\boldsymbol{\tau} + (P \odot S)\mathbf{1}$$

where $\odot$ denotes the Hadamard (element-wise) product and $\mathbf{1}$ is a vector of ones. We can solve for τ the equation

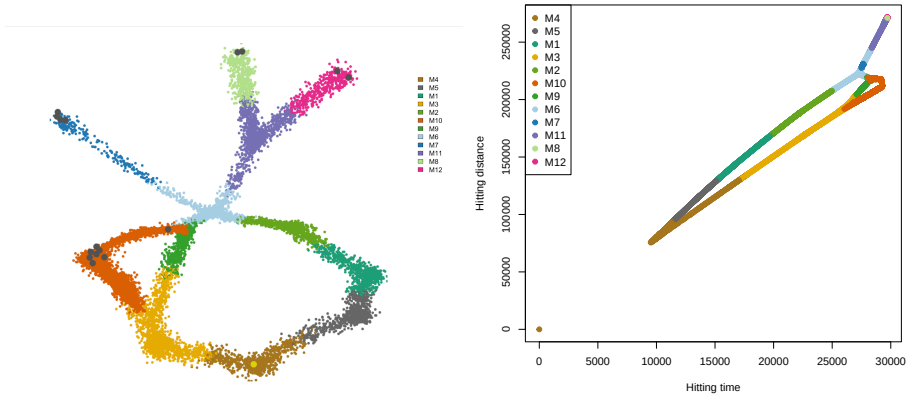
$$(I - P)\boldsymbol{\tau} = (P \odot S)\mathbf{1} \tag{4}$$

where I denotes the identity matrix. Note that if T is symmetric (which is by default enforced by setting $T_{ij} = max(T_{ij}, T_{ji})$) the Equation 4 can be rewritten as $D^{1/2}(I - P)D^{-1/2}D^{1/2}\tau = D^{1/2}(P \odot S)\mathbf{1}$, where $D^{1/2}(I - P)D^{-1/2}$ is a symmetric matrix. The Equation 4 is a sparse system which can be efficiently solved by iterative methods, such as the (bi)conjugate gradient method, even for graphs with millions of vertices. This allows to keep the resolution of pseudotime on the single-cell level.

Once pseudotime is estimated, we can use it to simulate finite random walks. We do it by creating a directed acyclic graph (DAG) $G' = (V, E')$, where $E' = \{(v_i, v_j) \in E,\ \tau(v_j) > \tau(v_i)\}$. Random walks on a DAG are necessarily finite, and their final vertices are suitable candidates of developmental endpoints. We simulate large number of walks on G' and

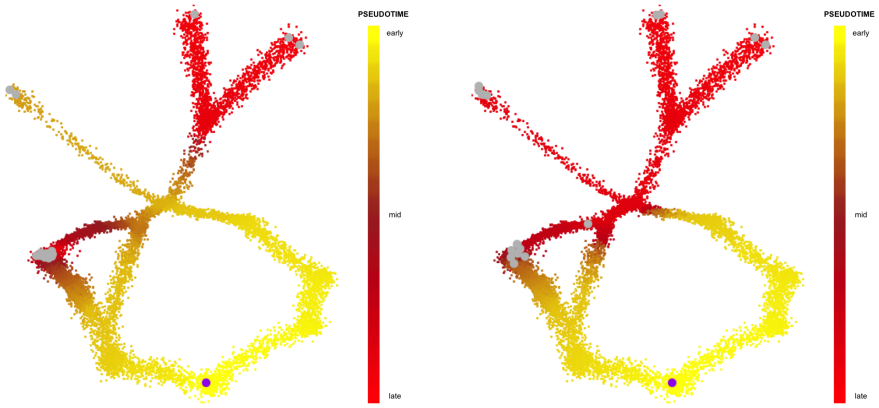
record the endpoints for further analysis (e.g. on Figure 2 in gray), where we study the possible trajectories (represented by sets of spatially coherent random walks) by which the endpoints can be reached.

The main motivation for introducing the *expected hitting distance* was our need for robust pseudotime estimation, even for the cases of significant accumulation of cells at some stages of development. The *expected hitting time* tends to rise quickly within compact, numerically large populations, and this would yield undesirable points of attraction when simulating developmental pathways in our downstream analysis. To illustrate this issue, we took the up-sampled *dyntoy* dataset (see subsection 1.1 and Figure 2) and set the point closest to the centroid of population *M4* as v_0. If pseudotime is calculated as *expected hitting time*, population *M10* forms a sink which 'traps' any random walk entering it (see the color gradient on Figure Figure 2c). In contrast, pseudotime based on *expected hitting distance* allows random walks to pass through population *M10* (Figure 2d). Upon inspection of Figure 2b, we see that passing through *M10* would be equivalent to going back in time when using *expected hitting time* formulation.



(a) *vaevictis* embedding of *dyntoy* data with marked populations

(b) Comparison of 'expected hitting time' and 'expected hitting distance' plotted for each point, colored by population



(c) Pseudotime: hitting time

(d) Pseudotime: hitting distance

Figure 2: Pseudotime calculation strategies

3.2 On numerical aspects of the solution

One of the crucial advantages of describing *pseudotime* as a solution of a linear system $Ax = b$ (Equation 4) is the possibility to measure the accuracy of our solution as the relative error $\frac{\|Ax-b\|}{\|b\|}$. Although iterative methods can solve the system (provided the

solution exists) to any precision (limited by machine precision of floating point arithmetic), in practice the desired precision might not be possible to reach (e.g. because of an ill-conditioned system).

Although this happened rarely in our experiments, it is an issue worthy of attention. In most cases, non-convergence can be rectified by changing the kernel function (for instance, Gaussian kernel seems to work better than exponential kernel on data from flow cytometry) or adjusting the transformation of the data (an *asinh* transformation is usually a sensible choice).

To summarize, the reported error of the numerical solution for pseudotime values offers an important diagnostic tool in pseudotime estimation.

4 Topological clustering of random walks

Our approach to clustering random walks is based on the idea presented in Pokorny et al. (2014). Detailed descriptions of relevant mathematical concepts and tools (algebraic topology, persistent homology) can be found in Hatcher (2000); Edelsbrunner and Harer (2022).

4.1 Basic notions of simplicial homology

For convenience, we define basic terms used throughout this document here.

Definition 3 (p-simplex)**.** *We define a p-**simplex** as a convex hull of $p+1$ affinely independent points.*
*A p-simplex σ is a point for $p = 0$, a line segment for $p = 1$, a triangle for $p = 2$, a tetrahedron for $p = 3$, etc. We use the notation $\sigma = [v_0, v_2, \ldots, v_p]$ where v_i are vertices. The convex hull of a nonempty subset of vertices is called a **face** of the simplex.*

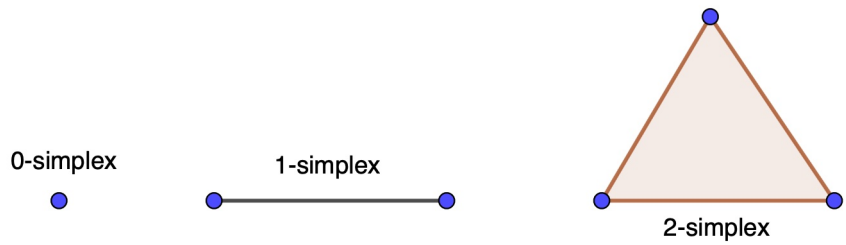


Figure 3: Simplices

Definition 4 (Simplicial complex)**.** *A* **simplicial complex** K *is a set of simplices that satisfies the following conditions:*

1. *Every face of a simplex from* K *is also in* K*.*
2. *The non-empty intersection of any two simplices* $\sigma_1, \sigma_2 \in K$ *is a face of both* σ_1 *and* σ_2*.*

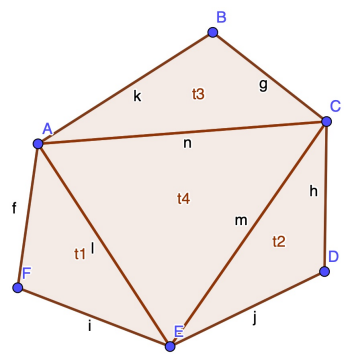


Figure 4: Simplicial complex

Definition 5 (p-chain group)**.** *For a simplicial complex* K*, the group* $C_p(K)$ *of* p**-chains** *of* K *is defined as a formal linear combination of* p*-simplices:*

$$C_p(K) = \left\{ \sum_i m_i \sigma_i | m_i \in \mathbb{Z}/2\mathbb{Z} \right\},$$

where σ_i are p-simplices of K.

Note *that the coefficients $m_i \in \mathbb{Z}/2\mathbb{Z}$. In general, these coefficients may be from any ring. This choice simplifies many things (e.g. no issues with orientation arise, as $-1 = 1$), but we lose some descriptive power.*

Definition 6 (Boundary)**.** *The* **boundary homomorphism** $\partial_p : C_p(K) \to C_{p-1}(K)$ *is defined simplex-wise for a simplex $\sigma = [v_0, v_2, ..., v_p]$ as*

$$\partial\sigma = \sum_{j=0}^{p} [v_0, \ldots, v_{j-1}, \hat{v}_j, v_{j+1}, \ldots, v_p],$$

where $\hat{v}_j$ indicates deletion of vertex v_j.

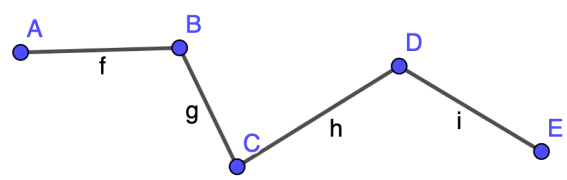


Figure 5: 1-chain $c = f + g + h + i = [A, B] + [B, C] + [C, D] + [D, E]$

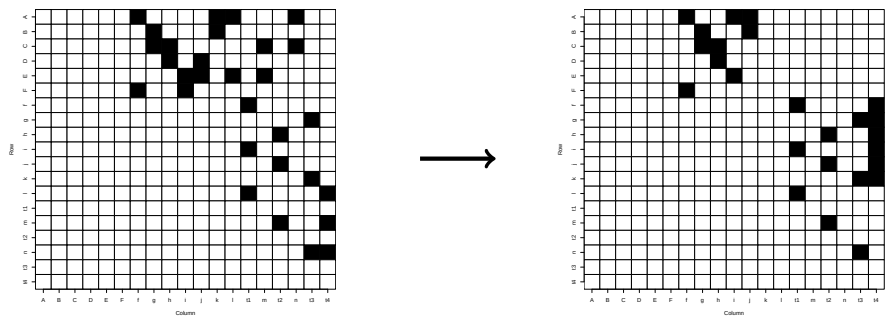
$$\begin{aligned}\partial c = \partial f + \partial g + \partial h + \partial i &= \partial[A, B] + \partial[B, C] + \partial[C, D] + \partial[D, E] \\ &= A + B + B + C + C + D + D + E = A + E\end{aligned}$$

(coefficients in $\mathbb{Z}/2\mathbb{Z}$)

The boundary operator can be expressed as a matrix

$$\partial_{i,j} = \begin{cases} 1, & \text{if } \sigma_i \text{ is a co-dimension 1 face of } \sigma_j \\ 0, & \text{otherwise} \end{cases}$$

Definition 7 (Reduced (boundary) matrix)**.** *Let R be a matrix with elements in $\mathbb{Z}/2\mathbb{Z}$ and Low(j) be the row index of the lowest 1 in the column j of the matrix R and Low(j)*



(a) Boundary matrix (b) Reduced boundary matrix

Figure 6: Boundary matrix of simplicial complex from Figure 4, 1s are marked as black squares

undefined on zero columns. We say that matrix R is **reduced** *if for any two non-zero columns i, j, $i \neq j$ $Low(i) \neq Low(j)$. See Figure 6.*

Matrix reduction can be performed by efficient algorithms and is essential for calculation of persistence (Definition 11). Matrix reduction corresponds to left-to-right column addition of columns representing boundaries of the same dimension, hence the columns of *reduced boundary matrix* correspond to boundaries of simplicial complexes (sums of boundaries of some p-simplices).

Definition 8 (Boundaries & Cycles)**.** *We call*

p-**boundaries***: the group $B_p = Im\ \partial_{p+1}$*

p-**cycles***: the group $Z_p = Ker\ \partial_p$*

p-**th homology group**[1]*: $H_p = Z_p/B_p$*

Definition 9 (Filtration)**.** **Filtration** *is a sequence of simplicial complexes that are ordered by inclusion. Hence, if K_1 and K_2 are complexes where K_1 appears before K_2 in the filtration,*

[1]Note that by fundamental lemma of homology $\partial_p \partial_{p+1} c = 0$ holds for any $p+1$-chain c and consequently $B_p \subset Z_p$.

then $K_1 \leq K_2 : K_1$ *is a sub-complex of* K_2.

By **filtration of a simplicial complex** K *we mean a sequence*

$$\emptyset = K_0 \subseteq K_1 \subseteq K_2 \subseteq \ldots \subseteq K_n = K$$

These inclusions induce a sequence of homology groups connected by homomorphisms

$$f_p^{i,j} : H_p(K_i) \to H_p(K_j),\ i \leq j$$

Filtration thus gives a sequence of homology groups

$$0 = H_p(K_0) \to H_p(K_1) \to H_p(K_2) \to \ldots \to H_p(K_n) = H_p(K).$$

Definition 10 (Filtration induced by a monotonic function)**.** *Let* K *be a simplicial complex and* $f : K \to \mathbb{R}$ *a monotonic function on* K*, that is* $f(\sigma) \leq f(\tau)$ *whenever* σ *is a face* τ*. Let* $a_1 < a_2 \ldots < a_n$ *be the values of function* f *on the simplices in* K *and* $a_0 = -\infty$*, then the sequence* $K_i = f^{-1}(-\infty, a_i]$ *is a* **filtration of** K **induced by** f*. We call* a_i **filtration values***.*

Definition 11 (Persistence and essential classes)**.** *Given a filtration,* p**-th persistent homology groups** *are defined as* $H_p^{i,j} := Im f_p^{i,j}$*, for* $0 \leq i \leq j \leq n$*. (i.e.* $H_p^{i,j} = Z_p(K_i)/(B_p(K_j) \cap Z_p(K_i))$*).*

We say that a homology class $\gamma \in H_p(K_i)$ *is* **born** *at* K_i *if* $\gamma \notin H_p^{i-1,i}$*. Furthermore we say, that a class* γ *born at* K_i **dies** *entering* K_j *if* $f_p^{i,j-1}(\gamma) \notin H_p^{i-1,j-1}$ *but* $f_p^{i,j}(\gamma) \in H_p^{i-1,j}$*.*

If the filtration is induced by a monotonic function we define the **persistence** *of class* γ

which was born at K_i and died at K_j as pers$(\gamma) = a_j - a_i$. Sometimes it is convenient to consider **index persistence** $= j - i$.

Throughout the text we will refer to the subcomplexes or corresponding filtration values where a homology class is born or where it dies as **birth** *and* **death** *of the class, respectively.*

We say that a p-cycle c represents an **essential class** *in the simplicial complex K iff there is no $(p+1)$-chain $k^{'} \in C_{p+1}(K)$ such that $c = \partial k^{'}$. In other words, essential classes are persistent homology classes which never die.*

If we consider a *reduced boundary matrix* with columns and rows ordered by filtration, then every non-zero column j corresponds to a *death* of a homology class. Moreover, this class was *born* at $K_{\mathrm{Low}(j)}$, thus the *index persistence* of this homology class is $j - \mathrm{Low}(j)$.

4.2 Triangulation of a point cloud & random walks

The triangulation of a high-dimensional point cloud is a computationally intensive task. The complexity of building triangulations used in lower-dimensional problems (such as *Delauney triangulation*) grows exponentially with the dimension, while other triangulations, better suited for high-dimensional problems (like the *Vietoris-Rips complex*) can become very large if we need sufficient cover of the space. We decided, therefore, to use the *witness complex* (Silva and Carlsson (2004)), implemented in Kachanovich (2015). *Witness complex* is built from a smaller set of landmark points and uses the original point cloud to witness the existence of connections between points. In our implementation we first cluster the points into a large number of clusters (625 by default) by k-means clustering, and use the centroids of the clusters as landmark points. Obviously, the vertices of *witness complex* differ from the vertices in G' where the random walks were simulated. We need to triangulate the random walks as well. This procedure is straightforward: we simply contract the

vertices to the corresponding cluster and remove loops from the new contracted random walks. Thus, after this *triangulation* step, each random walk corresponds to a connected 1-chain in the *witness complex*.

A witness complex can be built up to a preset dimension of simplices. Since we need only 2-simplices to classify 1-chains, we limit the dimension to 2, and effectively create a 2-skeleton. The filtration values of simplices are then calculated the same way as they are in alpha complex filtration (Rouvreau (2015)).

The choice of landmark points (clusters) can limit the resolution of the interactive aggregation of random walks. We did not encounter such an issue in practice and the simplicial complex built with default parameters always offered rich enough structure to achieve the desired level of detail. Potential issues can however be rectified by increasing the number of clusters or by introducing a custom clustering vector. Of note the triangulation of random walks takes place during the interactive session and the filtration can be recalculated independently of all other modules.

4.2.1 Data denoising

Technical noise is always present in single-cell data. The presence of unwanted data points in sparse regions which are used for characterization of the shape of our point cloud can make the use of persistent homology less effective at describing our data. In order to reduce this noise and to help capture the topology correctly, we apply the denoising technique, suggested in Carlsson et al. (2008).

Taking advantage of a previously built k-NNG, we replace the coordinates of each data point with the averaged coordinates of some l of its nearest neighbors. The denoised version of our expression matrix is used to construct the witness complex.

4.3 Cycle representation

In order to classify random walks with respect to homology classes, we need a suitable representation of their dissimilarity. The basic idea of this approach is outlined in Pokorny et al. (2014). We, however, improve on the cited results by allowing for *essential classes*, and consequently the classification of random walks with different endpoints (e.g. developmental branching).

Every random walk can be understood as a connected 1-chain with at least one vertex in its boundary (the cell-of-origin vertex v_0) common to all other random walks. Consider first a set of 1-chains $\{c^0, c^1, \ldots, c^N\}$ with a common boundary (i.e. origins and endpoints of all corresponding triangulated random walks are the same). We can now fix a reference 1-chain (e.g. c^0) and consider 1-*cycles* $\{c^0 + c^1, c^0 + c^2, \ldots, c^0 + c^N\}$. If all cycles in this set are boundaries of some 2-chain, they can be uniquely expressed as linear combinations (over $\mathbb{Z}/2\mathbb{Z}$) of columns of the *reduced boundary matrix* (as described in Pokorny et al. (2014)). However, if some of these *cycles* represent an *essential class*, we need a more elaborate solution, implemented in Algorithm 1. If we encounter a *cycle* which is not a *boundary* during the calculation of the representation, we simply form an auxiliary *cycle* representing the "essential part" and add it as another column into the *reduced boundary matrix*. The filtration value corresponding to the *death* of this class is ∞, but for the purpose of visualization on the persistent diagram we assign arbitrary values to all *essential classes*, so that they have the same *death/birth* ratio and the *death* filtration value both higher than any non-essential classes.

If we fix a reference 1-chain c^0, we can uniquely represent this way any 1-chain c^i with the same boundary by expressing the *cycle* $c^0 + c^i$ by a set of columns of (possibly updated) *reduced boundary matrix*. Note that $c^0 + c^0$ is an empty 1-chain. This representation will be used in subsection 4.4 to hierarchically cluster random walks.

What remains to be discussed is the case of classification of random walks with different endpoints. We implement two options. The first option is to simply choose the endpoint with highest *pseudotime*, set it as the endpoint of all random walks in question, and connect the other endpoint(s) to it by the shortest path within the simplicial complex. This approach is convenient in the case of closely clustered endpoints that represent the same developmental fate. The second option is to add a 1-simplex between selected endpoints. Since there were no 1-simplices connecting these endpoints (otherwise the random walks in question would end in the endpoint with the higher *pseudotime*), this will effectively create one or more new *essential classes*. Random walks differing by an *essential class* will be split into major branches at the root of the dendrogram (see subsection 4.4).

We close this subsection by a detailed description of the algorithms used and by providing a proof of correctness. Since we use the field $\mathbb{Z}_2$ as the coefficients ring, we can understand simplicial complexes (and consequently p-chains) as sets of simplices. To simplify further exposition we introduce following notation and arithmetic:

Definition 12 (Notation and arithmetic of p-chains & Low^{-1})**.** *Given a simplicial complex $K = \{\sigma_1, \sigma_2, \ldots, \sigma_N\}$ and a p-chain $c = \{\sigma_{i_1}, \sigma_{i_2}, \ldots, \sigma_{i_k}\}$, we define the* **index representation** *c_I of c as the tuple $c_I = (i_1, i_2, \ldots, i_K)$.*
For any two p-chains a_I, b_I we define (in agreement with the arithmetic over $\mathbb{Z}_2$)

$$a_I + b_I = a_I \triangle b_I \text{ (the symmetric difference of the 2 sets).}$$

By a slight abuse of notation, we will define the **index representation of a set of p-chains** *$C = \{c^1, \ldots, c^k\}$ as $C_I := (c^1 + c^2 + \ldots + c^k)_I$.*
We will also assume that the indices in the index representation tuple are always ordered with respect to the filtration in descending order, i.e. for any p-chain c, $c_I[1]$ has the highest

filtration value among all simplices forming the p-chain.

For the reduced boundary matrix R let us further define the map Low^{-1}:

$Low^{-1}(i) = j$ if i is the highest row index of a non-zero element in a column j of matrix R

and $Low^{-1}(i) = -1$ if there is no such column.

Algorithm 1 One cycle representation

Input: c_I cycle, R reduced boundary matrix with corresponding Low^{-1} function
Output: r representation (tuple of column indices) of the cycle c_I as columns of (updated) boundary matrix R, updated boundary matrix R

```
procedure GET_REPRESENTATION_OF_CYCLE(c_I,R)
    r ← ∅
    e_I ← ∅
    b_I ← ∅
    N ← number of columns of R
    while c_I ≠ ∅ do
        j ← Low^{-1}(c_I[1])
        if j>0 then
            r ← r ∪ j
            c_I ← c_I + R[,j]_I
            b_I ← b_I + R[,j]_I
            e_I ← e_I
        else
            e_I ← e_I + c_I[1]
            c_I ← c_I + c_I[1]
            r ← r
        end if
    end while
    if e_I ≠ ∅ then
        R ← [R, e_I] (adding a column with 1s at positions e_I and 0s elsewhere)
        Low^{-1} ← update Low^{-1} function with respect to the new R.
        r ← r ∪ (N + 1)
    end if
    return r, R
end procedure
```

Algorithm 2 Cycles representation

Input: $C = \{c_I^1, c_I^2, \ldots, c_I^L\}$ set of cycles, R reduced boundary matrix with corresponding Low^{-1} function

Output: Representation $= \{r^1, r^2, \ldots, r^L\}$ representations (tuples of column indices of (updated) boundary matrix R) of cycles C

```
for i ∈ 1 ... L do
    r^i, R ← GET_REPRESENTATION_OF_CYCLE(c_I^i, R)
end for
return Representation
```

Proposition 4.1: Correctness of algorithms 1 and 2

i) Algorithm 1 finds unique representation for any p-cycle c_I in maximum of $\max(c_I)$ steps in the form $c_I = \hat{b}_I + \hat{e}_I$, where $\hat{b}_I$ is a p-boundary and $\hat{e}_I$ is a p-cycle representing essential class(es).

ii) Matrix R is updated in the i-th step of algorithm 2 *iff* $\forall j : 0 \leq j < i$ the cycle $c_I^i + c_I^j$ represents an essential class ($c_I^0 := \emptyset$).

Proof. Let us denote the original reduced boundary matrix R_0 and the number of columns of R_0, N_0. Notice that for any e_I constructed by algorithm 1 there was no column with lowest non-zero element at $\max(e_I)$ before the update, hence the matrix R is always *reduced* and the columns remain linearly independent after the update(s). Consequently, every element in their span has a unique representation.

(i) Since $c_I[1] = \max(c_I)$ and if $j = \text{Low}^{-1}(c_I[1]) > 0$ then $\max(R[,j]_I) = \max(c_I)$, the value $\max(c_I)$ is reduced at least by 1 in every iteration.

By construction $\emptyset = c_I + b_I + e_I$, hence e_I constructed by algorithm 1 is a cycle. If $e_I \neq \emptyset$, the matrix R is updated by a column containing a cycle representing an essential class. All cycles not representing an essential class can be expressed as a unique sum of boundaries and hence e_I would be $\emptyset$.

Consider a representation r corresponding to the cycle c_I. Let us decompose r into

disjoint sets $r = r_b \cup r_e$, such that $\min(r_e) > N_0$, $\max(r_b) \leq N_0$ and define $\hat{e}_I := R[, r_e]_I$, $\hat{b}_I := R[, r_b]_I$.

(ii) If there exists c_I^j, $0 \leq j < i$ such that the cycle $c_I^i + c_I^j$ does not represent an essential class, then by definition there exists a p-boundary b such that $c_I^i = c_I^j + b$. Since c_I^j has a representation within R and b can be expressed as sum of columns of R_0, no matrix update is necessary. Conversely, if $\forall 0 \leq j < i$ the cycle $c_I^i + c_I^j$ represents an essential class, then c_I^i cannot be expressed in the terms of the columns of matrix R and e_I constructed by algorithm 1 is non-empty. □

4.4 Topological hierarchical clustering of random walks

Consider a set of L random walks and corresponding 1-chains $\{c^0, c^1, c^2, \ldots, c^L\}$. Let us select c^0 as a reference 1-chain (thus having trivial representation) and the representations obtained by algorithm 2 be $\{\emptyset, r^1, r^2, \ldots, r^L\}$. Each r^i corresponds to a set of homological classes each with its own *birth* and *death* filtration values. If we set a threshold on the *death* value and delete all homological classes which *died* before this point from the representation, a natural hierarchical clustering stems from the following idea. Initially, each leaf of the dendrogram contains all random walks with the same representation, then every time a homological class dies as we are increasing the threshold, two leaves/branches of the dendrogram may merge. As soon as all [2] homological classes *die*, the dendrogram is complete.

In practice, we proceed differently. We first choose (using an interactive persistence diagram) homology classes we deem significant based on their persistence. Then we order the homology classes present in any representation in decreasing order with respect to

[2] In practice we set threshold on column indices of corresponding (updated) *reduced boundary matrix*. Thus even *essential classes die* after a finite number of steps

their *death* value. Finally, we split the random walks into two branches[3] based on the presence/absence of the homology class with highest *death* value in their representation. The class with the highest *death* value is dropped and we re-iterate the process on each of the two branches.

5 Connectome

First, a multigraph is created from vertices corresponding to clusters. One directed edge is added between any pair of vertices per each traversal by a random walk. Self-loops are deleted. Then, a new graph is plotted where the edges with the same direction between same vertices are merged, setting edge weights to the number of edges before merging (by default, only the majority-direction edge is kept for any pair of vertices). Vertices of the resulting graph are plotted as pie charts of cell populations present in each cluster. The graph layout employs the same 2D representation(s) as *tviblindi* GUI to facilitate the interpretation. Terminal nodes are automatically detected as vertices with fewer outgoing than incoming edges. The widths of edges represent their weights.

Although connectome offers only a low resolution overview of the dataset, the result is informative even for very complex topologies (Figure 7). On this dataset connectome was run as `Connectome(tv1,qq=0.8)` i.e. only arrows with width above the 80% percentile were plotted.

6 Graphical user interface (GUI)

An essential part of *tviblindi* is GUI which enables the on-demand aggregation of random walks and real-time resolution adjustments. In this section we describe the main features

[3] If one branch is empty, we drop the class and re-iterate

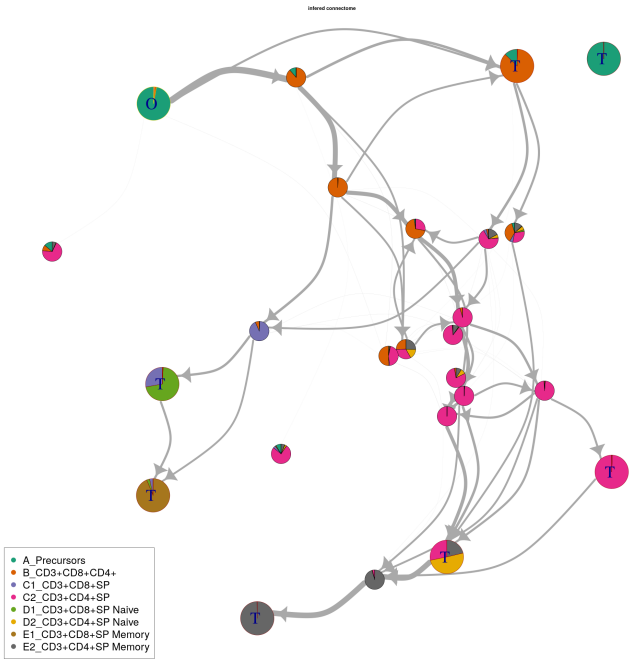


Figure 7: Connectome applied on the human thymus dataset (subsection 1.2)

of this GUI and the analysis process. (See also supplementary videos.) In this text we will refer to the annotated screenshots of the GUI on pages 29 and 30 of this document by numbers in circles, e.g. (1), and squares, e.g. [1], respectively. Please consult section 7 or the documentation of the *tviblindi* package for more details on any function mentioned in the text (function names are typeset as `verbatim`). Note that pop-up tooltips are displayed for button in the GUI when running.

The user usually starts with adjusting visualization and selecting a path model. As every run of the function `DimRed` adds a 2D layout to the *tviblindi* analysis object, we can choose the most appropriate visualization in the drop-down menu (1). Menu (2) allows the user to choose the path model (pseudotime and random walks model, for different choice of cell-of-origin, different parameter choices for *pseudotime* calculation etc.). The 2-dimensional layout plot is optimized for large datasets (millions of events), but button

(9) can increase point size when needed. Labels of points (7) serve mainly for visualizing labeled populations on the plot invoked by (8), and for tracking population frequencies along the trajectory ((21) second pane).

This first step of the analysis itself is the choice of endpoints (3). The user selects some black dots in the layout via rectangular selection at once, or iteratively by pressing (4) after each selection. The selected endpoints are listed (as row indices in the data matrix) together with the number of random walks which end in a particular endpoint. Pressing (4) moves the endpoints in the current rectangular selection (*Selected terminal nodes* frame) to *Terminal nodes marked for further analysis.*

Once the user is satisfied with the endpoints selection, they can calculate their representation by pressing (5). Checkbox (6) switches between the two options for the multiple endpoints treatment (see subsection 4.3).

At this point the endpoints of interest are selected and the representation of all random walks ending in any of these endpoints are calculated. The user can then switch to the pane *Homology classes by persistence selection* [1] and select homology classes they deem significant, based on their persistence. By default, the y axis of the diagram corresponds to the *death/birth* ratio[4]. This ratio describes the significance of the hole (sparse region) corresponding to the homology class. The larger this value, the sparser the interior of the hole relative to the density of its boundary, and the higher the significance for the random walk classification. The x axis corresponds to the *death* value of each homology class plotted: this value corresponds to the *size* of the corresponding hole, and to the height of the branching on the dendrogram [4]/(13). For the artificial data used on the screenshots, the situation is clear: there are two *essential classes*, one corresponding to the larger hole in

[4]This gives better resolution for homology classes with large differences in size than the more standard *death - birth*. This can be changed using checkbox [3], resulting in a common persistence diagram.

the data and one created by the selection of two disparate developmental fates[5]. Another significant homology class corresponds to the smaller hole in the data.

The selection process is equivalent to the selection of endpoints, and by pressing button 2 the dendrogram 4 is updated.

Using the dendrogram 13, the user can select leaves/branches to aggregate random walks into trajectories. Two trajectories can be built at the same time by selecting a trajectory group A or B 11. The selection process is the same as before: pressing the "+" button 14 adds selected walks into the selection into the active trajectory, the "fire" button resets the trajectory. Note the slider 23, which adjusts the complexity of the dendrogram by setting a threshold on the minimal fraction of random walks for a node to be shown. Nodes below this threshold are merged.

The user can adjust the level of detail by selecting significant homology classes in 1. In some cases it is more convenient to focus on a smaller subset of random walks. This can be achieved by pressing the button 17: this makes the analysis focus only on random walks in the active trajectory and the user can analyze its topology in detail, e.g. if inspecting finer branching processes. To go back to the entire set of random walks, the user can press the "fire" button 4 and start over with selection of endpoints.

Selected trajectories are visualized in a 2-dimensional embedding (layout) in pane 18. Change in signal from specific markers along a trajectory can be tracked 21. The development of those markers is shown as a line plot connecting the average values of a specific marker (or points on specific random walks, see below 5) within a pseudotime segment. By default, the pseudotime values are divided into uniform segments (each containing the same number of points) whose number can be set changed in 20. To focus on either early or late stages of the development, adjust the pseudotemporal scale using 19: the

[5]Note that *essential classes* are infinite from the point of view of homology and their *death* values are chosen arbitrarily for visualization purposes

higher the value, the more resolution is put on early stages, and vice-versa. If the user is interested in *where* in the layout a certain part of development is projected, they can select the segments of interest and press 22. If multiple markers are tracked, the points of all trajectories in the selected segments are highlighted. If only one marker is tracked 5, we can see individual random walks and highlight only points from selected walks. If the user wants to manually clean a selected trajectory based on the evolution of the signal levels from a specific marker, they can manually select some random walks and remove them from the trajectory by pressing the "flash" button near 6 "flag".

It is often useful to export the trajectories/selected points for further analysis. Highlighted points can be pinned by pressing the "pin" button 24 (the "thrash" deletes all pinned points). Trajectories can be pinned using 15. When pressing the "pin" button, the user is prompted to name the trajectory or group of points. Pinned results are stored as list slots in the *tviblindi* object or can be exported as an extended FCS file by pressing the "download" button 16 (the "trash bin" button clears pinned trajectories). Once pressed, the user is prompted to name the exported FCS file and the results are stored either in a new FCS file built from the data matrix or added to an existing FCS file (if provided: see section section 7). The following is stored in the FCS file: the active 2D embedding, pseudotime values from the active "PATH MODEL", all sets of labels, all pinned trajectories and all pinned highlighted points.

Button 11 saves the corresponding figure as either a PNG or an SVG file 12.

tviblindi

LAYOUT (1) 1_vaevictis | PATH MODEL (2) M4_hitting_distance | LABELS (7) default | POINT SIZE (9) S M L (8) | IMAGE EXPORT PNG SVG (12) | tviblindi_2022-02-19

Terminal nodes selection | Homology classes by persistence selection

(3)

PSEUDOTIME: early, mid, late

(4) (5) (8) (11)

Add 1-simplices when connecting termini (6) experimental

Selected terminal nodes

```
4575 (label: M8; 361 walks terminate here)
6546 (label: M7; 43 walks terminate here)
2916 (label: M7; 58 walks terminate here)
1388 (label: M7; 91 walks terminate here)
4058 (label: M8; 44 walks terminate here)
3797 (label: M7; 42 walks terminate here)
```

Terminal nodes marked for further analysis

```
4575 (label: M8; 361 walks terminate here)
6546 (label: M7; 43 walks terminate here)
2916 (label: M7; 58 walks terminate here)
1388 (label: M7; 91 walks terminate here)
4058 (label: M8; 44 walks terminate here)
3797 (label: M7; 42 walks terminate here)
```

(13) Whole dendrogram | Zoom

177, 71, 157, 122, 42, 70

(14) (16) (17) (11)

0 batches pinned

A | B

Min trajectory count % per leaf

(23) 0 10 100

0 10 20 30 40 50 60 70 80 90 100

Selected dendrogram nodes by counts

42

Marked pathways in A (15) N = 177

1, 6, 16, 18, 30, 33, 35, 36, 42, 43, 45, 46, 50, 51, 54, 59, 65, 66, 69, 71, 74, 78, 80, 83, 84, 85, 94, 96, 98, 103, 104, 106, 111, 113, 117, 121, 122, 124, 129, 132, 133, 134, 135, 137, 141, 143, 146, 153, 154, 156, 166, 181, 183, 184, 195, 196, 197, 199, 202, 206, 209, 210, 214, 218, 219, 220, 222, 228, 231, 239, 240, 247, 257, 258, 269, 271, 274, 275, 280, 281, 285, 287, 288, 289, 290, 293, 294, 295, 298, 301, 304, 308, 314, 322, 324, 326, 328, 337, 338, 340, 343, 344, 348, 354, 362, 364,

Marked pathways in B (15) N = 42

23, 29, 40, 41, 120, 125, 127, 138, 152, 157, 175, 193, 213, 221, 232, 245, 251, 252, 260, 268, 270, 278, 320, 330, 365, 382, 383, 400, 420, 421, 435, 457, 482, 494, 498, 511, 526, 547, 570, 593, 603, 613

(18) (22) (22) (24) (11)

Scale exp (19) 1 | # segs (20) 25

Larger text

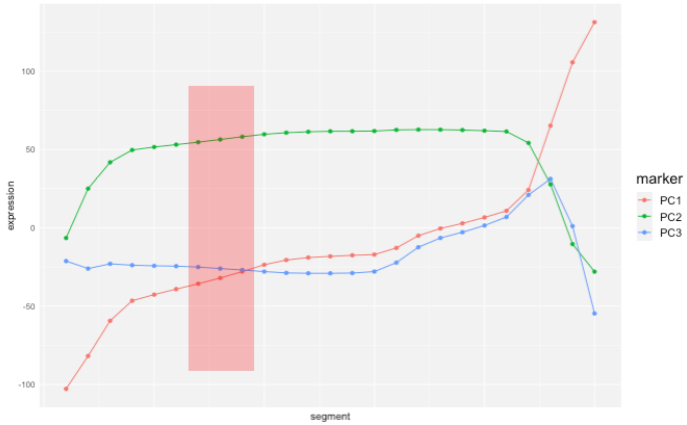
Marker expression tracking (21) | Population tracking

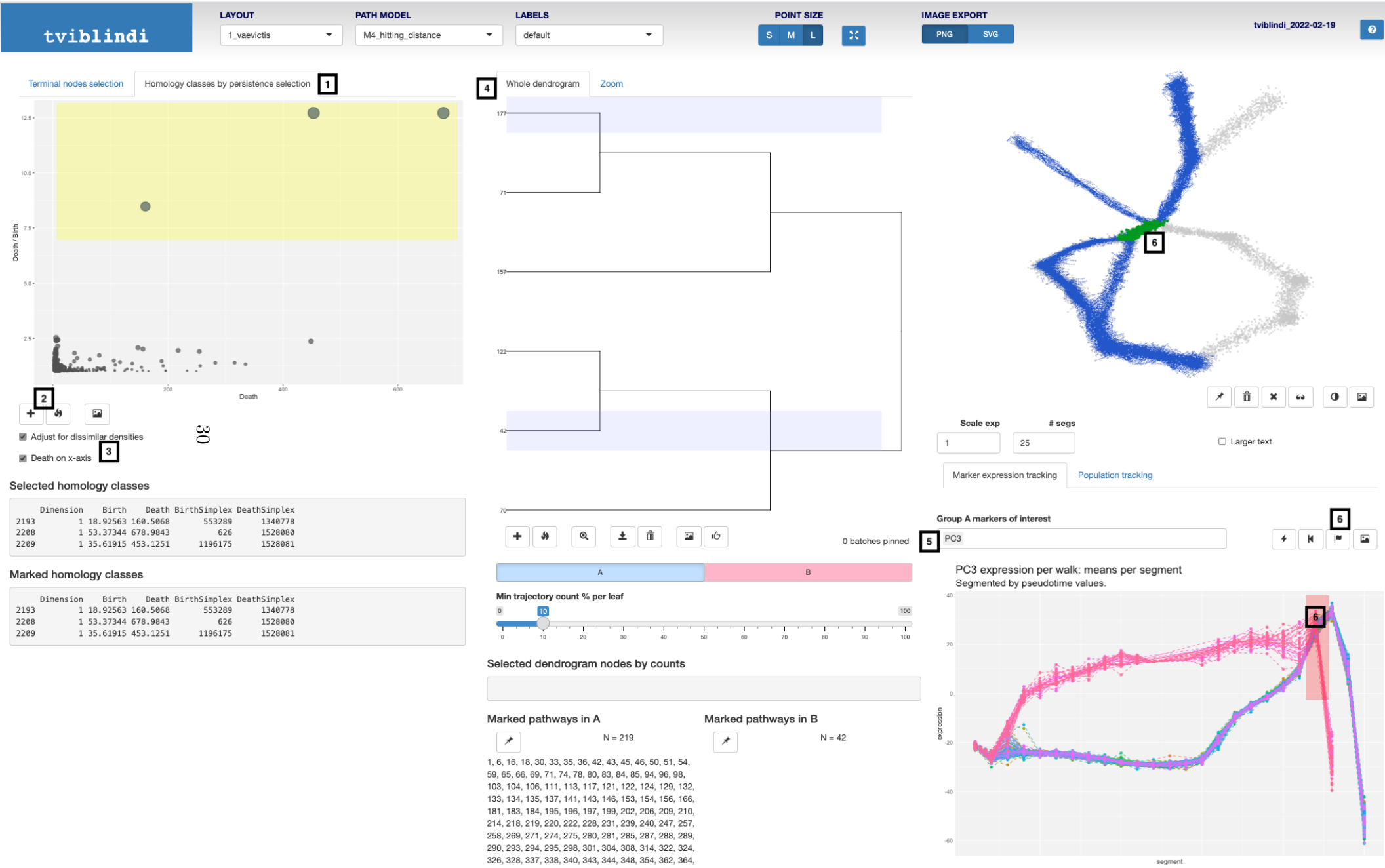
Group A markers of interest

PC1 PC2 PC3 (11) (22)

Multiple markers expression

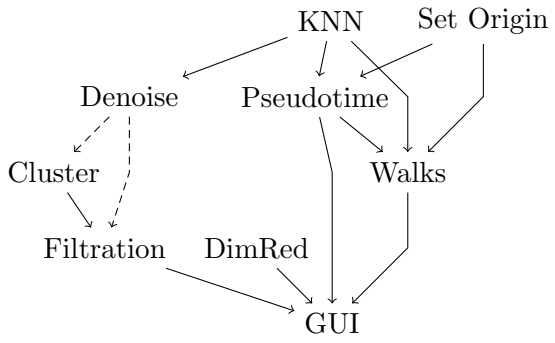
Segmented by pseudotime values.




tviblindi
LAYOUT
1_vaevictis
PATH MODEL
M4_hitting_distance
LABELS
default
POINT SIZE
S M L
IMAGE EXPORT
PNG SVG
tviblindi_2022-02-19
Terminal nodes selection
Homology classes by persistence selection
1
Death / Birth
Death
2
Adjust for dissimilar densities
Death on x-axis
3
Selected homology classes
Dimension Birth Death BirthSimplex DeathSimplex
2193 1 18.92563 160.5068 553289 1340778
2208 1 53.37344 678.9843 626 1528080
2209 1 35.61915 453.1251 1196175 1528081
Marked homology classes
Dimension Birth Death BirthSimplex DeathSimplex
2193 1 18.92563 160.5068 553289 1340778
2208 1 53.37344 678.9843 626 1528080
2209 1 35.61915 453.1251 1196175 1528081
4
Whole dendrogram
Zoom
0 batches pinned
A
B
Min trajectory count % per leaf
Selected dendrogram nodes by counts
Marked pathways in A
N = 219
1, 6, 16, 18, 30, 33, 35, 36, 42, 43, 45, 46, 50, 51, 54, 59, 65, 66, 69, 71, 74, 78, 80, 83, 84, 85, 94, 96, 98, 103, 104, 106, 111, 113, 117, 121, 122, 124, 129, 132, 133, 134, 135, 137, 141, 143, 146, 153, 154, 156, 166, 181, 183, 184, 195, 196, 197, 199, 202, 206, 209, 210, 214, 218, 219, 220, 222, 228, 231, 239, 240, 247, 257, 258, 269, 271, 274, 275, 280, 281, 285, 287, 288, 289, 290, 293, 294, 295, 298, 301, 304, 308, 314, 322, 324, 326, 328, 337, 338, 340, 343, 344, 348, 354, 362, 364,
Marked pathways in B
N = 42
6
Scale exp
segs
1
25
Larger text
Marker expression tracking
Population tracking
Group A markers of interest
PC3
5
PC3 expression per walk: means per segment
Segmented by pseudotime values.
expression
segment

7 Running the analysis

The analytical pipeline consists of several steps which are not completely interdependent and can be (re)run in a different order if necessary (see following graph of the dependence structure (a dashed connection shows that the denoising step is optional).



The *R* code needed to run a generic *tviblindi* analysis is shown below.

```
tv1 <- tviblindi(
    data = d,       # matrix of row-wise data, with named columns
    labels = l,     # vector of string labels per each row of d
    fcs = p,        # path to FCS file if d comes from its expression matrix
    events_sel = i  # indices of rows of FCS file, if only a subset is used
)
Set_origin(
    tv1,
    label = o,      # name of progenitor population
    origin_name = n # path model name (new)
)
KNN(tv1)            # build a k-NN graph
Denoise(tv1)        # reduce noise before witness complex construction
```

```
Cluster(tv1) # k-means clustering by default: k = 625 clusters
Filtration(tv1)     # build witness complex
DimRed(tv1)         # dimensionality reduction
Pseudotime(
    tv1,
    origin_name = n # path model name (existing)
)
Walks(
    tv1,
    N = 5000,       # number of finite random walks to simulate
    origin_name = n # path model name (existing)
)

launch_shiny(tv1)   # launch GUI for interactive analysis
```

All functions starting with a capital letter are generic S3 methods and the documentation could be accessed via the class suffix, e.g. `?KNN.tviblindi`. The parameter `origin_name` is optional and serves to distinguish different choices of the cell-of-origin, pseudotime calculation and random walk simulation (different path models; see section 6). The first command deserves clarification. The `fcs`, `events_sel` parameters are optional and connect the *tviblindi* object to an existing FCS file to be 'enhanced' as a part of the analysis. If omitted, the GUI will create a new FCS file out of the data matrix if the user chooses to export an FCS file. Otherwise if these parameters are provided, analysis results will be added to an existing FCS file as additional columns. If only a subset of rows from the FCS file is used for the analysis, the parameter `events_sel` records the indices of these rows as an integer vector.

8 Performance evaluation

8.1 Sensitivity to hyperparameters

8.1.1 Number of nearest neighbors

The pseudotime calculation is robust with respect to the number of nearest neighbors considered. The *Spearman correlation coefficient* between pseudotime estimated using 10 nearest neighbors and 50 nearest neighbors was 0.9986367 for the data from subsection 1.2.

The choice of the number of nearest neighbors k used for *random walks* simulation has significant effect on the number of detected endpoints. Lower k leads to higher number of endpoints, while higher k tends to smooth the topology of the graph leading to more global view. Since the *random walks* simulation does not have any effect on the results of other modules (section 7), the user can experiment with this setting very efficiently. But even when larger k causes spurious connection between unrelated population of cells, the topological clustering of *random walks* allows for the separation of different trajectories.

8.1.2 Relaxation parameter for *witness complex* construction

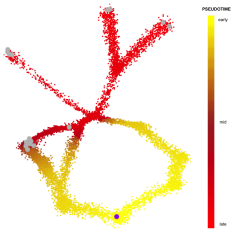
The relaxation parameter `alpha2` (function `Filtration`) controls the size of the simplicial complex. Lower values of `alpha2` lead to smaller simplicial complex which might not give sufficient resolution for the downstream analysis. On the other hand, smaller simplicial complexes lead to significant speedup.

The default setting is very conservative and creates very large simplicial complex (roughly 1-2 orders of magnitude larger than necessary, according to our tests) while maintaining feasible running times (subsection 8.3).

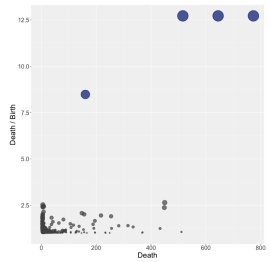
8.2 Comparison to state-of-the-art methods

We evaluate how *tviblindi* performs in comparison to selected state-of-the-art TI methods. As we focus on analysis of large single-cell datasets, we have chosen methods which scored well in previously published benchmarks, and scale well with sample size Saelens et al. (2019); Stassen et al. (2021). Namely we compared *tviblindi* to *Monocle3* (v1.3.1) Cao et al. (2019), *Stream* (v1.1) Chen et al. (2019), *Palantir* (v1.0.0) Setty et al. (2019), *VIA* (v0.1.23) Stassen et al. (2021) and *PAGA* (scanpy==1.9.3) Wolf et al. (2019). All methods were run with either default or recommended setting (when available) for the corresponding dataset. When the implementation enabled a custom embedding, the final output was projected using *vaevictis* projection used in the main text to facilitate the interpretation. For our comparison we use datasets which were thoroughly analyzed by *tviblindi* in the main text.

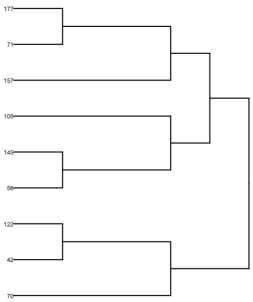
8.2.1 *tviblindi* on up-sampled *dyntoy* dataset



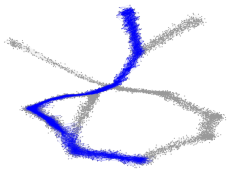
(a) Pseudotime & Termini



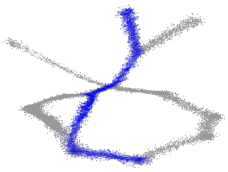
(b) Persistence diagram



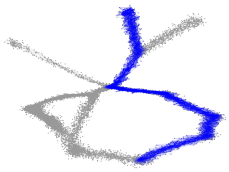
(c) Dendrogram of random walks



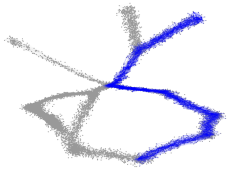
(d)



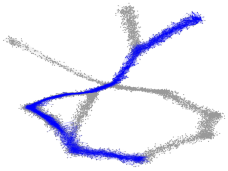
(e)



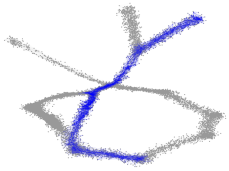
(f)



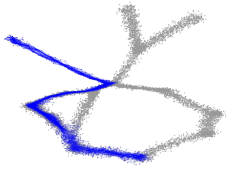
(g)



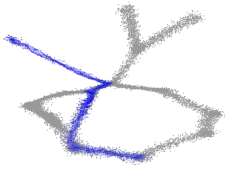
(h)



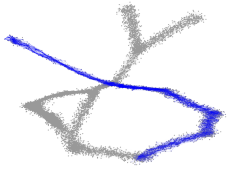
(i)



(j)



(k)



(l)

Figure 8: *tviblindi* applied on up-sampled *dyntoy* dataset

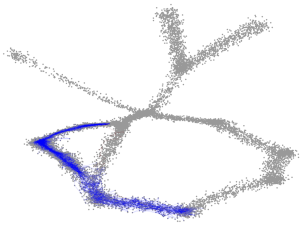


Figure 9: 4% of random walks ended in the up-sampled population

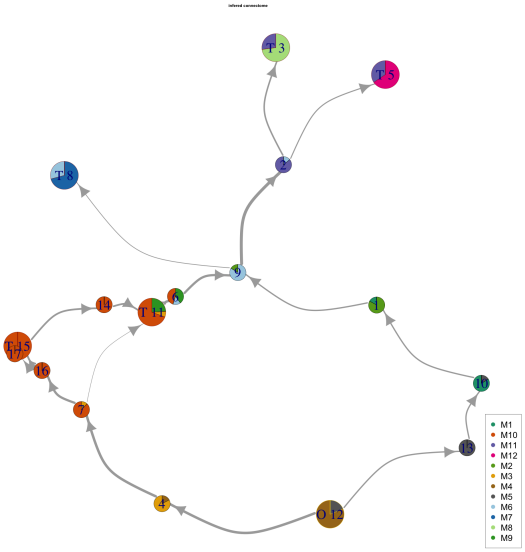


Figure 10: Connectome applied on up-sampled dyntoy dataset

8.2.2 Monocle 3

Monocle 3 relies heavily on the dimensionality reduction. The drawbacks of this approach become apparent when we apply *Monocle 3* on the artificial dataset while using the default method *UMAP* (Figure 11, Figure 12). The artifacts (e.g. over-separation of populations) introduced by the dimensionality reduction may hamper the interpretability of the result. On the other hand if we use *vaevictis*, *Monocle 3* captures the basic skeleton of dataset and misses only some connections (Figure 13). At the same time the inference of pseudotime

based on the graph representation is very robust with respect to uneven distribution of cells when using appropriate dimensionality reduction (Figure 14). When applied the biological dataset the shortcomings are more subtle. In the subsection 1.2 dataset *Monocle* does not detect β selection which is supported by low number of events. On the other hand, it over-complicates the interpretation of the dynamics within large CD4 SP and CD8 SP populations Figure 15. *Monocle 3* offers interactive interrogation of the *principal graph* and calculation of *pseudotime* along selected branches (not shown). However the resolution is set beforehand by the dimensionality reduction, clustering and graph construction and cannot be adjusted during the interactive section. In contrast to *tviblindi* which keeps the level of resolution on a single *random walk* and uses the structure of large simplicial complex for on-demand aggregation.

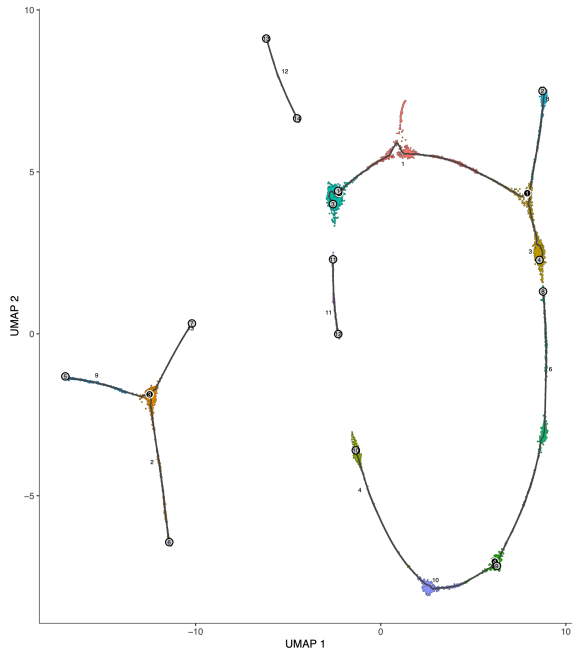


Figure 11: *Dyntoy* dataset subsection 1.1, *UMAP* dimensionality reduction

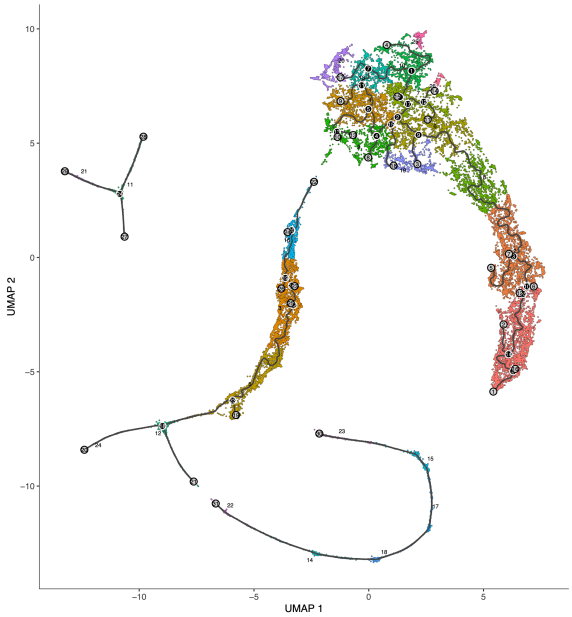


Figure 12: Up-sampled *dyntoy* dataset subsection 1.1, *UMAP* dimensionality reduction

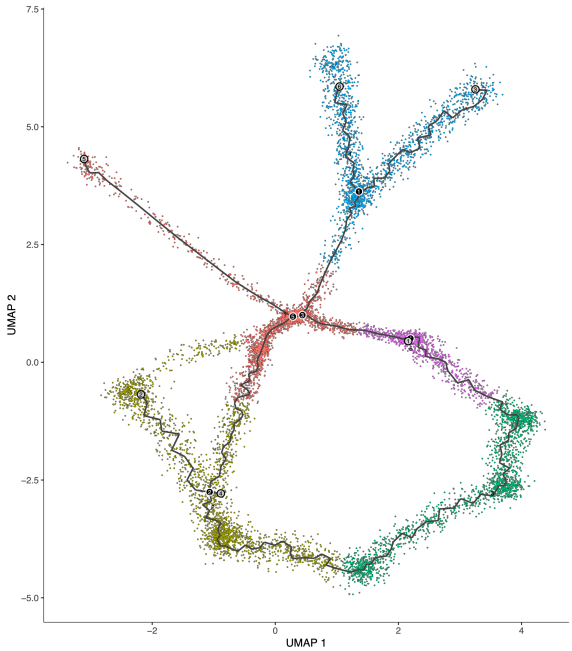


Figure 13: *Dyntoy* dataset subsection 1.1, *vaevictis* dimensionality reduction

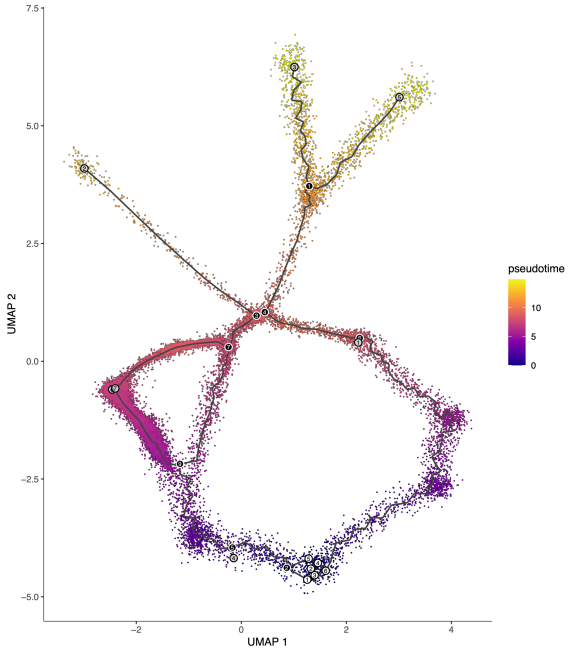


Figure 14: Up-sampled *dyntoy* dataset subsection 1.1 with inferred pseudotime, *vaevictis* dimensionality reduction

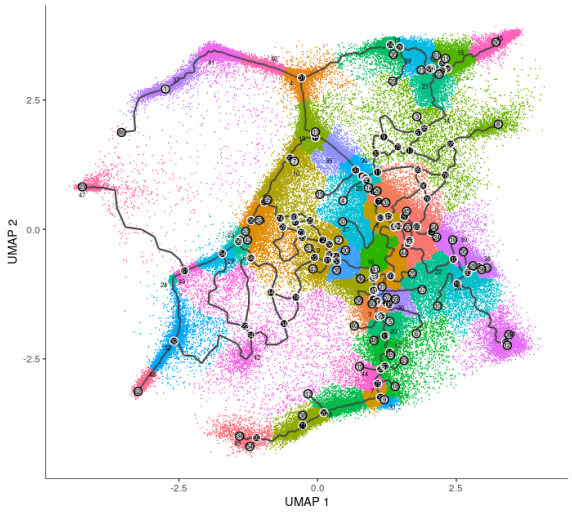


Figure 15: Human thymus & PBMC subsection 1.2, *vaevictis* dimensionality reduction

8.2.3 Stream

Since the underlying structure of the method is a *minimal spanning tree*, *Stream* is not well suited for the analysis of non-tree trajectories (Figure 16, Figure 17). *Stream* successfully discovers developmental terminal points (Progenitors, apoptotic cells and CD4 SP/CD8 SP in peripheral blood) in the human T-cell development data. However the discovered structure is too simplistic and skewed to the most abundant population (immature CD4 SP thymocytes) (Figure 18).

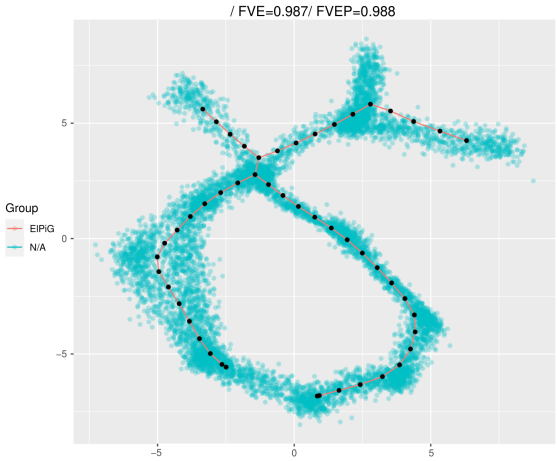


Figure 16: *Dyntoy* dataset subsection 1.1

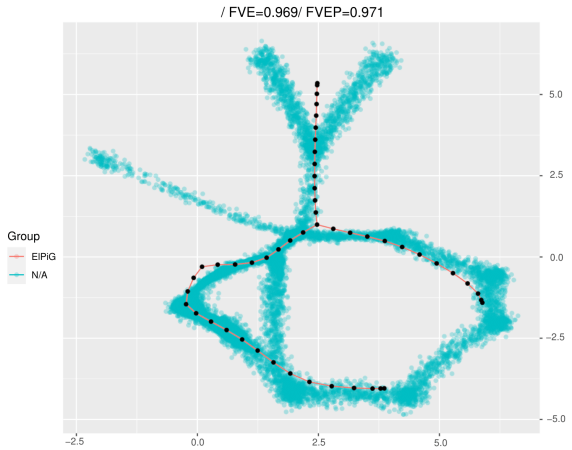


Figure 17: Up-sampled *dyntoy* dataset subsection 1.1

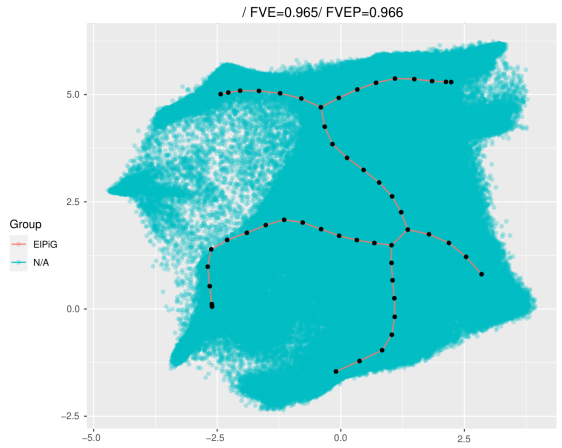


Figure 18: Human thymus & PBMC subsection 1.2

8.2.4 Palantir

Palantir is very resilient to the uneven distribution of the cells in all datasets when calculating the pseudotime. It correctly discovers candidate endpoints in the biological dataset (Figure 21) but lacks the toolbox to interrogate subtle features such as complex branching. Although the pseudotime is inferred correctly even in the artificial dataset, *palantir*

struggles to detect some of the endpoints (Figure 19, Figure 20).

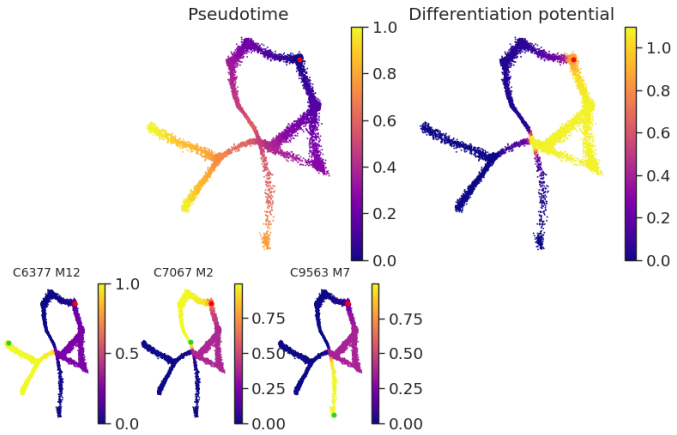


Figure 19: *Dyntoy* dataset subsection 1.1

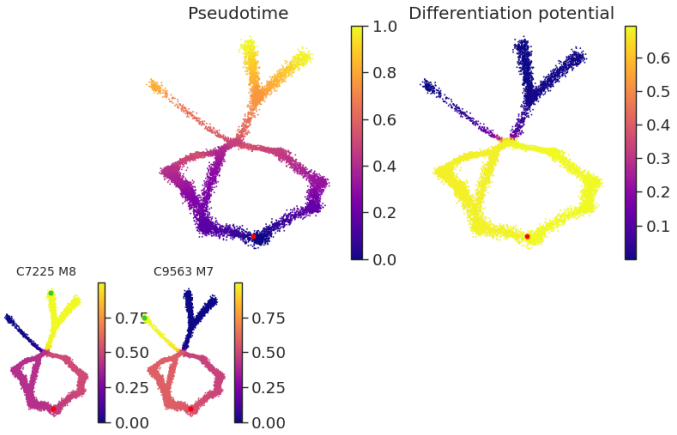


Figure 20: Up-sampled *dyntoy* dataset subsection 1.1

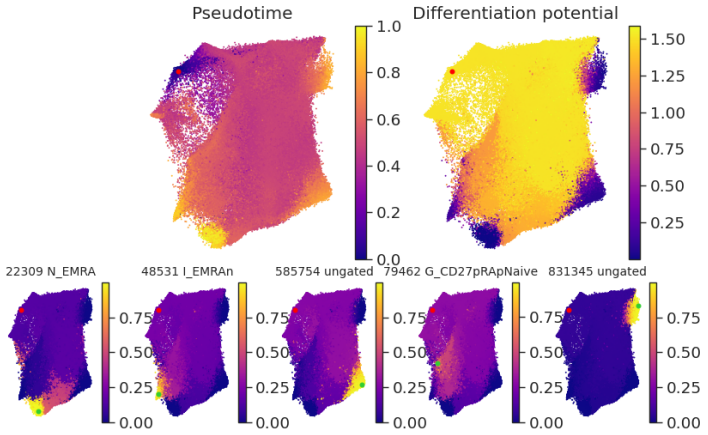


Figure 21: Human thymus & PBMC subsection 1.2

8.2.5 VIA

VIA is the only of the tested method with running times comparable to *tviblindi* (see subsection 8.3). It performs relatively well on artificial datasets (Figure 22, Figure 23), although it sometimes misses obvious connections. It is also susceptible to the aforementioned issues with unevenly distributed cells, giving the highest pseudotime value to the over-abundant population *M10* (Figure 24). In the case of the biological dataset *VIA* is able to detect candidate endpoints and some (but not all) subtle features (such as β selection; Figure 24). However the lack of an interactive interrogation or resolution adjustment of the results makes *VIA* difficult to use on complex datasets.

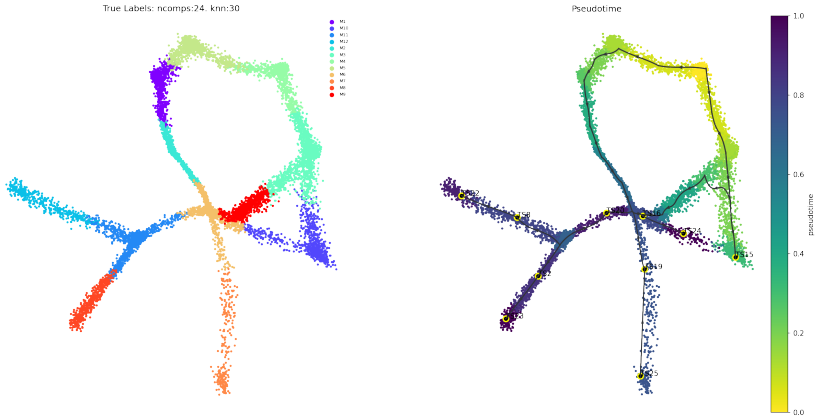


Figure 22: *Dyntoy* dataset subsection 1.1

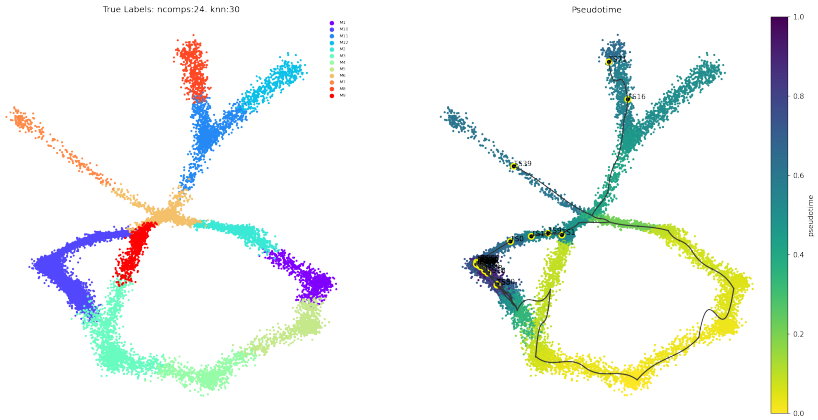


Figure 23: Up-sampled *dyntoy* dataset subsection 1.1

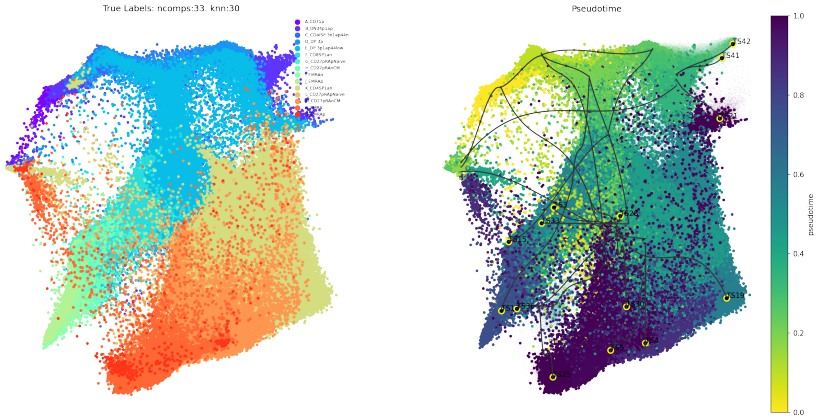


Figure 24: Human thymus & PBMC subsection 1.2

8.2.6 PAGA

PAGA performs well on the artificial dataset (Figure 25). However when applied on the up-sampled dataset, the result is surprisingly bad (Figure 26). This is especially surprising when compared with the result of the connectome tool (Figure 10). Our implementation uses the same clustering approach as *PAGA* and we used the same number of nearest neighbors in our analysis. Our approach (inspired by *PAGA*) should improve the result only by the introduction of better layout, directionality & automatic terminal node detection. However the resolution should be comparable.

In general the lack of sensible layout and the lack of terminal node detection makes *PAGA* difficult to interpret on complex datasets (Figure 27).

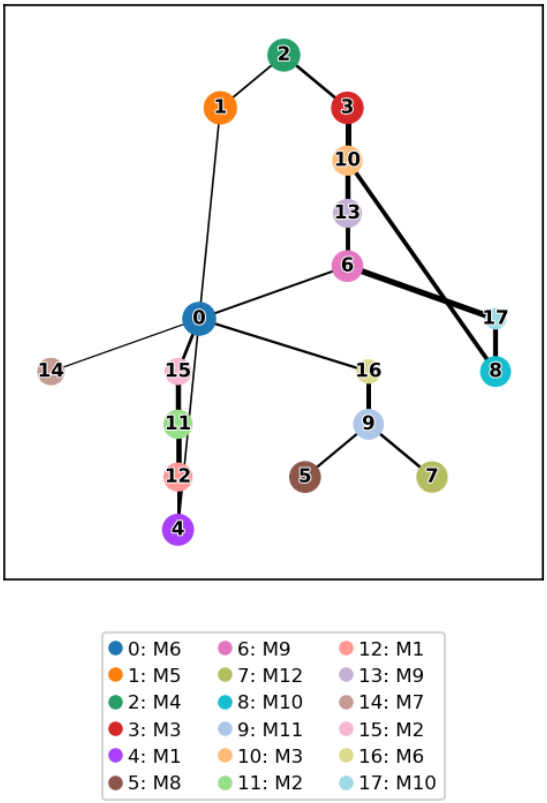
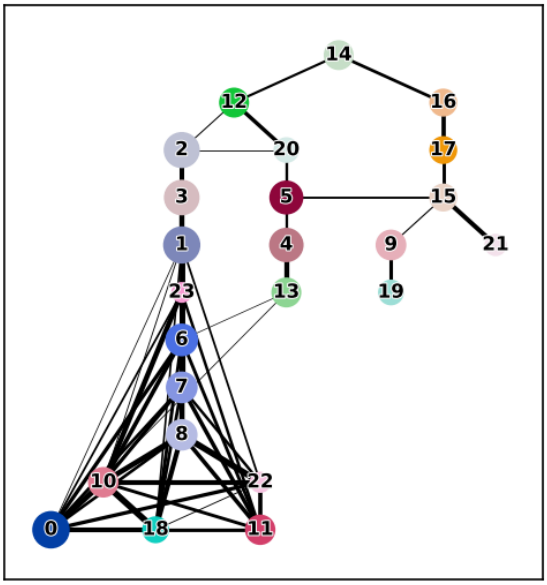


Figure 25: *Dyntoy* dataset subsection 1.1



0: M10	6: M10	12: M3	18: M10
1: M10	7: M10	13: M10	19: M8
2: M10	8: M10	14: M5	20: M9
3: M10	9: M11	15: M6	21: M7
4: M10	10: M10	16: M1	22: M10
5: M10	11: M10	17: M2	23: M10

Figure 26: Up-sampled *dyntoy* dataset subsection 1.1

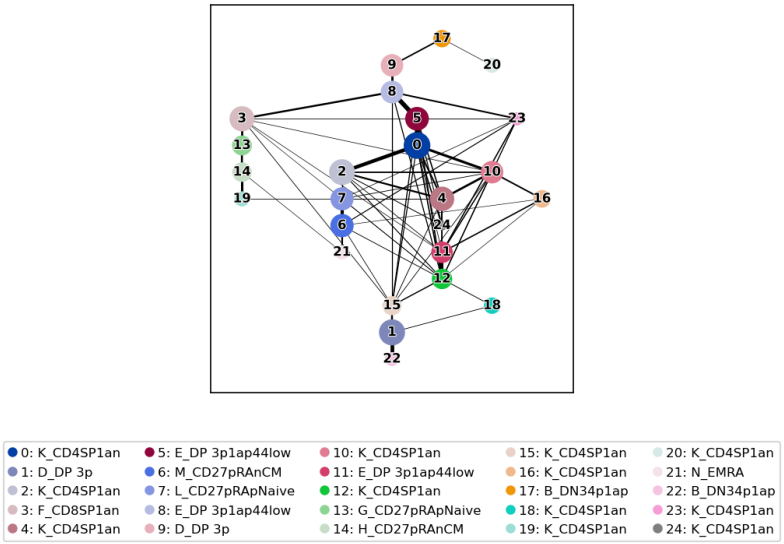


Figure 27: Human thymus & PBMC subsection 1.2

8.3 Running times

tviblindi outperforms tested TI methods on medium-sized and large datasets (Table 1) in terms of running time. All methods were run in a dedicated Docker container with 20 CPUs and RAM limited to 200 GB.

Table 1: Running times (minutes). Evaluated on the dataset from subsection 1.2.

n cells	**tviblindi**	**VIA (coarse)**	**VIA (fine)**	**Palantir**	**Monocle 3**	**STREAM**	**PAGA**
50000	0.9	0.7	2.1	3.2	4.9	10.7	0.8
100000	1.7	1.4	2.8	6.5	20.4	20.8	2
200000	4.1	3.1	6.1	14	74	32.7	6
500000	9.7	11.25	19.7	33.4	X	62	20.4
All (1,182,80 2)	22	28.1	51.8	87.9	X	136.7	46.1

The current implementation of *tviblindi* has effectively linear complexity with respect to the number of events for datasets with less then 10^7 events Figure 28. There is no clear theoretical reason for the complexity to be super-linear and we are currently focusing on the implementation to pinpoint possible overhead.

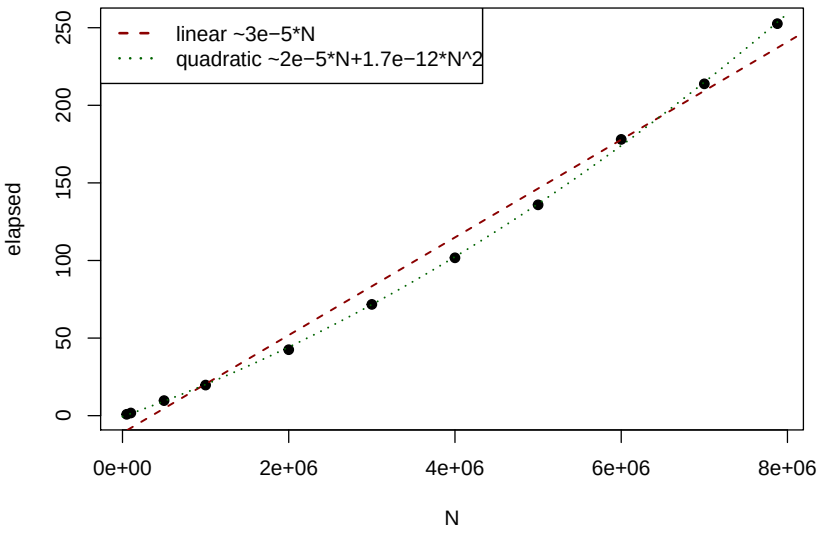


Figure 28: Running times of *tviblindi* with respect to the number of events (minutes).

9 Software acknowledgement

Apart from tools directly referenced in the text, the following tools and libraries were used in the implementation of *tviblindi*: The CGAL Project (2019); Seel (2019); Bauer et al. (2017); Bates et al. (2023); Eddelbuettel et al. (2023b); Bates and Eddelbuettel (2013); Kulichova and Kratochvil (2023); Eddelbuettel et al. (2023a); Bailey (2022); Perrier et al. (2023); Pedersen et al. (2022); Wickham et al. (2019); de Vries and Ripley (2022); Galili (2015); Wickham and Seidel (2022); Warnes et al. (2022); Urbanek (2022); Csardi and Nepusz (2006); Eddelbuettel and Sanderson (2014); Wilke (2020); Sali and Attali (2020); Maechler et al. (2022); Meyer and Perrier (2022); Melville (2023); Ellis et al. (2023); Huber et al. (2015); Ushey et al. (2023)

Bibliography

Bailey, E., 2022: *shinyBS: Twitter Bootstrap Components for Shiny*. URL https://CRAN.R-project.org/package=shinyBS, r package version 0.61.1.

Bates, D., and D. Eddelbuettel, 2013: Fast and elegant numerical linear algebra using the RcppEigen package. *Journal of Statistical Software*, **52 (5)**, 1–24, doi:10.18637/jss.v052.i05.

Bates, D., M. Maechler, and M. Jagan, 2023: *Matrix: Sparse and Dense Matrix Classes and Methods*. URL https://CRAN.R-project.org/package=Matrix, r package version 1.6-0.

Bauer, U., M. Kerber, J. Reininghaus, and H. Wagner, 2017: Phat - persistent homology algorithms toolbox. *J. Symb. Comput.*, **78**, 76–90, URL http://dblp.uni-trier.de/db/journals/jsc/jsc78.html#BauerKRW17.

Cannoodt, R., and W. Saelens, 2022: *dyntoy: Generating simple toy data of cellular differentiation*. R package version 0.9.9.

Cao, J., and Coauthors, 2019: The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, **566 (7745)**, 496–502.

Carlsson, G., T. Ishkhanov, V. De Silva, and A. Zomorodian, 2008: On the local behavior of spaces of natural images. *International Journal of Computer Vision*, **76 (1)**, 1–12, doi:10.1007/s11263-007-0056-x.

Chen, H., and Coauthors, 2019: Single-cell trajectories reconstruction, exploration and mapping of omics data with STREAM. *Nat. Commun.*, **10 (1)**, 1903.

Csardi, G., and T. Nepusz, 2006: The igraph software package for complex network research. *InterJournal*, **Complex Systems**, 1695, URL https://igraph.org.

de Vries, A., and B. D. Ripley, 2022: *ggdendro: Create Dendrograms and Tree Diagrams Using 'ggplot2'*. URL https://CRAN.R-project.org/package=ggdendro, r package version 0.1.23.

Ding, J., A. Condon, and S. P. Shah, 2018: Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. *Nature communications*, **9 (1)**, 1–13.

Eddelbuettel, D., J. W. Emerson, and M. J. Kane, 2023a: *BH: Boost C++ Header Files*. URL https://CRAN.R-project.org/package=BH, r package version 1.81.0-1.

Eddelbuettel, D., and C. Sanderson, 2014: Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis*, **71**, 1054–1063, doi:10.1016/j.csda.2013.02.005.

Eddelbuettel, D., and Coauthors, 2023b: *Rcpp: Seamless R and C++ Integration*. URL https://CRAN.R-project.org/package=Rcpp, r package version 1.0.11.

Edelsbrunner, H., and J. L. Harer, 2022: *Computational topology: an introduction*. American Mathematical Society.

Ellis, B., P. Haaland, F. Hahne, N. Le Meur, N. Gopalakrishnan, J. Spidlen, M. Jiang, and G. Finak, 2023: *flowCore: flowCore: Basic structures for flow cytometry data*. doi:10.18129/B9.bioc.flowCore, URL https://bioconductor.org/packages/flowCore, r package version 2.12.0.

Förster, Y.-P., L. Gamberi, E. Tzanis, P. Vivo, and A. Annibale, 2022: Exact and approximate mean first passage times on trees and other necklace structures: a local equilibrium approach. *Journal of Physics A: Mathematical and Theoretical*, **55 (11)**, 115 001, doi:10.1088/1751-8121/ac4ece, URL https://dx.doi.org/10.1088/1751-8121/ac4ece.

Galili, T., 2015: dendextend: an r package for visualizing, adjusting, and comparing trees of hierarchical clustering. *Bioinformatics*, doi:10.1093/bioinformatics/btv428, URL https://academic.oup.com/bioinformatics/article/31/22/3718/240978/dendextend-an-R-package-for-visualizing-adjusting, https://academic.oup.com/bioinformatics/article-pdf/31/22/3718/17122682/btv428.pdf.

Hatcher, A., 2000: *Algebraic topology*. Cambridge Univ. Press, Cambridge, URL https://cds.cern.ch/record/478079.

Huber, and Coauthors, 2015: Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, **12 (2)**, 115–121, URL http://www.nature.com/nmeth/journal/v12/n2/full/nmeth.3252.html.

Kachanovich, S., 2015: Witness complex. *GUDHI User and Reference Manual*, GUDHI Editorial Board, URL http://gudhi.gforge.inria.fr/doc/latest/group__witness__complex.html.

Kulichova, T., and M. Kratochvil, 2023: *scattermore: Scatterplots with More Points*. R package version 1.2.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik, 2022: *cluster: Cluster Analysis Basics and Extensions*. URL https://CRAN.R-project.org/package=cluster, r package version 2.1.4 — For new features, see the 'Changelog' file (in the package source).

Melville, J., 2023: *uwot: The Uniform Manifold Approximation and Projection (UMAP) Method for Dimensionality Reduction*. URL https://CRAN.R-project.org/package=uwot, r package version 0.1.16.

Meyer, F., and V. Perrier, 2022: *shinybusy: Busy Indicators and Notifications for 'Shiny'*

Applications. URL https://CRAN.R-project.org/package=shinybusy, r package version 0.3.1.

Pedersen, T. L., V. Nijs, T. Schaffner, and E. Nantz, 2022: *shinyFiles: A Server-Side File System Viewer for Shiny*. URL https://CRAN.R-project.org/package=shinyFiles, r package version 0.9.3.

Perrier, V., F. Meyer, and D. Granjon, 2023: *shinyWidgets: Custom Inputs Widgets for Shiny*. URL https://CRAN.R-project.org/package=shinyWidgets, r package version 0.7.6.

Pokorny, F. T., M. Hawasly, and S. Ramamoorthy, 2014: Multiscale Topological Trajectory Classification with Persistent Homology. *Robotics: Science and Systems*, doi:10.15607/rss.2014.x.054.

Rouvreau, V., 2015: Alpha complex. *GUDHI User and Reference Manual*, GUDHI Editorial Board, URL http://gudhi.gforge.inria.fr/doc/latest/group_alpha_complex.html.

Saelens, W., R. Cannoodt, H. Todorov, and Y. Saeys, 2019: A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.*, **37 (5)**, 547–554.

Sali, A., and D. Attali, 2020: *shinycssloaders: Add Loading Animations to a 'shiny' Output While It's Recalculating*. URL https://CRAN.R-project.org/package=shinycssloaders, r package version 1.0.0.

Seel, M., 2019: dD geometry kernel. *CGAL User and Reference Manual*, 4.14 ed., CGAL Editorial Board, URL https://doc.cgal.org/4.14/Manual/packages.html#PkgKernelD.

Setty, M., V. Kiseliovas, J. Levine, A. Gayoso, L. Mazutis, and D. Pe'er, 2019: Characterization of cell fate probabilities in single-cell data with palantir. *Nat. Biotechnol.*, **37 (4)**, 451–460.

Silva, V. d., and G. Carlsson, 2004: Topological estimation using witness complexes. *SPBG'04 Symposium on Point - Based Graphics 2004*, M. Gross, H. Pfister, M. Alexa, and S. Rusinkiewicz, Eds., The Eurographics Association, doi:10.2312/SPBG/SPBG04/157-166.

Stassen, S. V., G. G. K. Yip, K. K. Y. Wong, J. W. K. Ho, and K. K. Tsia, 2021: Generalized and scalable trajectory inference in single-cell omics data with VIA. *Nat. Commun.*, **12 (1)**, 5528.

Szubert, B., J. E. Cole, C. Monaco, and I. Drozdov, 2019: Structure-preserving visualisation of high dimensional single-cell datasets. *Scientific reports*, **9 (1)**, 1–10.

The CGAL Project, 2019: *CGAL User and Reference Manual*. 4.14 ed., CGAL Editorial Board, URL https://doc.cgal.org/4.14/Manual/packages.html.

Urbanek, S., 2022: *jpeg: Read and write JPEG images*. URL https://CRAN.R-project.org/package=jpeg, r package version 0.1-10.

Ushey, K., J. Allaire, and Y. Tang, 2023: *reticulate: Interface to 'Python'*. URL https://CRAN.R-project.org/package=reticulate, r package version 1.30.

Warnes, G. R., and Coauthors, 2022: *gplots: Various R Programming Tools for Plotting Data*. URL https://CRAN.R-project.org/package=gplots, r package version 3.1.3.

Wickham, H., and D. Seidel, 2022: *scales: Scale Functions for Visualization*. URL https://CRAN.R-project.org/package=scales, r package version 1.2.1.

Wickham, H., and Coauthors, 2019: Welcome to the tidyverse. *Journal of Open Source Software*, **4 (43)**, 1686, doi:10.21105/joss.01686.

Wilke, C. O., 2020: *cowplot: Streamlined Plot Theme and Plot Annotations for 'ggplot2'*. URL https://CRAN.R-project.org/package=cowplot, r package version 1.1.1.

Wolf, F. A., and Coauthors, 2019: PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.*, **20 (1)**, 59.