Gene-level alignment of single cell trajectories informs the progression of *in vitro* T cell differentiation

Authors: Dinithi Sumanaweera[1,†], Chenqu Suo[1,2,†], Daniele Muraro[1], Emma Dann[1], Krzysztof Polanski[1], Alexander S. Steemers[1,3], Jong-Eun Park[1,4], Bianca Dumitrascu[5,6], Sarah A. Teichmann[1,7]*

Affiliations:
[1]Wellcome Sanger Institute; Wellcome Genome Campus, Hinxton, Cambridge, UK.
[2]Department of Paediatrics, Cambridge University Hospitals; Hills Road, Cambridge, UK.
[3]Princess Máxima Center for Pediatric Oncology; Utrecht, Netherlands
[4]Graduate School of Medical Science and Engineering, Korea Advanced Institute of Science and Technology (KAIST); Daejeon, Korea.
[5]Department of Statistics, Columbia University; New York, US
[6]Irving Institute for Cancer Dynamics, Columbia University; New York, US
[7]Theory of Condensed Matter, Cavendish Laboratory, Department of Physics, University of Cambridge; Cambridge, UK.
[†]These authors contributed equally to this work.
*Corresponding authors. st9@sanger.ac.uk (S.A.T.).

Abstract:

Single cell data analysis can infer dynamic changes in cell populations, for example across time, space or in response to perturbation. To compare these dynamics between two conditions, trajectory alignment via dynamic programming (DP) optimization is frequently used, but is limited by assumptions such as a definite existence of a match. Here we describe **Genes2Genes**, a Bayesian information-theoretic DP framework for aligning single-cell trajectories. **Genes2Genes** overcomes current limitations and is able to capture sequential matches and mismatches between a reference and a query at single gene resolution, highlighting distinct clusters of genes with varying patterns of gene expression dynamics. Across both real life and simulated datasets, **Genes2Genes** accurately captured different alignment patterns, and revealed that T cells differentiated *in vitro* matched to an immature *in vivo* state while lacking the final TNFα signaling. This use case demonstrates that precise trajectory alignment can pinpoint divergence from the *in vivo* system, thus providing an opportunity to optimize *in vitro* culture conditions.

Introduction

Recent advances in single-cell genomics, the mainstay of which is single-cell RNA sequencing (scRNA-seq), have revolutionized our understanding of biology and opened up new avenues of research[1]. With their single-cell resolution and ability to observe thousands of genes simultaneously, these new technologies enable the identification of transition cell states and the study of dynamic cellular processes (e.g. cell differentiation/development; cellular response to perturbations). The computational task of deriving a 'timeline' for each dynamic process (e.g. based on transcriptomic similarity) is referred to as 'pseudotime trajectory inference'[2]. The next challenge is to then compare and align two (or more) trajectories, as when exploring similarities between *in vitro* cell differentiation and *in vivo* cell development, or the perturbation responses in control groups *versus* drug treatment groups (**Fig. 1a**). The benefit is obvious; for instance, identifying genes that differ between *in vitro* and *in vivo* systems can guide us to refine *in vitro* cell differentiation. Trajectory comparison poses a time series alignment problem which is dynamic programmable[3]. The goal is to find an optimal set of pairwise sequential correspondences between two trajectories.

Currently, dynamic time warping (DTW) is often used to align two single-cell trajectories. Several notable attempts[4–9], such as CellAlign[5], employ DTW to identify correspondences between two different profiles, allowing the detection of timing differences in biologically similar processes[10]. Current practice is to interpolate the gene expression time series prior to DTW, and then minimize the Euclidean distance of expression vectors between the matched time points to find an optimal alignment. While DTW is a powerful approach with numerous uses, we are motivated to overcome its main limitations, i.e., (1) requiring every time point in each trajectory to match with at least one time point in the other (capturing time-warps only); (2) disregarding missing data or substantial differences between two series, occurring in the form of insertions or deletions (indels); and (3) using a distance metric which relies only on the difference of mean expression, without considering their distributions. We specifically address these by developing a new DP framework that formally handles matches and mismatches in a principled way. It overcomes the need to impose any ad hoc thresholds[5] and/or post hoc processing of a DTW output[11] to capture differential regions in gene expression.

Warps and indels are fundamentally distinct (**Fig. 1b,c**). This is particularly highlighted in discussions about integrating DTW with the gap concept[12,13] (as in the area of biological sequence alignment[14,15]). Both matches and mismatches inform our understanding of temporal gene expression dynamics. A mismatch could either imply missing data or differential expression, which could be interesting to investigate further. For instance, a sudden rise or drop in expression of one system relative to the other might indicate that it is transitioning through a different cell state. A mismatch also occurs when a considerable fraction of cells in one system have a significantly different expression for some genes compared to that of the other, causing distributional differences. This is particularly clear when iPSC-derived organoid trajectories are compared to *in vivo* references, due to the upregulation of pluripotency markers in early stages of organoid development. On the other hand, warps occur due to the differences in the relative speeds of cell maturation in diverse contexts. Matching and mismatching can also inform divergence and convergence patterns in expression (e.g. **Fig. 1d**), allowing their computational separation for further downstream analysis.

Here we present Genes2Genes (G2G; **Fig. 2**), a novel framework for aligning single-cell transcriptomic trajectories of a reference and query system at single-gene resolution. G2G utilizes a DP alignment algorithm that accounts for matches, warps and indels by combining the classical Gotoh's biological sequence alignment algorithm[15] and DTW[16]. It is inspired by the concepts discussed in the related literature[17–19], and employs a new Bayesian information-theoretic measure based on minimum message length inference[20–22] to quantify the distance between two gene expression distributions assumed to be Gaussian. G2G facilitates: (1) generating descriptive pairwise alignments at gene-level, (2) identifying gene clusters of similar alignment patterns, (3) identifying genes with differential dynamic expression profiles, (4) exploring associated biological pathways, and (5) deriving an aggregate alignment across all or a subset of genes.

We first validate G2G's ability to accurately align and capture different patterns using simulated datasets. We further demonstrate how G2G captures mismatches and offers gene-level alignment, through benchmarking against related methods on a published real dataset[23] and a simulated negative control[11]. We next show how G2G quantitatively assesses the progression of *in vitro* human T-cell differentiation in an in house cultured artificial thymic organoid (ATO) compared to an *in vivo* reference of human T cell development. We find that the TNFα signaling pathway in the final stage of *in vivo* T cell maturation is not recapitulated *in vitro*, and identify potential transcription factors for optimizing *in vitro* cell engineering.

Results

Genes2Genes (G2G) aligns single-cell trajectories with dynamic programming, employing a Bayesian information-theoretic measure

Dynamic programming (DP) remains central to many sequence alignment algorithms. G2G is a new DP framework to infer and analyze gene trajectory alignments between a single-cell reference and query. Given a reference sequence R (a time series with time points: $R_1, R_2, \dots R_j, \dots\ R_{|R|}$) and query sequence Q (a time series with time points: $Q_1, Q_2, \dots Q_j, \dots Q_{|Q|}$), a computational alignment between them can inform us of the one-to-one correspondences (matches), one-to-many/many-to-one correspondences (expansion/compression warps) and mismatches (insertions and deletions) between their time points in linear order (**Fig. 1b**). A general DP algorithm finds their optimal alignment by constructing a pairwise alignment cost matrix and generating the optimal path with the minimum cost (**Fig. 1c**). This relies on a scoring scheme to quantify correspondences between every pair of R and Q time points.

Unlike the current DP alignment approaches (i.e. DTW and DNA/protein alignment), G2G implements a DP algorithm that handles both matches (including warps) and mismatches jointly, and runs it between the reference and query for each gene. This algorithm extends Gotoh's three-state sequence alignment algorithm[15] to five-states (**Fig. 1b**) for accommodating warps. Our DP scoring scheme incorporates a Bayesian information-theoretic cost function based on the minimum message length (MML) inference[20–22] (top left of **Fig. 2, Supplementary Fig. 1**), and the state transition probabilities from a five-state probabilistic finite state machine (**Extended Data Fig. 1**). The MML criterion allows us to compute a cost for matching any reference time point R_j and query time point Q_i based on their corresponding gene expression distributions. It evaluates their distributional differences in terms of both mean and variance, acknowledging that one trajectory may be noisier than the other. The five-state machine allows us to compute a cost of assigning an alignment state for R_j and Q_i out of the five possible states of alignment.

Overview of the G2G framework

The G2G framework is composed of several components to support single-cell trajectory comparison, which include input preprocessing, DP alignment algorithm, alignment clustering and downstream analysis (**Fig. 2**). It takes the input as log1p normalized single-cell gene expression matrices of a reference system and a query system, and their pseudotime estimates. G2G then performs interpolation to smoothen each gene expression trajectory. This first min-max normalizes the pseudotime axis over which we take a predefined number of equispaced interpolation points, similar to CellAlign[5]. For each interpolation time point, we estimate a Gaussian distribution of gene expression, taking into account all cells, kernel-weighted[5] by their pseudotime distance to this interpolation time point. Compared to the existing methods, our approach fits the entire distribution instead of only estimating the mean expression level at each interpolation point. The interpolated gene trajectories of the reference and query are then aligned using our DP algorithm. This generates optimal trajectory alignments for all input genes, described by five-state alignment strings (**Fig. 1d** and top right matrix of **Fig. 2**). These strings are then binary encoded to compute their pairwise Hamming distances, and genes displaying similar alignment patterns are clustered together using agglomerative hierarchical clustering. Alignments and their cluster memberships together allow us to proceed with further downstream analysis such as gene set overrepresentation analysis. (See Methods for details). The algorithmic novelties, definitions and descriptive level of the generated alignments altogether fundamentally distinguishes G2G from the current alignment approaches.

G2G accurately identifies different patterns of alignment with simulated data experiments

To benchmark how well G2G captures different alignment patterns, we performed two simulated data experiments on: (1) an artificial dataset (**Fig. 3a,e**, **Extended Data Fig. 2,** and **Supplementary Table 2-3**), and (2) a real biological dataset with artificial perturbations (**Fig. 3f,g, Extended Data Fig. 3,** and **Supplementary Table 4-17**). See Methods for details.

Experiment 1

We simulated 3500 trajectory pairs under three main classes of pattern: (1) *Matching* (500 genes), (2) *Divergence* (1500 genes), and (3) *Convergence* (1500 genes), using Gaussian Processes and suitable kernels[24,25], to test how well G2G aligns and captures them. Each trajectory consists of 300 data points spread across the pseudotime range [0,1]. In the *Divergence* and *Convergence* group, we have an equal number of pairs bifurcating at three time points (approximately at bifurcation time point $t_b \in [0.25, 0.5, 0.75]$, indicating early, mid, and late bifurcation, respectively). We first examined G2G performance on each of the seven classes of pattern, in identifying matched and mismatched regions accurately (with 50 interpolated time points – shortest possible alignment length L_{min}= 50 and longest possible alignment length L_{max}= 100), followed by clustering of all alignments.

Statistics for expected match and mismatch regions: For *Matching* pairs (**Fig. 3b**), 86.4% alignments give 100% alignment, while the rest show at least one false mismatch with an average length of 4.74 (~9.4% L_{min}). The mean matching percentage (the percentage of total matching, including one-to-one matches and warps, in the alignment string output, which we term 'alignment similarity') is 98.8%. For *Divergence* (**Fig. 3c**), we expect to see a full match at the beginning (start-match) followed by a full mismatch at the end (end-mismatch), where the match/mismatch length depends on the approximate location of bifurcation. Thus we report the distributions of start-match lengths (that follows a false mismatch if there is any), end-mismatch lengths, and start-mismatch lengths (detecting false mismatches) in all alignments across the three bifurcating locations. We observed no end matches, demonstrating that G2G accurately mismatches the differentially expressed region. However, 33.13% of the divergent alignments give at least one false start-mismatch (30.4% for $t_b = 0.25$, 33.2% for $t_b = 0.5$, and 35.8% for $t_b = 0.75$). However, their median mismatch length is 1 (with mean 2.29 = ~4.6% L_{min}). All distributions fall within their expected ranges of length with a few outliers. As the bifurcation point moves towards the maximum pseudotime, the number of matches increases while the number of mismatches decreases as expected. In contrast, *Convergence* alignments (**Fig. 3d**) (where we expect start regions to fully mismatch while end regions fully match) show only a 3.5% of all convergent alignments with false end-mismatches (2.8% for $t_b = 0.25$, 5.3% for $t_b = 0.5$, and 2.4% for $t_b = 0.75$). Their median mismatch length is 4 (with mean 11.84 = ~23.7% L_{min} , mainly due to just 4 outliers with complete mismatching alignments arising from t_b=0.75 case). Again, the distributions of end-match lengths and start-mismatch lengths across all subgroups fall within the expected ranges. We also see no false start matches. Only one 0.5 *Convergence* alignment (0.06%) showed a single mismatch within an expected match region. In conclusion, for *Matching*, *Convergence* and *Divergence* patterns, G2G is able to generate correct alignments with relatively high accuracy.

Clustering alignments: **Fig. 3e** shows the pairwise alignment distance matrix, which demonstrates a clear separation of the seven pattern classes. Hierarchical agglomerative clustering of the alignments at 0.2 distance threshold results in 11 clusters, capturing all the distinct patterns with only 0.34% mis-clustering rate. 21 alignments of early *Divergence* and late *Convergence* appear in 4 pure subclusters due to warps, confirming G2G's ability to distinguish between usual and outlying patterns. (Further results in **Extended Data Fig. 4**).

Overall, G2G has a good detection accuracy of the expected matches and mismatches across all the seven classes of trajectory alignment patterns we evaluated. Clustering results of their five-state alignment strings also confirm the utility of such descriptive outputs.

Experiment 2

To test the ability to detect matching patterns in real scRNA-seq data, we used a scRNA-seq dataset of E15.5 murine pancreatic development[26] and considered gene expression profiles of 769 genes varying in expression during beta-cell differentiation. We randomly split cells into query and reference, and simulated the presence of a mismatch in the form of a deleted portion (perturbation scenario 1) or changed portion (perturbation scenario 2) of increasing size (pseudotime was equally divided into 50 bins with an increasing number of bins being perturbed) at the beginning of the trajectory (**Fig. 3f**). We then performed alignments with G2G (under 50 interpolation time points) and calculated the percentage of match calling, i.e., alignment similarity (**Fig. 3g**). We found that on average, G2G recovered the matching region accurately across genes, with 91% mean alignment similarity when no perturbation was introduced, and 86% mean alignment similarity for mismatches smaller than 20% of pseudotime bins (10/50 bins) across the two different perturbation scenarios. For perturbation scenario 1, the alignment similarity decreases with increasing deletion sizes as expected across smaller perturbation sizes. However, the detected mismatch length is shorter than expected for deletions larger than 20%. This is due to the relative non-varying gene expression trends between pseudotime bin 10 to 20 (**Extended Data Fig. 3a**) and hence it causes warps instead of mismatches. For perturbation scenario 2, the alignment similarity has an expected maximum and minimum. For example, if the perturbation size = 5 in the query, the minimum mismatch segment to expect is: "`IIIII`", whereas the maximum mismatch segment to expect is: "`IIIIIDDDDD`" (illustrated in **Extended Data Fig. 3b**). Accordingly, the observed trend generally follows the expected trends, falling within the expected ranges for larger perturbations sizes.

In general, we observe that the alignment accuracy drops when the underlying assumption of a smooth trajectory breaks, and/or when there are significant inconsistencies in the cell densities across the trajectories. As apparent from this experiment, we also note that it is extremely challenging to simulate perturbations using real life datasets for trajectory benchmarking, as they may introduce warps instead of expected mismatches as shown in perturbation scenario 1. Consequently, it is difficult to guarantee a specific linear ordering of matches and mismatches. Overall, the results support that G2G has a good accuracy rate of match and mismatch detection.

G2G captures mismatches and offers gene-level resolution alignment

To benchmark our method against the widely-known DTW-based method, CellAlign[5], we performed G2G alignment on the same dataset[23] used by CellAlign (**Fig. 4a**), which includes time-course data of murine bone marrow-derived dendritic cells stimulated with PAM3CSK (PAM) or lipopolysaccharide (LPS) to simulate responses to different pathogenic components. The main difference between CellAlign and G2G is that, CellAlign only considers matches and warps through DTW, while G2G's algorithm unifies matches and mismatches through a single DP algorithm by imposing the notion of gaps on top of DTW via a probabilistic framework and Gotoh's algorithm[15]. Further conceptual differences between G2G and CellAlign are summarized in **Supplementary Table 1**. Of note, G2G outputs both individual gene-level alignments and an aggregate (average) alignment path over all gene-level alignments, unlike CellAlign which outputs only a single, high-dimensional alignment across all genes. This is particularly useful when there is heterogeneity in alignment patterns among different genes.

G2G's ability to capture mismatches is revealed in aligning genes from the "core antiviral module" of the PAM/LPS dataset (**Supplementary Table 18-22**). CellAlign demonstrated a 'lag' in expression in PAM stimulation compared to LPS[5]. This is also captured by the G2G aggregate alignment whereby later PAM pseudotime points were mapped to earlier LPS pseudotime points (**Fig. 4b**). In addition, G2G identified mismatches in the early and late pseudotime points.

In the early pseudotime points, the gene expression was consistently low in the PAM condition, whereas some of the cells stimulated by LPS were already showing elevated expression at early time points (**Fig. 4c**). These have also been noticed in the original paper and were described as "precocious expressers"[23]. The mismatch in the late pseudotime points of LPS stimulation was caused by the peaked expression, while the expression of the PAM-stimulated cells was still on the rise and had not reached a peak yet (**Fig. 4c**).

In the case of aligning genes from the "peaked inflammatory module" (**Supplementary Table 23-30**), **Fig. 4d** shows the main average path of alignment generated by G2G. In addition, genes were clustered based on their alignment patterns (method illustrated in **Fig. 2**; results in **Extended Data Fig. 5-6**), and the aggregate alignment path for each cluster was different from the main average path (**Fig. 4d**). Representative genes from different clusters are shown in **Fig. 4e**. Although they all had matches for some pseudotime points, there are subtle differences in the length and position of the time points that are matched.

In addition, genes can be examined via the similarity between their expression profiles along the query and reference pseudotime trajectories. This can be computed through their percentage of alignment similarity. We identified genes with low alignment similarity and high log fold change between query and reference data, such as *CCRL2* and *C5AR1*. *CCRL2* expression started at a much higher level and peaked early in the LPS condition, whereas following PAM stimulation it grew as a slow incline. This stark difference suggests that *CCRL2* is a more LPS-specific response gene, consistent with previous reports in murine dendritic cells, macrophages, glial cells, astrocytes and microglia stimulated with LPS[27,28]. On the other hand, these data suggest that *C5AR1* is a PAM-specific response gene. In the case of the *TNF* gene, although the differential expression log fold change is almost negligible, the low alignment similarity revealed different expression dynamics. This difference would not be picked up by differential gene expression, hence highlighting the importance of such an alignment. For the genes that have high alignment similarity such as *CD44*, the log fold change should be small due to consistent matching across the pseudotime.

The above results showcase the benefit of G2G alignments over DTW alignments, i.e., to capture mismatched regions in gene expression trajectories. While the DP matrix of DTW may reflect the time points of relative differences and similarities in gene expression between two trajectories, we need extra effort subjected to thresholding to perform local DTW alignment and extract mismatches. TrAGEDy[11], the most recent extension on CellAlign[5,11], performs such ad hoc processing of DTW outputs (See feature comparison in **Supplementary Table 1**). We next compared G2G with CellAlign and TrAGEDy on a simulated dataset containing two trajectories with no shared process[11](referred to as a negative control dataset). G2G successfully quantifies that they are very dissimilar, with a low average of 34.5% alignment similarity across all the genes (**Supplementary Fig. 2a, Supplementary Table 52**). G2G also generates an aggregate alignment solely with insertions and deletions, which agrees with the expectation (**Supplementary Fig. 2b**). CellAlign is unable to produce a meaningful output for this dataset due to its DTW assumption of no mismatch. The alignment from TrAGEDy includes segments of one-to-one matches. This could be caused by the assumption of a definite match in between, in TrAGEDy post hoc thresholding. It is further validated by testing on three simulated genes with completely mismatched trajectories (**Supplementary Fig. 2c**). While TrAGEDy falsely generates matched subregions, G2G gives accurate outputs of 100% mismatch.

Overall, the above results over a real biological dataset and a negative control validate that G2G can accurately detect mismatches.

In vivo, *in vitro* human T cell development comparison using G2G reveals differences in TNFα signaling

We next applied G2G to compare *in vitro* and *in vivo* human T cell development. Thymus is the key site for T cell development in humans, where lymphoid progenitors differentiate through stages of double negative (DN) and double positive (DP) T cells to acquire T cell receptor (TCR) (illustrated in **Fig. 5a**). If the TCR recognizes self antigen presented on MHC via the process of positive selection, the developing T cells further differentiate through abT(entry) cells and finally mature into single positive (SP) T cells. There are different subsets of SP T cells, including CD4+T, CD8+T and regulatory T (Treg) cells, as well as the newly recognized unconventional type 1 and type 3 innate T cells and CD8AA[29,30]. To investigate human T cell development in a model system, we differentiated induced pluripotent stem cells (iPSCs) into mature T cells using the artificial thymic organoid (ATO) system[31]. We previously harvested differentiated cells from week 3, 5, and 7 and reported that the mature T cells in ATO were most similar to the *in vivo* type 1 innate T cells. To explore this further, we performed scRNA-seq analysis of differentiated cells harvested at regular intervals throughout the differentiation, i.e., including the early time points as well

(**Fig. 5b**). Cell types were annotated (low-level annotation in **Fig. 5c** with more refined annotation in **Extended Data Fig. 7a**) using a combination of the logistic regression based predictions with CellTypist[32] (**Extended Data Fig. 7b**) and marker gene analysis (**Extended Data Fig. 8**). The ATO system captures the differentiation from stem cells, through mesodermal progenitors, endothelium, to haematopoietic lineage, and then further down to T cell lineage (**Fig. 5b**).

We then combined the *in vitro* ATO data with the relevant *in vivo* cell types from our developing human immune atlas[30] (hereafter referred to as the pan fetal reference) and integrated using scVI[33] onto a common latent embedding (**Fig. 5d,e**). For ATO data, the pseudotime was estimated using a Gaussian Process Latent Variable Model (GPLVM)[34] with sampling times as priors (**Fig. 5f**). GPLVM has previously been successfully applied in single-cell trajectory inference to incorporate useful priors[35–39]. The pan fetal reference cells' pseudotime was computed similarly by estimating their time priors from the nearby ATO cells (see Methods).

Alignment between *in vitro* ATO data and *in vivo* pan fetal reference was performed with G2G (under 15 equispaced time points in [0,1] pseudotime range) using all transcription factor (TF) genes[40] (**Supplementary Table 31**), as many TFs function as 'master regulators' of cell states and have been used to induce cell differentiation. The aggregate alignment result for all TFs showed a mismatch at the beginning and a mismatch at the end (**Fig. 6a**). We explored this further by performing gene set overrepresentation among the most mismatched genes (alignment similarity < 40%, **Fig. 6b, Supplementary Table 32**), and found that pluripotency and TNFα signaling pathways were the two most significant gene sets. Many of the genes in the pluripotency signaling pathway, such as the well-known pluripotent genes *POU5F1*, *NANOG* and *SOX2*[41], were expressed at the beginning of ATO development but missing from the reference (**Fig. 6c**). This is expected given that the *in vitro* differentiation started from iPSCs, whereas the earliest cells from the *in vivo* reference were haematopoietic stem cells (HSCs). On the other hand, *HHEX*, which is known to be expressed in HSC[42,43] and early DN T cells[44] demonstrates another pattern: matching between *in vivo* and *in vitro* HSC and DN T cells as expected although the maximum *HHEX* expression in *in vitro* cells was lower than that of *in vivo* cells (**Fig. 6c**).

With regard to the TNFα signaling via NFκB pathway, many of the genes such as *FOSB, JUNB* and *NR4A2* show an increasing trend at the last stage of *in vivo* T cell development, and this increase is missing in the *in vitro* T cells (**Fig. 6d**). TNFα activation of NFκB pathway has been implicated in the final functional maturation of murine T cells within the thymus[45,46] and indicates that, while the ATO system captures the single positive T cell development through positive selection, other *in vivo* signaling necessary for maturation, such as TNFα pathway, might be missing. By systematically comparing the *in vivo* data to *in vitro* data with G2G, it gives us potential targets for further *in vitro* optimization. There are exceptions to this overall pattern, such as *KLF2*, whose expression is higher in *in vitro* T cells than that in *in vivo* (**Fig. 6d**). This might be the result of each gene being regulated by more than one signaling pathway. We remark that although it is possible to recover the difference between TNFα signaling by doing direct differential gene expression between cell subsets, e.g., end products of ATO vs *in vivo* T cells, a key advantage of using the trajectory alignment is that we could pinpoint where along the differentiation did the mismatch occur. This would in turn inform us when to introduce the TNFα in *in vitro* optimization.

Using G2G, genes are also clustered based on their alignment patterns (**Extended Data Fig. 9**, **Supplementary Table 31,33-45**). Cluster 4 captures the pluripotent genes, and the majority show a complete mismatch alignment pattern. Cluster 10 genes are enriched in the Hippo signaling pathway, and many of its genes show insertions at the beginning of the pseudotime. Hippo signaling has been implicated in stem cell biology and pluripotency regulation[47], and the observed mismatch might again be explained by the stem cell stage present in the organoid but absent from the *in vivo* reference. Interestingly, cluster 0 genes show mismatches in the middle time points. This might represent a missing cell state, e.g., *BATF2* is expressed sparsely in endothelial cells which are present in the *in vitro* but not in the *in vivo* system (**Extended Data Fig. 10**).

We further repeated the analysis restricting to T cell lineages, i.e., DN T cells onwards (**Extended Data Fig. 11a-d, Supplementary Table 46-47**). TNFα signaling via NFκB pathway remains the most significantly enriched gene set among the mismatched genes.

The above alignments were performed using *in vivo* type 1 innate T cells and the relevant precursors, as we previously found that the *in vitro* mature T cells were most similar to the *in vivo* type 1 innate T cells[30]. However, *in vitro* cell differentiation to conventional CD8+T cells might also provide promising routes for cell therapies. We therefore performed another G2G alignment using *in vivo* conventional CD8+T cells and the relevant T lineage precursors (DN T cells onwards), with the results shown in **Extended Data Fig. 11e-h** (and **Supplementary Table 48-49**). The most significantly enriched gene set among the mismatched genes is also TNFα signaling pathway. To further explore the differences in the two alignment results, we focused on genes that showed the most dissimilar alignment results (genes that had alignment similarity differences > 0.5 or < -0.5) (**Fig. 6e**). Three of the genes, *SOX4, FOXP1* and *ARID5B* had large $\log_2$ fold change differences (absolute $\log_2$ fold change > 1) between type 1 innate T cells and CD8+T cells. For these three genes, the expression dynamics of *in vitro* T cell development are more similar to those of *in vivo* type 1 innate T cells, whereas *in vivo* CD8+T cells had higher *SOX4*, *FOXP1* and lower *ARID5B* expression in the last stages of development (**Fig. 6e**). While the role of *SOX4* in CD8+T cell development is unclear, *FOXP1* has been shown to maintain a quiescent profile in naive CD8+T cells[48,49] , and our results are in keeping with a more activated profile in type 1 innate T cells. *ARID5B* has been reported to regulate metabolic programming and promote IFNγ production in NK cells[50]. The higher expression in type 1 innate T cells might explain some of their NK-like features[29,30]. On the other hand, for *BHLHE40*, which is downstream of TNFα signaling and other pro-inflammatory cytokines[51], its expression dynamic in *in vitro* T cell development is more similar to that in CD8+T cells, while the *in vivo* type 1 innate T cells have increased expression at the end.

Overall, G2G alignment between *in vivo* and *in vitro* human T cell development revealed potential targets for further optimization of *in vitro* T cell differentiation (illustrated in **Fig. 6f**).

Discussion

Genes2Genes offers a structured alignment framework to compare single-cell pseudotime trajectories at single-gene resolution. We validated G2G's accuracy in identifying mismatches and different alignment patterns through extensive simulation studies. We have also benchmarked against current state-of-art methods in trajectory alignment using a real-life dataset and a simulated negative control dataset from the literature. In addition, we demonstrated G2G's potential in identifying genes and pathways that can guide the refinement of T cell differentiation in an organoid protocol.

Given cell-by-gene matrices of reference and query systems along with their corresponding pseudotime estimates, G2G generates a five-state alignment string for each gene of interest by running a DP algorithm that handles both matches and mismatches. The gene sets to compare can be all expressed genes against each other, or restricted to gene sets of interest such as e.g. TFs, regulons, highly variable genes or genes associated with certain biological/signaling pathways of interest. Compared to existing alignment approaches, G2G outputs more descriptive and direct results highlighting both matched (including warps) and mismatched regions of a gene over time. G2G provides a powerful addition to the current repertoire of comparative analysis toolboxes for any pseudotime alignment task, e.g., *in vivo*/*in vitro*, treatment/control, cross-species etc.

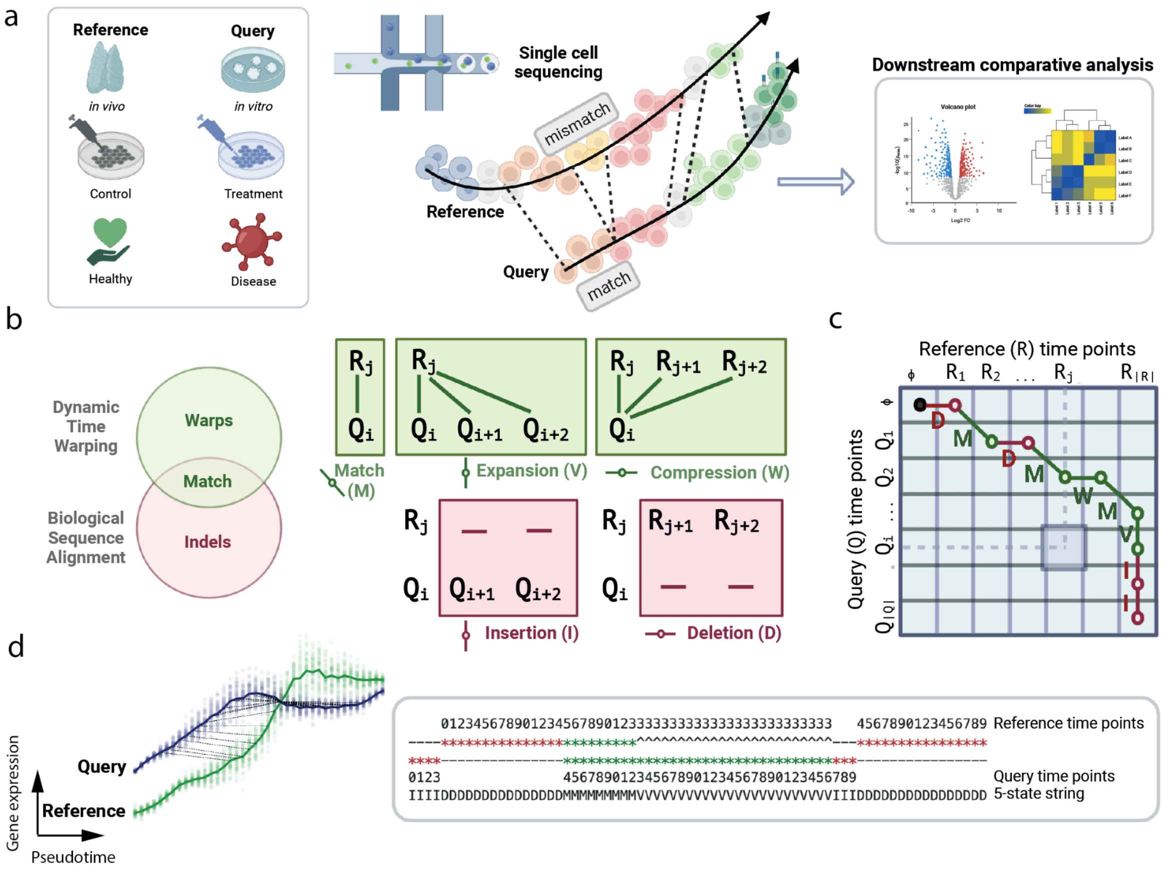
An important feature of G2G are the gene-specific alignments. Most existing methodologies produce a single alignment for all genes by computing high-dimensional Euclidean distances over their mean gene expression vectors. Such metrics suffer from 'the curse of dimensionality' by losing accuracy as the number of genes increases[52]. Importantly, in many contexts, an overall alignment across all genes masks gene heterogeneity along trajectories in the reference and query systems. Alpert et al (2018)[5] discuss choosing the right alignment resolution, recommending alignment of the largest gene set that shows significant differential expression (DE genes) over time. This is to remove stably expressed genes which may add noise and skew the alignment results. Our method goes further and fully resolves all gene groups with distinct matching and mismatching patterns at different stages along trajectories (**Fig. 4d**, **Extended Data Fig. 4-6**).

The gene-specific alignment of G2G also allows users to cluster genes based on their matching patterns to form groups with broadly similar alignments. We show that pathway overrepresentation analysis on each gene cluster can reveal specific biological signaling pathways that are driving the differences in pseudotime trajectories at different stages. These pathways and gene sets can be a starting point for protocol intervention strategies in the case of *in vivo*/*in vitro* alignments, and for mechanistic molecular interpretation of differences between trajectories in other cases.

The reliability of a trajectory alignment depends on how trustworthy the given pseudotime estimates are[2]. Different pseudotime estimates can produce different alignment distributions. While our initial G2G framework provides proof-of-concept by demonstrating the power of gene-level DP alignment to discover differential genes along pseudotime trajectories, future work is needed to develop suitable methods to calibrate its input (i.e. pseudotime estimates and interpolation), and better parameter optimization strategies to ensure more reliable alignment distributions. For instance, an adaptive window size for the Gaussian kernel-based interpolation may optimize the method's sensitivity to the variance of expression in the nearby cells. Furthermore, the current G2G version can only compare two linear trajectories without considering branching processes. We are aware of other efforts in aligning branched processes with DTW based tree alignment[8]. The output from such alignments, i.e., identified pairs of correspondences, could be further inputted into G2G for a comprehensive pairwise lineage alignment to capture mismatches.

In summary, G2G provides a formal trajectory alignment for single-cell transcriptomic data and is able to capture matches and mismatches at single-gene level. It enables a deeper understanding of the diversity of gene-level trajectory alignments across single-cell datasets. The G2G package is easy-to-use and freely available online at https://github.com/Teichlab/Genes2Genes with a tutorial. We have demonstrated that regenerative medicine can specifically benefit from such trajectory comparisons by extracting cues to guide refinement of *in vitro* cell engineering to recapitulate *in vivo* development. We envision that the software will be useful to the community for exploring other biological scenarios such as cell activation/stimulation responses in control and disease, generating new insights to advance our understanding of cell development and function in health and disease.

Main Figures



```
    01234567890123456789012333333333333333333333333   4567890123456789 Reference time points
----************************^^^^^^^^^^^^^^^^^^^^^^^^---****************
****---------------************************************----------------
0123               456789012345678901234567890123456789                Query time points
IIIIDDDDDDDDDDDDDDDMMMMMMMMMVVVVVVVVVVVVVVVVVVVVVVVVIIIDDDDDDDDDDDDDDDD 5-state string
```

Fig. 1 | Computational alignment of single-cell transcriptomic trajectories. a, Trajectory alignment is important for comparing different single-cell reference and query systems that dynamically change. This could be between *in vivo* cell development and *in vitro* cell differentiation, or between control and drug-treated cells in response to the same perturbation, or between the responses to vaccination or pathogen challenge in healthy *versus* diseased individuals. A complete alignment between them can capture matches and mismatches for further downstream analysis. **b,** Schematic illustration of the different states of alignment and their theoretical origins. Left: dynamic time warping and biological sequence alignment are complementary to each other[14–16], where both address matches yet capture either warps or indels respectively. Right: between discrete time points in R (reference trajectory) and Q (query trajectory), there may exist 5 different states of alignment: matches (1-1 correspondences), warps (1-to-many expansion or many-to-1 compression correspondences) and mismatches (insertions/deletions denoting a significant difference in one system compared to the other). **c,** Example alignment path across a pairwise time point matrix between R and Q trajectories. Diagonal lines (green) refer to matches (M); vertical lines refer to either insertions (I) (red) or expansion warps (V) (green); horizontal lines refer to deletions (D) (red) or compression warps (W) (green). Any matrix cell (i, j) denotes the pairing of a reference time point R_j and query time point Q_i. **d,** Example gene alignment plot (left) and the corresponding five-state alignment string generated by G2G (right). Plots display interpolated log1p normalized expression (y-axis) between a reference (green) and query (blue) against their pseudotime (*x*-axis) for the gene *SERTAD2* in the PAM/LPS dataset from Shalek et al (2014)[23]. The bold lines represent mean expression trends, while the faded data points are 50 random samples from the estimated expression distribution at each time point. The black dashed lines visualize matches and warps between time points. The boxed alignment string describes matches and mismatches for the gene (stars represent time points, and ^ represent a repeated time point from the left due to a warp).

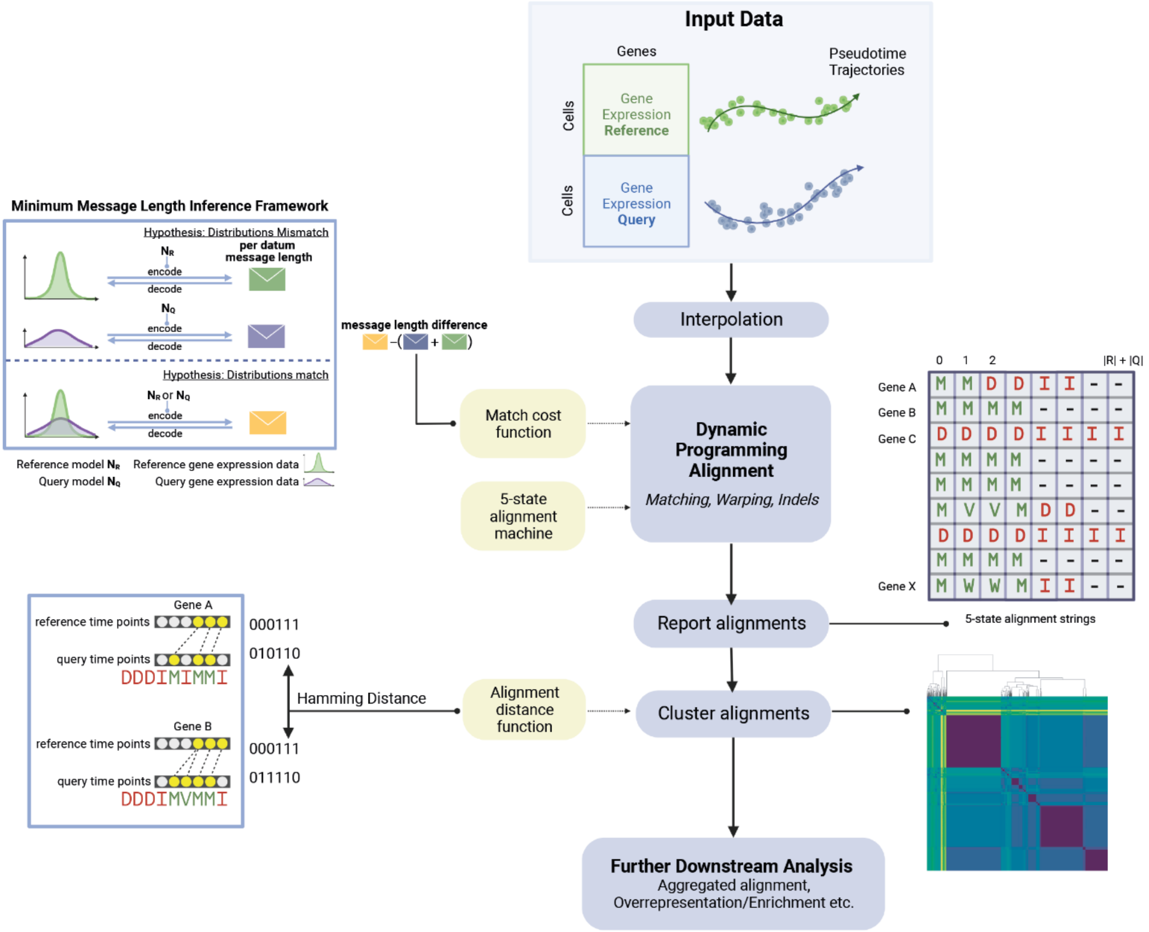


Fig. 2 | Overview of the Genes2Genes (G2G) alignment framework for comparing single-cell transcriptomic trajectories. Schematic illustration of G2G workflow. Given log1p normalized cell by gene expression matrices of a reference (R) and query (Q) system and their pseudotime estimates, G2G infers individual alignments for a list of genes of interest. It first interpolates data by extending mean based interpolation in Alpert et al (2018)[5] to distributional interpolation, and then runs Gotoh's dynamic programming (DP) algorithm[15] adapted for a five-state alignment machine (**Extended Data Fig. 1a**) defining a match (M), compression warp (W), expansion warp (V), insertion (I) and deletion (D). All reported alignments are then clustered and used to deliver statistics on the overall degree of alignment between R and Q, supporting further downstream analyses. Top left, the DP recurrence relations utilize a match cost function defined under minimum message length (MML)[21] statistical inductive inference. Given a hypothesis (a distribution model) and data, MML can define the total message length of encoding them for lossless compression along an imaginary message transmission. G2G defines two hypotheses: (1) Φ: R_j and Q_i time points mismatch, and (2) A: R_j and Q_i time points match. Under Φ, the message length is the sum of independent encoding lengths of their corresponding interpolated expression data and distributions. Under A, the message length is their joint encoding length of the corresponding interpolated expression data with a single Gaussian distribution (either of R_j or Q_i). The match cost is taken as the difference of A and Φ per datum encoding lengths. Bottom left, G2G encodes the five-state alignment strings using a binary scheme and computes a pairwise Hamming distance matrix to cluster all gene-level alignments. Top right: example output of five-state alignment strings for each gene. Bottom right, example clustermap showcasing the clustering result of alignment strings. The color represents the pairwise Hamming distance, and the clustering is performed using Agglomerative hierarchical clustering.

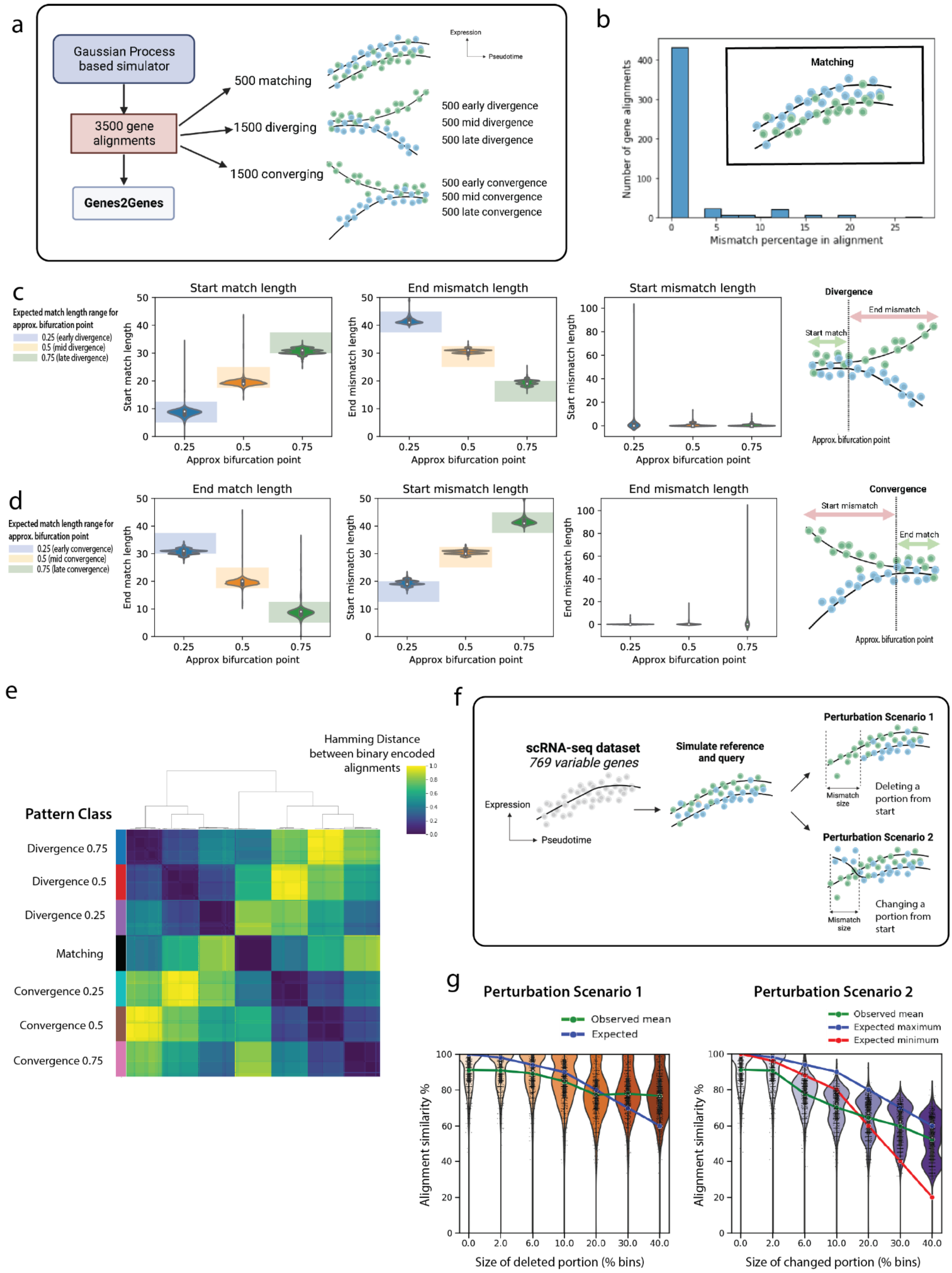


Fig. 3 | G2G accurately identifies different patterns of alignment with simulated data experiments. a, Experiment 1 uses a Gaussian Process based simulator to generate 3500 simulated pairs of reference and query gene trajectories to test G2G on. These include three main classes of alignment pattern: (1) *Matching*, (2) *Divergence* and (3) *Convergence*. The *Divergence* and *Convergence* groups further sub-categorised based on their approximate location of bifurcation (0.25, 0.5, and 0.75 within [0,1] pseudotime range), resulting in seven pattern classes (each with 500 alignments). **b,** Alignment frequency histogram of mismatch percentages observed in the 500 alignments for the simulated *Matching* class. **c,** Distributions of start match lengths (following a false mismatch if there is any), end mismatch lengths (prior to a false match if there is any), and start mismatch lengths (number of mismatches starting from time point 0) in the 1500 *Divergence* alignments across the three bifurcation subgroups. 50 equispaced time points on pseudotime [0,1] were used for distribution interpolation and alignment. In the case of early divergence, i.e., bifurcation at 0.25, there are 2 cases showing complete mismatch, i.e., mismatch length of 100. **d,** Distributions of end match lengths (prior to a false mismatch if there is any), start mismatch lengths (following a false match if there is any), and end mismatch lengths (number of mismatches before time point 1) in the 1500 *Convergence* alignments across the three bifurcation subgroups. 50 equispaced time points on pseudotime [0,1] were used for distribution interpolation and alignment. In the case of late convergence, i.e., bifurcation at 0.75, there are 4 cases showing complete mismatch, i.e., mismatch length of 100. **e,** The clustergram of the pairwise Hamming distance matrix across all alignments, which clearly shows the separate classes of pattern. **f,** Experiment 2 uses the mouse pancreas development dataset (Beta lineage) scRNA-seq dataset [26] to generate perturbed pairs of alignment from the expected *Matching* alignments. Perturbation scenario 1 deletes the start region from the reference trajectory, whereas perturbation scenario 2 changes the start region from the reference trajectory. **g,** The alignment similarity distributions for varying sizes (percentage of 50 pseudotime bins) of perturbation under perturbation scenario 1 (left) and perturbation scenario 2 (right). Each point represents a gene (total number of genes n = 769). In each plot, the observed average alignment similarity across different perturbation sizes is shown by the green line. For perturbation scenario 1 (left), the blue line shows the expected alignment similarity across different perturbation sizes. For perturbation scenario 2 (right), there are two expected lines: maximum (in blue) and minimum (in red). The maximum mismatch length is expected when both reference and query time points form insertions and deletions, making the maximum expected length *size**2. The minimum mismatch length is expected when only the changed reference time points are mismatched as insertions, while the corresponding query time points are matched to the non-perturbed reference time points (illustrated in **Extended Data Fig. 3b**).

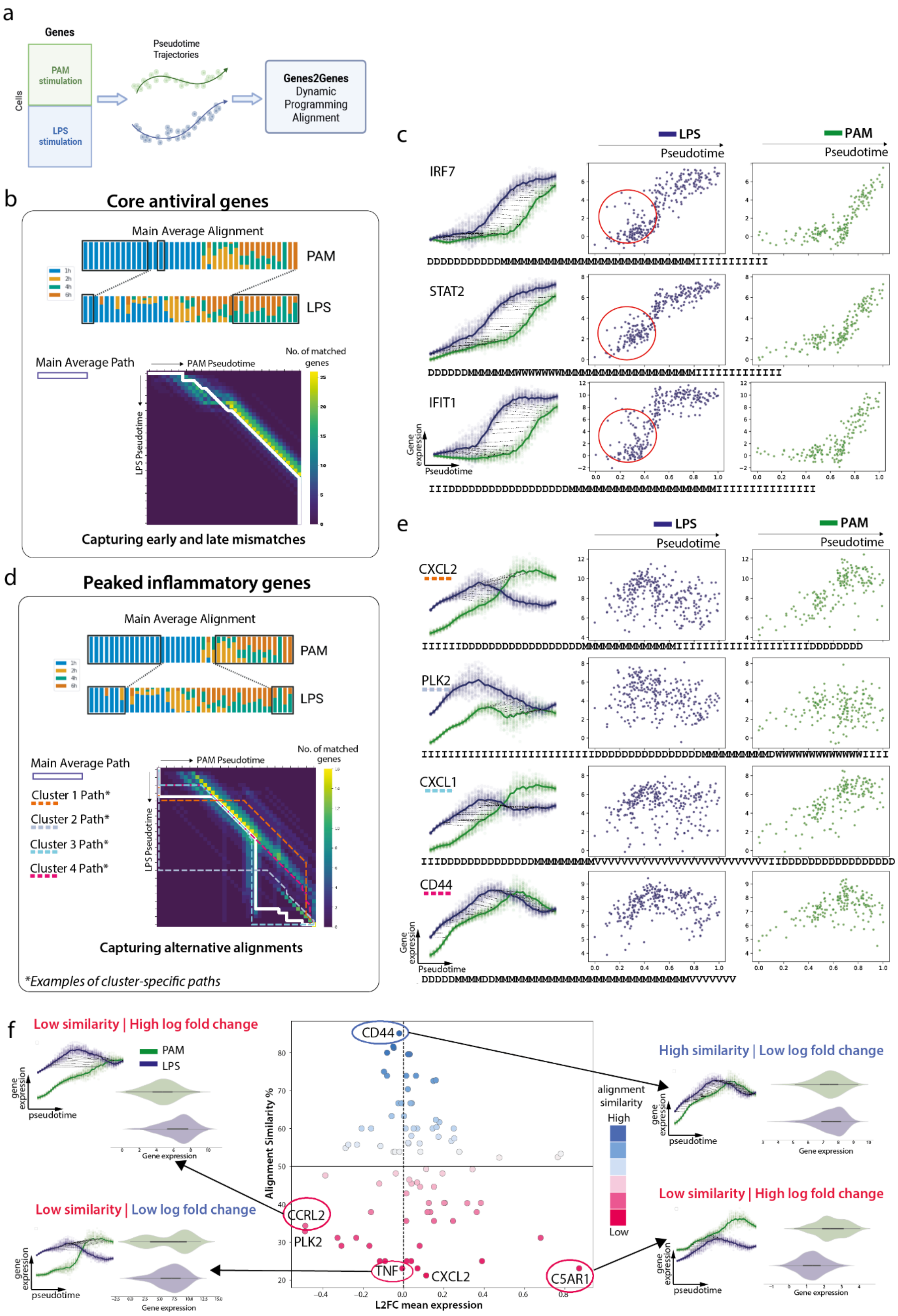
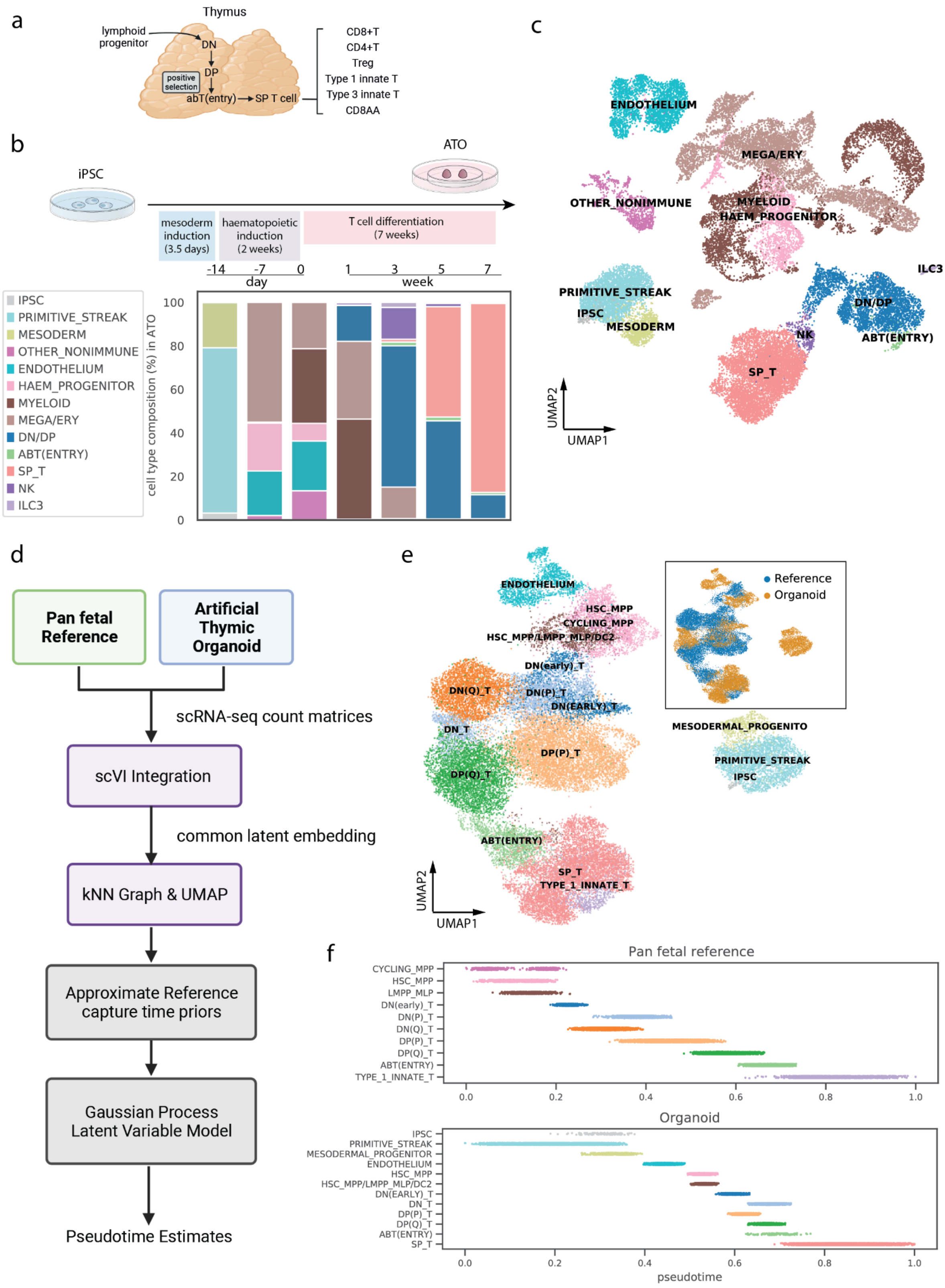

a
Genes
Cells
PAM stimulation
LPS stimulation
Pseudotime Trajectories
Genes2Genes Dynamic Programming Alignment
b
Core antiviral genes
Main Average Alignment
PAM
LPS
1h
2h
4h
6h
Main Average Path
PAM Pseudotime
LPS Pseudotime
No. of matched genes
Capturing early and late mismatches
c
LPS
PAM
Pseudotime
IRF7
STAT2
IFIT1
Gene expression
d
Peaked inflammatory genes
Main Average Alignment
PAM
LPS
Main Average Path
Cluster 1 Path*
Cluster 2 Path*
Cluster 3 Path*
Cluster 4 Path*
PAM Pseudotime
LPS Pseudotime
No. of matched genes
Capturing alternative alignments
*Examples of cluster-specific paths
e
LPS
PAM
Pseudotime
CXCL2
PLK2
CXCL1
CD44
Gene expression
f
Low similarity | High log fold change
PAM
LPS
gene expression
pseudotime
Gene expression
CD44
Alignment Similarity %
L2FC mean expression
CCRL2
PLK2
TNF
CXCL2
C5AR1
High similarity | Low log fold change
alignment similarity
High
Low
Low similarity | Low log fold change
Low similarity | High log fold change

Fig. 4 | G2G captures mismatches and offers gene-level resolution alignment. a, G2G alignment was performed on a time-course data[23] of murine bone marrow-derived dendritic cells stimulated with PAM or LPS. Both the gene expression data and the inferred pseudotime were taken from Alpert et al. 2018[5]. **b,** Top: schematic illustration of the aggregate alignment result for all the genes in the "core antiviral module". The stacked bar plots represent the cell compositions at each time point (50 equispaced time points on pseudotime [0,1]), colored by the time of sampling post stimulation. Boxed segments represent mismatched time points. The black dashed lines represent matches and warps between time points. Bottom: pairwise time point matrix between PAM and LPS pseudotime. The color represents the number of genes showing match or warp for the given pair of a PAM time point and an LPS time point. The white line represents the main average alignment path. **c,** Gene expression plots for three representative genes (*IRF7*, *STAT2* and *IFIT1*) from the "core antiviral module" in LPS-stimulated (blue) and PAM-stimulated (green) data along their pseudotime. Left column: the interpolated log1p normalized expression (*y*-axis) against pseudotime (*x*-axis). The bold lines represent mean expression trends, while the faded data points are 50 random samples from the estimated expression distribution at each time point. The black dashed lines represent matches and warps between time points. Right two columns: the actual log1p normalized expression (*y*-axis) against pseudotime (*x*-axis). Each point represents a cell. The five-state alignment string for each gene is shown below the expression plots. Red circles highlight the cells with high expression values at early time points, which are referred to as 'precocious expressers'. **d,** The same plots as in **b** for genes in the "peaked inflammatory module". In the pairwise time point matrix, the white line represents the main average alignment path. The genes are also clustered based on their alignment results (**Extended Data Fig. 5**), and dashed lines with different colors represent examples of cluster-specific alignment paths. **e,** The same plots as in **c** for a representative gene from each cluster shown in **d**. **f,** Plot of alignment similarity (*y*-axis) against $\log_2$ fold change of mean expression (*x*-axis) for all genes in the "peaked inflammatory module" (middle). The color also represents the alignment similarity. The surrounding plots show the interpolated log1p normalized expression (*y*-axis) against pseudotime (*x*-axis) on the left, and the violin plot of total gene expression on the right for four selected genes (*CD44*, *CCRL2*, *TNF*, *C5AR1*).



612

613 **Fig. 5 | *in vivo*, *in vitro* human T cell development data integration and pseudotime**
614 **inference. a,** Schematic illustration of T cell development in the human thymus. **b,** Top:
615 schematic showing the experimental set-up of T cell differentiation from iPSCs in ATOs.
616 Bottom: barplot of cell type composition in ATO at different time points during
617 differentiation. **c,** UMAP visualization of different cell types in the ATO dataset (low-level

annotation, number of cells n = 31,483), with more refined annotation in **Extended Data Fig. 7a**. **d,** Workflow of integrating *in vitro* (i.e. ATO) and *in vivo* (i.e. pan fetal reference from Suo et al. 2022[30]) human T cell development data and pseudotime inference using GPLVM. **e,** Main: UMAP visualization of integrated *in vivo* and *in vitro* human T cell development data, colored by the cell types. Right insert: the same UMAP visualization colored by the data source. **f,** Stripplot of the inferred pseudotime (*x*-axis) against different cell types (*y*-axis), colored by the cell types, of *in vivo* pan fetal reference data (top) and *in vitro* organoid data (bottom).

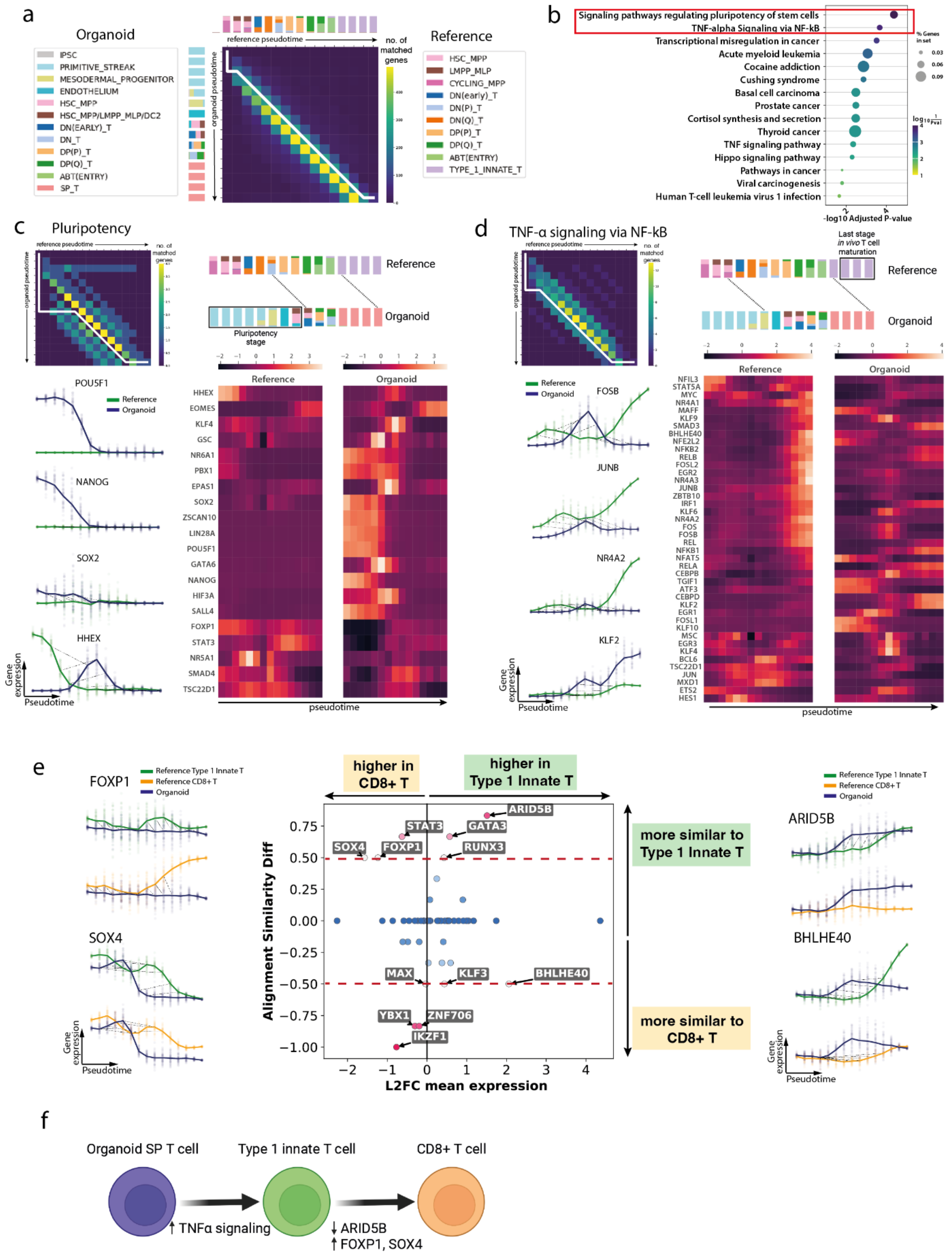

a
Organoid
IPSC
PRIMITIVE_STREAK
MESODERMAL_PROGENITOR
ENDOTHELIUM
HSC_MPP
HSC_MPP/LMPP_MLP/DC2
DN(EARLY)_T
DN_T
DP(P)_T
DP(Q)_T
ABT(ENTRY)
SP_T
reference pseudotime
organoid pseudotime
no. of matched genes
Reference
HSC_MPP
LMPP_MLP
CYCLING_MPP
DN(early)_T
DN(P)_T
DN(Q)_T
DP(P)_T
DP(Q)_T
ABT(ENTRY)
TYPE_1_INNATE_T
b
Signaling pathways regulating pluripotency of stem cells
TNF-alpha Signaling via NF-kB
Transcriptional misregulation in cancer
Acute myeloid leukemia
Cocaine addiction
Cushing syndrome
Basal cell carcinoma
Prostate cancer
Cortisol synthesis and secretion
Thyroid cancer
TNF signaling pathway
Hippo signaling pathway
Pathways in cancer
Viral carcinogenesis
Human T-cell leukemia virus 1 infection
-log10 Adjusted P-value
% Genes in set
0.03
0.06
0.09
c
Pluripotency
Reference
Organoid
Pluripotency stage
POU5F1
NANOG
SOX2
HHEX
Gene expression
Pseudotime
HHEX
EOMES
KLF4
GSC
NR6A1
PBX1
EPAS1
SOX2
ZSCAN10
LIN28A
POU5F1
GATA6
NANOG
HIF3A
SALL4
FOXP1
STAT3
NR5A1
SMAD4
TSC22D1
pseudotime
d
TNF-α signaling via NF-kB
Last stage in vivo T cell maturation
Reference
Organoid
FOSB
JUNB
NR4A2
KLF2
NFIL3
STAT5A
MYC
NR4A1
MAFF
KLF9
SMAD3
BHLHE40
NFE2L2
NFKB2
RELB
FOSL2
EGR2
NR4A3
JUNB
ZBTB10
IRF1
KLF6
NR4A2
FOS
FOSB
REL
NFKB1
NFAT5
RELA
CEBPB
TGIF1
ATF3
CEBPD
KLF2
EGR1
FOSL1
KLF10
MSC
EGR3
KLF4
BCL6
TSC22D1
JUN
MXD1
ETS2
HES1
pseudotime
e
FOXP1
Reference Type 1 Innate T
Reference CD8+ T
Organoid
SOX4
higher in CD8+ T
higher in Type 1 Innate T
Alignment Similarity Diff
L2FC mean expression
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−2
−1
0
1
2
3
4
ARID5B
STAT3
GATA3
SOX4
FOXP1
RUNX3
MAX
KLF3
BHLHE40
YBX1
ZNF706
IKZF1
more similar to Type 1 Innate T
more similar to CD8+ T
ARID5B
BHLHE40
f
Organoid SP T cell
Type 1 innate T cell
CD8+ T cell
↑ TNFα signaling
↓ ARID5B
↑ FOXP1, SOX4

Fig. 6 | *in vivo*, *in vitro* human T cell development alignment with G2G. a, Aggregate alignment result for all TFs between *in vitro* organoid (i.e. ATO) and *in vivo* reference (i.e. pan fetal reference from Suo et al. 2022[30]) human T cell development data shown in the pairwise time point matrix between organoid and reference pseudotime. The color represents the number of genes showing match or warp for the given pair of an organoid time point and a reference time point. The white line represents the main average alignment path. The stacked bar plots represent the cell compositions at each time point (15 equispaced time points on pseudotime [0,1]), colored by the cell types, for reference (top) and organoid (left) separately. **b,** Gene set overrepresentation results of the most mismatched genes from alignment in **a**, with the gene set names in *y*-axis, and -log$_{10}$(adjusted P-value) in *x*-axis. The size of the point represents the percentage of genes from that gene set being within the list of most mismatched genes. The color represents the -log$_{10}$(adjusted P-value). Two most significant gene sets were boxed in red. **c,** Alignment results for all genes in the pluripotency signaling pathway. Top left: pairwise time point matrix between organoid and reference pseudotime. The color represents the number of genes showing match or warp for the given pair of an organoid time point and a reference time point. The white line represents the main average alignment path. Top right: schematic illustration of the aggregate alignment result. The stacked bar plots represent the cell compositions at each time point, colored by the cell types. The black dashed lines represent matches and warps between time points. Boxed segment represents the mismatched pluripotency stage in the organoid. Bottom left: the interpolated log1p normalized expression (*y*-axis) against pseudotime (*x*-axis) for selected genes. Bottom right: heatmap of the smoothened (interpolated) and z-normalized mean gene expression along the pseudotime. **d,** The same plots as in **c** for all genes in the TNFα signaling via NFκB pathway. The boxed segment in the right top plot represents the mismatched last stage *in vivo* T cell maturation. **e,** Plots showing the alignment differences between *in vivo* conventional CD8+T lineage *versus in vitro* organoid, and *in vivo* type 1 innate T cell lineage *versus in vitro* organoid. Middle: plot of alignment similarity difference (*y*-axis) against log$_2$ fold change of mean expression between CD8+T and type 1 innate T cells (*x*-axis). The color reflects the absolute value of alignment similarity difference. Surrounding plots: the interpolated log1p normalized expression (*y*-axis) against pseudotime (*x*-axis) showing the alignment between *in vivo* type 1 innate T cell lineage (green) and *in vitro* organoid (blue) (top), and the alignment between *in vivo* CD8+T lineage (orange) and *in vitro* organoid (blue) (bottom), for four selected genes. **f,** Schematic illustration of potential targets for further optimization of *in vitro* T cell differentiation towards either type 1 innate T cells or conventional CD8+T cells.

References

1. Schier, A. F. Single-cell biology: beyond the sum of its parts. *Nat. Methods* **17**, 17–20 (2020).
2. Saelens, W., Cannoodt, R., Todorov, H. & Saeys, Y. A comparison of single-cell trajectory inference methods. *Nat. Biotechnol.* **37**, 547–554 (2019).
3. Bellman, R. The theory of dynamic programming. *Bulletin of the American Mathematical Society* vol. 60 503–515 Preprint at https://doi.org/10.1090/s0002-9904-1954-09848-8 (1954).
4. Aach, J. & Church, G. M. Aligning gene expression time series with time warping algorithms. *Bioinformatics* **17**, 495–508 (2001).
5. Alpert, A., Moore, L. S., Dubovik, T. & Shen-Orr, S. S. Alignment of single-cell trajectories to compare cellular expression dynamics. *Nat. Methods* **15**, 267–270 (2018).
6. Alpert, A. *et al.* Alignment of single-cell trajectories by tuMap enables high-resolution quantitative comparison of cancer samples. *Cell Syst* **13**, 71–82.e8 (2022).
7. Do, V. H. *et al.* Dynamic pseudo-time warping of complex single-cell trajectories. Preprint at https://doi.org/10.1101/522672.
8. Sugihara, R., Kato, Y., Mori, T. & Kawahara, Y. Alignment of single-cell trajectory trees with CAPITAL. *Nat. Commun.* **13**, 5972 (2022).
9. Cacchiarelli, D. *et al.* Aligning Single-Cell Developmental and Reprogramming Trajectories Identifies Molecular Determinants of Myogenic Reprogramming Outcome. *Cell Systems* vol. 7 258–268.e3 Preprint at https://doi.org/10.1016/j.cels.2018.07.006 (2018).
10. Tritschler, S. *et al.* Concepts and limitations for learning developmental trajectories from single cell genomics. *Development* **146**, (2019).
11. Laidlaw, R. F., Briggs, E. M., Matthews, K. R., McCulloch, R. & Otto, T. D. TrAGEDy: Trajectory Alignment of Gene Expression Dynamics. Preprint at https://doi.org/10.1101/2022.12.21.521424.
12. Sankoff, D. & Kruskal, J. B. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. (Addison-Wesley, 1983).
13. Chen, L. Similarity search over time series and trajectory data.

https://www.academia.edu/download/30803148/thesis-leichen.pdf.

14. Needleman, S. B. & Wunsch, C. D. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Molecular Biology* 453–463 Preprint at https://doi.org/10.1016/b978-0-12-131200-8.50031-9 (1989).
15. Gotoh, O. An improved algorithm for matching biological sequences. *Journal of Molecular Biology* vol. 162 705–708 Preprint at https://doi.org/10.1016/0022-2836(82)90398-9 (1982).
16. Vintsyuk, T. K. Speech discrimination by dynamic programming. *Cybernetics* vol. 4 52–57 Preprint at https://doi.org/10.1007/bf01074755 (1972).
17. Allison, L., Wallace, C. S. & Yee, C. N. Finite-state models in the alignment of macromolecules. *J. Mol. Evol.* **35**, 77–89 (1992).
18. Durbin, R. *et al. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. (Cambridge University Press, 1998).
19. Sumanaweera, D., Allison, L. & Konagurthu, A. S. Statistical compression of protein sequences and inference of marginal probability landscapes over competing alignments using finite state models and Dirichlet priors. *Bioinformatics* **35**, i360–i369 (2019).
20. Wallace, C. S. & Boulton, D. M. An Information Measure for Classification. *The Computer Journal* vol. 11 185–194 Preprint at https://doi.org/10.1093/comjnl/11.2.185 (1968).
21. Wallace, C. S. & Freeman, P. R. Estimation and Inference by Compact Coding. *Journal of the Royal Statistical Society: Series B (Methodological)* vol. 49 240–252 Preprint at https://doi.org/10.1111/j.2517-6161.1987.tb01695.x (1987).
22. Wallace, C. S. *Statistical and Inductive Inference by Minimum Message Length*. (Springer Science & Business Media, 2005).
23. Shalek, A. K. *et al.* Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. *Nature* **510**, 363–369 (2014).
24. Penfold, C. A. *et al.* Branch-recombinant Gaussian processes for analysis of perturbations in biological time series. *Bioinformatics* vol. 34 i1005–i1013 Preprint at https://doi.org/10.1093/bioinformatics/bty603 (2018).
25. Website. Wilson, Andrew Gordon. n.d. ‘The Change Point Kernel.’ Accessed February 7, 2023.

http://www.cs.cmu.edu/~andrewgw/changepoints.pdf.

26. Bastidas-Ponce, A. *et al.* Comprehensive single cell mRNA profiling reveals a detailed roadmap for pancreatic endocrinogenesis. *Development* **146**, (2019).

27. Otero, K. *et al.* Nonredundant role of CCRL2 in lung dendritic cell trafficking. *Blood* **116**, 2942–2949 (2010).

28. Schioppa, T. *et al.* Molecular Basis for CCRL2 Regulation of Leukocyte Migration. *Front Cell Dev Biol* **8**, 615031 (2020).

29. Park, J.-E. *et al.* A cell atlas of human thymic development defines T cell repertoire formation. *Science* **367**, (2020).

30. Suo, C. *et al.* Mapping the developing human immune system across organs. *Science* **376**, eabo0510 (2022).

31. Montel-Hagen, A. *et al.* Organoid-Induced Differentiation of Conventional T Cells from Human Pluripotent Stem Cells. *Cell Stem Cell* **24**, 376–389.e8 (2019).

32. Domínguez Conde, C. *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science* **376**, eabl5197 (2022).

33. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).

34. Titsias, M. & Lawrence, N. D. Bayesian Gaussian Process Latent Variable Model. in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* 844–851 (PMLR, 13--15 May 2010).

35. Kumasaka, N. *et al.* Mapping interindividual dynamics of innate immune response at single-cell resolution. Preprint at https://doi.org/10.1101/2021.09.01.457774.

36. Lalchand, V. *et al.* Modelling Technical and Biological Effects in scRNA-seq data with Scalable GPLVMs. in *Proceedings of the 17th Machine Learning in Computational Biology meeting* (2022).

37. Ahmed, S., Rattray, M. & Boukouvalas, A. GrandPrix: Scaling up the Bayesian GPLVM for single-cell data. *Bioinformatics* **35**, (2018).

38. Verma, A. & Engelhardt, B. E. A robust nonlinear low-dimensional manifold for single cell

RNA-seq data. Preprint at https://doi.org/10.1101/443044.

39. Lönnberg, T. *et al.* Single-cell RNA-seq and computational analysis using temporal mixture modelling resolves Th1/Tfh fate bifurcation in malaria. *Sci Immunol* **2**, (2017).

40. Lambert, S. A. *et al.* The Human Transcription Factors. *Cell* **175**, 598–599 (2018).

41. Niwa, H. How is pluripotency determined and maintained? *Development* **134**, 635–646 (2007).

42. Bedford, F. K., Ashworth, A., Enver, T. & Wiedemann, L. M. HEX: a novel homeobox gene expressed during haematopoiesis and conserved between mouse and human. *Nucleic Acids Res.* **21**, 1245–1249 (1993).

43. Jackson, J. T. *et al.* Hhex Regulates Hematopoietic Stem Cell Self-Renewal and Stress Hematopoiesis via Repression of Cdkn2a. *Stem Cells* **35**, 1948–1957 (2017).

44. Hosokawa, H. & Rothenberg, E. V. How transcription factors drive choice of the T cell fate. *Nat. Rev. Immunol.* **21**, 162–176 (2021).

45. Webb, L. V., Ley, S. C. & Seddon, B. TNF activation of NF-κB is essential for development of single-positive thymocytes. *J. Exp. Med.* **213**, 1399–1407 (2016).

46. Xing, Y., Wang, X., Jameson, S. C. & Hogquist, K. A. Late stages of T cell maturation in the thymus involve NF-κB and tonic type I interferon signaling. *Nat. Immunol.* **17**, 565–573 (2016).

47. Mo, J.-S., Park, H. W. & Guan, K.-L. The Hippo signaling pathway in stem cell biology and cancer. *EMBO Rep.* **15**, 642–656 (2014).

48. Feng, X. *et al.* Foxp1 is an essential transcriptional regulator for the generation of quiescent naive T cells during thymocyte development. *Blood* **115**, 510–518 (2010).

49. Kaminskiy, Y., Kuznetsova, V., Kudriaeva, A., Zmievskaya, E. & Bulatov, E. Neglected, yet significant role of FOXP1 in T-cell quiescence, differentiation and exhaustion. *Front. Immunol.* **13**, 971045 (2022).

50. Cichocki, F. *et al.* ARID5B regulates metabolic programming in human adaptive NK cells. *J. Exp. Med.* **215**, 2379–2395 (2018).

51. Cook, M. E., Jarjour, N. N., Lin, C.-C. & Edelson, B. T. Transcription Factor Bhlhe40 in Immunity and Autoimmunity. *Trends Immunol.* **41**, 1023–1036 (2020).

52. Verleysen, M. & François, D. The Curse of Dimensionality in Data Mining and Time Series

Prediction. *Computational Intelligence and Bioinspired Systems* 758–770 Preprint at https://doi.org/10.1007/11494669_93 (2005).

53. Garfinkel, A., Shevtsov, J. & Guo, Y. *Modeling Life: The Mathematics of Biological Systems*. (Springer, 2017).

54. Shannon, C. E. A Mathematical Theory of Communication. *Bell System Technical Journal* vol. 27 623–656 Preprint at https://doi.org/10.1002/j.1538-7305.1948.tb00917.x (1948).

55. Allison, L. Coding Ockham's Razor. Springer Preprint at https://doi.org/10.1007/978-3-319-76433-7 (2018).

56. Collier, J. H. Statistical inductive inference of protein structural alignments. (Monash University, 2016).

57. Conway, J. H. & Sloane, N. J. A. On the Voronoi Regions of Certain Lattices. *SIAM Journal on Algebraic Discrete Methods* vol. 5 294–305 Preprint at https://doi.org/10.1137/0605031 (1984).

58. Rajapaksa, S. *et al.* On the reliability and the limits of inference of amino acid sequence alignments. *Bioinformatics* **38**, i255–i263 (2022).

59. Chen, E. Y. *et al.* Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool. *BMC Bioinformatics* vol. 14 Preprint at https://doi.org/10.1186/1471-2105-14-128 (2013).

60. Fang, Z., Liu, X. & Peltz, G. GSEApy: a comprehensive package for performing gene set enrichment analysis in Python. *Bioinformatics* **39**, (2023).

61. Liberzon, A. *et al.* The Molecular Signatures Database (MSigDB) hallmark gene set collection. *Cell Syst* **1**, 417–425 (2015).

62. Kanehisa, M., Furumichi, M., Sato, Y., Ishiguro-Watanabe, M. & Tanabe, M. KEGG: integrating viruses and cellular organisms. *Nucleic Acids Research* vol. 49 D545–D551 Preprint at https://doi.org/10.1093/nar/gkaa970 (2021).

63. Lange, M. *et al.* CellRank for directed single-cell fate mapping. *Nat. Methods* **19**, 159–170 (2022).

64. Cannoodt, R., Saelens, W., Deconinck, L. & Saeys, Y. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nat. Commun.* **12**, 1–9 (2021).

65. Wolock, S. L., Lopez, R. & Klein, A. M. Scrublet: Computational Identification of Cell Doublets

in Single-Cell Transcriptomic Data. *Cell Syst* **8**, 281–291.e9 (2019).

66. Heaton, H. *et al.* Souporcell: robust clustering of single-cell RNA-seq data by genotype without reference genotypes. *Nat. Methods* **17**, 615–620 (2020).

67. Tirosh, I. *et al.* Dissecting the multicellular ecosystem of metastatic melanoma by single-cell RNA-seq. *Science* **352**, 189–196 (2016).

68. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).

69. Website. Bingham, Eli, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Rohit Singh, Paul Szerlip, Paul Horsfall, Noah D. Goodman, Neeraj Pradhan, and Theofanis Karaletsos. 2019. 'Pyro: Deep Universal Probabilistic Programming.' Journal of Machine Learning Research: JMLR. https://www.jmlr.org/papers/volume20/18-403/18-403.pdf.

70. Polański, K. *et al.* BBKNN: fast batch alignment of single cell transcriptomes. *Bioinformatics* **36**, 964–965 (2020).

Methods

Genes2Genes (G2G): A new alignment framework for single-cell trajectories

As described in the main text, Genes2Genes performs dynamic programming (DP) alignment independently for all genes of interest, between a reference trajectory R and a query trajectory Q. In other words, each gene-level (i.e. gene-specific) trajectory alignment is an independent DP task of pairwise time series alignment. The aim is to generate an optimal sequence of matched time point pairs and mismatched time point pairs between R and Q for each gene. As illustrated in **Fig. 1b**, there are five different alignment states which denote these matches and mismatches between two time points. For each time point in any gene trajectory, there is a respective expression distribution, as explained by an observed dataset of single-cell (scRNA-seq) measurements. G2G evaluates the similarities of these reference and query expression distributions over time, to determine a match or mismatch between their time points.

The following sections introduce the problem of pairwise time series alignment, and describe the main components of our G2G framework (**Fig. 2**) which operate together to produce optimal gene-specific alignments.

Pairwise time series alignment for trajectory comparison

A trajectory is a continuous path of change through some feature space, along some axis of progression (such as time)[53]. In single-cell transcriptomics, this feature space is usually defined by genes, and a trajectory through a high-dimensional gene space can describe the transcriptomic state of a cell as a function of time. A temporal (e.g. pseudotime) ordering of a set of single cells represents a discretization of the respective cell state trajectory, and their entire gene expression dataset forms a multivariate time series. On the other hand, their expression of a single gene forms a univariate time series. In this work, we consider a pairwise alignment of univariate time series, which allows us to perform gene-specific trajectory alignment.

Given two time series (sequences), reference R and query Q of length (i.e. a finite number of time points) $|R|$ and $|Q|$, their pairwise alignment describes sequential correspondences between their time points. As an optimization problem, computational alignment has two key properties: (1) an optimal substructure, and (2) overlapping set of subproblems, which make it dynamic programmable (DP)[3]. Property (1) means, the optimal alignment of any two prefixes $R_{1..j}$ and $Q_{1..i}$ depends on the optimality of three sub-alignments: (i) $R_{1..j-1}$ and $Q_{1..i-1}$, (ii) $R_{1..j-1}$ and $Q_{1..i}$, and (iii) $R_{1..j}$ and $Q_{1..i-1}$. Property (2) means, there exists subproblems (prefix alignments) that are overlapping. DP begins optimizing the alignment of prefixes, starting from null (Φ) sequences until it completes an alignment of the entire two sequences. During this process, it computes overlapping subproblems only once and reuses them through a memoization (history) matrix H. In the standard DP alignment algorithm, any cell (i, j) in H stores the optimal alignment cost of the two prefix sequences: $R_{1:j}$ and $Q_{1:i}$, by optimizing an objective function which quantifies the alignment through a set of recurrence relations. Once all H matrix cells are computed, the optimal alignment can be retrieved by backtracking, starting from the right-most bottom cell $(|Q| + 1, |R| + 1)$ until reaching the matrix cell $(0,0)$. The time complexity of this algorithm depends on how its alignment scoring scheme is designed. (The standard scheme has a quadratic complexity to find the best alignment out of the factorially growing all possible number of alignments).

Preprocessing a trajectory time series by distributional interpolation

Interpolation is a necessary preprocessing step that a time series has to undergo prior to taking part in an alignment. This is to ensure smoothly changing and uniformly distributed data that are in phase (i.e. having the same rate of sampling) at least approximately; otherwise a reliable alignment cannot be guaranteed[5,12]. Here we chose to extend the mean gene expression based interpolation method used by CellAlign[5] to a distributional interpolation, for preprocessing a reference and query time series of gene expression before their alignment.

Given a pseudotime series t of (log1p normalized) expression in some gene g_j of a single-cell dataset, our distributional interpolation method first min-max-normalizes the pseudotime of t as to be in the range of $[0,1]$. Then, m equally spaced artificial (interpolated) time points are determined within $[0,1]$, where for each artificial time point t', we estimate a Gaussian distribution (of mean $g_j(t')_{mean}$ and standard deviation $g_j(t')_{std}$) by using the Gaussian kernel-based weighted approach. For each cell i annotated with pseudotime t_i , an associated weight is computed w.r.t each artificial time point t' as:

$$w_i = exp(-\frac{(t_i - t')^2}{window\_size^2})$$

, where default $window\_size$ = 0.1. The below equations are then used to compute the Gaussian distribution parameters $g_j(t')_{mean}$ and $g_j(t')_{std}$:

$$g_j(t')_{mean} = \frac{1}{\Sigma w_i} \sum_{i=1}^{n} w_i \, g_j(t_i)$$

$$g_j(t')_{std} = c_{t'} \sqrt{n \frac{\sum_{i=1}^{n} w_i \, [g_{j\_mean} - g_j(t_i)]^2}{(n-1)\Sigma w_i}}$$

$$\text{where } g_{j\_mean} = \frac{\sum_{i=1}^{n} g_j(t_i)}{n},$$

, n is the total number of cells, and $c_{t'}$ is the weighted cell density (abundance of cells) at the interpolated time point t'. The weighted cell density computed as:

$$c_{t'} = \frac{\sum_{i=1}^{n} w_i}{n},$$

is the expected weight of a cell at t'. A higher expected weight is indicative of a higher cell density. This way, we account for cell abundance when deciding the variance for the interpolated point (otherwise with a very low number of cells, we may get a very high variance). Next, we generate $k = 50$ random data points from the Gaussian distribution $N(g_j(t')_{mean} , g_j(t')_{std})$ for each interpolated time t', representing the interpolated distribution of single-cell gene expression. Note: In this work, we use a fixed number of interpolated time points m for both reference and query. As m controls the resolution of the alignment, it should suffice to represent the entire trajectory. This interpolation also adds an O(nm) time complexity due to taking a weighted contribution from all cells at each t'. To overcome this, a general solution is to subsample datasets before the analysis and/or to reduce the number of contributing cells by considering only the nearest neighborhood.

The reference pseudotime series and query pseudotime series of each gene is preprocessed using the above described distributional interpolation method. The interpolated time series are then input to our G2G dynamic programming algorithm as detailed in the next section.

A new dynamic programming algorithm for time series alignment of a single gene

Here we describe a new dynamic programming (DP) algorithm to generate an alignment between a reference time series R and query time series Q of log-normalized expression of a specified gene. This algorithm jointly adapts Gotoh's sequence alignment[15] with classical dynamic time warping (DTW)[16] to accommodate five states of alignment (**Fig. 1b**), i.e., one-to-one match (m), many-to-one warp (w_d), one-to-many warp (w_i), insertion (i), and deletion (d) between time points in the two time series. We denote the five-state space as Ω = [m, w_d, w_i, d, i]. Our approach unifies matches and mismatches within a single DP algorithm unlike DTW which only handles matches (including warps).

The DTW algorithm originated from the speech recognition domain[16] under the family of dynamic programming algorithms for optimization[3]. It has been extensively used to align time series with shifts (warps). Sankoff and Kruskal (1983)[12] had previously discussed how to capture warps and indels both from a single alignment algorithm. They provided a DP recurrence relation involving evaluations of the five alignment states to decide the optimal state for each pair of R and Q time points when aligning two time series. Extending this idea further, we implemented the Gotoh's O($|R||Q|$) DP algorithm to generate an optimal five-state alignment hypothesis for R and Q time series by:

- Defining a Bayesian information-theoretic measure of distance between two gene expression distributions using the minimum message length statistical inductive inference framework[12,21].
- Defining a five-state machine that models state transitions in the alignment hypothesis across the five different matching and mismatching states, defining one-to-one matches, warps (i.e. many-to-one compression matches and one-to-many expansion matches), and insertions-deletions (indels).

The scoring scheme of the DP algorithm evaluates every pair of reference time point j and query time point i to generate an optimal alignment across all time points. This involves computing two types of cost: (1) the cost of matching the two time points, j and i (denoted by $Cost_{match}(i,j)$) based on their respective (interpolated) gene expression distributions, and (2) the cost of assigning an alignment state $x \in \Omega$ for the two time points j and i. The following sections first detail on how we compute these costs, and then describe how our DP optimization works using this scoring scheme.

Note: The reference time point j and query time point i are also denoted by R_j and Q_i in the main text.

The DP scoring scheme

The cost of match between the reference time point j and query time point i

We expect a match between the reference time point j and query time point i if they have similar distributions of gene expression. Thus, to score the likelihood of a match, we define a distance measure between the two gene expression distributions corresponding to the j and i time points, respectively. To compute this distance, we first take the interpolated single-cell expression datasets at time point j of R (denoted by $R(j)$) and time point i of Q (denoted by $Q(i)$). We already know the mean (μ) and standard deviation (σ) statistics for $R(j)$ and $Q(i)$ separately, as they were estimated during the time series interpolation step. Thus we define the Gaussian distribution $N(\mu_{R(j)}, \sigma_{R(j)})$ for $R(j)$, and the Gaussian distribution $N(\mu_{Q(i)}, \sigma_{Q(i)})$ for $Q(i)$ using their respective μ and σ statistics. Accordingly, if $D_{R(j)} = [d_1, d_2, \dots, d_{|R(j)|}]$ and $D_{Q(i)} = [d_1, d_2, \dots, d_{|Q(i)|}]$ are the expression data vectors of the $R(j)$ and $Q(i)$ datasets, respectively, then:

$$d_k \sim N(\mu_{R(j)}, \sigma_{R(j)}) \;\; \forall \, d_k \; \in D_{R(j)}$$

$$d_k \sim N(\mu_{Q(i)}, \sigma_{Q(i)}) \; \forall \, d_k \; \in D_{Q(i)}$$

To be brief, we denote $N(\mu_{R(j)}, \sigma_{R(j)})$ distribution by $N_{R(j)}$, and $N(\mu_{Q(i)}, \sigma_{Q(i)})$ by $N_{Q(i)}$.

Next, we implement the cost function, $Cost_{match}(i,j)$, to consider both:

1. **Data:** $D_{R(j)}$ and $D_{Q(i)}$ expression vectors of the $R(j)$ and $Q(i)$ datasets, respectively,
2. **Models:** Gaussian distributions, $N_{R(j)}$ and $N_{Q(i)}$,

when computing the distance between $R(j)$ and $Q(i)$. To do so, we use the minimum message length (MML) criterion[21,22] and define a Bayesian information-theoretic distance measure. **Fig. 2** (top left) illustrates an abstract overview of our MML framework and its place in the overall G2G alignment framework. **Supplementary Fig. 1a** further expands this illustration to explain how the $Cost_{match}(i,j)$ computation works for a pair of reference time point j and query time point i, as detailed by the following sections.

Primer on minimum message length inference (MML)

MML[20-22] is an inductive inference paradigm for model comparison and selection, grounded on Bayesian statistics, information and coding theory. It facilitates designing hypothesis test schemes specific to a problem domain. Given a hypothesis (model) H and some data D, it lays an imaginary message transmission from a sender who jointly encodes H and D, aiming for their lossless decoding at a recipients' side. Bayes theorem defines their joint probability as:

$$Pr(H,D) = Pr(H) \cdot Pr(D|H) = Pr(D) \cdot \; Pr(H|D).$$

Separately, Shannon information defines the optimal length of a message that encodes some event E with a probability $Pr(E)$ as:

$$I(E) = -log_e(Pr(E))$$

measured in nits, where I denotes information. By applying the Shannon information[54] to Bayes theorem, we can map the respective probability elements in $Pr(H,D) = Pr(H) \cdot Pr(D|H)$ onto the information space, describing the amount of Shannon information needed to encode H and D jointly as:

$$I(H,D) = I(H) + I(D|H) \quad \textbf{– Equation (1)}$$

This gives a two-part total message length of encoding H and D jointly. The first part $I(H)$ refers to the message length of encoding the hypothesis H itself, whereas the second part $I(D|H)$ refers to the message length of encoding the data points in D using H.

When there are two hypotheses, H_1 and H_2, that describe the same data D, MML enables us to select the best hypothesis that gives a model-complexity vs. model-fit tradeoff, by evaluating a compression statistic $\Delta = I(H_1,D) - I(H_2,D)$. Here, Δ is also the log odds posterior ratio between the two hypotheses.

$$\Delta = log(\frac{Pr(H_2,D)}{Pr(H_1,D)}) = log(\frac{Pr(D)Pr(H_2|D)}{Pr(D)Pr(H_1|D)}) = log(\frac{Pr(H_2|D)}{Pr(H_1|D)}) \textbf{ – Equation (2)}$$

If Δ>0, this implies that the hypothesis H_2 is e^{Δ} times more likely than H_1, and vice versa.

Casting the cost of matching between reference time point j and query time point* i *under MML

Given the expression data D (containing both $D_{R(j)}$ and $D_{Q(i)}$) and their estimated Gaussian distributions ($N(\mu_{R(j)}, \sigma_{R(j)})$ and $N(\mu_{Q(i)}, \sigma_{Q(i)})$ denoted by $N_{R(j)}$ and $N_{Q(i)}$, respectively), we formulate two different hypotheses:

1. **Hypothesis A:** assumes that the two time points match, and thus explains data D with a single, representative Gaussian distribution $N(\mu_*, \sigma_*)$ denoted by N_* (which is either $N_{R(j)}$ or $N_{Q(i)}$).
2. **Hypothesis Φ:** assumes that the two time points mismatch, and thus explains data $D_{R(j)}$ with $N_{R(j)}$, and data $D_{Q(i)}$ with $N_{Q(i)}$, independently.

We then compute the two message lengths: $I(A,D)$ and $I(\Phi,D)$ according to **Equation 1** in the above described MML formulation:

$$I(A,D) = I(A) + I(D|A) \textbf{ – Equation (3)}$$

$$I(\Phi,D) = I(\Phi) + I(D|\Phi) \textbf{ – Equation (4)}$$

where, $A = [N_*]$ and $\Phi = [N_{R(j)}, N_{Q(i)}]$. $I(A,D)$ is the total message length of encoding A and D jointly, where $I(A)$ refers to the encoding length of the parameters μ_*, σ_* of N_*distribution. $I(D|A)$ refers to the encoding length of all data points in D based on their likelihood under $N(\mu_*, \sigma_*)$. Accordingly, we can re-write and expand **Equation 3** to:

$$\begin{aligned} I(A,D) &= I(N_*) + I(D|N_*) \\ &= I(\mu_*, \sigma_*) + I(D|\mu_*, \sigma_*) \end{aligned}$$

On the other hand, $I(\Phi,D)$ is the total message length of encoding Φ and D jointly, where $I(\Phi)$ refers to the sum of the independent encoding lengths of parameters $\mu_{R(j)}, \sigma_{R(j)}$ of $N_{R(j)}$ and $\mu_{Q(i)}, \sigma_{Q(i)}$ of $N_{Q(i)}$. $I(D|\Phi)$ refers to the sum of the independent encoding lengths of all data points in $D_{R(j)}$ and all data points in $D_{Q(i)}$ based on their likelihood under their respective Gaussian distributions: $N(\mu_{R(j)}, \sigma_{R(j)})$ and $N(\mu_{Q(i)}, \sigma_{Q(i)})$. Accordingly, we can re-write and expand **Equation 4** to:

$$I(\Phi, D) = I(N_{R(j)}) + I(N_{Q(i)}) + I(D|N_{R(j)}, N_{Q(i)})$$

$$= I(\mu_{R(j)}, \sigma_{R(j)}\ ) + I(\mu_{Q(i)}, \sigma_{Q(i)}\ ) + I(D_{R(j)}|\mu_{R(j)}, \sigma_{R(j)}) + I(D_{Q(i)}|\mu_{Q(i)}, \sigma_{Q(i)})$$

Note: See the next section for the equations used to compute each term in **Equation 3** and **Equation 4.**

Next, we normalize each total message length to compute a per datum message length (i.e. entropy), by dividing them by the total number of datapoints (single-cells) in D.

$$I(A, D)_{entropy} = \frac{I(A, D)}{|D_{R(j)}| + |D_{Q(i)}|}$$

$$I(\Phi, D)_{entropy} = \frac{I(\Phi, D)}{|D_{R(j)}| + |D_{Q(i)}|}$$

Note: to make the $I(A, D)_{entropy}$ measure symmetric, we take the average:

$$I(A, D)_{entropy} = \frac{I(N_{R(j)}, D)_{entropy} + I(N_{Q(i)}, D)_{entropy}}{2} \text{ nits per datum}$$

Afterwards, we compute a compression statistic Δ, which is taken as our $Cost_{match}(i, j)$:

$$\Delta = I(A, D)_{entropy} - I(\Phi, D)_{entropy}$$

$$Cost_{match}(i, j) = \Delta$$

As in **Equation 2,** this reflects the log odds posterior ratio:

$$I(A, D) - I(\Phi, D) = log[\frac{Pr(D) \cdot Pr(\Phi|D)}{Pr(D) \cdot Pr(A|D)}] = log[\frac{Pr(\Phi|D)}{Pr(A|D)}]$$

When $R(j)$ and $Q(i)$ are very dissimilar, the total encoding length under hypothesis A (i.e. the time points match) results in a larger value compared to that of hypothesis Φ (i.e. the time points mismatch). Thus, $Cost_{match}(i, j)$ increases as the distributions deviate from each other (**Extended Data Fig. 1b, Supplementary Fig. 1b,c**).

Computing the total encoding message length for any Gaussian model N_* and data D of size N

The above described $Cost_{match}(i, j)$ distance measure is computed using the standard MML Wallace Freeman approximation[21,54] defined for a Gaussian distribution[22,55,56]. As defined by **Equation 1,** for any dataset D and a hypothesis H that describes $D = [x_1, x_2, \dots, x_N]$ under a Gaussian distribution $N(\mu, \sigma)$ with parameters $\vec{\theta} = (\mu, \sigma)$, the total message length of encoding H and D jointly is given by:

$$I(H, D) = I(\vec{\theta}, D) = I(\vec{\theta}) + I(D|\vec{\theta})$$

MML Wallace Freeman approximation expands this to:

$$I(\vec{\theta}, D) = \frac{d}{2} log(\kappa_d) - log[h(\vec{\theta})] + \frac{1}{2} log(det[Fisher(\vec{\theta})]) + L(\vec{\theta}) + \frac{d}{2}$$

, where d is the number of free parameters ($d = 2$ for a Gaussian), and κ_d is the Conway lattice constant[57] (κ_d is $\frac{5}{36\sqrt{3}}$ for $d = 2$). Here: $h(\vec{\theta})$ is the prior over the parameters. μ is defined with a uniform prior over a predefined range of length R_μ. $log(\sigma)$ is defined with a uniform prior over a predefined range of length R_σ. Accordingly,

$$h(\vec{\theta}) = h(\mu) \cdot h(\sigma) \ = \ (\frac{1}{R_\mu})(\frac{1}{\sigma R_\sigma})$$

$$\Rightarrow I[h(\vec{\theta})] \ = -log[h(\vec{\theta})] \ = \ log(\sigma) \ + log(R_\mu R_\sigma)$$

We use R_μ=15.0 and R_σ= 3.0 as reasonable for log normalized gene expression data.

$L(\vec{\theta})$ is the negative log likelihood:

$$L(\vec{\theta}) = N \, log(\sigma) \ + \frac{N}{2} log(2\pi) \ + \frac{1}{2\sigma^2} \sum_{i=1}^{N} (x_i - \mu)^2 \ \ - \sum_{i=1}^{N} log(\epsilon)$$

where, ϵ is the precision of measurement for each data point (taken as ϵ=0.001).

$det[Fisher(\vec{\theta})]$ is the determinant of the expected Fisher matrix (i.e. the matrix of the expected second derivatives of the negative log likelihood function). This determinant has the closed form: $\frac{2N^2}{\sigma^4}$.

The cost of alignment state assignment for the reference time point *j* and query time point *i*

The DP scoring scheme also involves computing a cost of assigning a certain alignment state $x \in \Omega$ =[m, w_d, w_i, d, i] for the two time points j and i. This state assignment cost is computed as the amount of Shannon information[54] required to encode state x given the assigned state y for the preceding time points. As previously said, in information theory, Shannon information defines the optimal length of a message that encodes some event E with a probability $Pr(E)$ as: $I(E) = -log_e(Pr(E))$ measured in nits, where I denotes information. Accordingly, the cost of assigning state x given a previous state y is: $I(x|y) = -log_e(Pr(x|y))$. We define a five-state machine (**Extended Data Fig. 1a**) to explain these conditional probabilities of state assignments (a.k.a state transitions).

This finite state machine extends the general three-state alignment machine[17,18] which has a match (m) state, delete (d) state, and insert (i) state, by adding two new states: compression warp (w_d) state and expansion warp (w_i) state (**Fig. 1b**). The warp states are equivalent to the match state but are extensions to accommodate one-to-many and many-to-one matches between the two series, respectively. As in the three-state alignment machine[19], we enforce symmetry, while prohibiting an invalid transition from an indel state to a warp state. That is, we do not allow i→w_d and d→w_i, as they can be covered by a single m state in the first place. On the other hand, we have the choice of allowing d→w_d and i→w_i, as there can be a legitimate case of a warp match after an insertion or deletion. Note: all the outgoing transitions of each state in this finite state machine add up to a probability of 1. It also treats <i and d> and <w_i and w_d> equivalently. Consequently, there are 23 total number of state transitions in this machine, yet there are only three free transition probability parameters: $[Pr(m|m), Pr(i|i), Pr(m|i)]$, due to the symmetry and characteristics of the machine. These probabilities control the expected lengths of a match and a mismatch. In this work, we have chosen reasonable values [$Pr(m|m)$=0.99, $Pr(i|i)$=0.25, $Pr(m|i)$=0.4] based on manual tuning. However, we note that these parameters can be automatically inferred using an added layer of optimization and time complexity on top of the main DP optimization, which will be an interesting future direction to follow.

Altogether, our G2G DP scoring scheme utilizes the $Cost_{match}(i,j)$ function, and the above described state assignment costs (i.e. all possible state transition costs evaluated as $I(x|y)$ for all possible $y \rightarrow x$ state transitions under the five-state machine), are then used to define the recurrence relations of our DP algorithm.

Dynamic programming recurrence relations We formulate the DP problem using five history matrices $[Hist_m, Hist_{w_d}, Hist_{w_i}, Hist_d, Hist_i]$, where each matrix corresponds to each alignment state in Ω, respectively. Any history matrix, $Hist_x$ for state $x \in \Omega$, has the dimensions $(|Q| + 1 \times |R| + 1)$. In other words, the columns correspond to the time points in the reference series R, while the rows correspond to the time points in the query series Q. Each cell $Hist_x(i,j)$ stores the optimal cost of aligning the prefix time series $R[1..j]$ and $Q[1..i]$ ending in state x. The DP recurrence relations to compute each matrix cell (i,j) of each history matrix for $i > 0, j > 0$ are:

$$\mathtt{Hist}_{\mathtt{m}}(i,j) = \mathtt{min}\begin{cases} \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{m}}(i-1,j-1) + I(\mathtt{m}|\mathtt{m}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{w_d}}(i-1,j-1) + I(\mathtt{m}|\mathtt{w_d}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{w_i}}(i-1,j-1) + I(\mathtt{m}|\mathtt{w_i}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{d}}(i-1,j-1) + I(\mathtt{m}|\mathtt{d}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{i}}(i-1,j-1) + I(\mathtt{m}|\mathtt{i}) \end{cases}$$

$$\mathtt{Hist}_{\mathtt{w_d}}(i,j) = \mathtt{min}\begin{cases} \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{m}}(i,j-1) + I(\mathtt{w_d}|\mathtt{m}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{w_d}}(i,j-1) + I(\mathtt{w_d}|\mathtt{w_d}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{w_i}}(i,j-1) + I(\mathtt{w_d}|\mathtt{w_i}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{d}}(i,j-1) + I(\mathtt{w_d}|\mathtt{d}) \end{cases}$$

$$\mathtt{Hist}_{\mathtt{w_i}}(i,j) = \mathtt{min}\begin{cases} \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{m}}(i-1,j) + I(\mathtt{w_i}|\mathtt{m}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{w_d}}(i-1,j) + I(\mathtt{w_i}|\mathtt{w_d}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{w_i}}(i-1,j) + I(\mathtt{w_i}|\mathtt{w_i}) \\ \mathtt{Cost}_{match}(i,j) + \mathtt{Hist}_{\mathtt{i}}(i-1,j) + I(\mathtt{w_i}|\mathtt{i}) \end{cases}$$

$$\mathtt{Hist}_{\mathtt{d}}(i,j) = \mathtt{min}\begin{cases} \mathtt{Hist}_{\mathtt{m}}(i,j-1) + I(\mathtt{d}|\mathtt{m}) \\ \mathtt{Hist}_{\mathtt{w_d}}(i,j-1) + I(\mathtt{d}|\mathtt{w_d}) \\ \mathtt{Hist}_{\mathtt{w_i}}(i,j-1) + I(\mathtt{d}|\mathtt{w_i}) \\ \mathtt{Hist}_{\mathtt{d}}(i,j-1) + I(\mathtt{d}|\mathtt{d}) \\ \mathtt{Hist}_{\mathtt{i}}(i,j-1) + I(\mathtt{d}|\mathtt{i}) \end{cases}$$

$$\mathtt{Hist}_{\mathtt{i}}(i,j) = \mathtt{min}\begin{cases} \mathtt{Hist}_{\mathtt{m}}(i-1,j) + I(\mathtt{i}|\mathtt{m}) \\ \mathtt{Hist}_{\mathtt{w_d}}(i-1,j) + I(\mathtt{i}|\mathtt{w_d}) \\ \mathtt{Hist}_{\mathtt{w_i}}(i-1,j) + I(\mathtt{i}|\mathtt{w_i}) \\ \mathtt{Hist}_{\mathtt{d}}(i-1,j) + I(\mathtt{i}|\mathtt{d}) \\ \mathtt{Hist}_{\mathtt{i}}(i-1,j) + I(\mathtt{i}|\mathtt{i}) \end{cases}$$

As previously described, $Cost_{match}(i,j)$ measures the distance between the two interpolated gene expression distributions corresponding to the reference time point j and query time point i. The cost term $I(x|y)\ \forall\ x,y \in \Omega$ refers to the Shannon information of a state transition $y \rightarrow x$ (e.g. $I(i|m)$ is the cost of $m \rightarrow i$, computed as $-log_e[Pr(i|m)]$), based on the five-state alignment machine as explained before. Prior to computing the above relations, the history matrices are initialized as:

$$\mathtt{Hist}_{\mathtt{x}}(i,j)_{\mathtt{x}\in\{\mathtt{m},\mathtt{w_d},\mathtt{w_i}\}} = \begin{cases} 0 & \text{for } i>0, j>0 \\ \infty & \text{otherwise} \end{cases}$$
$$\mathtt{Hist}_{\mathtt{i}}(i,j) = \mathtt{Hist}_{\mathtt{i}}(i-1,0) + I(\mathtt{i}|\mathtt{i})$$
$$\mathtt{Hist}_{\mathtt{d}}(i,j) = \mathtt{Hist}_{\mathtt{d}}(0,j-1) + I(\mathtt{d}|\mathtt{d})$$

Note: For the cases of <$i = 1$ and $j = 1$> (i.e. before the first state transition), we assign a uniform transition cost: $I(m) = I(i) = I(d) = -log_e(1/3)$. All the five history matrices are computed by running the aforementioned DP algorithm. We then generate the optimal alignment Y^* as a five-state string by backtracking, starting from the cell:

$$\min \begin{cases} \mathtt{Hist}_{\mathtt{m}}(|Q|,|R|) \\ \mathtt{Hist}_{\mathtt{w_d}}(|Q|,|R|) \\ \mathtt{Hist}_{\mathtt{w_i}}(|Q|,|R|) \\ \mathtt{Hist}_{\mathtt{d}}(|Q|,|R|) \\ \mathtt{Hist}_{\mathtt{i}}(|Q|,|R|) \end{cases}$$

Note: The optimal alignment cost landscape matrix L can be visualized by constructing:

$$L(i,j) = min_{\forall x \in \Omega}\{Hist_x(i,j)\}$$

Y^* is the optimal alignment between R and Q time series that minimizes the total DP alignment cost under our DP scoring scheme and recurrence relations. In other words, Y^* describes the optimal set of reference time point and query time point pairs that are matched, as well as the optimal set of reference time points and query time points that are mismatched. The kth character in Y^*, i.e., $Y^*[k]$, gives the alignment state for the corresponding reference and query time points ($Y^*[k] \in \Omega$ =[m, w_d, w_i, i, d]). Let the set of matched time point pairs (i,j) in Y^* be denoted by $T_{matched}$. Then, the total alignment cost of Y^* is the sum of the total match cost (C_{match}) and the total state assignment cost (C_{state}), where:

$$C_{match}(Y^*) = \sum_{\forall (i,j)\in T_{matched}} Cost_{match}(i,j)$$

$$C_{state}(Y^*) = \sum_{k=1}^{|Y^*|} I(Y^*[k]|Y^*[k-1])$$

under our scoring scheme.

Overall, the optimal alignment Y^* is generated by optimizing the following objective function:

$$Y^* = argmin\ \forall \mathrm{Y}\in\mathbf{Y}\ \ \{C_{match}(Y) + C_{state}(Y)\}$$

where **Y** is the space of all possible five-state alignment hypotheses.

Note on using a custom $Cost_{match}(i,j)$ function: Our MML-based $Cost_{match}(i,j)$ function defines a distribution-based distance measure to compute the cost of matching the reference time point j and query time point i based on their gene expression distributions (as explained in the previous sections). Considering expression distributions rather than just the mean expression values allows us to make technical/batch variations implicit. However, we note that this can be any cost function (e.g. KL-divergence) which can measure the distance between two Gaussian distributions. However, our MML-based compression statistic Δ enables us to define a complete description of each hypothesis, which considers both model-complexity and data-fit. On the other hand, KL-divergence is equivalent to the expected log-likelihood ratio, which does not take the complexity of model parameters into account.

Reporting alignment statistics over gene-level alignments

G2G generates individual alignments for all genes of interest by running the DP alignment algorithm (detailed in the previous section) independently for each gene between the reference and query. Each optimal alignment output is a five-state alignment string describing the matches and mismatches. These gene-level alignments are further analyzed to generate useful statistics and insights as below.

Distribution of Alignment similarities The percentage of total matching (i.e. one-to-one matches and warps) (termed as 'alignment similarity' percentage) in each five-state gene-alignment string, as well as its average across all genes, provide quantitative measures of the degree of concordance between the reference and query. We also generate a single aggregate alignment across all genes using each of their optimal alignment landscapes. Recall that any matrix cell (i, j) in the optimal alignment landscape (i.e. $L(i, j) = min_{\forall x \in \Omega}\{Hist_x(i, j)\}$) refers to the optimal ending alignment state of the prefix time series $R_{1..j}$ and $Q_{1..i}$. Thus, across such matrices of all genes, there is a five-state frequency distribution for each (i, j). To generate an average alignment, we start a traversal from the right-most bottom cell $(|Q| + 1, |R| + 1)$, and choose the most probable alignment state $x \in \Omega$ for $R_{|R|}$ and $Q_{|Q|}$ time points as the most frequent state across all genes. According to this state, we traverse to the next matrix cell (i.e. if it is m, we go to $(i - 1, j - 1)$; if it is d, we go to $(i, j - 1)$ and so on). By the time we reach the cell $(0,0)$, we have a representative five-state alignment string.

Clustering five-state alignment strings Given a set of five-state alignment strings (i.e. gene-specific alignments), we employ a string clustering approach to identify groups of genes that show similar temporal matching and mismatching patterns. This requires the definition of a distance measure between two alignment paths. While the polygonal area based distance measure[58] is ideal for three-state alignment strings, it is unable to distinguish between warps and indels. Thus, we use a binary encoding scheme that transforms each five-state alignment string into a binary vector of size $|R| + |Q|$. This is done by traversing through the alignment path, recording for each trajectory, the match/mismatch state of their respective pseudotime points (i.e a match state $x \in [m, w_d, w_i]$ is encoded by 1; a mismatch state $x \in [i, d]$ is encoded by 0). The resultant binary strings of R and Q are then concatenated to numerically represent their alignment path. Next, the binary representation of each gene-specific alignment is used to compute a pairwise Hamming distance matrix between all pairs of alignments, which is then input to standard agglomerative hierarchical clustering (under the average linkage method; using the Python *sklearn.cluster* package). The threshold parameter for linkage distance controls the level at which the cluster merge stops, allowing inspection of the clusters at different levels of a clustering hierarchy.

Pathway overrepresentation analysis The alignment similarity percentage of each gene-specific alignment allows us to rank all the genes, from which we select the top k mismatching genes to analyze their pathway overrepresentation. The identified clusters of genes are also analyzed. We use the GSEApy Enrichr[58–60] wrapper against the *MSigDB_Hallmark_2020*[61] and *KEGG_2021_Human* pathway genesets[61,62]. For all analyses, a 0.05 significance threshold of the adjusted P-value (with the default FDR correction method used by GSEApy) was applied.

Datasets

Datasets for simulated experiments

Simulating pairwise datasets with different alignment patterns using Gaussian Processes

We modeled log-normalized expression of a gene x as a function f of time t using a Gaussian Process (GP), a stochastic process where any finite instantiation of it follows a multivariate Gaussian distribution. In other words, it is a distribution of functions, from which we can sample an $f(t)$:

$$f(t) \sim GP(\vec{\mu}(t), K(t,t'))$$

where, μ is the mean vector, and $K(t,t')$ is a kernel function which evaluates a covariance matrix covering every pair of finite time points where the f(t) is evaluated. The characteristics of this function are controlled by the class of the kernel being used (e.g. a Radial Basis Function (RBF) kernel for generating smooth, non branching functions; a change point kernel for generating branching functions). Therefore, a GP with an appropriate kernel is ideal to simulate different trajectory patterns in single-cell gene expression across pseudotime. Following the standard textbook and kernels discussed in literature[24,25], we implemented a simulator for three different types of pairwise trajectory patterns: (1) *Matching*, (2) *Divergence*, and (3) *Convergence*, across a pseudotime range [0,1] with 300 total number of simulated cells for each trajectory.

Generating a *Matching* pair of reference and query gene trajectories We used a GP with a constant c mean vector $\overrightarrow{\mu_c}$ ($c \in [0.5{,}9.0]$ uniform random sampled) and RBF kernel K to first sample a function $\mu(t)$ that describes an average expression value for each time point. Next, we sampled two gene expression trajectories: $GEX_{ref}(t)$ and $GEX_{query}(t)$, from a GP where the mean is $\mu(t)$ and the covariance matrix is $\sigma^2 I$ ($\sigma \in [0.05{,}1.0]$ uniform random sampled, and I= identity matrix).

$$\mu(t) \sim N(\vec{\mu}_c, K)$$

$$GEX_{ref}(t) \ \sim N(\mu(t), \sigma^2 I \ )$$

$$GEX_{query}(t) \sim N(\mu(t), \sigma^2 I)$$

Generating a *Divergence* pair of reference and query gene trajectories Here we use a Change Point (CP) kernel which imposes a bifurcation in the trajectory as it reaches an approximate time point t_{CP} (a.k.a. change point). The idea is to activate one covariance function before t_{CP} and another covariance function after t_{CP}. We used the below CP kernel[24,25] K_{CP}:

$$K_{CP}(t,t') = aK_1(t,t') + a'K_2(t,t')$$

where,

$$a = \sigma(t)\sigma(t')$$

$$a' = [1 - \sigma(t)]\,[1 - \sigma(t')]$$

$$\sigma(x) = \frac{1}{1+exp(-s(x-t_{CP}))} \text{ (sigmoid function)}$$

with s acting as a steepness parameter, deciding how steep the change point is. Penfold et al (2018)[24] defines a branching process by enforcing a zero kernel (K_1) imposed before t_{CP} and another suitable kernel (K_2) afterwards. We used an RBF for K_2. Following is the generative process starting with a base mean function $\mu(t)$ sampled from a separate GP with a constant c mean vector $\overrightarrow{\mu_c}$ ($c \in [0.5, 9.0]$ uniform randomly sampled), and an RBF kernel K.

$$\mu(t) \sim N(\overrightarrow{\mu_c}, K)$$

$$f_1(t) \sim N(\mu(t), K_{CP})$$

$$f_2(t) \sim N(\mu(t), K_{CP})$$

$$GEX_{ref}(t) \sim N(f_1(t), \sigma^2 I)$$

$$GEX_{query}(t) \sim N(f_2(t), \sigma^2 I)$$

Next, two functions were sampled from a GP with base $\mu(t)$ and CP(t,t'), which were then used as mean vectors to generate $GEX_{ref}(t)$ and $GEX_{query}(t)$ with covariance matrix $\sigma^2 I$ (σ=0.3). This was run for [t_{CP}=0.25, t_{CP}=0.5, t_{CP}= 0.75] to obtain 3 different groups of *Divergence* with varying bifurcation points. We use a constant σ which is not too low and not too high, and also apply a filtering criteria to ensure that the final dataset includes pairs of simple and clear divergence (a stable ground truth with no complex patterns). Pairs are filtered through basic heuristics such as the difference between mean expression before divergence and at the end terminals of reference and query.

Extended Data Fig. 2 shows that the branching effect may start approximately before the change point. Therefore, we expect the early non-divergent segment to continue at least until time point $i < t_{CP}$ where we begin to see > 0.01 covariance in the change point kernel. Accordingly, given our approx_bifurcation_start_point = i, we expect the range of match lengths to fall between a lower limit = $n\_total\_pseudotime\_points \times i$ and upper limit = $n\_total\_pseudotime\_points \times change\_point$.
Equivalently, we expect mismatch lengths to fall between $n\_total\_pseudotime\_points \times (1 - i)$ and upper limit = $n\_total\_pseudotime\_points \times (1 - change\_point)$.

Generating a *Convergence* pair of reference and query gene trajectories For *Convergence,* we simply inverted the above generated divergent trajectory pairs, as the *Convergence* and *Divergence* patterns are complementary to each other.

Simulating mismatches on real scRNA-seq data

We downloaded the mouse pancreas development dataset[26] from the CellRank python package[63]. We subset the dataset to include only cells in the beta-cell lineage, using the annotations from the original authors (selecting cells labeled as "Ngn3 low EP", "Ngn3 high EP", "Fev+", "Beta"), retaining 1845 cells. To select genes varying along beta-cell differentiation, we ran the CellRank pipeline (v1.5.1) to compute lineage drivers, following the package tutorials (https://cellrank.readthedocs.io/en/stable/auto_examples/estimators/compute_lineage_drivers.html) and selected genes significantly associated with differentiation potential to beta-cells (at 1% FDR we selected 769/2000 highly variable genes). For the pseudotime axis, we used the diffusion pseudotime estimated by the CellRank authors. To simulate trajectories for alignment, we divided the diffusion pseudotime (between 0 and 1) equally into 50 bins, assigned cells to the bins based on their estimated pseudotime and randomly split cells into query and reference datasets in each bin. To simulate deletions of *n* bins, we excluded query cells from the first *n* pseudotime bins (i.e. cells where the pseudotime value was less or equal to the upper margin of the *n*th bin). To simulate mismatches of *n* bins, we found the pseudotime bin with highest mean expression for the gene of interest in the query cells, and calculated mean (*max_mean*) and standard deviation (*max_std*) of expression of query cells for this bin; then, for each of the first *n* pseudotime bins, we substitute expression values of the query cells with a sample from a normal distribution with mean = *max_mean* + *max_std* and standard deviation = *max_std* . Pseudotime values for the query cells were min-max normalized to [0,1] after perturbation. We then ran the G2G alignment on each tested gene and calculated their alignment similarity (match calling) percentages.

Datasets for benchmarking G2G

Dendritic cell stimulation dataset

The normalized single-cell expression datasets of PAM/LPS stimulation and their pseudotime estimates were downloaded from the CellAlign[5] github repository and converted into *Anndata* objects. These contain two gene sets: 'core antiviral module' (99 genes) and 'peaked inflammatory module' (89 genes), pre-selected from the original publication[23] and referred to as global and local modules, respectively by Alpert et al (2018)[5]. The datasets include 179 PAM-stimulated cells and 290 LPS-stimulated cells for which the pseudotime has been estimated by CellAlign authors using Diffusion-maps.

Simulated dataset containing trajectories with no shared process

This is a simulated, negative control dataset which was generated using the published script by Laidlaw et al (2023) (in the TrAGEDy[11] Git repository). Their script uses DynGen[64], a single-cell data simulator for dynamic processes. The resulting negative dataset contains two trajectories simulated under two different gene regulatory networks and TF activity, ensuring that there is no shared process between them. The reference and query trajectories have 619 genes across 2000 and 1940 cells, respectively.

We reproduced the high-dimensional alignments from CellAlign and TrAGEDy over the 619 genes in this dataset by re-running the scripts provided by Laidlaw et al (2023). Here, CellAlign uses Euclidean distance, whereas TrAGEDy uses Spearman correlation. Note: For gene-level alignment, TrAGEDy was run using the Euclidean distance as in CellAlign; this is because Spearman correlation is mathematically undefined for a single gene dimension. We input log normalized, scaled gene expression data to CellAlign, following its documentation.

Dataset preparation for *in vivo in vitro* T cell development comparison

Cell cultures for artificial thymic organoid (ATO) and single-cell RNA sequencing experiment

MS5 line transduced with human DLL4 was obtained from G. Crooks (UCLA) as a gift. The MS5-hDLL4 cells were cultured in DMEM (Gibco) with 10% FBS. Two iPSC lines were used in this study. Cell lines HPSI0114i-kolf_2 (Kolf) and HPSI0514i-fiaj_1 (Fiaj) were obtained from the Human Induced Pluripotent Stem Cell initiative (HipSci: www.hipsci.org) collection. All iPSC lines were cultured on vitronectin (diluted 1:25 in PBS; Gibco) coated plates, in TeSR-E8 media (Stemcell Technologies).

We followed the PSC-ATO protocol as previously described[31]. iPSC cells were harvested as a single-cell suspension and seeded (3×10^6 cells per well) in GFR reduced Matrigel (Corning) - coated 6-well plates in X-VIVO 15 media (Lonza), supplemented with rhActivin A, rhBMP4, rhVEGF, rhFGF (all from R&D Systems), and ROCK inhibitor (Y27632; LKT Labs) on day −17, and only rhBMP4, rhVEGF and rhFGF on days −16 and −15. Cells were harvested 3.5 days later (day−14) , and isolated by FACS for $CD326^{-}CD56^{+}$ (PE anti-human CD326 antibody, Biolegend, 324205; APC anti-human CD56 antibody, Biolegend, 318309) human embryonic mesodermal progenitors (hEMPs).

Isolated hEMPs were combined with MS5-hDLL4 at a ratio of 1:50. Two or three cell-dense droplets (5×10^5 cells in 6 µl hematopoietic induction medium) were deposited on top of an insert in each well of a six-well plate. Hematopoietic induction medium composed of EGM2 (Lonza) supplemented with ROCK inhibitor and SB blocker (TGF-β receptor kinase inhibitor SB- 431542; Abcam) was added into the wells outside the inserts so that the cells sat at the air-liquid interface. The organoids were then cultured in EGM2 with SB blocker for 7 days (days −14 to −7), before the addition of cytokines rhSCF, rhFLT3L, rhTPO (all from Peprotech) between days −6 to 0. These 2 weeks formed the hematopoietic induction phase. On day 1, media was changed again to RB27 (RPMI supplemented with B27 (Gibco), ascorbic acid (Sigma-Aldrich), penicillin/streptomycin (Sigma-Aldrich) and glutamax (Thermo Fisher Scientific)) with rhSCF, rhFLT3L and rhIL7. The organoids can be maintained in culture for 7 more weeks in this medium.

For dissociation of organoids on day −7, they were removed from culture insert and incubated in digestion buffer, which consisted of collagenase type IV solution (StemCell Technologies) supplemented with 0.88mg/ml collagenase/dispase (Roche) and 50U DNase I (Sigma), for 20 minutes at 37°C. Vigorous pipetting was performed in the middle of the incubation and at the end. After complete disaggregation, single cell suspension was prepared by passing through a 50-µm strainer.

For dissociation of organoids from day 0 onwards, a cell scraper was used to detach ATOs from cell culture insert membranes and detached ATOs were then submerged in cold flow buffer (PBS (Gibco) containing 2% (v/v) fetal bovine serum (FBS; Gibco) and 2 mM EDTA (Invitrogen)). Culture inserts were washed and detached ATOs were pipetted up and down to form single-cell suspension before passing through a 50-µm strainer.

Cells were then stained with designed panels of antibodies and analyzed by flow cytometry. FACS was performed at the same time and live human DAPI^{-}anti-mouse CD29^{-} (APC/Cy7 anti-mouse CD29 antibody, Biolegend, 102225) cells were sorted for day −7, day 0 and week 3 samples, and live (DAPI–) cells were sorted for week 5 and week 7 samples before loading onto each channel of the Chromium chip from Chromium single cell V(D)J kit (10X Genomics). The metadata for all the ATO samples can be found in **Supplementary Table 50.** For the day −14 sample, some sorted (both hEMP and the rest of DAPI^{-} fraction) and unsorted cells were stained with hashtag antibodies (TotoalSeq-C antibodies from Biolegend, see **Supplementary Table 51**, following 10X cell surface protein labeling protocol) before being mixed together with some mouse stromal cells (MS5-hDLL4) for 10X loading. For week 1 sample, hashtag antibodies were added in at the same time as the FACS antibodies i.e., before sorting.

Single-cell cDNA synthesis, amplification and gene expression (GEX) and cell surface protein (CITE-seq) libraries were generated following the manufacturer’s instructions. Sequencing was performed on the Illumina Novaseq 6000 system. The gene expression libraries were sequenced at a target depth of 50,000 reads per cell using the following parameters: Read1: 26 cycles, i7: 8 cycles, i5: 0 cycles; Read2: 91 cycles to generate 75-bp paired-end reads.

ATO data preprocessing and annotation

Raw scRNA-seq reads were mapped with cellranger 3.0.2 with combined human reference of GRCh38.93 and mouse reference of mm10-3.1.0. Low quality cells were filtered out (minimum number of reads = 2000, minimum number of genes = 500, maximum number of genes = 7000, Scrublet[65] (v0.2.3) doublet detection score < 0.15, mitochondrial reads fraction < 0.2). Cells where the percentage of counts from human genes was < 90% were considered as mouse cells and excluded from downstream analysis. Cells were assigned to different cell lines (Kolf, Fiaj) using genotype prediction with Souporcell (v2.4.0)[66]. The mapping outputs of the eight samples were merged, with the sample ID prepended to the barcode IDs in both the BAM and barcodes.tsv to prevent erroneous cross-sample barcode overlap. Souporcell was run with `--skip_remap True --K 2` and the common variants file based on common (>= 2% population allele frequency) SNPs from 1000 genomes data, as distributed in the tool's repository. We selected 2 clusters due to the already known 2 cell lines. Next the data went through the standard pipeline of filtering out genes (cell cycle[67] genes, genes detected in less than 3 cells), and normalizing the per cell total count to 10,000 followed by log1p transformation and scaling to zero mean and unit variance (with max_value = 10 to clip after scaling), using SCANPY[68]. The final dataset had 31,483 ATO cells with 23,526 genes which were input to CellTypist[6] (for prediction using pre-trained logistic regression classifier – *Pan_Fetal_Human* model under majority voting). We then obtained a Uniform Manifold Approximation and Projection (UMAP) embedding for this dataset based on its scVI[33] batch corrected embedding (v0.14.5 with 10 latent dimensions (default), 2 hidden layers, 128 nodes per hidden layer (default), and 0.2 dropout rate for the neural network), and subsetted cells to non-hematopoietic lineage, T/ILC/NK lineage, and other hematopoietic lineage cells (**Extended data Fig. 8**) using their Leiden clustering. By default, scVI models gene counts with zero-inflated negative binomial distribution, defines a normal latent distribution, and handles batch effects. For each lineage, scVI latent dimensions and UMAP embedding were re-computed and cell types were annotated by inspecting both the CellTypist results and marker gene expression.

Joint Embedding of reference and organoid in preparation for pseudotime estimation

We downloaded the annotated human fetal atlas dataset from https://developmental.cellatlas.io/fetal-immune and extracted the cell types (79,535 cells in total) representing the T cell developmental trajectory starting from progenitor cells towards Type 1 Innate T cells (T1 dataset), including Cycling MPP, HSC_MPP, LMPP_MLP, DN(early) T, DN(P) T, DN(Q) T, DP(P) T, DP(Q) T, ABT(entry) and Type 1 innate T cells. We then compiled a reduced representation (20,384 cells) that preserve their underlying cell type composition. This was done by random subsampling from each cell type (with minimum sample size = 500 cells, aiming ~20,000 total number of cells) based on their originally published annotations. Such stratified-sampling approach is practical for dealing with massive single-cell datasets to reduce computational resource requirements. Separately, we extracted the cell types from the ATO dataset (19,013 cells) representing the trajectory starting from iPSCs towards SP T cells, including iPSC, primitive streak, mesodermal progenitor, endothelium, HSC_MPP, HSC_MPP/LMPP_MLP/DC2, DN(early) T, DN T, DP(P) T, DP(Q) T, ABT(entry), SP T cells.

Then the T1 and ATO datasets were merged and preprocessed together by filtering out cells with more than 8% total mitochondrial UMI, cell cycle genes[67] and genes expressed in less than 3 cells (`min_cells = 3`). Next, the highly variable genes were selected after per cell count normalization to 10,000 reads per cell and log1p normalization. The T1 pan fetal reference had 33 batches (due to different 10X chemistry 3' *versus* 5' and different donors), while the ATO had only 2 batches (due to 2 cell lines, Kolf and Fiaj). This merged dataset was then input to constructing a joint latent embedding using the scVI (v0.14.5) variational autoencoder implementation[33] (with 10 latent dimensions (default), 2 hidden layers, 128 nodes per hidden layer (default) and 0.2 dropout rate for the neural network). The joint embedding was then taken to build the cell neighborhood graph and UMAP embedding using SCANPY[68]. The final T1 and ATO datasets comprise 20,327 cells and 17,176 cells respectively. 18,436 cells of T1 and 10,089 cells of ATO belong to T cell lineage, i.e., DN T onwards.

We followed a similar preprocessing for the pan fetal reference representing the trajectory towards CD8+T (CD8 dataset) (including Cycling MPP, HSC_MPP, LMPP_MLP, DN(early) T, DN(P) T, DN(Q) T, DP(P) T, DP(Q) T, ABT(entry) and CD8+T cells). The initially extracted CD8 subset (83,177 cells) was reduced to 20,412 cells, which was then merged with the 19,013 ATO cells and subjected to the same filtering and normalization as for the T1+ATO merge prior to scVI integration. The final CD8 dataset comprises 20,324 cells of which, 18,490 cells are DN T onwards.

Pseudotime estimation using the Gaussian Process Latent Variable Model

The differentiation pseudotime was estimated separately for T1 reference, CD8 reference and ATO by employing the Gaussian Process Latent Variable Model (GPLVM)[34,37]. GPLVM is a probabilistic non-linear dimensionality reduction method which models observed gene expression as a function $f(X)$ of a set of latent covariates X. It enables us to incorporate Gaussian time priors when estimating pseudotime as a latent dimension. We used the Pyro[69] GPLVM implementation (Pyro v1.8.0) with Sparse Gaussian Process inference (32 inducing points) and Radial Basis Function kernel to obtain a 2D latent embedding, where the first latent dimension corresponds to pseudotime and the second latent dimension corresponds to a second level of latent effects (e.g. batch). The first dimension was assigned a Gaussian prior with cell capture times as the mean. The second latent dimension was zero initialized to allow for a second level of latent effects (e.g. batch). The model used Adam optimizer to infer the optimal latent embedding with 2000 iterations (where the loss curve reasonably converged).

For the ATO, the GPLVM was initialized with cell capture days as the prior. Since there was no temporal data present in the pan fetal reference, we first approximated time prior for each reference cell as the weighted average of their k-nearest organoid neighborhood (kNN) capture time. A k=3 organoid neighborhood for a reference cell was obtained using the cKDTree based search method implemented in BBKNN[70] on their scVI based UMAP embedding. Contribution of each organoid neighbor was weighted according to its distance. (kNN distance vector was softmax transformed, and the normalized reciprocal of each distance was taken as the associated weight, enforcing less contribution from distant neighbors towards the weighted average). This approximation may introduce outliers due to the spatial arrangement of different cell types in the UMAP. Thus, we leveraged the known cell-type annotations to refine the approximation by assigning each reference cell with the average approximated capture time of its cell type. These approximated capture times were scaled to be in [0,1] range and input as the mean prior to the previously described GPLVM. For T1 and CD8 GPLVMs, the input gene space was 2608 genes and 2616 genes respectively (same as at scVI integration). To ensure no outliers, the GPLVM estimated pseudotime was further refined by correcting outliers of each cell type using the cell-type specific average of estimated pseudotime. Outliers were selected based on the Interquartile Range (IQR) rule (i.e.

1.5 times IQR below the first quartile and above the third quartile of the cell-type specific pseudotime distribution).

Genes2Genes alignment

For the complete T1 vs ATO comparison using G2G, the total common gene space of 20,240 genes was considered upon filtering genes with less than 3 cells expressed, 10,000 total count per cell normalization, and log1p normalization. For the DN T onwards comparison, there were 17,718 genes for T1 vs. ATO, and 20,183 genes for CD8 vs. ATO. All pseudotime estimates were min-max normalized to ensure [0,1] range. These total gene spaces were subsetted to include only the transcription factors[40] (1371 TFs) and relevant signaling pathways focused in this work. G2G alignment was performed using 15 equispaced pseudotime points.

Software packages used in the work

Genes2Genes framework and all analysis related code (including plot generation) were implemented using the standard Python libraries (Numpy, Pandas, Seaborn, scikit-learn), GPyTorch, GSEApy. Illustrations were made using Adobe Illustrator 2023 and BioRender.

Code and data availability

Genes2Genes is implemented as an open-source package in Python 3 (v3.8) with tutorial available at: https://github.com/Teichlab/Genes2Genes. Code and data used to generate figures and perform analyses in the manuscript are available at: https://github.com/Teichlab/G2G_notebooks and https://drive.google.com/drive/folders/15LKmo3yRB-cR8Aq3aE59Taq2r0KtJSdX?usp=sharing. Raw sequencing data for newly generated sequencing libraries have been deposited in ArrayExpress (accession number E-MTAB-12720).

Acknowledgements

We thank the Crooks lab (A. Montel-Hagen, S. Lopez, and G. Crooks from University of California, Los Angeles) for their kind help in setting up the ATO experiments; N Huang, K Meyer, AM Cujba, L Dratva, R Lindeboom, R Elmentaite, A Maartens from the Wellcome Sanger Institute, CH Ek from University of Cambridge, Z Miao from Guangzhou Laboratory, Y Chen and T Wang from London School of Economics and Political Science for their helpful discussions; and BioRender.com for graphical illustrations. We gratefully acknowledge the Sanger Flow Cytometry Facility and Sanger Core Sequencing pipeline for support with sample processing and sequencing library preparation. We acknowledge Wellcome Sanger Institute as the source of HPSI0114i-kolf_2 and HPSI0514i-fiaj_1 human induced pluripotent cell lines which were generated under the Human Induced Pluripotent Stem Cell Initiative funded by a grant from the Wellcome Trust and Medical Research Council, supported by the Wellcome Trust (WT098051) and the NIHR/Wellcome Trust Clinical Research Facility, and acknowledge Life Science Technologies Corporation as the provider of Cytotune. This publication is part of the Human Cell Atlas - www.humancellatlas.org/publications.

Funding

D.S. is supported by the Marie Curie Individual Fellowship; This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No: 101026506. C.S. is supported by a Wellcome Trust Ph.D. Fellowship for Clinicians. S.A.T. is funded by Wellcome (WT206194), the ERC Consolidator Grant ThDEFINE (646794). This work was also supported by a grant from the Wellcome Sanger Institute’s Translation Committee Fund.

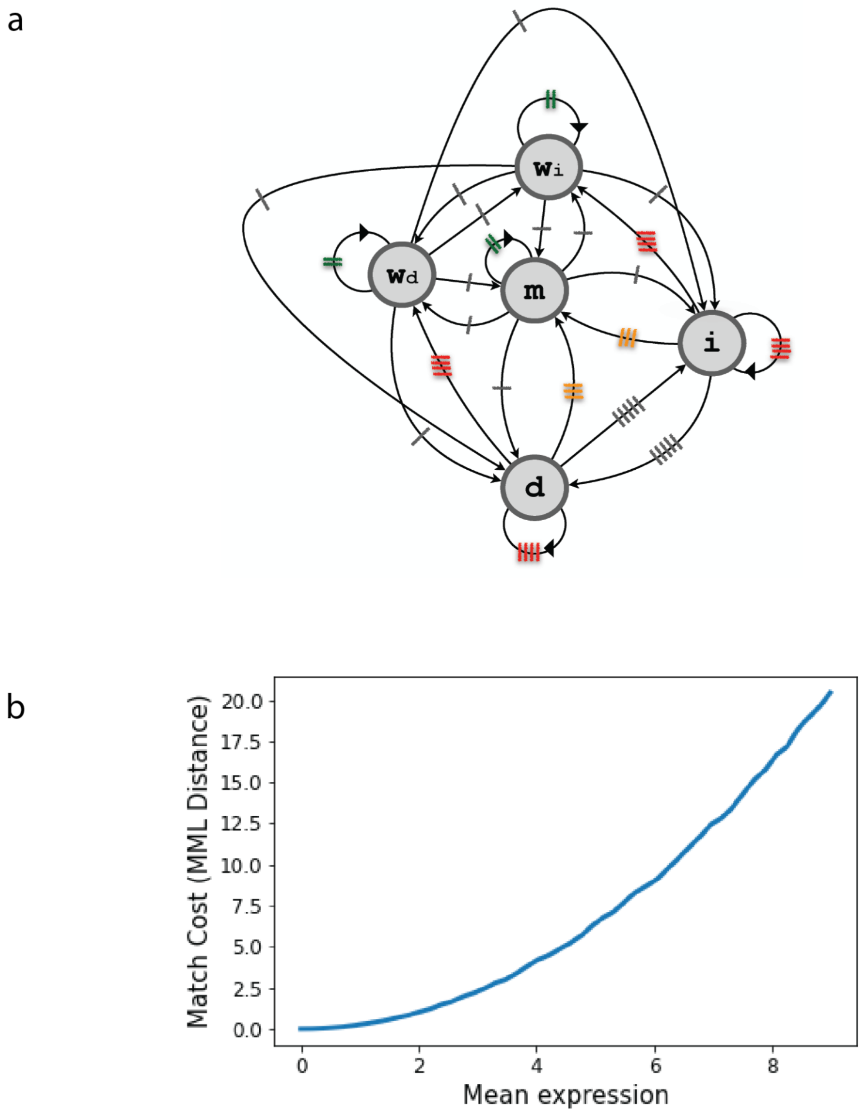
Author contributions

D.S., C.S. and S.A.T. conceived the initial project. D.S. and C.S. set up and directed the study. D.S., C.S., E.D. and K.P. performed bioinformatic analyses. D.S. designed and developed the software. C.S. A.S.S. and J.P. performed cell culture experiments. D.M., E.D., B.D. and S.A.T. provided intellectual input. S.A.T. acquired funding. D.S. and C.S. wrote the manuscript. All authors read and/or edited the manuscript.

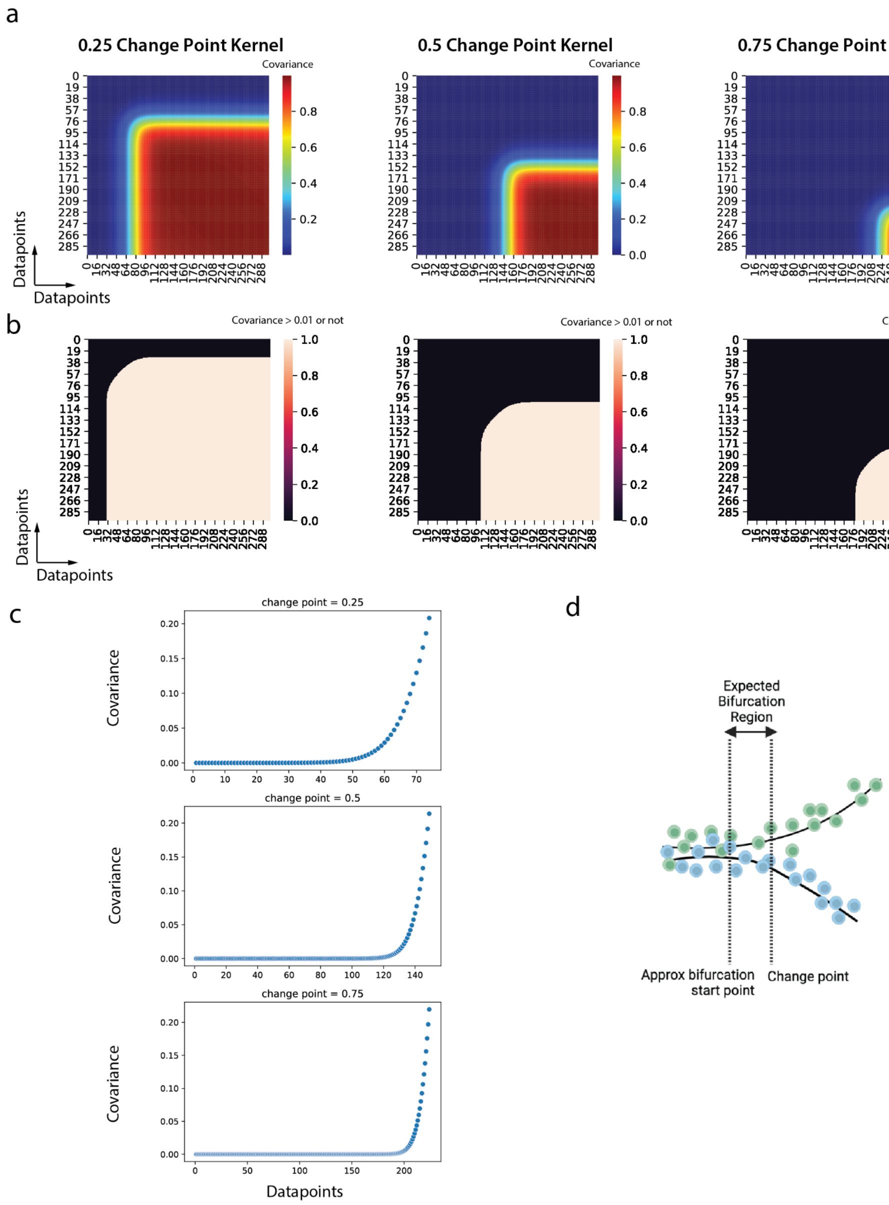
Competing interests

In the past three years, S.A.T. has received remuneration for Scientific Advisory Board Membership from Sanofi, GlaxoSmithKline, Foresite Labs and Qiagen. S.A.T. is a co-founder and holds equity in Transition Bio.

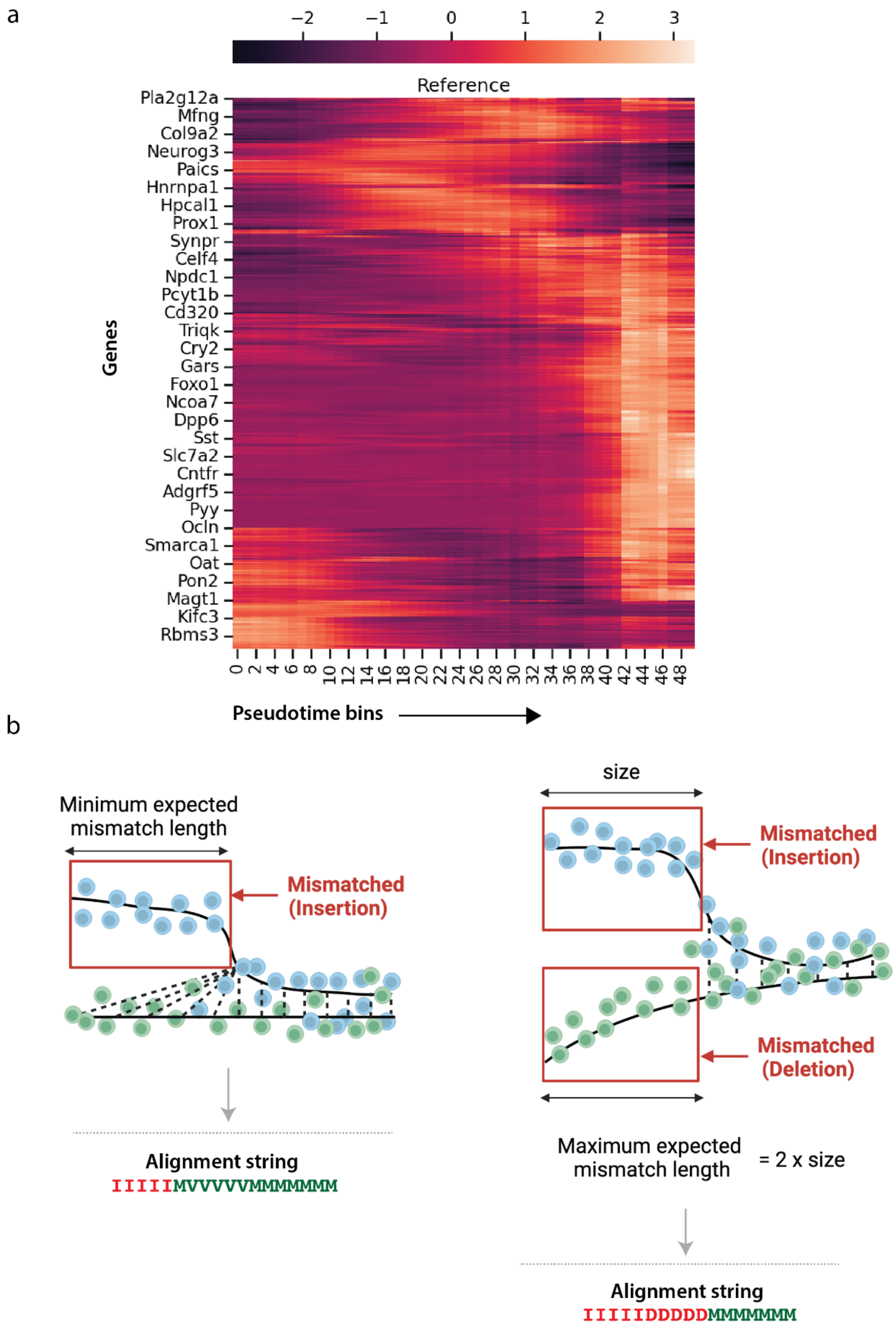
Extended Data Figures



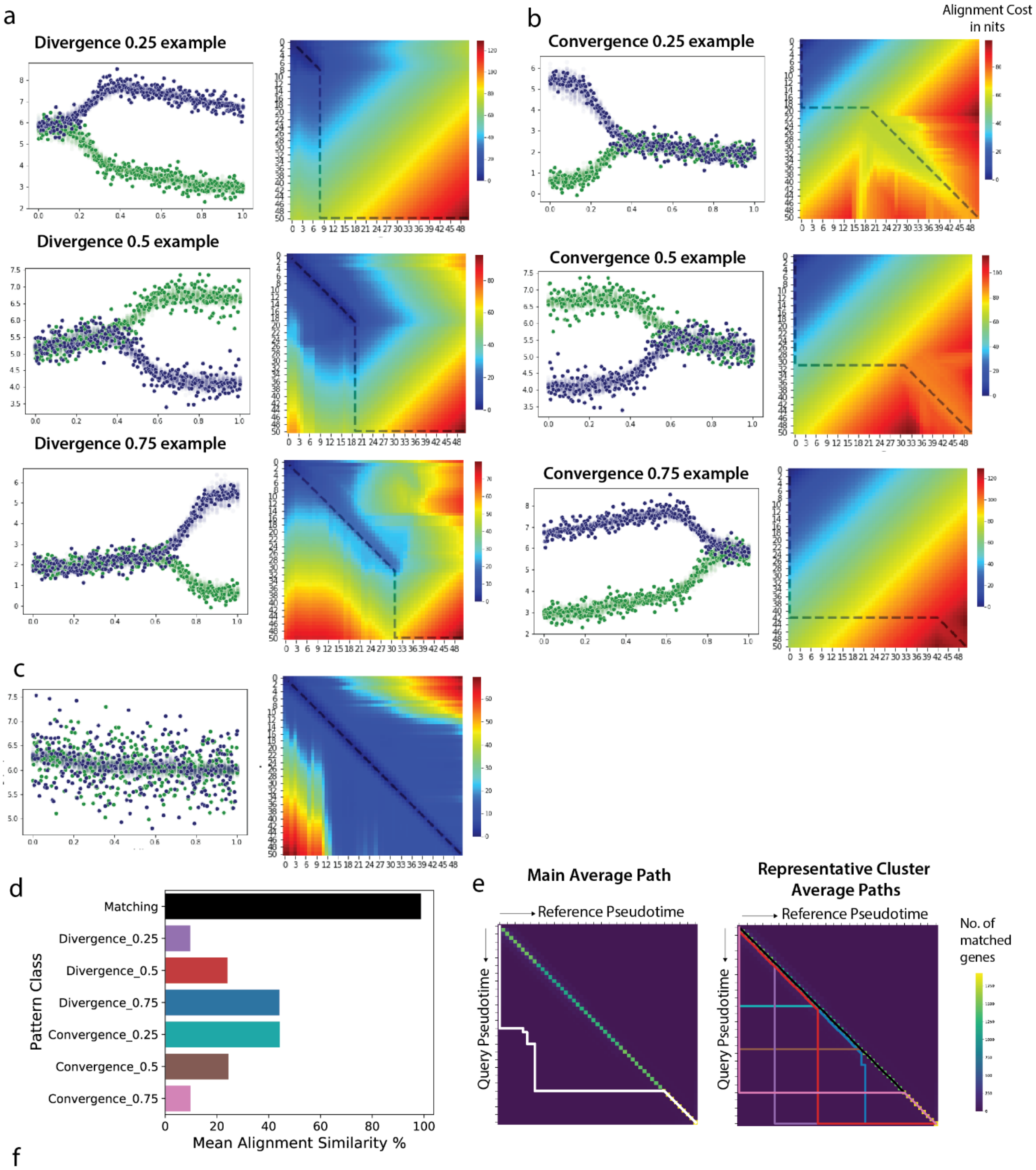
Extended Data Fig. 1 | The dynamic programming algorithm in G2G relies on a five-state machine and a Bayesian information-theoretic match cost function. a, This is a (five-state) finite state machine which can generate a string over the alphabet, Ω = [m, w_d, w_i, d, i] that describes all the 5 possible states of alignment (**Fig. 1b**) between two time points. Each arrow represents a state transition. Arrows with the same color tag implies equal probability of state transition. **b,** Behavior of the minimum message length (MML) inference based match cost function used by G2G. It plots the MML distance (i.e. match cost) between the standard Gaussian distribution $N(0,1)$ and $N(\mu, 1)$ Gaussian distribution for $\mu \in [0,9]$ at 50 equispaced points. 5000 data points have been randomly sampled from each $N(\mu, 1)$ distribution to represent itself. More illustrative examples are shown in **Supplementary Fig. 1b,c.**



Extended Data Fig. 2 | Simulating the bifurcation of reference and query trajectories using change point kernels. a, Change point kernel heatmaps for each approximate bifurcation point (change point) $\in [0.25,0.5,0.75]$. **b,** The same change point kernels binarized based on the 0.01 covariance threshold (top), **c,** The average covariance plotted for each $i\ \times i$ sub square matrix from $i = 0$ to $i =$ change point, showing that the branching effect can approximately start before the specified change point. **d,** Expected bifurcation region is taken from the point where we begin to see > 0.01 covariance in the change point kernel, until the particular change point.



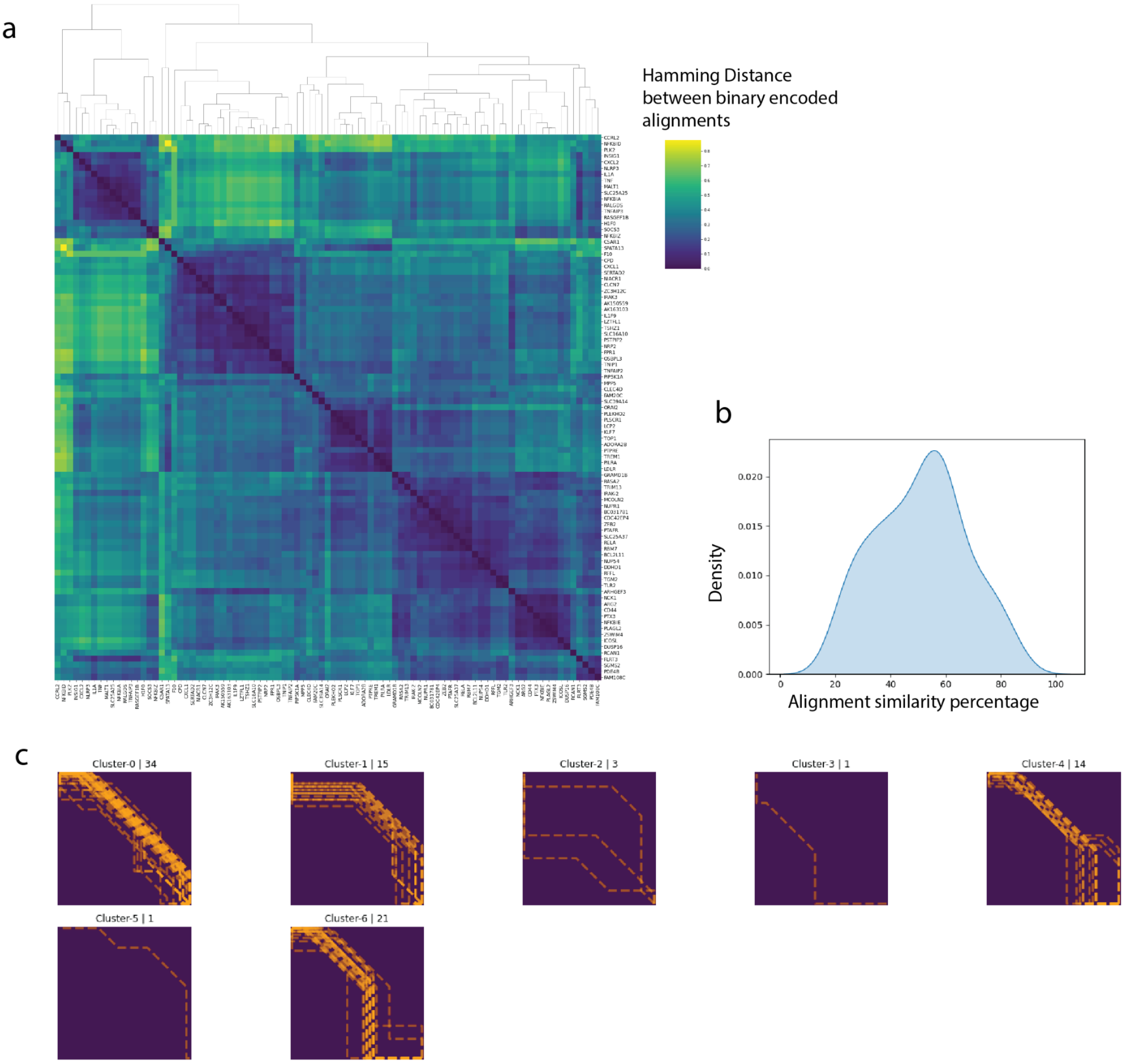
Extended Data Fig. 3 | Simulation Experiment 2: simulating perturbations across 769 genes in the scRNA-seq dataset of E15.5 murine pancreatic development[26]. a, Overall smoothened (interpolated) and z-normalized mean gene expression along the pseudotime (equally divided into 50 bins) for all genes in the dataset. **b,** Example illustrations of the two types of trajectory alignment that gives the minimum expected mismatch length and the maximum expected mismatch under the perturbation scenario 2 **(Fig. 3f,g),** where a start portion of a particular size in the query trajectory (in blue) is changed with respect to the reference trajectory (in green).



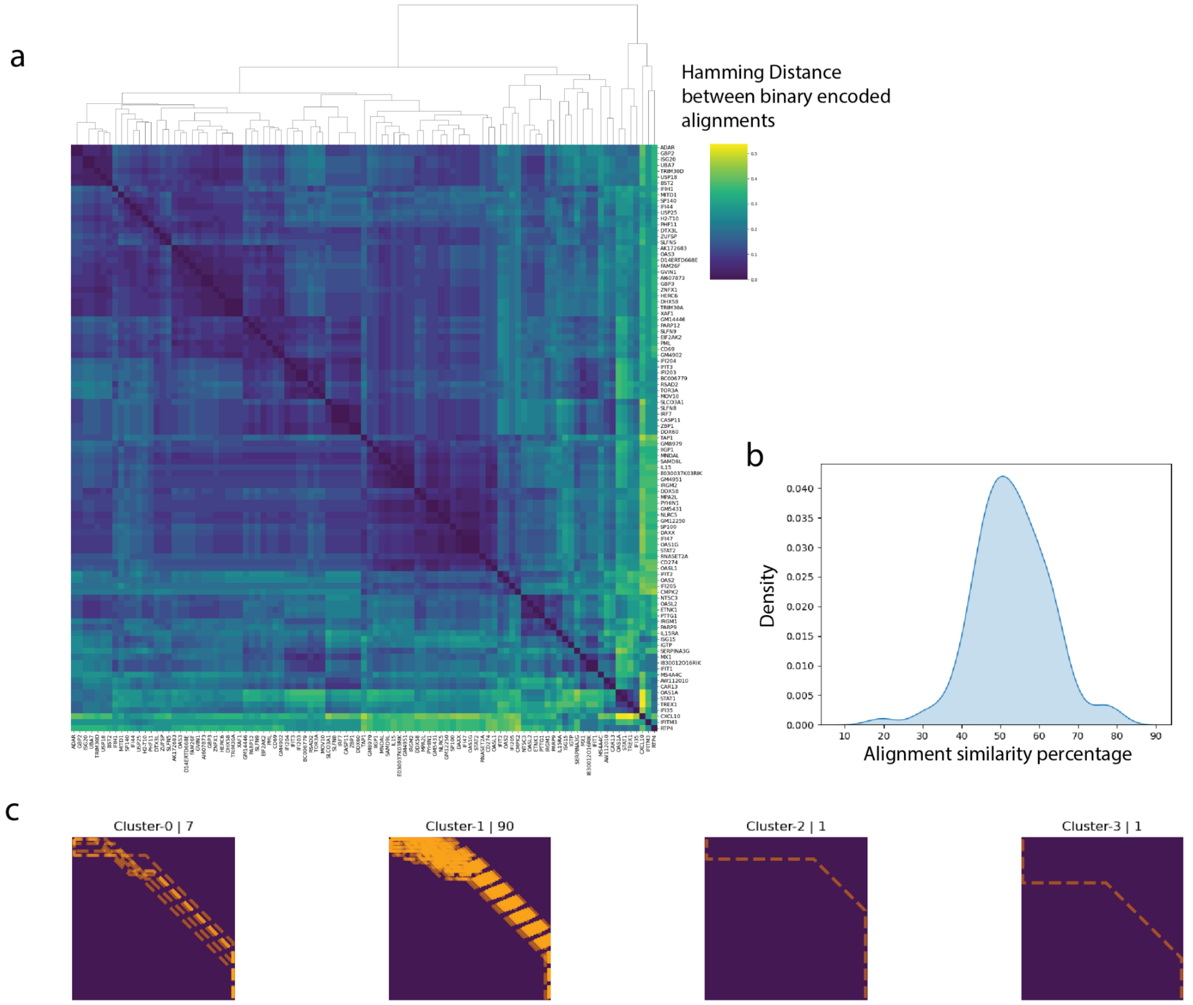
f

Pattern Class	Alignment Similarity %	Average alignment of the G2G representative cluster
Matching	100%	MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
Divergence 0.25	18%	IMMMMMMMMMIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Divergence 0.5	40%	IMMMMMMMMMMMMMMMMMMMMIIIIIIIIIIIIIIIIIIIIIIIIIIIIIDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Divergence 0.75	62%	IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMIIIDIIIIIIIIIIIIIIIDDDDDDDDDDDDDDDDDD
Convergence 0.25	60%	IIIIIIIIIIIIIIIIIIIIDDDDDDDDDDDDDDDDDDDDMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
Convergence 0.5	38%	IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDMMMMMMMMMMMMMMMMMMM
Convergence 0.75	16%	IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDMMMMMMMM

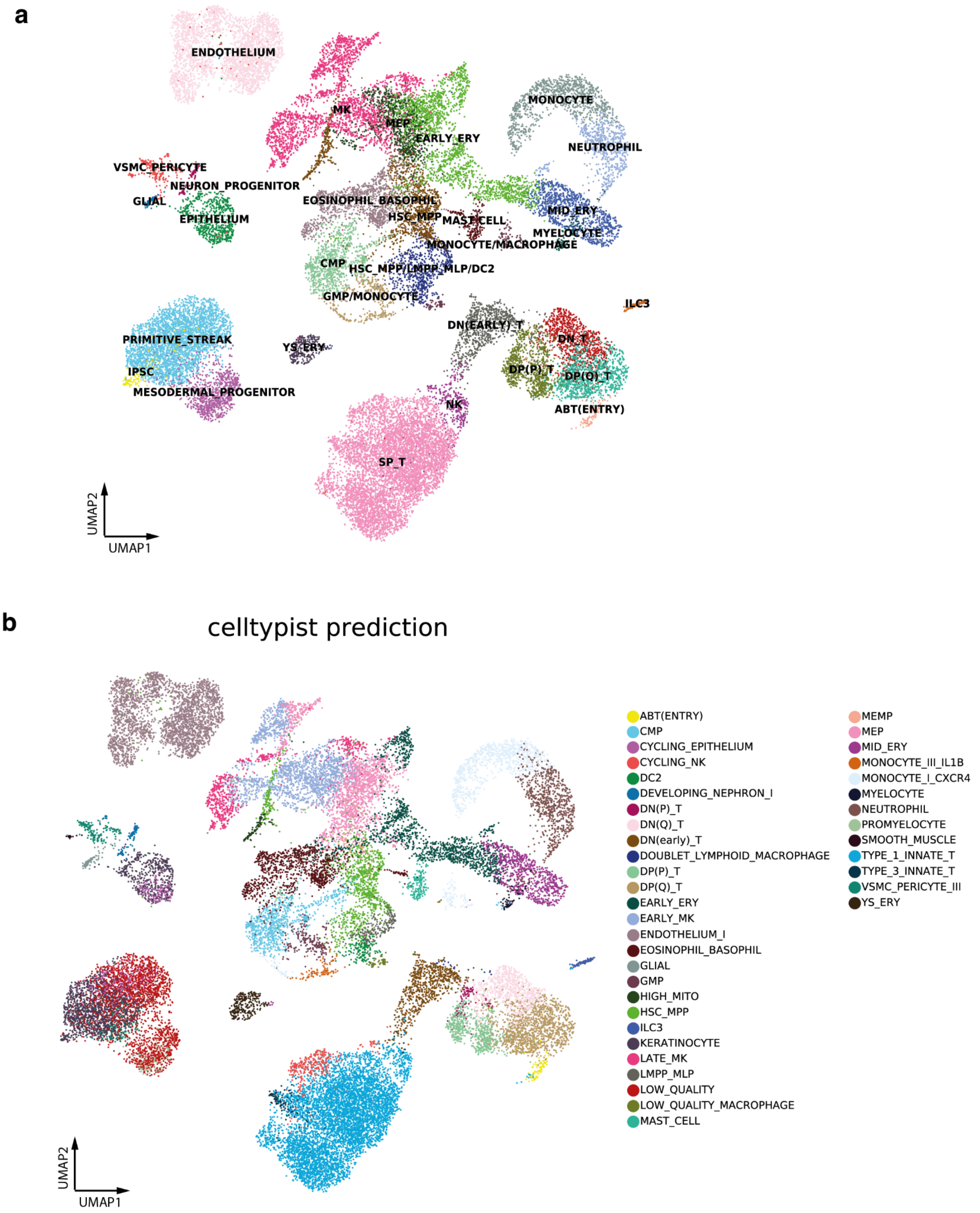
Extended Data Fig. 4 | Simulation Experiment 1: G2G alignments for different classes of pattern give expected results. a, Example simulated divergent pairs of reference (green) and query (blue) trajectories for approximate bifurcation point ∈ [0.25,0.5,0.75] (left) and their alignment cost landscapes with the optimal alignment path highlighted by the black dashed line (right). **b,** Example simulated convergent pairs for approximate bifurcation point $\in [0.25, 0.5, 0.75]$ (left) and their alignment cost landscapes with the optimal alignment path highlighted by the black dashed line. (right). **c,** Example simulated matching pair (left) and its alignment cost landscape with the optimal alignment path highlighted by the black dashed line. **d,** Mean percentage of alignment similarity observed in the 7 classes of alignment pattern. **e,** On the pairwise total time-point match count matrix between reference and query trajectories, the main average alignment path across all the 1500 alignments (left), and the individual, cluster-specific average alignment paths for the 7 representative clusters that accurately capture the 7 classes of pattern (right). Note: the color codes for the paths follow the same in d. **f,** The five-state alignment strings of the 7 cluster-specific average alignments with their alignment similarity percentages.



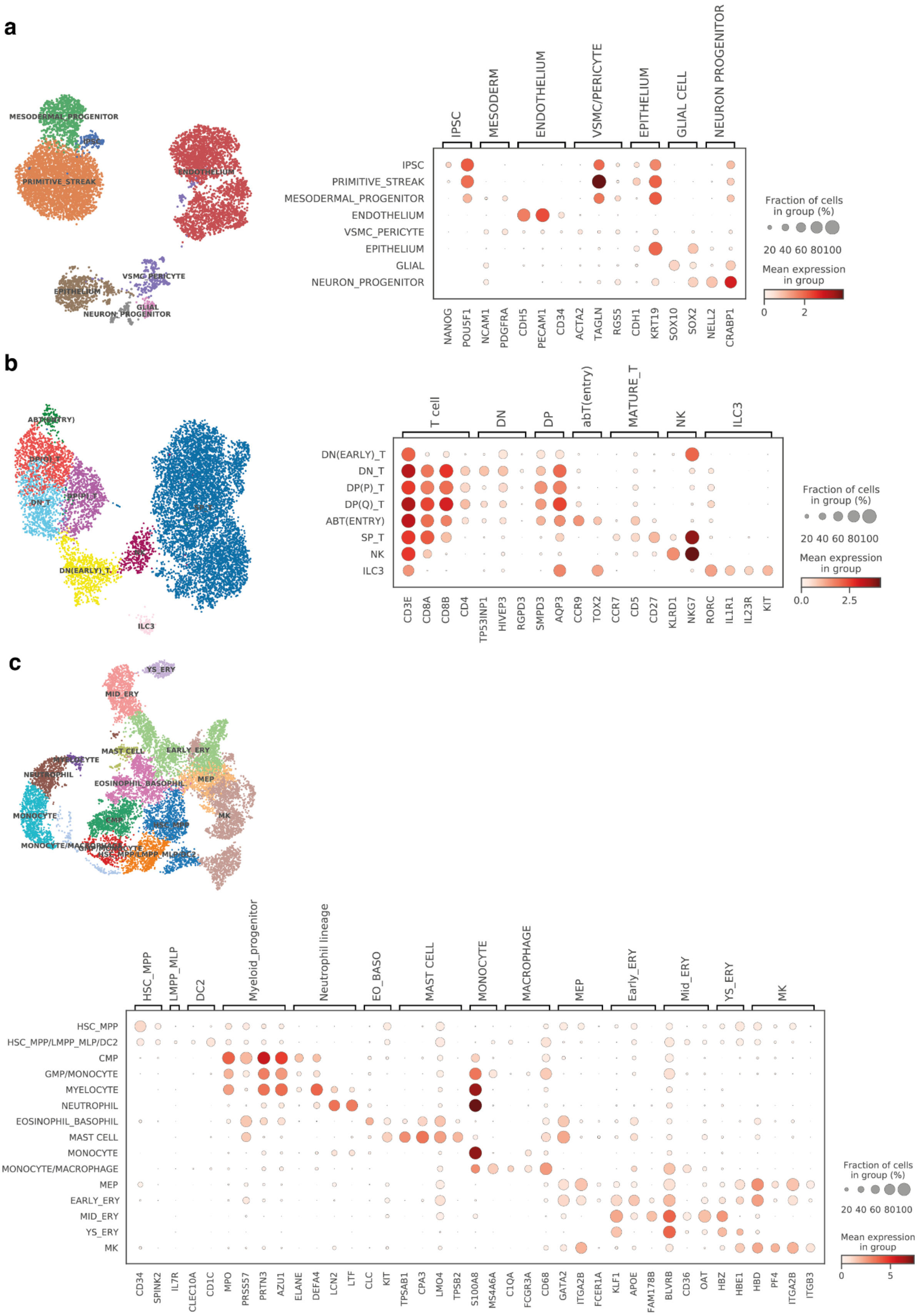
Extended Data Fig. 5 | The PAM vs. LPS 'peaked inflammatory module' alignment clustering outputs. a, The clustermap of the pairwise Hamming distance matrix of all 89 gene alignments. **b,** Density plot of the alignment similarity distribution (i.e. distribution of the percentage of matches/warps across all the alignment outputs). **c,** Seven gene alignment clusters were identified from hierarchical agglomerative clustering at a 0.3 distant threshold. Each plot titled by "Cluster-x | n" is the pairwise matrix of reference and query time points, visualizing alignment paths for all the genes (one alignment per gene and a total of n genes in the cluster) in cluster x. *C5AR1* (Cluster-3) and *RCAN1* (Cluster-5) stand out as single-gene clusters.



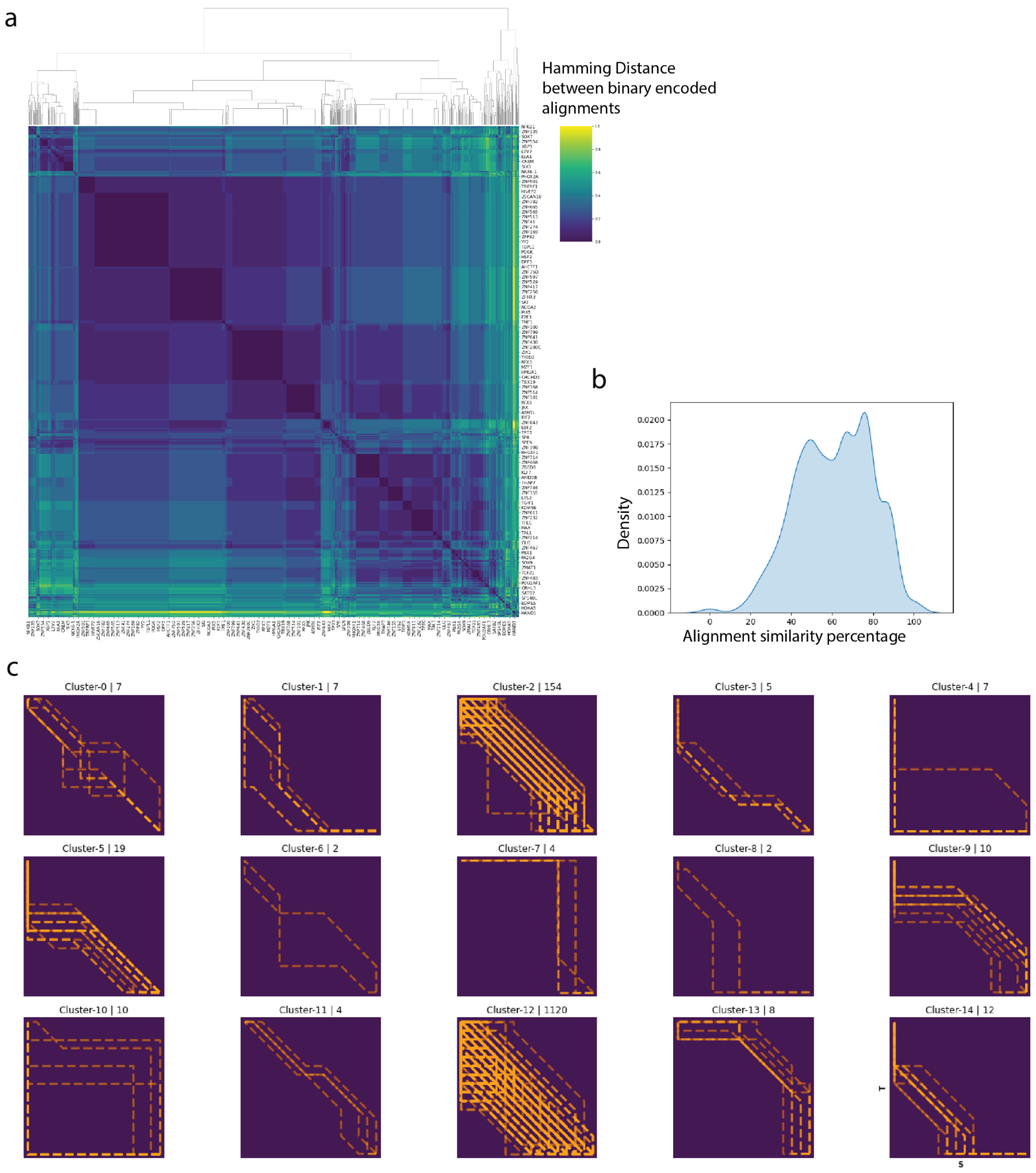
Extended Data Fig. 6 | The PAM vs. LPS 'core antiviral module' alignment clustering outputs. a, The clustermap of the pairwise Hamming distance matrix of all 99 gene alignments. **b,** Density plot of the alignment similarity distribution (i.e. distribution of the percentage of matches/warps across all the alignment outputs). **c,** Four gene alignment clusters were identified from hierarchical agglomerative clustering at a 0.25 distant threshold. Each plot titled by "Cluster-x | n" is the pairwise matrix of reference and query time points, visualizing alignment paths for all the genes (one alignment per gene and a total of n genes in the cluster) in a cluster x. *CXCL10* (Cluster-2) and *IFITM3* (Cluster-3) stand out as single-gene clusters.



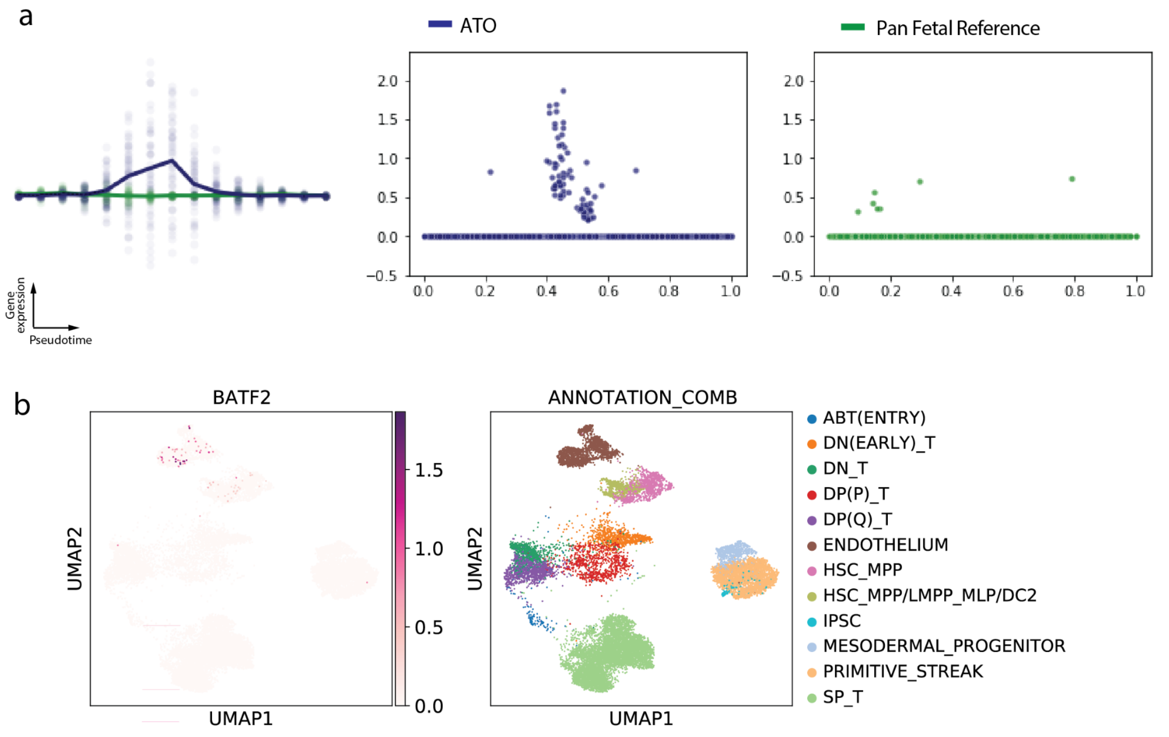
Extended Data Fig. 7 | Analysis of artificial thymic organoid scRNA-seq data. a, UMAP visualization of different cell types in the ATO (refined annotation). iPSC: induced pluripotent stem cell, HSC_MPP: hematopoietic stem cell, and multipotent progenitor, LMPP_MLP: lymphoid-primed multipotent progenitor and multi lymphoid progenitor, DC: dendritic cell, CMP: common myeloid progenitor, GMP: granulocyte and monocyte progenitor, MK: megakaryocyte, MEP: megakaryocyte erythroid progenitor, YS_ERY: yolk sac-like erythrocyte, EARLY_ERY: early erythrocyte, MID_ERY: mid-stage erythrocyte, DN(EARLY) T: early double negative T cell, DN T: double negative T cell, DP(P) T: proliferating double positive T cell, DP(Q) T: quiescent double positive T cell, SP T: single positive T cell, NK: natural killer cell, ILC: innate lymphoid cell. **b,** Predicted annotations from logistic regression model with CellTypist using the developing human immune atlas[30] as the training dataset, overlaid on the same UMAP plot as in **a**.


a
MESODERMAL_PROGENITOR
IPSC
PRIMITIVE_STREAK
ENDOTHELIUM
VSMC_PERICYTE
EPITHELIUM
GLIAL
NEURON_PROGENITOR
IPSC
MESODERM
ENDOTHELIUM
VSMC/PERICYTE
EPITHELIUM
GLIAL CELL
NEURON PROGENITOR
NANOG
POU5F1
NCAM1
PDGFRA
CDH5
PECAM1
CD34
ACTA2
TAGLN
RGS5
CDH1
KRT19
SOX10
SOX2
NELL2
CRABP1
Fraction of cells in group (%)
20 40 60 80100
Mean expression in group
0
2
b
ABT(ENTRY)
DP(Q)_T
DP(P)_T
DN_T
DN(EARLY)_T
NK
SP_T
ILC3
T cell
DN
DP
abT(entry)
MATURE_T
NK
ILC3
CD3E
CD8A
CD8B
CD4
TP53INP1
HIVEP3
RGPD3
SMPD3
AQP3
CCR9
TOX2
CCR7
CD5
CD27
KLRD1
NKG7
RORC
IL1R1
IL23R
KIT
Fraction of cells in group (%)
20 40 60 80100
Mean expression in group
0.0
2.5
c
YS_ERY
MID_ERY
MAST CELL
EARLY_ERY
MYELOCYTE
NEUTROPHIL
EOSINOPHIL_BASOPHIL
MEP
MONOCYTE
CMP
HSC_MPP
MK
MONOCYTE/MACROPHAGE
GMP/MONOCYTE
HSC_MPP/LMPP_MLP/DC2
HSC_MPP
LMPP_MLP
DC2
Myeloid_progenitor
Neutrophil lineage
EO_BASO
MAST CELL
MONOCYTE
MACROPHAGE
MEP
Early_ERY
Mid_ERY
YS_ERY
MK
HSC_MPP
HSC_MPP/LMPP_MLP/DC2
CMP
GMP/MONOCYTE
MYELOCYTE
NEUTROPHIL
EOSINOPHIL_BASOPHIL
MAST CELL
MONOCYTE
MONOCYTE/MACROPHAGE
MEP
EARLY_ERY
MID_ERY
YS_ERY
MK
CD34
SPINK2
IL7R
CLEC10A
CD1C
MPO
PRSS57
PRTN3
AZU1
ELANE
DEFA4
LCN2
LTF
CLC
KIT
TPSAB1
CPA3
LMO4
TPSB2
S100A8
MS4A6A
C1QA
FCGR3A
CD68
GATA2
ITGA2B
FCER1A
KLF1
APOE
FAM178B
BLVRB
CD36
OAT
HBZ
HBE1
HBD
PF4
ITGA2B
ITGB3
Fraction of cells in group (%)
20 40 60 80100
Mean expression in group
0
5

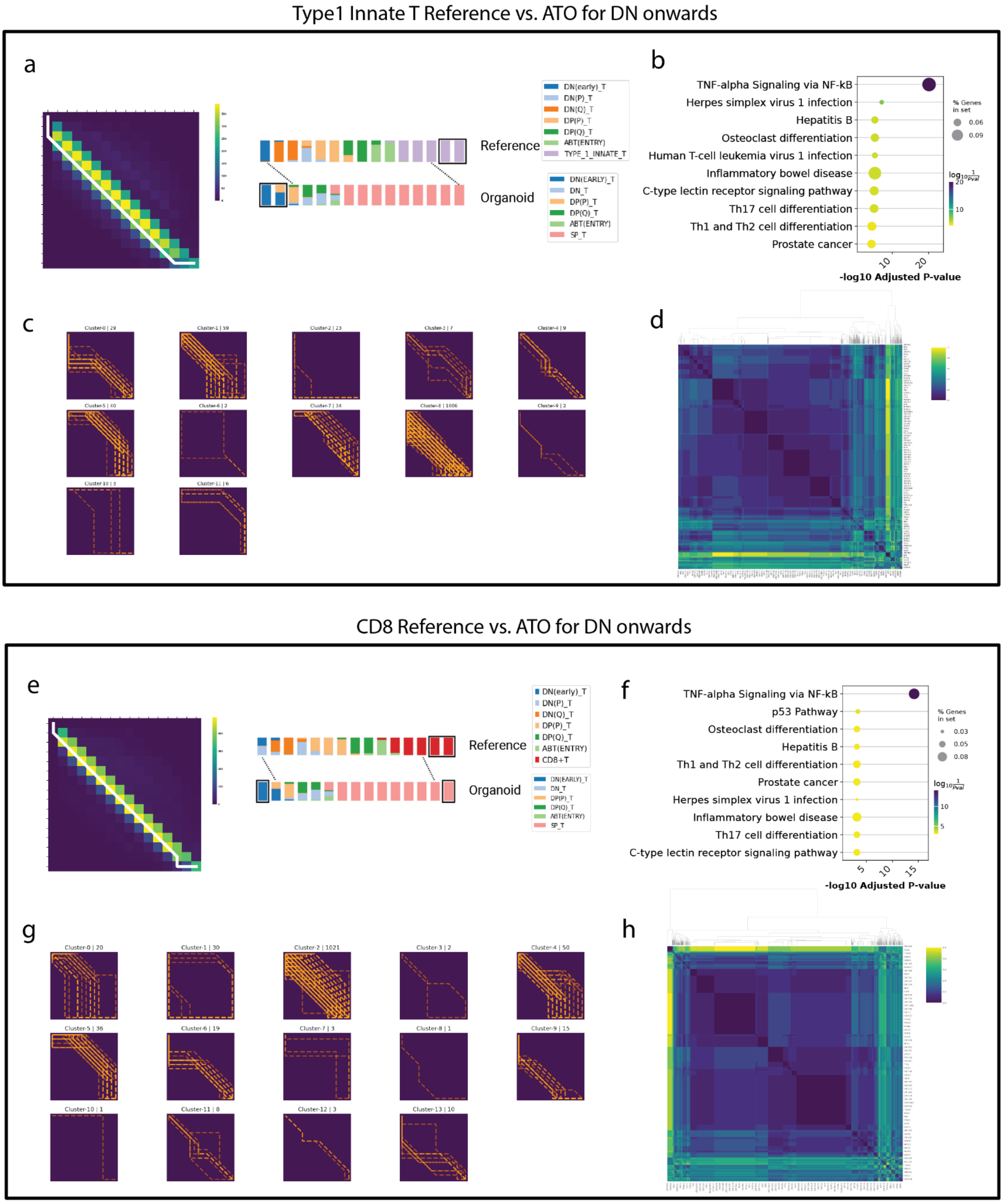
Extended Data Fig. 8 | Annotation of artificial thymic organoid scRNA-seq data. For each subset lineage embedding generated through scVI, we show UMAP embeddings of cells colored by annotated cell populations and dot plots of mean expression (log-normalized counts, dot color) and fraction of expressing cells (dot size) of marker genes (columns) used for cell population annotation (rows). **a,** Annotation of non-hematopoietic cells. **b,** Annotation of T/ILC/NK lineage cells. **c,** Annotation of other hematopoietic cells that are not in **b**.



Extended Data Fig. 9 | Pan fetal reference vs artificial thymic organoid alignment clustering outputs. a, The clustermap of the pairwise Hamming distance matrix of all 1371 transcription factor alignments. **b,** Density plot of the alignment similarity distribution (i.e. distribution of the percentage of matches/warps across all the alignment outputs). **c,** Fifteen gene alignment clusters were identified from hierarchical agglomerative clustering at a 0.3 distant threshold. Each plot titled by “Cluster-x | n” is the pairwise matrix of reference and query time points, visualizing alignment paths for all the genes (one alignment per gene and a total of n genes in the cluster) in a cluster x.

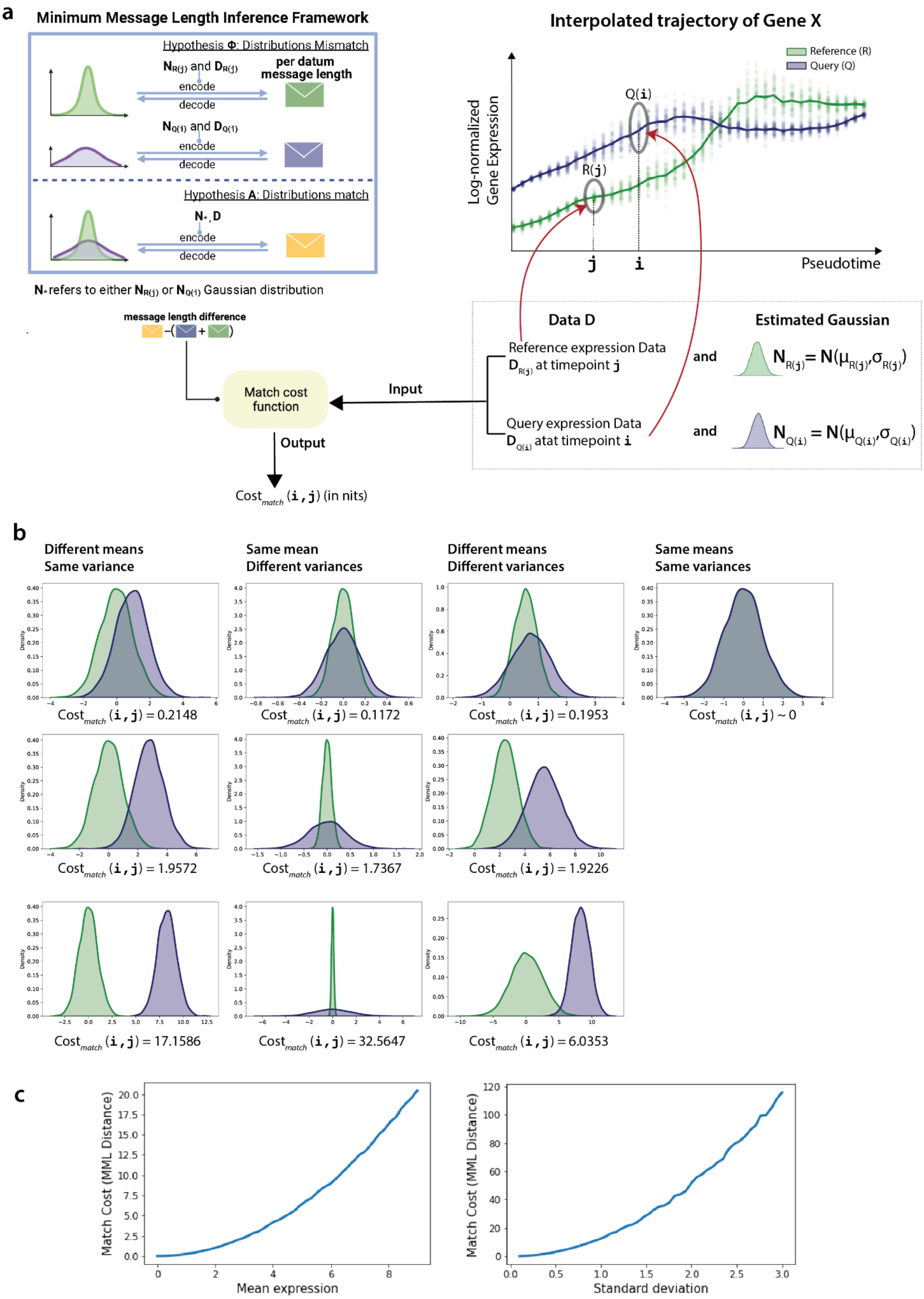


Extended Data Fig. 10 | *BATF2* expression in *in vitro* ATO and *in vivo* pan fetal reference. a, Left panel: the interpolated log1p normalized expression (*y*-axis) against pseudotime (*x*-axis) for *BATF2*. Right two panels: the actual log1p normalized expression (*y*-axis) against pseudotime (*x*-axis). Each point represents a cell. **b,** The same UMAP visualization as in **Fig. 5e**, subsetted to *in vitro* cells from ATO, colored by the *BATF2* gene expression value (left) or the cell types (right).

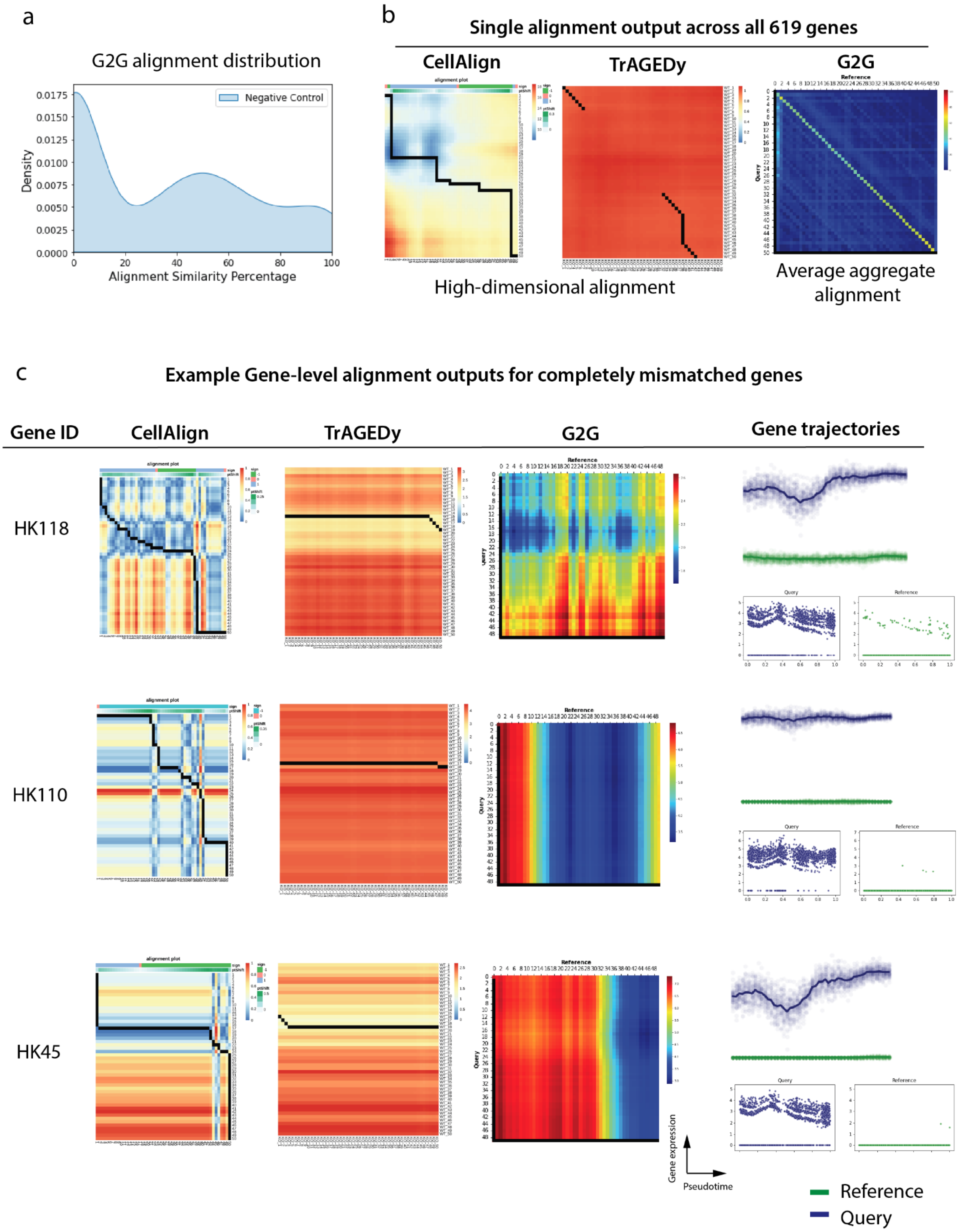


Extended Data Fig. 11 | Analysis of the pan fetal reference vs artificial thymic organoid (ATO) alignment from DN T cells onwards. a, Aggregate alignment between Type 1 Innate T cell reference and ATO across 1220 human transcription factors. **b,** Gene set overrepresentation results of the most mismatched genes from alignment in **a**. **c,** Twelve gene alignment clusters were identified for **a** from hierarchical agglomerative clustering at a 0.3 distant threshold. Each plot titled by "Cluster-x | n" is the pairwise matrix of reference and query time points, visualizing alignment paths for all the genes (one alignment per gene and a total of n genes in the cluster) in a cluster x. **d,** The clustermap of the pairwise Hamming distance matrix of all TF alignments in **a**. **e,** Aggregate alignment between CD8+ Reference and ATO across 1219 human transcription factors. **f,** Gene set overrepresentation results of the most mismatched genes from alignment in **e**. **g,** Fourteen gene alignment clusters were identified for **e** from hierarchical agglomerative clustering at a 0.3 distant threshold. Each plot titled by “Cluster-x | n” is the pairwise matrix of reference and query time points, visualizing alignment paths for all the genes (one alignment per gene and a total of n genes in the cluster) in a cluster x. **h,** The clustermap of the pairwise Hamming distance matrix of all TF alignments in **e**.

Supplementary Figures



Supplementary Fig. 1 | Minimum message length (MML) inference based distance function to compute the cost $Cost_{match}(i,j)$ of matching a reference time point j and query time point i. a, The top right plot gives the interpolated log-normalized expression (y-axis) of a particular gene X in the observed single-cell data of a reference R and query Q, against their pseudotime estimates (x-axis). The bold lines represent mean expression trends. The faded data points represent the interpolated data (i.e. 50 random samples from the estimated Gaussian distribution at each time point). As detailed in the Methods of the main text, the scoring scheme of the G2G DP algorithm computes the cost of matching every pair of time points between R and Q based on their corresponding interpolated expression distributions. Here we consider an example reference time point j and query time point i, and their respective single-cell expression datasets, $R(j)$ and $Q(i)$, as circled in the plot. Their interpolated expression distributions are $N(\mu_{R(j)}, \sigma_{R(j)})$ and $N(\mu_{Q(i)}, \sigma_{Q(i)})$, denoted by $N_{R(j)}$ and $N_{Q(i)}$, respectively. Their interpolated expression data vectors are $D_{R(j)}$ and $D_{Q(i)}$. The top left is the schematic illustration of our MML framework, extending the overview schematic **Fig. 2** (top left) in the main text. Under the MML framework, we define two hypotheses: Hypothesis **A:** the (i,j) time points match, and Hypothesis Φ: the (i,j) time points mismatch. Next we compute: (1) the total (per datum) message length of encoding **A** and D jointly, and (2) the total (per-datum) message length of encoding Φ and D jointly. Then we define $Cost_{match}(i,j)$ to be the difference between those two message lengths, measured in nits (a unit of information). **b,** Example cases of distributional differences (caused by the difference in means, the difference in variance, the difference in both mean and variance) between $R(j)$ and $Q(i)$, and their $Cost_{match}(i,j)$ values measured in nits. When the mean and variance is the same, $Cost_{match}(i,j)$~0. **c,** Behavior of $Cost_{match}(i,j)$ as the difference between the distributions increases. Left plot: $Cost_{match}(i,j)$ between the standard Gaussian distribution $N(0,1)$ and $N(\mu,1)$ Gaussian distribution for $\mu \in [0,9]$ at 50 equispaced points. 5000 data points have been randomly sampled from each $N(\mu,1)$ distribution to represent itself. Right plot: $Cost_{match}(i,j)$ between the standard Gaussian distribution $N(0,1)$ and $N(0,\sigma)$ Gaussian distribution for $\sigma \in [0.1,3]$ at 50 equispaced points. 5000 data points have been randomly sampled from each $N(0,\sigma)$ distribution to represent itself.



Supplementary Fig. 2 | Benchmarking G2G against the DTW based approach: CellAlign[5] and its most recent extension, TrAGEDy[11], over a simulated negative control dataset from literature[11]. a, Density plot of the G2G alignment similarity distribution (i.e. distribution of the percentage of matches/warps across all the G2G gene-level alignment outputs inferred for the 619 genes in the negative control dataset). **b,** The single alignments produced by CellAlign, TrAGEDy and G2G for all the 619 simulated genes. CellAlign and TrAGEDy generate a high dimensional alignment, whereas G2G generates an average alignment across all the gene-specific alignments. The alignment paths are drawn (in black) on the pairwise time point matrix between the reference and query trajectories. The color of the CellAlign and TrAGEDy matrices represent their cost of aligning the corresponding pair of reference and query time points. In the G2G matrix, the color represents the number of genes showing match or warp for the corresponding pair of time points. In CellAlign and TrAGEDy alignment plots, the vertical and horizontal lines represent warps, whereas diagonal lines represent one-to-one matches. In the G2G alignment plots here, the full vertical line joined by the full horizontal line denotes an alignment path with all insertions followed by all deletions. **c,** CellAlign, TrAGEDy, and G2G outputs for three simulated genes (with IDs: *HK118*, *HK110*, *HK45*) that have completely mismatched trajectories. Each row gives the gene-specific alignment paths (highlighted in black) on the pairwise time point matrices produced by each method for the corresponding gene. The matrix color represents each methods' cost of aligning the corresponding pair of reference and query time points. For G2G, this is the $\log_{10}$ normalized nits compression (i.e. the difference between the match cost and the mismatch cost as described in the Methods and **Supplementary Fig. 1**). The right-most column ("Gene trajectories") shows the reference and query trajectories of the corresponding gene in each row. For each gene row, the top panel shows the interpolated log1p normalized expression (*y*-axis) against pseudotime (*x*-axis). The bold lines represent mean expression trends, while the faded data points are 50 random samples from the estimated expression distribution at each time point. The bottom panels show the actual log1p normalized expression (*y*-axis) against pseudotime (*x*-axis) for reference (right) and query (left). Each point represents a cell.

Supplementary Tables

Framework component	**CellAlign**[5]	**TrAGEDy**[11]	**Genes2Genes**
Input	(1) Log-normalized single-cell gene expression data (2) Pseudotime estimates of the cells inferred using any available method of choice		
Algorithm	Uses Dynamic Time Warping (DTW) algorithm.	Builds on top of CellAlign, and performs post-hoc changes to the DTW alignment to capture mismatches.	Combines DTW and Gotoh's biological sequence alignment[15] through a new dynamic programming algorithm.
Alignment states	Handles matches and warps only, subjected to a weight scheme with constant weight for warp open/extension	Identifies optimal start and end time points of the trajectories for DTW alignment to exclude regions of mismatch at the beginning and end. It further filters the DTW aligned regions based on alignment cost thresholding to identify mismatches.	Handles matches, warps and mismatches jointly, subjected to a five-state machine with state transition probabilities, handling gap/warp open/extension.
Trajectory Interpolation	Interpolates data using a Gaussian kernel-based weighted mean expression.	Extends CellAlign interpolation to use a cell density weighted window size.	Extends CellAlign interpolation to distributional interpolation using weighted variance.
Distance measure between a pair of reference and query time points	Uses min-max normalized, mean gene expression based Euclidean distance measure, to identify similar trends of expression dynamics across two conditions.	Uses Spearman correlation, which does not require gene expression scaling as done in CellAlign, but does not support gene-level alignment.	Uses a minimum message length inference based distributional distance measure, aiming to compare the gene expression distributions between two conditions.
Alignment output	Outputs only a single, high-dimensional alignment (across a given gene list). A single, gene-level alignment can be obtained by giving a single gene input.	Modified, high-dimensional DTW alignment after pruning the matches.	Outputs gene-specific alignments with an explicit alignment state description via a five-state alignment string for all the given genes. Can output an aggregate alignment for any given set of genes.
Alignment clustering	Can cluster genes only based on the pseudotime shifts in their	Does not explicitly discuss clustering.	Can cluster genes based on their five-state alignment strings,

	alignments.		covering both matches and mismatches.
Differential expression capture	Requires additional downstream tasks to extract differential genes and regions (e.g. local DTW alignment with user-defined similarity threshold).	Sliding window soft clustering approach to extract DE using t-test/Mann-Whitney U test.	A gene-specific alignment output itself is a direct description of the differential expression status along the time axis. Provides a ranked list of genes based on their alignment similarity percentage.

Supplementary Table 1 | List of features included in trajectory alignment frameworks. A table outlining and comparing the features of CellAlign[5], TrAGEDy[11] and G2G.

Supplementary Table 2
Gaussian Process based simulated gene alignments
GP_based_simulations_gene_alignments.csv

Supplementary Table 3
Gaussian Process based simulated gene alignment cluster statistics
GP_based_simulations_cluster_statistics.csv

Supplementary Table 4-17
Real data based simulations alignment statistics
pancreas_simulations_<perturbation_type>_size<perturbation_size>gene_alignments.csv

Supplementary Table 18
PAM vs. LPS ‘core antiviral module’ alignments
PAM_LPS_global_gene_alignments.csv

Supplementary Table 19-22
Overrepresentation analysis results of PAM vs. LPS ‘core antiviral module’ alignment clusters
PAM_LPS_global_cluster_<CLUSTER_ID>_pathways.csv

Supplementary Table 23
PAM vs. LPS ‘peaked inflammatory module’ alignments
PAM_LPS_local_gene_alignments.csv

Supplementary Table 24-30
Overrepresentation analysis results of PAM vs. LPS ‘peaked inflammatory module’ alignment clusters
PAM_LPS_locall_cluster_<CLUSTER_ID>_pathways.csv

Supplementary Table 31
T1 vs. ATO all alignments
SPT_all_gene_alignments.csv

Supplementary Table 32

Overrepresentation analysis results of T1 vs. ATO top 165 genes (under <40% alignment similarity threshold)

SPT_all_overrep_results_top_k_DE_threshold_0.4sim_165genes.csv

Supplementary Table 33-45

Overrepresentation analysis results of T1 vs. ATO alignment clusters

SPT_all_cluster_<CLUSTER_ID>_pathways.csv

Supplementary Table 46

T1 vs. ATO DN onwards all alignments

SPT_DN_gene_alignments.csv

Supplementary Table 47

Overrepresentation analysis results of T1 vs. ATO DN onwards top 130 genes (under <40% alignment similarity threshold)

SPT_dn_overrep_results_top_k_DE_threshold_0.4sim_130genes.csv

Supplementary Table 48

CD8+T vs. ATO DN onwards all alignments

CD8_DN_gene_alignments.csv

Supplementary Table 49

Overrepresentation analysis results of CD8+T vs. ATO DN onwards top 120 genes (under <40% alignment similarity threshold)

CD8_dn_overrep_results_top_k_DE_threshold_0.4sim_120genes.csv

Supplementary Table 50

ATO metadata

ATO_WT_manifest.csv

Supplementary Table 51

ATO CITE-seq metadata

ATO_hashtagging.csv

Supplementary Table 52

Negative control alignments

negative_control_gene_alignments.csv