nature portfolio

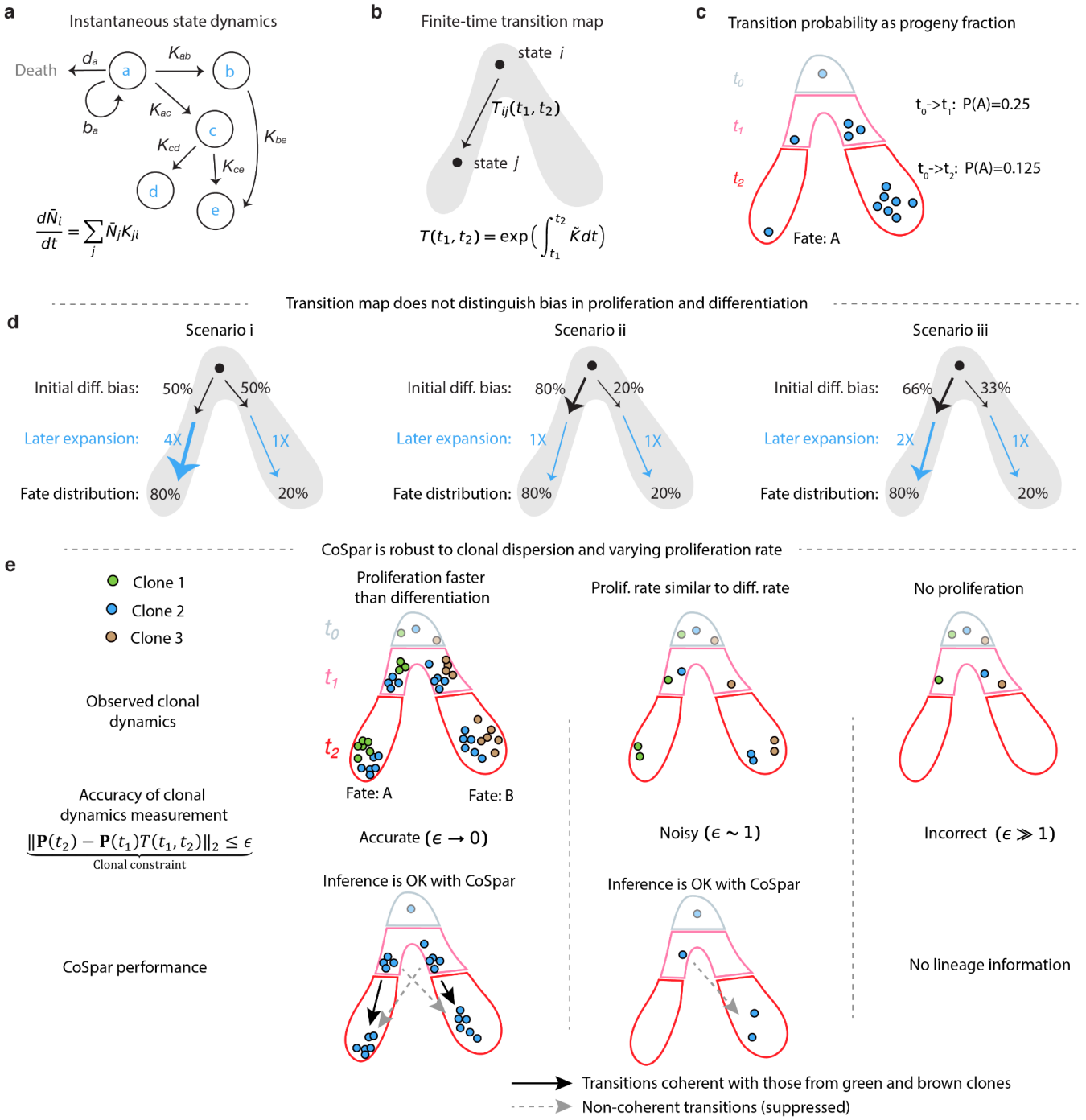
https://doi.org/10.1038/s41587-022-01209-1

Supplementary information

CoSpar identifies early cell fate biases from single-cell transcriptomic and lineage information

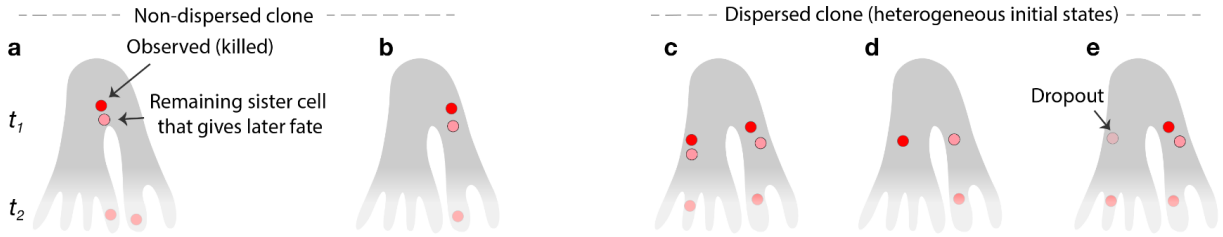
In the format provided by the authors and unedited

Supplementary Information

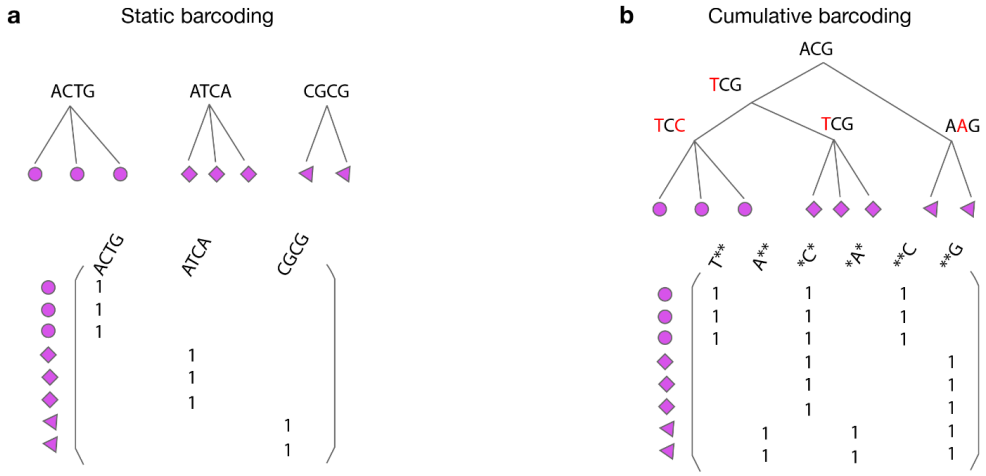


Supplementary Fig. 1. Models, assumptions and limitations of Coherent Sparse Optimization. a, Simple example of the class of stochastic models that CoSpar seeks to learn. In such models, each node represents an observed cell state. In practice, thousands of measured states are included; here only five are shown. At each state cells self-renew, die, or differentiate with state-specific rates. The mean fraction of cells in each state evolves according to coupled first-order equations as shown. See Supplementary Note 1 for details.

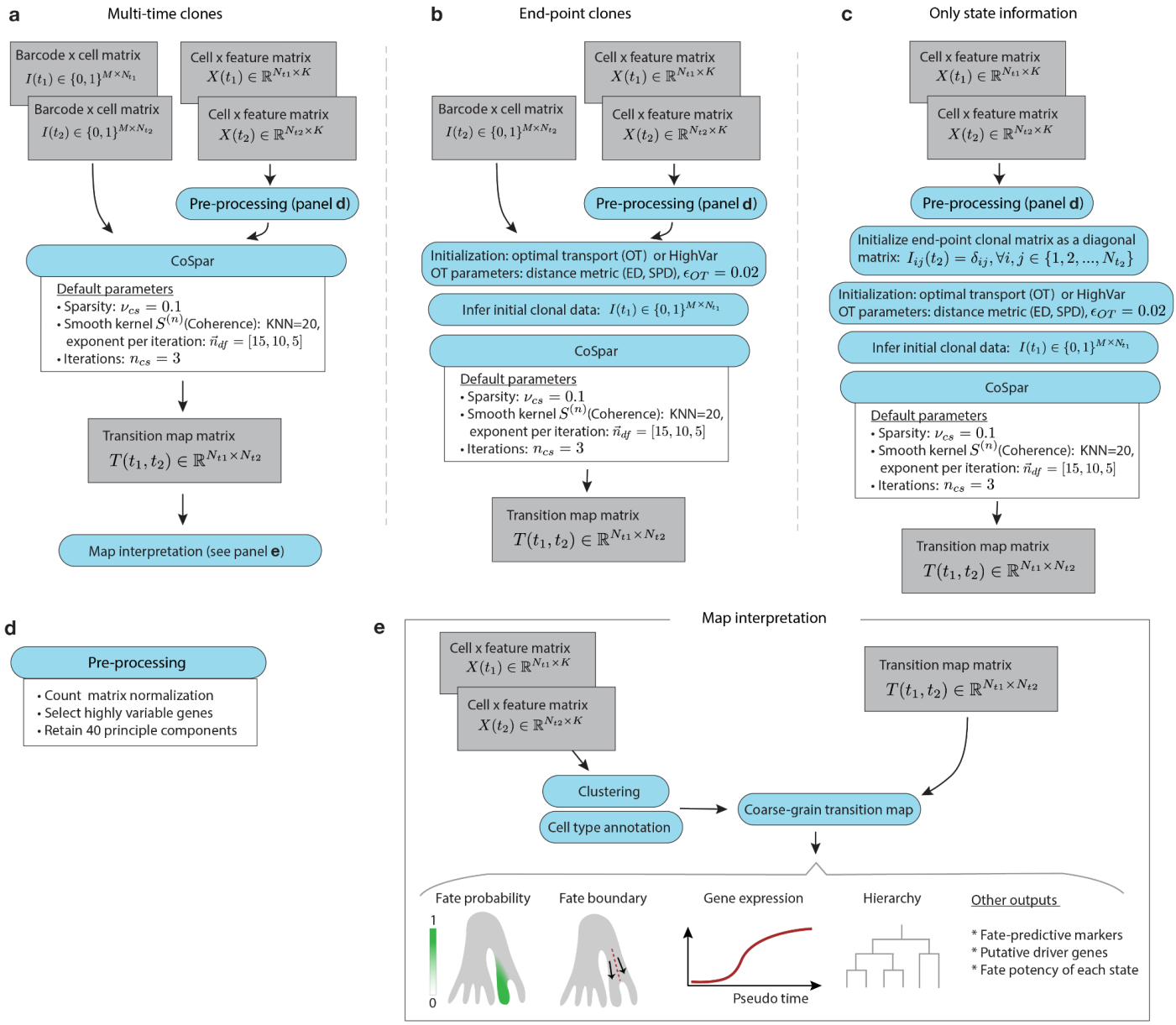
b, The empirically-observed finite-time transition map can be interpreted through its relation to the transition rate matrix *K* (see panel **a**). See Supplementary Note 1 for details.
c, Schematics illustrating the operational, experimentally-accessible definition of a transition probability, as the average fraction of progeny derived from an initial cell *i* at t_0 that differentiates into a target state *j* at later times. As defined, transition probabilities are sensitive to biases in fate choice, and to differential rates of cell division and cell loss.
d, Schematics exemplifying that transition maps cannot distinguish fate bias from differences in net rates of cell expansion (division – loss). Three different underlying dynamics lead to the same transition maps.
e, Schematics clarifying the robustness of CoSpar to clonal dispersion (demonstrated in Fig. 3). i), When cells undergo extensive proliferation prior to fate bifurcation and clonal sampling, each clone densely samples several differentiation trajectories. By imposing sparsity and coherence, CoSpar re-enforces a minimal number of transitions that explain dynamics across all clones. ii), At lower rates of proliferation, fewer cells from each clone are sampled, and it may lead to observing clonally-related cells at different time-points on different trajectories, as shown (blue clone sampled towards fate A at t_1, and towards fate B at t_2). By enforcing coherence between clones rooted in neighboring states, CoSpar may still recover a correct transition map. In this case, there is a trade-off in the CoSpar cost function between minimizing the clone transition map error and maximizing coherence. iii), Lacking proliferation, one cannot establish clonal relationships that constrain dynamic inference.



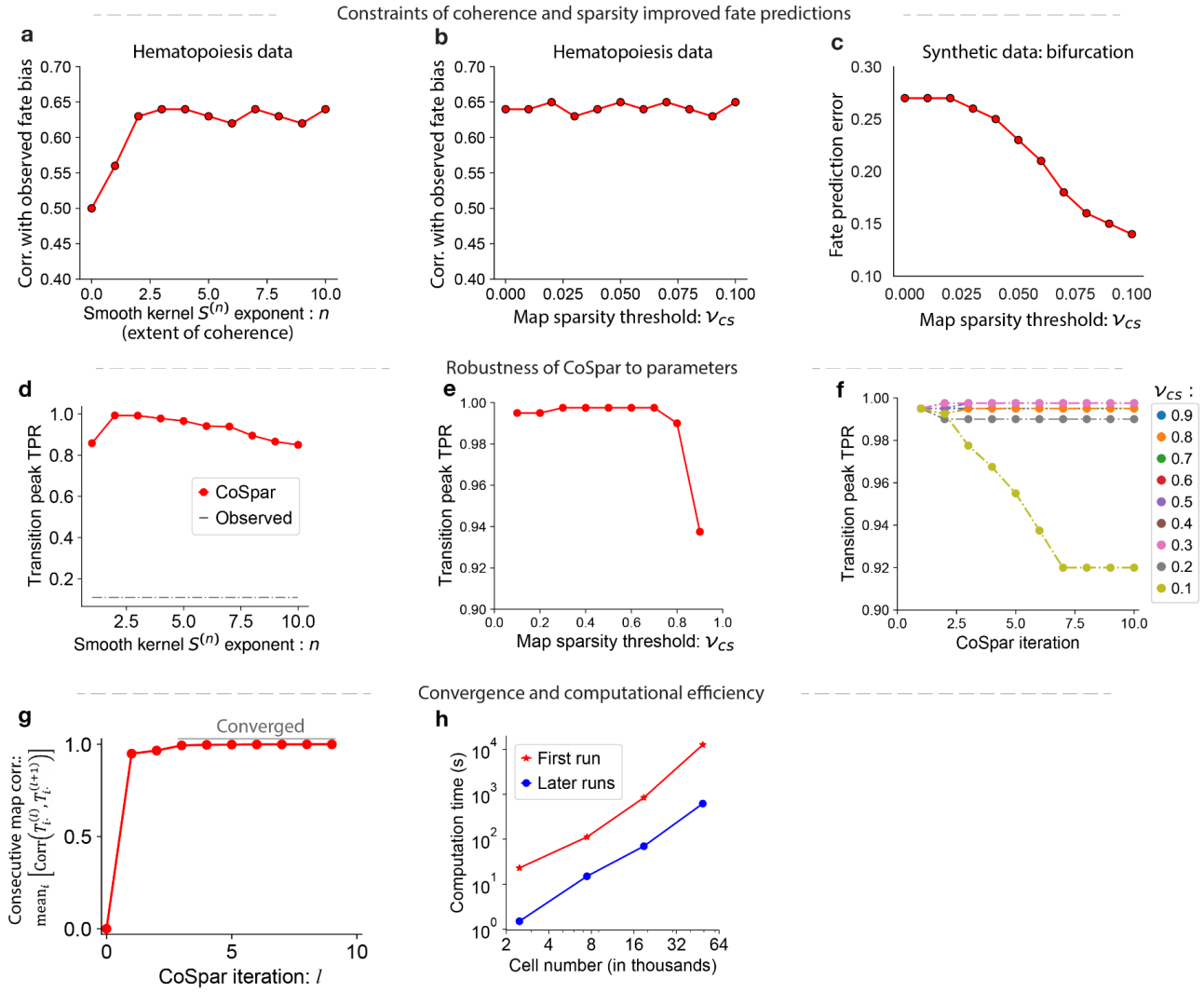
Supplementary Fig. 2. Illustration of early-time clonal dispersion, supporting Fig. 1f(v). When clones are observed at more than one time-point, clonally-related cells in the earliest observed time-point may be similar (non-dispersed), in which case the observed early state of a clone provides a good approximation of early state of clonal progeny observed later. Alternatively, clonally-related cells may have already become heterogeneous (dispersed) at the earliest observed time point. Dispersion introduces uncertainty as to the founder state of a clone, and thus introduces errors into inferred transition maps. **a,b** Examples of non-dispersed clones, where the observed initial cell (red) and the remaining sister cell (orange) are very similar in states. In these two examples, the cells observed at an early time point serve as good estimates of the ancestors of the cells observed later (faded, red). **c-e** Examples of dispersed clones, where the initial states (red=observed; orange=remaining sister cells) can be very different. In these cases, if some of the early cell states are unobserved due to drop-out (**d, e**), the apparent clonal transitions can be discordant with the underlying dynamics.



Supplementary Fig. 3. CoSpar accepts both static and cumulative barcoding data. **a,b** Illustration of static and cumulative barcoding, and their corresponding cell-by-barcode matrix.



Supplementary Fig. 4. Flowchart for using CoSpar with different experimental designs. Abbreviation: ED for Euclidean distance and SPD for shortest-path graph distance. See Methods for the definition of key parameters.



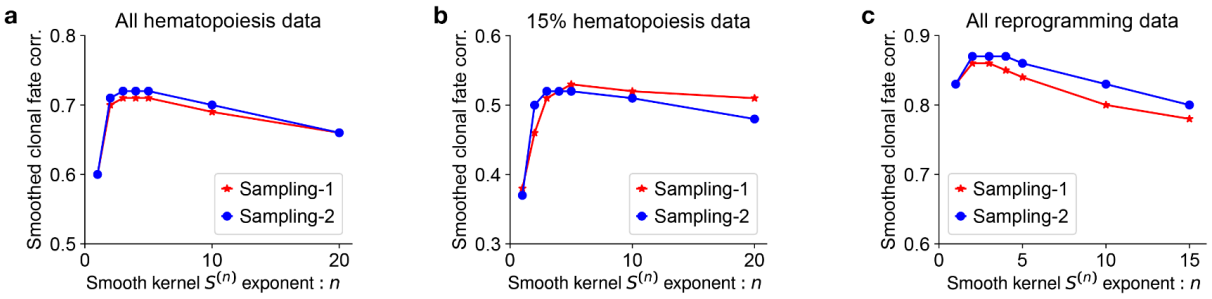
Supplementary Fig. 5. Evaluating CoSpar performance across parameter sweeps.

a-c, Evaluating the coherence and sparsity assumptions. The extent of coherence in CoSpar can be tuned with the smooth kernel exponent n, and the desired sparsity can be set with the sparsity threshold ν_{cs} (see Methods for parameter definitions). In **a** and **b**, we test the importance of these assumptions in the hematopoiesis dataset using a cross-validation test: apply CoSpar to the top 15% dispersed clones across all time points (training dataset), and train MPLClasifier with the inferred fate bias to predict the bias on the remaining 85% testing dataset. We evaluate the performance using the correlation between the predicted and the observed bias. **c**, The fate prediction error at different sparsity thresholds on the synthetic bifurcation dataset. The fate prediction error is defined as $Mean_i |Q_i - Q^*_i|$, where Q_i and Q^*_i are the predicted and expected progenitor bias for cell state i, respectively. This dataset is more dispersed than the hematopoietic dataset (Fig. 3**e**).

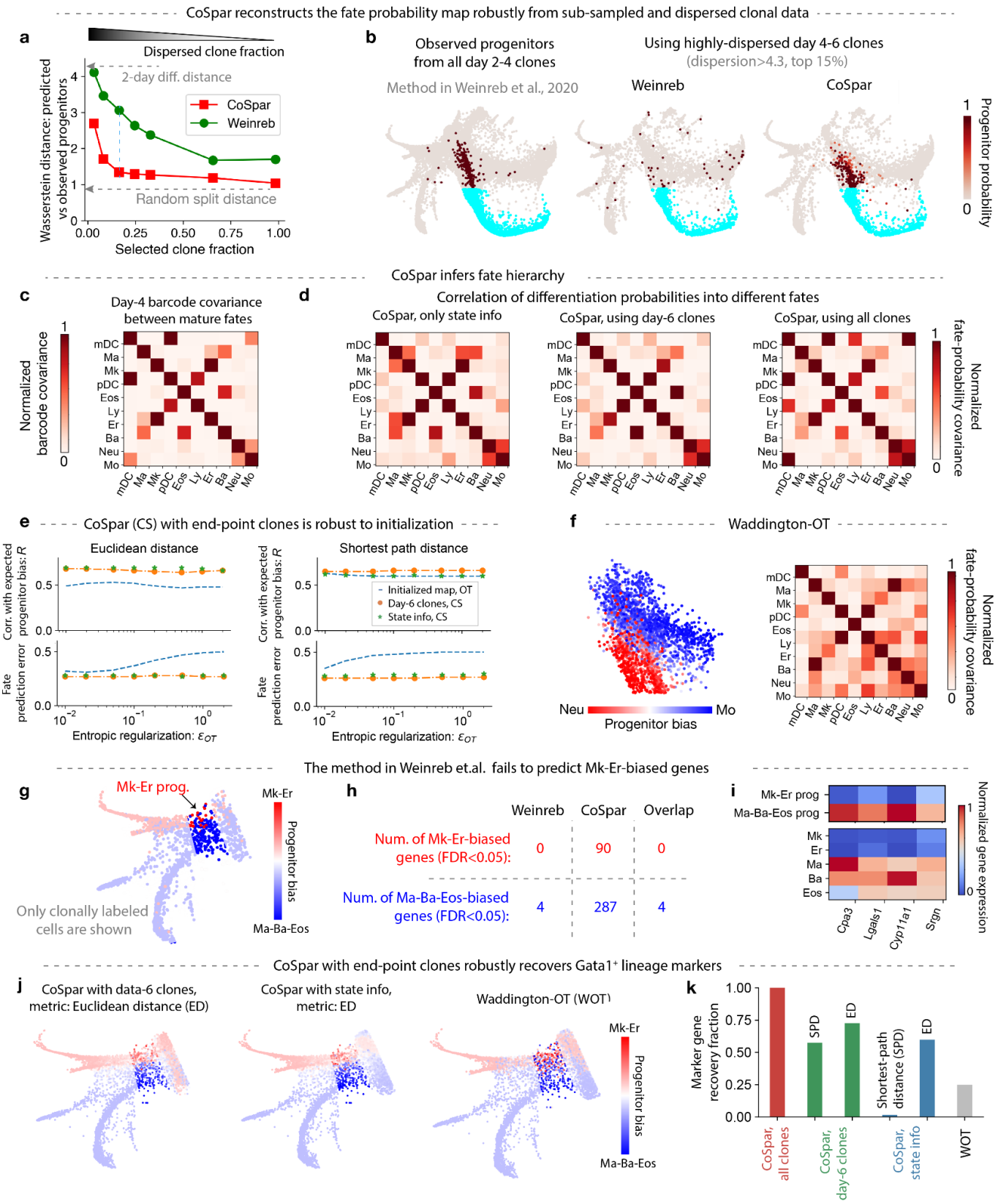
d-f, Performance of CoSpar using simulations as in Fig. 3**a-d** with a range of algorithm parameters: (**d**) smoothing kernel exponent; (**e**) sparsity threshold $\nu_{cs} \in [0, 1]$; (**f**) number of iterations, showing convergence.

g, Demonstration of algorithm convergence, seen in the correlation between maps from consecutive iterations against the number of iterations for the CoSpar algorithm (see Methods). The maps analyzed here correspond to those from the down-sampled hematopoietic dynamics (Fig. 4**h**).

h, Computational time to convergence, as a function of total cell number. In the first run, CoSpar will generate (and save) a similarity matrix, which is very costly (red curve). CoSpar can use similarity matrices generated previously to speed up computation (blue curve).



Supplementary Fig. 6. Establishing upper bounds for fate prediction after data loss. In this paper, performance of CoSpar was compared to previously published methods by discarding clonal data and then examining the fidelity of fate predictions in the face of data loss. Supporting the results reported in Figs. 4**g,i** and 5**h**, we obtain an upper bound for fate prediction, by randomly sampling 50% cells from the full ground-truth dataset in each case to predict the progenitor bias of remaining cells, with different smoothing exponents *n*. Prediction was carried out by first inferring the progenitor bias Q_i^{tr} from the training data (denoted by *tr*) to predict the bias Q_i^{tst} of the test data, by imputation via graph diffusion: $Q_i^{tst} = \sum_j S_{ij}^{(n)} Q_j^{tr}$. Results show that, in all the three cases considered, a smoothing exponent *n*=3 provided the best correlation between the imputed and actual values of Q_i^{tst}. These correlation values are indicated by the upper dashed grey lines in Figs. 4**g,i** and 5**h**.



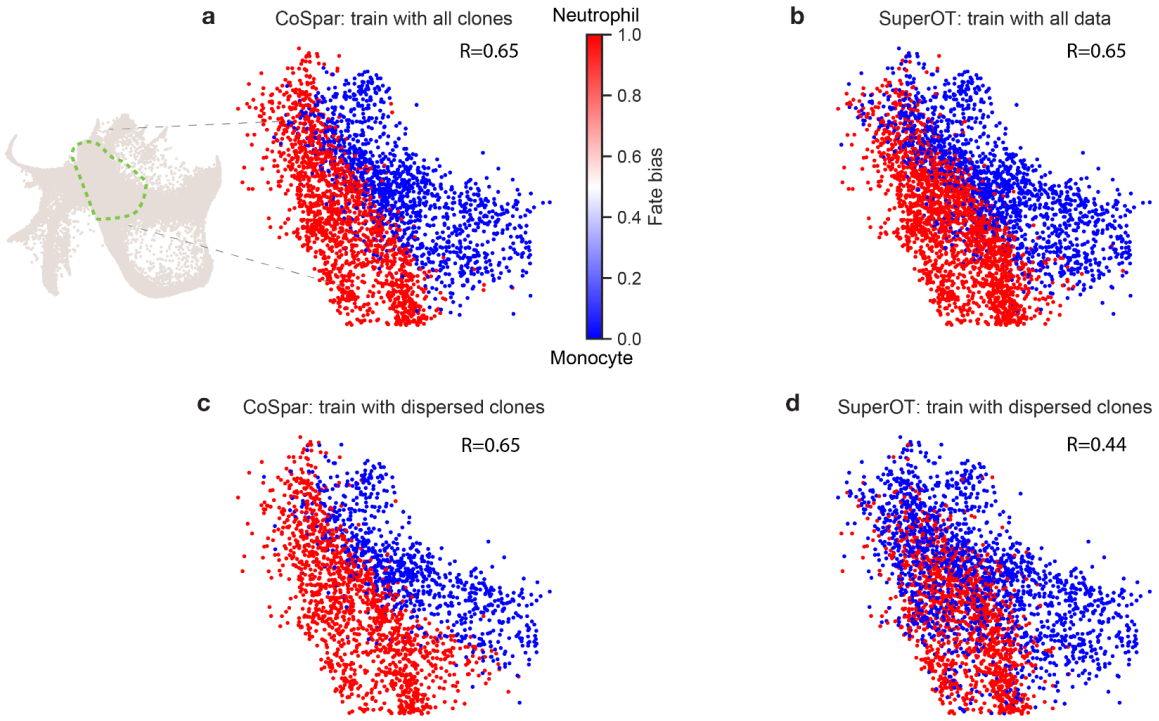
Supplementary Fig. 7. Benchmarking CoSpar in hematopoiesis.

a, CoSpar reconstructs transition maps from sub-sampled and dispersed clonal data. Here, we evaluate the prediction error as the Wasserstein distance between fraction of cell progeny predicted to occupy a given fate, compared to that obtained from the 'ground truth' transition map constructed using all clonal data rooted in day 2 clones (see main text). In

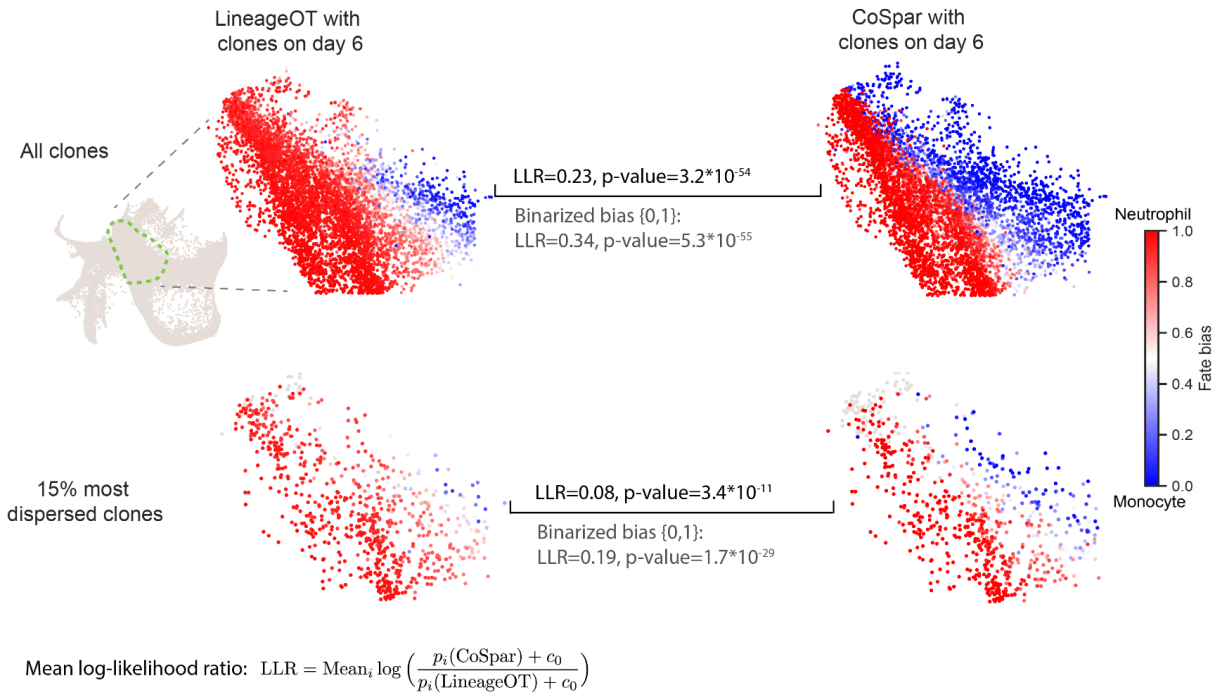
a, the prediction error is assessed for a decreasing fraction of day 4-6 clones, obtained by progressively excluding less dispersed clones that contribute the strongest signal (see Fig. 4**b**). Green curve is obtained by applying the method from the original paper. A lower bound on the error (random split distance) is the Wasserstein distance between random 50% partitions of the ground-truth data. The largest observed errors are comparable to the Wasserstein distance between populations separated by two days of progressive differentiation (upper grey arrow).
b, The ground truth and predicted fate maps for neutrophils cluster using the 15% most dispersed clones. These plots illustrate one value on the plot in **a**.
c, The normalized covariance of clonal barcode abundances between different cell types, calculated using all data on day 4 of differentiation[1].
d, The correlation of predicted transition probabilities of progenitors, inferred with CoSpar using different data indicated (See Methods).
e, Joint CoSpar optimization is robust to initialization and choice of distance metric. This panel accompanies Fig. 4**g**. The progenitor biases are calculated from the transition maps for different initialization choices of the transition map, and are evaluated using two metrics: correlation with expected fate bias and the fate prediction error (defined in Supplementary Fig. 5**c**). Optimal transport (OT) is used to initialize the transition map from state information alone prior to CoSpar. Plots scan the OT entropic regularization strength ϵ_{OT}.

f, Application of Waddington-OT (WOT) to hematopoiesis dataset. WOT was applied to the same data in Ref[2], where clonal data was used to tune the local cell proliferation rates. When WOT is applied without access to any clonal information, performance is degraded as seen by comparing the plots here to the ground truth. Plots are to be compared with those in panels **c**,**d** and Fig. 4**c**. WOT is applied with default parameters (ϵ_{OT} =0.05).

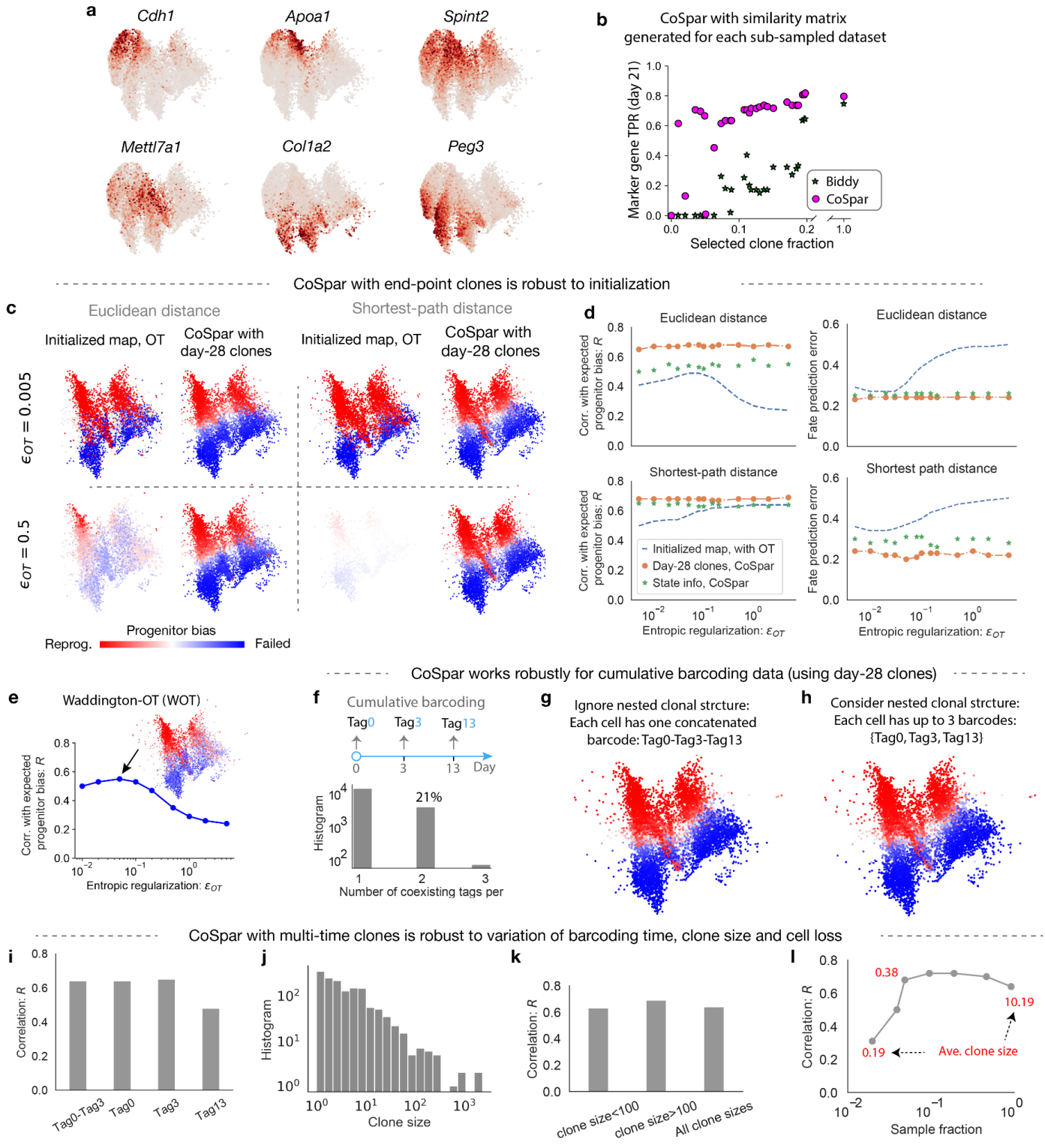
g-i, Predicting early fate boundaries in the Gata1+ lineages using the original method from Ref[2]. **g**, Predicted progenitor bias among the Gata1^{+} cells on the state embedding. **h**, Comparison of the number of differentially expressed genes (FDR<0.05) identified from different methods of clonal analysis. **i**, Gene expression heat map for all differentially expressed genes identified with the Weinreb method[2].
j,k Comparison of marker gene recovery using end-point clones or just state information. **j**, Prediction of progenitor fate bias from CoSpar with day-6 clones, CoSpar with only state information, and Waddington-OT. Here, CoSpar predictions are obtained with the Euclidean distance metric. **k**, Fraction of recovered marker genes for each method at the respective choice of distance metric (SPD: shortest-path distance, ED: Euclidean distance). As in **h**, we define the target set of markers as the 377 genes defined by CoSpar using all clonal information. We report the recovery fraction as the number of true positives divided by 377.



Supplementary Fig. 8. Benchmarking fate inference using SuperOT. SuperOT is a neuron-network-based method that integrates clonal data across time points with state information for fate prediction[3]. We use the hematopoiesis dataset to evaluate SuperOT in its ability to exploit clonal information across time points to predict progenitor fate bias between Neutrophil and Monocyte for cell states on day 2 and 4. Rather than predicting the probability of fate bias, SuperOT specifically seeks to associate early transcriptional states with later discrete fate labels, which can be interpreted to represent the majority fate of a given clone. To allow comparison of CoSpar with SuperOT, we extend CoSpar to generate a similar majority-fate prediction (See Methods). The correlation *R* between predicted and observed fate outcome is reported. In the first test (**a,b**), we use all available clonal data for both training and testing. In the second test (**c,d**), we restrict the test to the top 15% dispersed clones in the hematopoiesis dataset, and predict the fate bias of the cell states belonging to the remaining 85% clones. These tests indicate that SuperOT performance is comparable to CoSpar using all clonal data, but SuperOT is less robust when inferring a Transition Map using clonal data with weak fate biases. This difference in performance may occur because SuperOT, as currently implemented, relies on assigning each clone with a single fate outcome.



Supplementary Fig. 9. Benchmarking fate inference using LineageOT. LineageOT is a method that seeks to learn a transition map from an early time point to a later time point, using clonal data observed only at a later time point[4]. Here, we show UMAPs of scRNA-Seq data on hematopoietic differentiation (see Fig. 4 of main text), colored by the fate bias predicted by the LineageOT and CoSpar methods respectively. Top row shows predictions using the full dataset; bottom row shows predictions using only the 15% most dispersed clones to train the models (lower panels). In each case, a Transition Map is inferred, and from this map the fate bias of cells observed at day 4 of culture are calculated. LineageOT and CoSpar were run for a one time-point analysis with clonal information only on day 6 (left and right panels). We compared the model likelihoods using the Vuong Test[5], which evaluates the probability (p-value) that the two models are equally likely in light of the same test data (one-sided; see Supplementary Note 7 for test definition and Likelihood calculations). For the models trained with the full dataset of day-6 clones, we directly compared model predictions with the observed day-6 fates of each day-4 cell. For the sub-sampled test, the model predictions were then transferred to the remaining clones (the test set), by averaging the fate bias of the k-nearest training-set neighbors (k=20) of each test-set cell. The log-likelihood of the models is then evaluated from the observed fates in the test dataset (see Supplemental Note 7 for likelihood calculation; we used a pseudo-count c_0=0.01 to avoid numerical instability). The mean log-likelihood ratio (LLR) between the two models is shown, with the associated p-value of the Vuong test. We also present the LLRs and p-values using binarized fate bias prediction (i.e., binarized to be {0,1} at the threshold 0.5). In both calculations, CoSpar has a higher likelihood of predicting clonal outcomes than LineageOT that is statistically significant (Vuong test p-value<10^{-10}).



Supplementary Fig. 10. Benchmarking CoSpar in fibroblast reprogramming.

a, Expression of selected marker genes on UMAP visualizations from day 15, 21 and 28.

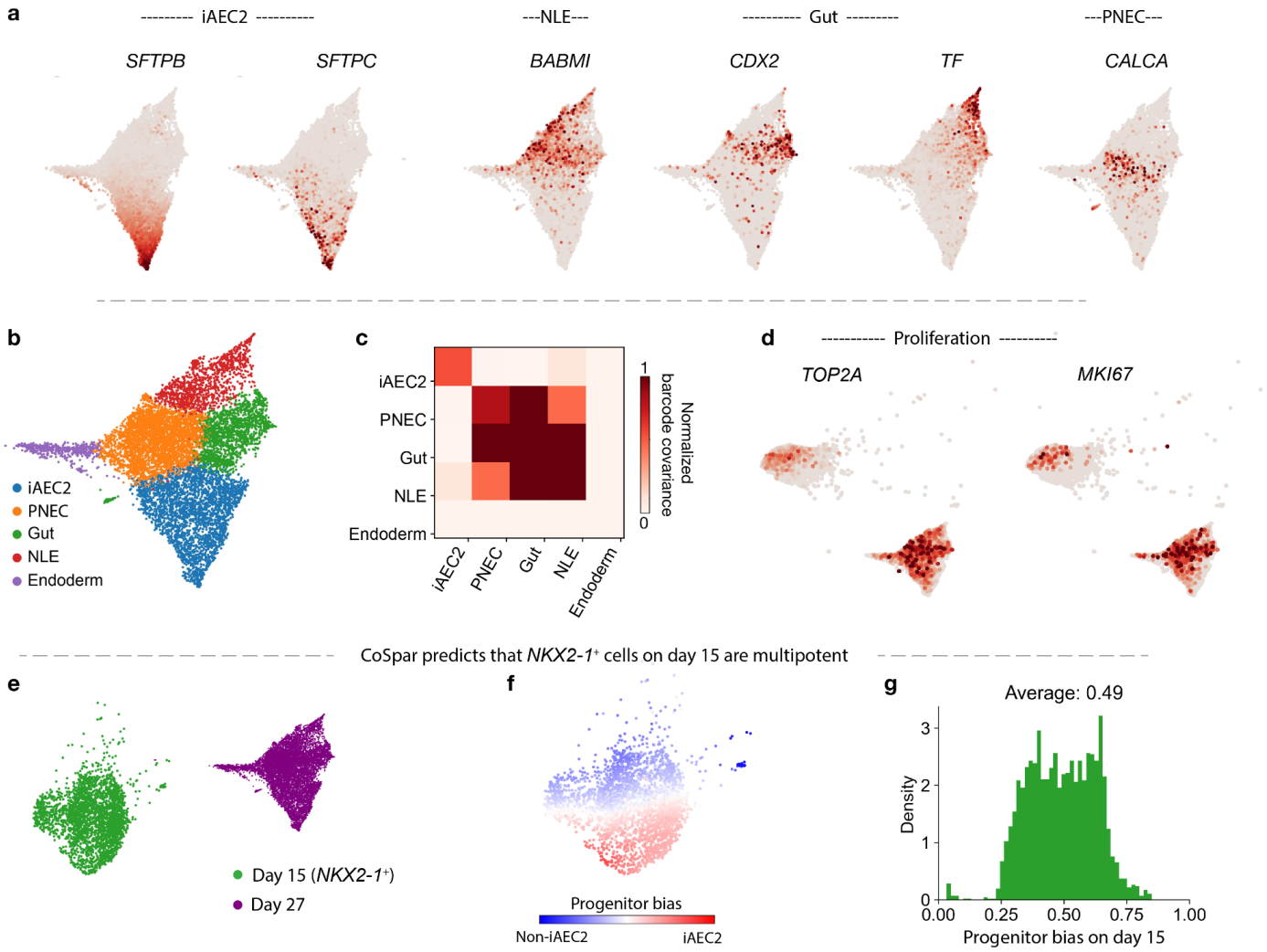
b, Reproduction of results in Fig. 5**e** using a similarity matrix obtained from each sub-sampled dataset. Results are seen to be robust to sub-sampling strategies.

c-e Transition maps inferred by CoSpar with access only to end-point clonal information are robust to the choice of initialization. These panels accompany Fig. 5**h**. **c**, Visualization of the progenitor bias derived from the initialized transition map and the corresponding CoSpar prediction, for different entropic regularizations and distance metrics as indicated. **d**, Parameter sweep quantifying the stability of the predicted progenitor bias. Two metrics are used: correlation

with expected fate bias and the fate prediction error (defined in Supplementary Fig. 5**c**). **e**, Progenitor bias prediction from Waddington-OT[6], which relies only on state information. Upper panel: the predicted progenitor bias on the state manifold at ϵ_{OT}=0.05. Lower panel: progenitor bias correlation with ground truth across different ϵ_{OT} values.

f-h, CoSpar analysis with clonal barcodes integrated at sequential time points from the reprogramming dataset[7]. The analysis was done with clonal data on day 28. **f**, The cumulative barcoding scheme in the reprogramming experiment. Cells were barcoded on day 0, 3, and 13. Bottom panel shows the histogram of the number of tags from different time points per cell. **g**, A progenitor bias prediction generated by concatenating all tags from all three time points into a single clonal barcode for each cell, thus ignoring the nested clonal structure in the data. **h**, Equivalent results of CoSpar analysis with nested clonal structure, carried out by treating Tag0, Tag3 and Tag13 as independent barcodes for a cell, such that each cell may have up to three barcodes.

i-l CoSpar with multiple clonal time points is robust to variation of barcoding time, clone size and cell loss, using correlation with expected progenitor bias (R) on day 15 and 21 as the metric (same as **d** and **e**). **i**, CoSpar gives similarly accurate predictions using barcoding tags from a single time point. **j**, The clone size distribution (each barcode is a tag0-tag3 concatenation). **k**, CoSpar performs robustly using clones with different sizes or their combination. **l**, CoSpar is robust to cell loss in a clone, modeled by sub-sampling the reprogramming dataset. The average clone sizes are annotated for selected data points (in red).



Supplementary Fig. 11. Marker gene expression and clonal structure during differentiation into alveolar cells and other endodermal cells.

a, Expression of genes associated (in Ref[8]) with iAEC2 cells, non-lung endoderm (NLE), gut endoderm, and pulmonary neuroendocrine cells (PNEC).

b, Leiden clustering of day-27 cell states. Clusters are named based on their corresponding gene expression.

c, Normalized barcode covariance on day 27 among all clusters, showing evidence of clonal partitioning of iAEC2 cells.

d, Expression of two representative genes marking proliferating cells (*TOP2A* and *MKI67*) on day 17 and 27 state manifold, showing that cells predicted by CoSpar to show low commitment on day 17 appear proliferating (Fig. 6**c**).

e-g, CoSpar predicts that lineage restriction occurs after day 15, except for a rare fraction of cells committed to non-iAEC2 fates. **e**, UMAP visualization of cell states on day 15 and 27. **f**, CoSpar-predicted progenitor bias among cells on day 15. **g**, Histogram of the progenitor bias on day 15 (shown in panel **f**). Unlike on day 17 (Fig. 6**c**), here progenitor bias is concentrated at 50%.

Supplementary Note

CONTENTS

Supplementary Note 1: Connecting transition maps to models of differentiation
Supplementary Note 2: The effect of noisy measurement on transition map inference
Supplementary Note 3: Coherent sparse optimization
Supplementary Note 4: CoSpar analysis of cumulative and evolving clonal barcodes
Supplementary Note 5: Joint inference of clonal origins and transition maps using clonal data at a single time-point
Supplementary Note 6: Transition map initialization with HighVar
Supplementary Note 7: Model comparisons for clonal fate prediction

References

Supplementary Note 1: Connecting transition maps to models of differentiation

This note grounds the finite-time transition map in a stochastic model of cell differentiation. In doing so it also clarifies what cannot be learnt from the transition map.

We begin by considering a Markov model of differentiation represented by an arbitrary graph of finite size, where each node represents a cell state. In this model, each cell probabilistically undergoes proliferation, death, and differentiation with rates that are specific to the cell state. A clone is a realization of such a stochastic branching process, seeded as a single barcoded cell in some cell state. Starting from a cell state i, k_{ij} is the differentiation rate to a different state j; b_i is the probability of a cell dividing into two cells; and d_i is the cell loss rate for cells in state i. We assume that these rates are first-order (independent of the number of cells in a state). These rates can vary with time to reflect changes in the tissue environment. Supplementary Fig. 1**a** shows a simplified example of such a model.

This model is useful in its simplicity, but it is clearly not general: being a Markov process, it assumes that we have a complete measurement of the variables that could affect state dynamics, such as the transcriptome, epigenome, and extracellular environment. This is unlikely to be true. Incomplete state measurement leads to a non-Markovian dynamics[9]. Nonetheless, our model may be a useful approximation as it generates predictions of biomarkers and fate regulators, and their correlation with fate bias.

Our goal in this paper is to learn the structure of such a graphical model (e.g. Supplementary Fig. 1**a**) and its rate constants, from LT-scSeq data. To learn a model from data, we focus most simply on the mean dynamics of cell number at each state. To do so, one could consider a complete stochastic description using the chemical master equation[10], which gives the distribution evolution over the extended state space $N \times X = \{(N_i, X_i) \ \ \forall\ i;\ \text{and } N_i = 1, 2, ...\}$, where N_i is the number of cells at state i and X_i is the corresponding state. However, because we assume a first-order model, there exists a closed-form equation for the dynamics of average cell number $\bar{N}_i(t)$ at state i and time t,

$$\frac{d}{dt}\bar{N}_i(t) = \sum_j \bar{N}_j(t) K_{ji}, \tag{1}$$

where $K_{ij} \equiv (1-\delta_{ij})k_{ij} + \delta_{ij}(b_i - d_i - \sum_{k\neq i} k_{ik})$, with $\delta_{ij} = \{1 \text{ if } i = j;\ \text{otherwise } 0\}$, is the instantaneous transition rate from state i to j that includes all cellular processes: division, cell death, and differentiation. This mean dynamics only captures the net effect of cell number change $(b_i - d_i)$, and does not distinguish whether it is from cell proliferation or loss.

To make contact with experiment, we represent the number of cells at each state as a fraction of the total cell number to obtain the cell density:

$$P_i(t) \equiv \frac{\bar{N}_i(t)}{\bar{N}(t)}, \tag{2}$$

where $\bar{N}(t) \equiv \sum_j \bar{N}_j(t)$ is the total cell number at time t. The dynamics of the cell density $P_i(t)$ is

$$\frac{d}{dt}P_i(t) = \sum_j P_j \tilde{K}_{ji}(t), \tag{3}$$

where $\tilde{K}_{ji}(t) \equiv K_{ji} - \delta_{ji}\bar{\alpha}(t)$, and $\bar{\alpha}(t) \equiv \sum_k P_k(t)(b_k - d_k)$ is the average growth rate of the population at time t. Diagonal elements in $\tilde{K}$ reflect whether net growth in each state is larger (positive) or smaller (negative) than the population average.

We now can ground the transition map T in terms of the model. Integrating Eq. (3) from time t_1 to t_2 leads to the relation

$$P_i(t_2) = \sum_j P_j(t_1) T_{ji}(t_1, t_2), \tag{4}$$

where the intrinsic finite-time transition map

$$T = \exp\Big(\int_{t_1}^{t_2} \tilde{K} dt\Big) \tag{5}$$

is obtained from matrix exponentiation of the corrected instantaneous transition rate matrix $\tilde{K}$.

The transition probability T_{ij} is the fraction of progenies from initial state i that ends at later state j (Supplementary Fig. 1**b**). To see this, we can sum over all states in Eq. (4), and noting that $\sum_i P_i(t) = 1$, we have $1 = \sum_j P_j(t_1) \sum_i T_{ji}$. This equation is valid for any distribution $P_j(t_1)$ and therefore the transition map satisfies the conservation property

$$\sum_j T_{ij} = 1. \tag{6}$$

Owing to its normalization (Eq. 6), the transition map that is experimentally accessible captures the most interesting property we want: the probability of a cell to differentiate into different cell types. A certain initial state i can transition to multiple states over time window t, i.e., T has multiple non-zero entries associated with the i-th row.

Nonetheless, it is important to note that T_{ij} is shaped both by differences in transition rates between states, and by the collective effect of proliferation and cell death along the trajectories between state i and j. Mathematically, although proliferation and cell death only affect the diagonal terms in the instantaneous transition matrix $\tilde{K}$, the matrix exponentiation in Eq. (5) will propagate this effect to the off-diagonal terms in the finite-time transition matrix T. For this reason, empirical transition maps alone obscure differences between biases in proliferation and choice towards competing fates, as illustrated in Supplementary Fig. 1**d**.

Supplementary Note 2: The effect of noisy measurement on transition map inference

In Eq. (5), the transition map is seen to emerge from stochastic state transitions accumulating over time. In practice, an inferred map is also shaped by sources of noise associated with measurement and subsequent dimensionality reduction of the data. In this note, we examine the errors propagated from different technical sources into the observed transition map T. As might be expected, we show that technical sources of error lead to a 'blurred' transition map, delocalized over the cell state graph. The smoothing kernels connecting the true and observed transition map can be understood as a matrix product of error kernels associated with each individual source of uncertainty.

a. Measurement errors. We will consider the errors associated with correctly assigning transition rates from a state i at time t_1 to state j at time t_2. Such a transition contributes to mass at matrix element $T_{ij}(t_1, t_2)$ of the transition map. At time t_2, errors in measurement re-assign cells from state j to another state k, with a probability ϵ_{jk} normalized such that $\sum_k \epsilon_{jk} = 1$. With such an error, the observed transition map now becomes $T^{(\text{obs.})}_{ij} = \sum_k T_{ik}\epsilon_{kj}$. A similar error may occur at t_1. Because technical errors may differ between time points, we will denote $\epsilon^{(i)}$ as the error in measuring the state of a cell at time t_i. Accounting for errors in two time points, the observed transition map now becomes:

$$T^{(\text{obs.})}_{ij} = \sum_{k,l} \epsilon^{(1)}_{ki} T_{kl} \epsilon^{(2)}_{lj}.$$

b. Clonal dispersion. In LT-scSeq experiments, the cells sampled at t_1 are clonally related to those that give rise to cells sampled at t_2. But being distinct, they may occupy different states. As above, we consider the error in estimating transition rates from state i at t_1 to state j at t_2. At t_1, a clonally-related state, k, is observed instead of state i, with a probability that we shall denote σ_{ik}. This probability satisfies normalization $\sum_k \sigma_{ik} = 1$. Accounting for this clonal dispersion, the observed transition map relates to the true transition map through the relation:

$$T^{(\text{obs.})}_{ij} = \sum_k \sigma_{ki} T_{kj}.$$

Note that because cells divide, more than one cell may be observed in a clone at time t_1. In this case, the error kernel σ_{ki} no longer has a unique definition because choices in constructing the transition map may assign more or less weight to particular cells within each clone. By enforcing local coherence, CoSpar strongly weights σ_{ki} towards states k that are close to i, thus reducing errors in the transition map as compared to using a 'naive' clonal analysis method such as we have previously reported[2], which weights all cells in a clone at t_1 equally.

Compounding clonal dispersion and measurement error, we recognize the the observed transition map has the form:

$$T^{(\text{obs.})}(t_1, t_2) = S_1^T T(t_1, t_2) S_2,$$

where $S_1 = \epsilon^{(1)}\sigma$ and $S_2 = \epsilon^{(2)}$.

Supplementary Note 3: Coherent sparse optimization

Our goal in dynamic inference is to learn the finite-time transition map, as defined in Eq. (4), for the set of observed cell states in a given experiment. In particular, we want to constrain the map inference with experimentally observed clones at different time points.

We first derive the mathematical constraints on T from multiple observed clones at time t_1 and t_2. In the above model of stochastic differentiation, cells in a clone are distributed across states with a time-dependent density vector $\vec{P}(t)$. The density distribution of each clone forms an independent constraint for the transition map according to Eq. (4). Given multiple clonal observations, we consider each observed cell transcriptome as a distinct state, and we can represent the density vector in this state space, i.e., $\vec{P}(t) \in \mathbb{R}^{N_t}$, where N_t is the total cell number at t, including those lacking clonal information. We then introduce $S(t) \in R^{N_t \times N_t}$ as a matrix of cell-cell similarity over all observed cell states at time t. Denoting $I(t) \in \{0,1\}^{M \times N_t}$ as a barcode-by-cell matrix of M clonal barcodes, the density profiles of observed clones $\mathbf{P}(t) \in R^{M \times N_t}$ are estimated as

$$\mathbf{P}(t) = I(t)S(t). \tag{7}$$

In matrix form, the constraint Eq. (4) from all observed clones then becomes

$$\mathbf{P}(t_2) \approx \mathbf{P}(t_1)T(t_1, t_2). \tag{8}$$

With enough clonal information, $T(t_1, t_2)$ could in principle be learnt by matrix inversion. However, the number of clonal barcodes (M) will always be far less than the number of cells profiled. To constrain the map, we require that: 1) T is a sparse matrix (Fig. 1**e**, left panel); 2) T is locally coherent (Fig. 1**e**, right panel); and 3) T is a non-negative matrix. With these requirements, the inference can be formulated as the following optimization problem:

$$\min_T ||T||_1 + \alpha||LT||_2, \text{ s.t. } \sum_m ||\vec{P}(t_2; m) - \vec{P}(t_1; m)T(t_1, t_2)||_2 \leq \epsilon;\ T \geq 0;\ \text{Normalization.} \tag{9}$$

Here, $L_{ij} = 1 - \bar{w}_{ij}/\sum_j \bar{w}_{ij}$ is the normalized graph laplacian, with w_{ij} the graph connectivity of the nearest neighbor kNN graph of cell states. $\vec{P}(t; m)$ is a row-vector representing the distributions of cell states within the m-th clone, or m-th column of the matrix $\mathbf{P}(t)$. We note that $\sum_m ||\vec{P}(t_2; m) - \vec{P}(t_1; m)T(t_1, t_2)||_2 = ||\mathbf{P}(t_2) - \mathbf{P}(t_1)T(t_1, t_2)||_2$, which gives the form of the cost function given in Fig. 2**a**. Note that Eq. (7) integrates the state information (encoded in S) and clonal information (encoded in I) into **P**. This local smoothing operation indirectly imposes coherent transitions in this system.

Before continuing, we note the relationship of this optimization problem to past literature. Absent the coherence constraint ($\alpha = 0$), this optimization problem reduces to sparse optimization by lasso regression. To our knowledge, only one study has explored the extension of lasso to enforce coherence with relation to a data embedding, called 'fused lasso' optimization[11]. Fused lasso is however different in three important ways from Eq. (9). First, it suppresses the first-order derivative of the inference target to promote coherence. Second, fused lasso was developed for 1-d or 2-d datasets, assuming a natural ordering for the observed cell states. Third, like lasso, the inference object of fused lasso is a vector. In contrast, the coherent sparse optimization in Eq. (9) is generalized to arbitrary graphs; it suppresses the second-order derivative (the curvature) to enforce coherence; and it is generalized to matrix inference.

We now discuss implementation of the optimization problem. Eq. (9) might be formulated as a quadratic programming problem, and be solved accordingly as in fussed lasso[11]. However, this strategy is very expensive computationally[11]. There could be ways to solve the optimization efficiently and exactly, and we leave it as an open problem. Instead, we provide an efficient yet heuristic way to solve the optimization. Specifically, we break down individual elements of the objective function, and propose a simple alternative for each of them.

1. *Sparsification.* Instead of including the sparsity term $||T||_1$ into the objective function, we directly apply a pre-defined thresholding to the transition map at each iteration: $T \leftarrow \theta(T, \nu)$, where

$$[\theta(T,\nu)]_{ij} = \begin{cases} T_{ij}, & \text{if } T_{ij} \geq \nu \max_j T_{ij} \\ 0, & \text{Otherwise} \end{cases} \tag{10}$$

2. *Transitions within clones.* To enforce Eq. (4) for each observed clone, we consider a clonal transition map π^m for the m-th clone, which allows only intra-clone transitions and conserves the total transition flux within a clone. We do so by projecting the transition map T and performing clone-wise normalization: $\pi^m \leftarrow \mathcal{P}_m(T)$:

$$[\mathcal{P}_m(T)]_{ij} = \frac{\tilde{\pi}^m_{ij}}{\sum_{i'j'} \tilde{\pi}^m_{i'j'}}, \tag{11}$$

where $\tilde{\pi}^m_{ij} = T_{ij}$ if the transition $i \to j$ occurs within clone m, and otherwise $\tilde{\pi}^m_{ij} = 0$. The composite map capturing all intra-clone transitions is then,

$$\mathcal{P}(T) = \sum_m \mathcal{P}_m(T) \tag{12}$$

A map constructed in this way, $\pi = \mathcal{P}(T)$, will satisfy the following equation approximately:

$$I(t_2) \approx I(t_1)\pi(t_1; t_2), \tag{13}$$

which is the clonal constraint for directly observed cell states[12]. The map $\pi(t_1; t_2)$ can be used to specify T, but being constrained to clones it is no longer coherent.

3. *Coherence.* To enforce coherence, we begin by noting that Eqs. (4), (7) and (13) together lead to the relationship $T(t_1; t_2) = S^{-1}_{t_1}\pi(t_1; t_2)S_{t_2}$. As similarity matrices S are generally non-invertable, we introduce a pseudo-inverse,

$$T(t_1; t_2) \approx S^{+}_{t_1}\pi(t_1; t_2)S_{t_2}. \tag{14}$$

Eq. (14) smoothes the transition map learnt within-clones, π, over nearby states to get a transition map T across all states. T is now a locally continuous map and satisfies the coherence constraint: similar initial cell states have similar fate outcomes.

This approach to calculating T leads to minimization of the term $\alpha||LT||_2$ in Eq. (9), although the parameter α establishing the relative weight of coherence is no longer explicitly identifiable in the procedure. It is instead reflected in the extent of smoothing.

These three steps, carried out sequentially and iteratively, define the CoSpar algorithm given in methods. Note that normalization is performed clone-wise in Eq. (12). The non-negativity constraint, $T \geq 0$, is implicitly satisfied in the above steps. In our strategy, Eq. (14) is the most time-consuming step as it involves multiplication of large matrices. CoSpar is nonetheless efficient as it carries out matrix multiplication *only* at Eq. (14), and we find that it converges within a few iterations (Supplementary Fig. 5**g**).

Supplementary Note 4: CoSpar analysis of cumulative and evolving clonal barcodes

Recent technical advances now enable the generation of clonal barcodes that encode nested clonal structure through ongoing barcode insertion or by gradual mutation of a clonal barcode[13]. When these clonal tagging (insertion/mutation) events occur across multiple cell division cycles, they will define clones identifiable by the earliest tagging event, with nested sub-clones that are identifiable through the acquisition of further clonal barcode insertions or mutations after one or more cell divisions. In this note, we discuss challenges in interpreting such structured clonal relationships in generating a transition map between cell states across two time points. We then explain how CoSpar accepts and uses nested clonal barcode data to do so.

The problem of learning a transition map between times t_1 and t_2 using nested clonal data at a single time point has been previously considered (LineageOT[4]). Clonal sub-structure resulting from division events after t_1 should not alter the transition map: this is because the map (merely) reports the fraction of progeny of each cell at t_1 to be observed at a given state at t_2, irrespective of the history of cell division and cell loss that occurred between t_1 to t_2. By contrast, each sub-clone from a division event prior to t_1 reflects a separate clonal constraint for transitions between t_1 to t_2, since each

sub-clone is rooted in a different cell state at t_1. For sub-clones that arise immediately prior to t_1, it is reasonable to expect that the progenitors of sub-clones would be similar at t_1. These general principles guide the use of nested clonal data for inferring a transition map.

Therefore, the timing of sub-clonal labeling relative to t_1 becomes important to the decision on how to interpret nested clonal information in constructing a transition map from t_1 to a later time point. In some published experiments, sub-clonal labels were introduced at defined time points[7,14]. If these time points occur after a time point of interest t_1, it becomes clear that such labeling should add no further information to a map rooted in t_1. If these time points occur prior to t_1, then sub-clonal relationships should be treated as distinct clones for the purpose of learning the transition map with CoSpar. Other recent methods, however, rely on stochastic clonal barcode integration or mutation[13], which may not allow estimating the timing of clonal barcoding. In such cases, further assumptions are needed to learn both the transition map and the timing of sub-clonal barcoding events. We have not attempted to solve this more general problem.

As a compromise, in implementing CoSpar we take the approach of treating all clonal relationships between cells with equal weight, irrespective of their nesting level. Consider a clone with three identified sub-clones. Such a clone can be represented by three barcodes, such that the cells in this clone could have any of three clonal barcode vectors: either the first barcode is present (1,0,0), or the first and second (1,1,0), or the first and third (1,0,1). Owing to drop-out events, other non-nested clonal patterns may appear such as (0,1,0), (0,0,1). All of these combinations define just three clonal constraints on the Transition Map: on all cells in the clone (1,*,*), on the first sub-clone (*,1,*), and one on the second sub-clone (*,*,1). The considerations above argue that *only* the (1,*,*) barcode should be treated as a constraint if the division event partitioning the second and third barcodes occur after t_1. And conversely, *only* the remaining two barcodes should be used if the same division event occurred prior to t_1. By including all three clonal barcodes, we effectively hedge against this uncertainty, while making use of all available sub-clonal information, and also utilizing information where drop-out events have occurred. Cells for which one may detect both the full clonal and the sub-clonal barcodes contribute to more than one penalty in the cost function, thus more strongly enforcing sub-clonal constraints.

This approach is implemented by default in CoSpar. CoSpar accepts a representation of the clonal data as a cell-by-barcode matrix, whether barcodes are strictly disjoint, or are instead nested reflecting cumulative barcoding/mutation. We have provided an illustration of the cell-by-barcode matrix for both scenarios (Supplementary Fig. 3). For cumulative barcoding, each independent variant in the tracer DNA is seen as a barcode (e.g., a given nucleotide at a given location, as shown in Supplementary Fig. 3**b**). The resulting cell-by-barcode matrix encodes all the available lineage information in the data.

This information is then utilized to enforce within-clonal transitions during each iteration of the CoSpar algorithm, described in Supplementary Notes 1-3, and in the Methods. This is specifically achieved with Eq. (3) in Methods, which we recall here gives the updated transition amplitude from state i to j as:

$$[PT]_{ij} = \min_m \frac{\tilde{\pi}^m_{ij}}{\sum_{i'j'} \pi^m_{i'j'}}$$

where $\tilde{\pi}^m_{ij}$ the transition map associated with barcode m that only allows intra-clone transitions (thus incorporating clonal observation). The normalization factor $\sum_{i'j'} \pi^m_{i'j'}$ aggregates all transitions within a clone, and thus the clonal relationships within sub-clones will be weighted more heavily than clonal relationships only within parent clones, because parent clones are larger, and because the same cells within sub-clones will be subject to more than one clonal constraint (i.e., from both parent and sub-clones).

We have explicitly tested the performance of CoSpar in dealing with cumulative mutations using the data set from (Biddy et al.[7]). This dataset was generated with 3 rounds of barcoding (or tagging) at days 0, 3 and 13 of culture. These can be considered as 3 independent mutations. In Supplementary Fig. 10**f-h**, we analyze these as nested barcodes. Using only the nested clonal information on day-28, CoSpar achieves as high a performance as the "benchmark" of two-time-point analysis.

Supplementary Note 5: Joint inference of clonal origins and transition maps using clonal data at a single time-point

The constraints used by CoSpar to learn a transition map are: 1) maximizing transitions between progenitors to clonally-related cells at the end point; 2) coherence and sparsity. When clonal data is available only at a single later time point, one must now simultaneously infer the identity of clonally-related cells at the initial time point. This problem becomes under-determined. To identify a biologically-reasonable solution, we can impose an additional constraint by further demanding a minimum global transport cost between the states at t_1 and those at t_2. This optimization problem now becomes fully determined and closely related to that pioneered by LineageOT[4], but with additional constraints of coherence and sparsity.

We are not aware of a fast and memory-efficient optimization to the joint problem of clonally-constrained, coherent, sparse optimal transport. An optimal solution may be reached by iterating over the CoSpar and transport constraints

infinitesimally and sequentially, until a map converges. Because the iteration is still very slow, we propose an approximate (non-convex) solution that utilizes the same constraints and gives rise to approximate transition maps in the benchmarking examples shown in the paper (Figs. 4, 5). Specifically, we first learn a transition map by fully enforcing global optimal transport. We then infer the most likely initial clonal states consistent with the initial map; and finally run CoSpar to learn the map consistent with the initial clonal states.

This approach replaces simultaneous optimization with sequential optimization. Although it does not solve the original problem, we nonetheless found that it generates transition maps with an accuracy comparable to those obtained from two time-point inference. Further, it is robust to how we initialize the transition map using optimal transport. Using either a Euclidean distance or a shortest-path graph distance, and using different entropic regularization parameters, we obtain a robust result (Figs. 4, 5).

Supplementary Note 6: Transition map initialization with HighVar

The HighVar method provides an approach to initialize the joint inference of T and $I(t_1)$ (see Methods). The approach is loosely motivated by the expectation that cells similar in gene expression between time points may share clonal origin. This expectation can be violated; we use it only to initialize numerical optimization.

HighVar consists of three steps: 1) Select highly variable genes that are expressed at both t_1 and t_2; 2) For each highly variable gene (indexed by m), threshold its expression to form a binary expression matrix $\hat{x}_{im} \in \{0, 1\}$ for all states observed at t_1 and t_2 to generate pseudo clonal data $\hat{I}(t_1)$ and $\hat{I}(t_2)$ from the binary expression matrix; 3) Run CoSpar with $\hat{I}(t_1)$ and $\hat{I}(t_2)$. The pseudo-clonal data $\hat{I}(t_1)$ and $\hat{I}(t_2)$ are discarded, and the resulting map T is used to initialize CoSpar with the true clonal data.

For the first step, we use the SPRING gene filtering function filter_genes with an adjustable gene variability percentile parameter HighVar_gene_pctl to select highly variable genes[15]. For the second step we discretize the gene expression of each highly-variable gene, sequentially, with a gene-specific threshold η_m:

$$\hat{I}_{im} = H\Big(x_i(m) - \eta_m\Big) \times Z_{im},$$

where $H(\cdot)$ is the Heaviside step function ($H(x) = 1$ if $x > 0$; otherwise 0), $Z_{im} = [1 - H(\sum_{m^*=0}^{m-1} \hat{I}_{im^*})]$ sums over previously considered genes to ensure that the same cell is not assigned to more than one pseudo-clone. The gene-specific threshold η_m is chosen such that every pseudo clone has the same number of cells at each time point N_t/M, where N_t is the number of observed cells at time t and M is the total number of highly variable genes (i.e., pseudo clones). In case N_t/M is not an integer, we use its ceil, i.e., $\lceil N_t/M \rceil$, and stop the clonal matrix update when all cells are clonally labeled.

Supplementary Note 7: Model comparisons for clonal fate prediction

In this note we show the application of Vuong's test[5] to test the null hypothesis that two competing models of clonal fate inference are equally likely in light of an observed data set. Vuong's test specifically considers non-nested models. This note has been applied to evaluate model likelihoods resulting from two algorithms: LineageOT[4] and CoSpar. The same approach could in principle be generalized to evaluate predictions of other models discussed in this paper and in future work. We introduced this test late in the study, and thus have not applied it to the majority of comparisons made in this paper. Results derived from this analysis are shown in Supplementary Fig. 9.

To compare models, one must assess the Likelihood of each model given observed data $\mathcal{D}$. The data consists of a set of observed clones with cells occupying one of M fates. The k-th clone has N_k cells, which are distributed into these fates as follows: $\vec{n}_k = \{n_k^{(1)}, \ldots, n_k^{(M)}\}$ with $\sum_j n_k^{(j)} = N_k$. The full dataset is $\mathcal{D} = \{\vec{n}_1, \ldots, \vec{n}_Z\}$ for Z clones in total.

We specifically wish to compare between models $\mathcal{M}_1, \mathcal{M}_2$ that predict the frequency at which cells in each clone should be observed in each fate. For this study, the predictions are derived via an inferred Transition Map, and are thus predicted from the state of a clone observed at an earlier time point. A Transition Map is learnt on a fraction of the clonal data, and then fate predictions are evaluated using clones reserved as the test data set $\mathcal{D}$. For the k-th clone, the prediction is encoded by a vector $\vec{p}_k = \{p_k^{(1)}, \ldots, p_k^{(M)}\}$ with $\sum_j p_k^{(j)} = 1$. The full predictions for one model are the set $\mathcal{M} = \{\vec{p}_1, \ldots, \vec{p}_Z\}$ for Z clones in total.

To compare models, we first calculate the Likelihood of the data given the model, and then invoke Bayes' theorem to obtain model Likelihoods. The probability of observing the fate vector $\vec{n}_k$ for the k-th clone, given a model $\mathcal{M}$, is the

multinomial:

$$P(\vec{n}_k|\vec{p}_k) = \binom{N_k}{n_k^{(1)}\ n_k^{(2)} \cdots n_k^{(M)}} \prod_{j=1}^{M} (p_k^{(j)})^{n_k^{(j)}}$$

We will assume that each clone is sampled independently, which is correct for the data sets described here. The probability of observing the full set of clones $\mathcal{D}$, given the full set of predictions $\mathcal{M}$ is thus the product of individual clone Likelihoods:

$$P(\mathcal{D}|\mathcal{M}) = \prod_{k=1}^{Z} P(\vec{n}_k|\vec{p}_k).$$

In comparing models, we can treat the Likelihood of each model as: $\mathcal{L} \propto P(\mathcal{D}|\mathcal{M})P(\mathcal{M})$ where $P(\mathcal{M})$ is the prior belief in the model. We take $P(\mathcal{M})$ to be uniform for all models.

The above expressions allow calculating a Log-Likelihood Ratio (LLR) between two alternative models:

$$\mathrm{LLR} = \log\left(\frac{\mathcal{L}_2}{\mathcal{L}_1}\right) = \sum_{k=1}^{Z}\sum_{j=1}^{M} n_k^{(j)} \log(q_k^{(j)}/p_k^{(j)})$$

Where p, q represent predicted frequency of cells in fate j in clone k in models $\mathcal{M}_1, \mathcal{M}_2$ respectively.

The question we wish to ask is whether a non-zero LLR represents a significant difference in the likelihood of the two models. For nested models, the likelihood ratio test can be used to address this question. The case we are dealing with here is one of non-nested models, which nonetheless address predictions on the same data. For such models, Vuong derives an approach to estimate the probability (p-value) that of the observed LLR value, under the null hypothesis that the two models are equally likely[5]. For readers' convenience, we point out the intuition behind Vuong's test. Consider each individual data point – in this case each individual cell for which we predict future fate – as a separate test of the model. We can define a log-Likelihood ratio (LLR) for the k-th cell (or clone) as:

$$\lambda_k = \log\left(\frac{\mathcal{L}_2^{(k)}}{\mathcal{L}_1^{(k)}}\right) = \sum_{j=1}^{M} n_k^{(j)} \log(q_k^{(j)}/p_k^{(j)})\,.$$

Note that the overall LLR can be re-written in terms of λ_k, as $\mathrm{LLR} = Z \times \bar{\lambda}$, where $\bar{\lambda}$ is the mean of the individual test λ_k values. Now we can consider the uncertainty in estimating $\bar{\lambda}$. With enough clonal observations, owing to the Central Limit Theorem, then the distribution of λ_k values should be normal $\bar{\lambda} \sim N(0, SEM_{\bar{\lambda}})$ under the null hypothesis that the models are equally likely. With this in mind, the Vuong test statistic is:

$$v = \frac{\bar{\lambda}}{\sigma_\lambda/\sqrt{Z}},$$

with λ_k as defined above for each clone, $\bar{\lambda} = \frac{1}{Z}\sum_{k=1}^{Z}\lambda_k$, and $\sigma_\lambda = \sqrt{\frac{1}{Z-1}\sum_{k=1}^{Z}(\lambda_k - \bar{\lambda})^2}$. And the (one-tailed) p-value is:

$$p = 1 - \mathrm{normcdf}(|v|) = (1 - \mathrm{erf}(|v|/\sqrt{2}))/2.$$

A version of this test has been implemented by Skipper Seabold at `https://gist.github.com/jseabold/6617976`.

---

[1] C. Weinreb and A. M. Klein, Proc. Natl. Acad. Sci. U. S. A. **117**, 17041 (2020).
[2] C. Weinreb, A. Rodriguez-Fraticelli, F. D. Camargo, and A. M. Klein, Science **367** (2020).
[3] N. Prasad, K. Yang, and C. Uhler, arXiv preprint arXiv:2007.12098 (2020).
[4] A. Forrow and G. Schiebinger, Nat. Commun. **12**, 1 (2021).
[5] Q. H. Vuong, Econometrica , 307 (1989).
[6] G. Schiebinger, J. Shu, M. Tabaka, B. Cleary, V. Subramanian, A. Solomon, J. Gould, S. Liu, S. Lin, P. Berube, *et al.*, Cell **176**, 928 (2019).
[7] B. A. Biddy, W. Kong, K. Kamimoto, C. Guo, S. E. Waye, T. Sun, and S. A. Morris, Nature **564**, 219 (2018).

[8] K. Hurley, J. Ding, C. Villacorta-Martin, M. J. Herriges, A. Jacob, M. Vedaie, K. D. Alysandratos, Y. L. Sun, C. Lin, R. B. Werder, *et al.*, Cell Stem Cell **26**, 593 (2020).
[9] S.-W. Wang, K. Kawaguchi, S.-i. Sasa, and L.-H. Tang, Phys. Rev. Lett. **117**, 070601 (2016).
[10] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[11] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, J. R. Stat. Soc. **67**, 91 (2005).
[12] One can appreciate that this equation is approximately satisfied because $I(t_1)\pi(t_1;t_2)$ gives a matrix with non-zero values at clonally observed states at t_2. Therefore $I(t_1)\pi(t_1;t_2)$ has the same sparse structure as $I(t_2)$ but will differ in the exact non-zero values because $I(t_2)$ is strictly binary.
[13] D. E. Wagner and A. M. Klein, Nat. Rev. Genet. (2020).
[14] B. Raj, D. E. Wagner, A. McKenna, S. Pandey, A. M. Klein, J. Shendure, J. A. Gagnon, and A. F. Schier, Nat. Biotech. **36**, 442 (2018).
[15] C. Weinreb, S. Wolock, and A. M. Klein, Bioinformatics **34**, 1246 (2018).