Supplementary Information for "A stochastic model dissects single-cell states in biological transition processes"

Jonathan W. Armond[1], Krishanu Saha[2], Anas Rana[1,4] Chris Oates[1,4,5], Rudolf Jaenisch[3,6], Mario Nicodemi[7,8], Sach Mukherjee[4,8]

November 21, 2013

Contents

[1]Centre for Complexity Science, University of Warwick, Coventry, UK

[2]Department of Biomedical Engineering, University of Wisconsin-Madison, Madison, WI, USA

[3]The Whitehead Institute for Biomedical Research, Massachusetts Institute of Technology, Cambridge, MA, USA

[4]Netherlands Cancer Institute, Amsterdam, The Netherlands

[5]Department of Statistics, University of Warwick, Coventry, UK

[6]Department of Biology, Massachusetts Institute of Technology, Cambridge, MA, USA

[7]Dip.to di Scienze Fisiche, Univ. di Napoli *"Federico II"*, INFN Napoli, Italy

[8]Corresponding authors

1 Introduction

In this Supplementary Information file we describe in further detail the model used in Main Text and present some further supporting results and methods. In Section 2 we discuss the gene expression data, normalization and gene filtering. Section 3 provides technical details concerning the STAMM (State Transitions using Aggregated Markov Models) methodology as well as empirical results investigating the robustness of reported results to data perturbation (additive noise and deletion of entire time-points). Section 4 briefly introduces the STAMM software used to produce the results. Section 6 describes model selection approaches to explore the number of model states.

In addition to Table S1 of this manuscript, which contains a list of core reprogramming genes discussed in the Main Text, three Microsoft Excel files accompany this document that are referred to in Main Text and SI. These files contain:

- Table S2: The full list of state-specific signatures for the four states in the model we discuss in the Main Text, over all genes considered (i.e. all estimated parameters β_{ij}), including the scores associated with profiles shown in Main Text Fig. 4.
- Table S3: The top ranked gene pairs by their state discriminatory power score.
- Table S4: Gene Set Enrichment Analysis results for the five profiles discussed in Main Text and shown in Fig. 4 therein.

We also include in this document the following **Supplementary Figures**:

- Fig. S1: Examples and brief description of the considered reprogramming systems.
- Fig. S2: Comparison of fit with parameter $\lambda = 0.1$ vs $\lambda = 0$.
- Fig. S3: Comparison of the stochastic model with different number of states fitted to the Samavarchi-Tehrani et al. [1] dataset that is focused on in the Main Text.
- Fig. S4: Gene ontology terms overrepresented in lists of genes specifically expressed in each state.
- Fig. S5: Robustness of results reported for the Samavarchi-Tehrani et al. [1] dataset to data perturbation.

- Fig. S6: Results using data from Samavarchi-Tehrani et al. [1] in which time indices were randomly permuted.
- Fig. S7: Comparison of STAMM with gene ranking using differential expression and temporal change criteria.
- Fig. S8: Fit of four-state model to a dataset due to Mikkelsen et al. [2].
- Fig. S9: Number of states in single-cell mRNA-seq data of inner cell mass to embryonic stem cell transition correlates with discriminatory power of gene pairs according to model fitted to Samavarchi-Tehrani et al. [1] data.
- Fig. S10: Effect of gene expression variability and detection thresholds on cell fraction measurements.
- Fig. S11: State-specific pairs of marker genes.

All computations were carried out in MATLAB R2009b or R2011b. Nonlinear least squares fitting was done using the function `lsqnonlin`; k-means clustering using the function `kmeans` (with 1000 re-starts and remaining settings as default).

2 Gene expression data

We used microarray data previously published by Samavarchi-Tehrani, et al. [1] (Samavarchi-Tehrani system). The Samavarchi-Tehrani system was based on reprogramming secondary MEFs by factor expression introduced by piggyBac transposon. Cultures were grown as polyclonal cultures. Gene expression levels were assayed at 0, 2, 5, 8, 11, 16, 21 and 30 days using microarrays. We also used microarray data from a reprogramming system due to Mikkelsen, et al. [2] (Mikkelsen system). The Mikkelsen system is a primary MEF system infected with inducible lentiviruses. Gene expression was assayed at 0, 4, 8, 12 and 16 days. A summary of the systems is shown in Fig. S1.

2.1 Normalization

Microarray data was standardized on a per gene basis. For each gene j we used transformed data $y_j(t) = (z_j(t) - \mu_j)/\sigma_j$, where $z_j(t)$ is the original data (log-transformed and pre-processed as described in the references) and μ_j and σ_j are the mean expression and standard deviation, respectively, across timepoints for gene j.

2.2 Gene filtering

For genome-wide fits to the Samavarchi-Tehrani et al. [1] dataset, we sought to remove genes that were uninformative with respect to dynamics or expressed at very low levels (with time-varying profiles therefore likely to be dominated by noise). We therefore removed genes that were in the bottom quartile of mean expression and then, from the remaining genes, those genes that were in the bottom quartile with respect to standard deviation across time. This left a total of 4383 genes that were used for genome-wide fits.

3 STAMM: model and estimation

In our model a latent stochastic process is used to describe state transitions at the single-cell level. Aggregation of these latent processes yields a likelihood at the level of the cell population that is then used to estimate model parameters from cell-population-averaged (i.e. homogenate) time-course data. This gives information on transition rates, the changing state fractions in the cell population as a function of time, and state-specific expression signatures. For simplicity, we describe the model here with reference to stem cell reprogramming and for a model with forward transitions only, as discussed in the Main Text.

3.1 Markov chain

Transitions at the single-cell level are described by a latent continuous-time Markov chain whose discrete state space is identified with the cell states of interest. We assume that the initial population is a homogeneous population of cells in an initial, MEF state and allow transitions to a second state with mean rate $w_{1,2}$ (Fig. 3a). Similarly, transitions occur between the i^{th} and $(i+1)^{\text{th}}$ states with rate $w_{i,i+1}$ until a cell reaches the final n^{th} state. The transition rates determine the probability $p_i(t)$ that a cell is in state i at time t (Fig. 3c), via the master equation (where we define $w_{0,1} = w_{n,n+1} = 0$):

$$\frac{dp_i(t)}{dt} = w_{i-1,i}p_{i-1}(t) - w_{i,i+1}p_i(t). \tag{1}$$

This model can be solved fully from (1) to give the probabilities $p_i(t)$. For the four-state model that we focus on, we have:

$$p_1(t) = e^{-w_{1,2}t} \tag{2}$$

$$p_2(t) = \frac{w_{1,2}}{w_{2,3} - w_{1,2}} \left(e^{-w_{1,2}t} - e^{-w_{2,3}t}\right) \tag{3}$$

$$p_3(t) = \frac{w_{1,2}w_{2,3}}{w_{2,3} - w_{1,2}} \left(\frac{e^{-w_{1,2}t} - e^{-w_{3,4}t}}{w_{3,4} - w_{1,2}} - \frac{e^{-w_{2,3}t} - e^{-w_{3,4}t}}{w_{3,4} - w_{2,3}}\right) \tag{4}$$

$$p_4(t) = 1 - p_1(t) - p_2(t) - p_3(t). \tag{5}$$

State-specific gene expression is described by parameters β_{ij} that represent the mean expression level for gene j in state i ("state-specific signatures"). For a large number of cells, and assuming independence between the latent, cell-level stochastic processes, we have population-averaged expression $x_j(t)$ of gene j at time t as $x_j(t) = \beta_{1j}p_1(t) + \beta_{2j}p_2(t) + \cdots + \beta_{nj}p_n(t)$ (here, for simplicity

observation noise is suppressed, we introduce noise and arrive at the likelihood below). We observe average expression at discrete times $t \in \{1 \dots T\}$. Letting $x_j = [x_j(1) \dots x_j(T)]^{\mathrm{T}}$ we can write

$$x_j = P\beta_j, \tag{6}$$

where P is a $T \times n$ matrix with entries $P_{ti} = p_i(t)$, and $\beta_j = [\beta_{1j} \dots \beta_{nj}]^{\mathrm{T}}$.

3.2 Parameter estimation

We estimated parameters $w_{i,i+1}$ and β_{ij} using a ℓ_1-penalized estimator related to the maximum *a posteriori* or MAP estimator for the Bayesian model described below. Specifically, for each gene j the likelihood was penalized by $\lambda \sum_{i=1}^{n} \beta_{ij}$, where λ is a tuning parameter used to control the extent of penalization. Non-linear least squares was used to maximise the resulting penalized likelihood. Setting λ to a small positive value regularizes estimation, reducing estimator variance (i.e. improving stability). Here, we set $\lambda = 0.1$ via cross-validation. Specifically, at each iteration one time-point was held out, and the entire procedure repeated for several candidate values of λ in the range 0.01 to 0.3. The λ with lowest overall cross-validation error was chosen. Comparing results reported with the unpenalized ($\lambda = 0$) case (i.e. a standard least squares fit), we found good agreement, showing that the penalty was not overly influential (Fig. S2).

To fit $w_{i,i+1}$ and β_{ij} for large sets of genes, we focused first on a small subset of k_1 genes that collectively captured observed classes of expression dynamics and fit parameters only for these genes. The subset was chosen by clustering the time-course data for the reprogramming-associated genes listed in Table S1 into k_1 clusters (we used K-means clustering with $k_1 = 7$ clusters; increasing k_1 did not significantly improve the K-means objective function). The gene with profile closest to each cluster centroid was chosen as a representative for that cluster. We then estimated parameters $w_{i,i+1}$ and β_{ij} for the k_1 representative genes. Noting that, conditional on rates $w_{i,i+1}$, eq. (1) from the Main Text is independent between genes j, we used rates estimated from the k_1 representative genes to sequentially fit the remaining genes by nonlinear least squares (estimated rates did not depend strongly on k_1, nor on the precise subset of representative genes). Following this procedure we fit a total of 4383 genes (see section 2 for gene filtering procedure; β_{ij}'s are listed in Table S3).

3.3 Robustness to data perturbation

We checked robustness to data perturbation in two ways: (i) a parametric approach using additive Gaussian noise and (ii) deletion of entire time-points. In each case, data were perturbed over several iterations. At each iteration, a perturbed dataset was created, as described below, and the models were fitted to the perturbed data. Then, results obtained across iterations were compared to assess robustness.

3.3.1 Additive noise

Here, data was perturbed by addition of (zero-mean, Gaussian) noise. At each iteration, noise was added to create a perturbed dataset, and models fitted to the perturbed data. Then, results were compared between iterations to assess robustness. Specifically, we re-estimated model parameters using the following perturbed dataset

$$y'_j(t) = y_j(t) + \eta \tag{7}$$

where $y_j(t)$ denote the original (log-transformed and standardized) data and $\eta \sim N(0, \sigma^2)$ with $\sigma = 0.2$. We repeated this procedure over ten iterations. Fig. S5a shows results obtained for the four-state model for the Samavarchi-Tehrani et al. [1] dataset. Results appear robust to additive noise.

3.3.2 Deletion of time-points

We tested robustness against deletion of an entire time-point from the time-course data. At each iteration, all data for one time-point from the Samavarchi-Tehrani et al. [1] data was removed, and the models fitted to the remaining data. Since the time-course comprises a total of 8 time points, such deletion represented a dramatic perturbation, with 1/8 of the data removed at each iteration. Results are shown in Fig. S5b. We find that while variability is increased compared with the milder perturbation provided by additive Gaussian noise, the main results remain robust.

3.3.3 Random permutation of time-points

We also tested the model behavior under random permutation of time points, i.e. random reordering of time indices for each gene expression value. We produced a set of randomized data where for each random sample time indices were permuted at random and the entire model was re-fit to the permuted data. Results are shown in Fig. S6. Model fitting error (i.e. residual sum-of-squares, see Section 6) for the permuted data is shown alongside the corresponding result

from the original data (i.e. with correct temporal order, as in Fig. 2a in Main Text). The condition number of the state signature matrix (a measure of the distinctness of state signatures, see Section 6 below) for the permuted and original data is also shown. We also show results from carrying out Bayesian model selection, as described in Main Text and below, on the permuted data. We find that both model fit and distinctness of state signatures are systematically worse under permutation of time indices. While Bayesian model selection applied to the original data showed clear evidence of intermediate states (Fig. 2c, Main Text), this is completely lost in the case of permuted data. These findings suggest that our simple model of transition dynamics captures real temporal structure in the data.

4 Using the STAMM software

The analysis presented in the Main Text was produced using a tool called STAMM (State Transitions using Aggregated Markov Models) implemented in MATLAB. STAMM enables the identification of cellular state transitions in time-course genomic data, the determination of state-specific expression signatures and offers a number of ways of analyzing the fitted model. Model fitting was carried out as described above and in Methods. STAMM was used to rank pulse and switch genes (for the reprogramming study presented in Main Text) as described in Methods.

Here we describe in brief the use of the STAMM tool. The software is easily adaptable to other datasets; instructions are provided with the software download available at `mukherjeelab.nki.nl/CODE/STAMM.zip`.

The basic procedure for using the software is as follows:

1. Read time course data into a MATLAB structure as detailed in README.txt.
2. Run stammClusterGenes to identify core expression profiles for initial fitting.
3. Run stammFitCluster to fit the core genes to data and then use the parameters obtained from those genes to fit the remaining genes.
4. Use stammPlotFit to display the fitted model against data.

An example session follows:

```
>> mkdir 'results'
>> data = stammPrepareSamavarchiTehraniData('s-t.csv',1,5);
Per gene normalization
Gene set size: 85

>> stammClusterGenes(data,7,'results/cluster.mat');
Cluster 1:
Gdf3 Tdgf1 Sall4 Utf1 Esrrb Gata4 Nanog Ctnnbl1 Slc2a1 Ctcf Dnmt3b Ezh2
Csnk2a1 Dnmt1 Prmt7 Cdc20 Mad2l1 Ccnf Fgf4 Bub1 Hprt1 Suz12 Eed Phc1

Cluster 2:
Inadl Cdh1 Epcam Ocln Crb3 Esrp1 Cldn3 Cldn4 Cldn7 Terf1 Alpl Prdm1
Kdm1 NrOb1 Nacc1 Zfp248 Rnf2 Tcf3
Cluster 3:
Cldn11 Cdkn2a Fut4 Gsk3b Ccnd1
Cluster 4:
Cdh2 Ncam1 Thy1 Tgfb1 Col5a2 Nes Notch1
Cluster 5:
Zeb1 Zeb2 Snai2 Snai1 Cdkn2b Stat3 Lifr
Cluster 6:
Tgfb2 Bmi1 Jag1
```

```
Cluster 7:
Zfp42 Lin28 Dnmt3l Dppa4 Pecam1 Tcf7 Klf2 Dazl Dppa2 Dppa3 Nodal Lefty1
Tcl1 Dppa5a Tbx3 Myst4 Nr6a1 Fgf5 Trim28 Klf5 Rest

Representative cluster genes: Gdf3 Cdh1 Ccnd1 Col5a2 Zeb1 Bmi1 Dppa4

>> stammFitCluster(data,'stammIps4StateFwd',100,'results/cluster.mat','test',0);
Fitting 7 genes from clustering (78 remaining):
Gdf3 Cdh1 Ccnd1 Col5a2 Zeb1 Bmi1 Dppa4
RSS = 6.612584
Fitting Lifr (77 remaining): RSS = 3.189699
Fitting Cldn11 (76 remaining): RSS = 4.163793
Fitting Inadl (75 remaining): RSS = 0.622107
Fitting Fut4 (74 remaining): RSS = 2.557118
Fitting Nr0b1 (73 remaining): RSS = 0.654309
Fitting Terf1 (72 remaining): RSS = 0.787887
Fitting Rnf2 (71 remaining): RSS = 0.968732
Fitting Cldn7 (70 remaining): RSS = 0.746953
Fitting Suz12 (69 remaining): RSS = 1.388244
Fitting Ocln (68 remaining): RSS = 0.748871
Fitting Myst4 (67 remaining): RSS = 2.643615
Fitting Crb3 (66 remaining): RSS = 0.519120
Fitting Epcam (65 remaining): RSS = 0.269488
Fitting Csnk2a1 (64 remaining): RSS = 1.305327
Fitting Dnmt3b (63 remaining): RSS = 1.177557
Fitting Ctnnbl1 (62 remaining): RSS = 0.542358
Fitting Sall4 (61 remaining): RSS = 0.285649
Fitting Esrp1 (60 remaining): RSS = 0.246314
Fitting Cdc20 (59 remaining): RSS = 0.783199
Fitting Alpl (58 remaining): RSS = 0.249918
Fitting Cldn3 (57 remaining): RSS = 0.920376
Fitting Cldn4 (56 remaining): RSS = 0.301463
Fitting Nanog (55 remaining): RSS = 0.057316
Fitting Zfp248 (54 remaining): RSS = 0.948551
Fitting Phc1 (53 remaining): RSS = 0.375057
Fitting Fgf4 (52 remaining): RSS = 1.666604
Fitting Nacc1 (51 remaining): RSS = 1.159014
Fitting Tdgf1 (50 remaining): RSS = 0.158517
Fitting Tgfb2 (49 remaining): RSS = 1.409963
Fitting Prdm1 (48 remaining): RSS = 2.013728
Fitting Esrrb (47 remaining): RSS = 0.479687
Fitting Cdh2 (46 remaining): RSS = 1.322494
Fitting Nr6a1 (45 remaining): RSS = 2.763035
Fitting Slc2a1 (44 remaining): RSS = 2.235182
Fitting Cdkn2a (43 remaining): RSS = 2.898367
Fitting Kdm1 (42 remaining): RSS = 0.529218
Fitting Mad2l1 (41 remaining): RSS = 1.042570
Fitting Ezh2 (40 remaining): RSS = 0.787894
Fitting Tcf3 (39 remaining): RSS = 4.034229
Fitting Utf1 (38 remaining): RSS = 0.308563
Fitting Ctcf (37 remaining): RSS = 0.883446
Fitting Prmt7 (36 remaining): RSS = 0.443753
Fitting Zfp42 (35 remaining): RSS = 1.407661
Fitting Ncam1 (34 remaining): RSS = 0.950230
Fitting Lefty1 (33 remaining): RSS = 1.467478
```

```
Fitting Nodal (32 remaining): RSS = 2.315962
Fitting Gata4 (31 remaining): RSS = 0.539001
Fitting Snai2 (30 remaining): RSS = 1.086668
Fitting Ccnf (29 remaining): RSS = 1.307029
Fitting Bub1 (28 remaining): RSS = 1.236361
Fitting Nes (27 remaining): RSS = 3.861888
Fitting Tgfb1 (26 remaining): RSS = 1.182227
Fitting Dnmt3l (25 remaining): RSS = 0.985564
Fitting Tcl1 (24 remaining): RSS = 0.717098
Fitting Zeb2 (23 remaining): RSS = 0.564076
Fitting Cdkn2b (22 remaining): RSS = 2.406205
Fitting Thy1 (21 remaining): RSS = 1.053211
Fitting Stat3 (20 remaining): RSS = 1.886879
Fitting Pecam1 (19 remaining): RSS = 2.126603
Fitting Snai1 (18 remaining): RSS = 1.165043
Fitting Notch1 (17 remaining): RSS = 2.257409
Fitting Eed (16 remaining): RSS = 1.400365
Fitting Tcf7 (15 remaining): RSS = 2.557572
Fitting Klf5 (14 remaining): RSS = 3.066951
Fitting Gsk3b (13 remaining): RSS = 2.614817
Fitting Dppa2 (12 remaining): RSS = 3.342786
Fitting Dazl (11 remaining): RSS = 3.342786
Fitting Jag1 (10 remaining): RSS = 2.616590
Fitting Lin28 (9 remaining): RSS = 3.342786
Fitting Dppa5a (8 remaining): RSS = 3.342786
Fitting Rest (7 remaining): RSS = 3.601873
Fitting Fgf5 (6 remaining): RSS = 3.342786
Fitting Tbx3 (5 remaining): RSS = 5.417233
Fitting Dppa3 (4 remaining): RSS = 2.657081
Fitting Trim28 (3 remaining): RSS = 1.422620
Fitting Klf2 (2 remaining): RSS = 3.593677
Fitting Dnmt1 (1 remaining): RSS = 0.174330
Fitting Hprt1 (0 remaining): RSS = 2.222912
Total RSS = 133.778416

>> stammPlotFit(data,'results/stammIps4StateFwd.mat','results');
```

5 Joint markers of cell state

The state-specific signatures obtained using our approach give an unbiased, genome-wide way to identify potential marker genes that could define individual states and help in the design of experiments aimed at isolating them. While cellular states have often been characterised by single genes, several genes taken together may represent more robust markers to discriminate distinct cell state. Thus, rather than considering only the top genes in the profile of each state, as listed in Table S2, to enable state identification by joint markers, we ranked pairs of genes by their ability to discriminate the four states (scored by the sum of the differences in gene expressions for each possible combination of two states, as discussed below). Examples are shown in Fig. S11. These joint markers suggest strategies by which to purify cell states using single-cell approaches with appropriate gene-specific reporters or cell-surface antibodies. A full list of gene pairs with state discriminatory scores and state-specific expression levels is provided in Table S3 as a resource for the design of such experiments.

5.1 Discriminatory score for marker gene pairs.

To quantify how well a given pair of genes (i, j) discriminates between states we used a score $d_{i,j}$ that was calculated as the sum of pairwise distances between the gene expression signatures for each state: for n states, $d_{i,j} = \sum_{k=1}^{n} \sum_{l=1}^{n} |\beta_{ki} - \beta_{lj}|$. Note that, since the data are standardized (see Supplementary Information), the score is not dependent on scale. A list of the top discriminatory pairs, with corresponding scores $d_{i,j}$ and state-specific signatures, is given in Table S3.

6 Number of model states

To explore number of model states, we looked at both simple empirical measures, including fit-to-data and distinctness of state signatures, as well as results from a Bayesian model selection procedure.

6.1 Fit-to-data and distinctness of state signatures

The sum of squared error (also known as "residual sum-of-squares" or RSS) for T timepoints and n genes is

$$RSS = \sum_{i=1}^{T} \sum_{j=1}^{n} (y_j(t_i) - \hat{y}_j(t_i))^2, \tag{8}$$

where y_j is the observed (log-transformed) expression level of gene j and $\hat{y}_j$ the corresponding output of the fitted model. Smaller RSS indicates a closer fit between the model and the data to which the model is fitted.

When models with many parameters are fitted to finite data, "over-fitting" may occur, leading to a situation where a low sum-of-squares (i.e. a good fit) does not necessarily mean good predictive power. In the present setting, over-fitting may occur, for example, by introduction of artifactual states. Such states increase the number of model parameters and may improve fit (to the data using which model parameters are estimated) but may not improve explanatory or predictive power. In addition to fit-to-data, we therefore also looked at the distinctness of state signatures. This was done by considering the mutual (linear) dependence of state signatures as quantified by the condition number of the matrix $\mathbf{B}$ formed by taking fitted state signatures β_{ij} as entries. The condition number is a standard quantity in numerical linear algebra and is defined as $C = \max(s_i)/\min(s_i)$, where s_i are the singular values of $\mathbf{B}$. If the expression signatures are non-distinct, the condition number will be larger.

6.2 Statistical model selection

As outlined in the Main Text and Methods therein, we used a Bayesian model selection procedure to explore number of model states. Since models with more states also have more parameters, to objectively choose between models, it is important to account for model complexity [3]. In Bayesian inference, uncertainty with respect to an unknown of interest is described via a probability distribution over the unknown, conditional on the data; this distribution is

known as the posterior probability distribution. In model selection, the posterior probability distribution is over a set of models that are under consideration. The relevant calculations take account of both fit-to-data and model complexity. This is important when choosing between models that differ with respect to model complexity (loosely, number of free parameters), as is the case here. Here, the number of states, n, defines the models of interest; we seek to obtain a posterior distribution over n given the time-course data.

6.2.1 Bayesian formulation

In what follows the dependence on model parameters is important; we therefore write the matrix P of state occupancy probabilities explicitly as a function $P(w)$ of transition rates w. Taking logs in (6) and making the noise model explicit we have $y_j = \log(x_j) = \log(P(w)\beta_j) + \epsilon_j,\ \ \epsilon_j \sim \mathcal{N}(0, \sigma_j^2 I)$, where $\mathcal{N}$ denotes a Normal density, σ_j^2 denotes gene-specific variance and I is the identity matrix.

Let M_n denote the model with n states and $\mathbf{y} = \{y_j\}$ denote observed data for all genes. Taking a flat prior over models $P(M_n) \propto 1$, the posterior probability over models is $P(M_n \mid \mathbf{y}) \propto p(\mathbf{y} \mid M_n)$. The quantity $p(\mathbf{y} \mid M_n)$ is known as the marginal likelihood and gives a score for each model that takes account of both fit-to-data and model complexity. Collecting together all model parameters as $\theta = (\{\beta_j\}, \{\sigma_j^2\}, w)$, the marginal likelihood is

$$p(\mathbf{y} \mid M_n) = \int p(\mathbf{y} \mid \theta, M_n) p(\theta \mid M_n)\, d\theta. \tag{9}$$

We assume prior independence between the transition rates and gene-level parameters (β_j, σ_j^2); this corresponds to the assumption that knowing transition rates does not *a priori* give information about the state signatures or gene-level variances. The parameter priors are $\sigma_j \sim \text{InverseGamma}(a_j, b_j)$, $\beta_{ij} \sim \text{Exponential}(\kappa)$ and $w_{i,i+1} = \text{Gamma}(\alpha_1, \alpha_2)$. To reflect the approximate timescale of reprogramming transitions we set $\alpha_1 = 2, \alpha_2 = 0.15$; this corresponds to a prior mean transition time of ≈ 6.7 days. For a moderate level of shrinkage we set $\kappa = 2$; noise hyperparameters a_j, b_j were set to (i) give prior expected signal to noise ratio (SNR) equal to d, i.e. $\mathbb{E}_{\sigma_j}[S_j/\sigma_j] = d$, where S_j denotes the mean of the (absolute log-transformed) data for gene j, i.e. $S_j = \frac{1}{T}\sum_t |y_j(t)|$ and (ii) assign equal prior mass to SNRs greater and smaller than the prior mean d. Larger d corresponds to an assumption of lower noise and favours more complex models *a priori*; to avoid bias in favour of multiple states we therefore set d conservatively to 20. We investigated sensitivity to hyper-parameters and found that hyper-parameter specification was influential,

as expected in a small sample size setting, but that results across a range of values (κ, d) were consistent with one or two intermediate states (i.e. $n = 3, 4$).

6.2.2 Markov chain Monte Carlo sampling

Markov chain Monte Carlo (MCMC) algorithms are widely used to numerically solve integrals and play a key role in computational Bayesian statistics. Here, we used a MCMC approach to obtain the posterior distribution over number of states. This involves calculating the marginal likelihood, which is obtained by solving the integral in (9). Since the dimension of θ can be large, convergence of standard samplers may be poor [4]. Lack of convergence can then lead to results that are artefacts of algorithm initial conditions and very poor approximations to the true posterior distribution. We therefore used a recent Monte Carlo scheme which combines population MCMC with thermodynamic integration, shown by [5] to produce low variance, low bias estimators. We briefly introduce the method here, as used in the present application, and refer the interested reader to [5, 6] for further technical details. The following material assumes familiarity with basic ideas in Bayesian statistics.

The marginal likelihood is $p(\mathbf{y}|M_n)$. For notational simplicity in the following we leave conditioning on the model M_n implicit throughout, referring to the marginal likelihood as $p(\mathbf{y})$. As above, all model parameters are collected together as $\theta = (\{\beta_j\}, \{\sigma_j^2\}, w)$.

Tempering here refers to the bridging between the prior and posterior distributions, via a parameter γ:

$$p_\gamma(\boldsymbol{\theta}) \propto p(\mathbf{y}|\boldsymbol{\theta})^\gamma p(\boldsymbol{\theta}). \tag{10}$$

Assuming existence of the densities, p_γ defines a path in the space of distributions which connects the prior p_0 to the posterior p_1. Through physical analogy, γ is sometimes referred to as an "inverse temperature" parameter, and controls the tempering schedule. Simulated tempering exploits the observation that

$$\log p(\mathbf{y}) = \int_0^1 \mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{y},\gamma} \log p(\boldsymbol{y}|\boldsymbol{\theta}) d\gamma \tag{11}$$

where the expectation here is with respect to the "power posterior" $p_\gamma(\boldsymbol{\theta})$ [5]. In practice this integral is evaluated using a fine discretisation of $0 = \gamma_1 < \gamma_2 < \cdots < \gamma_N = 1$. Discretisation sacrifices the unbiased property of the extimator, with the precise choice of discretisation known to effect both the bias and the variance of the estimator, as discussed in detail by [5]. For our scheme we took $\gamma_i = (i/N)^c$ (we used $c = 5, N = 30$).

For the statistical model which we consider, it is not possible to obtain the integrand $\mathbb{E}_{\boldsymbol{\theta}|\boldsymbol{y},\gamma} \log p(\boldsymbol{y}|\boldsymbol{\theta})$ in closed form; rather this must also be estimated. A naive strategy would be to construct independent estimators

$$\hat{\mathbb{E}}_{\boldsymbol{\theta}|\boldsymbol{y},\gamma_i} \log p(\boldsymbol{y}|\boldsymbol{\theta}) \approx \frac{1}{M} \sum_{m=1}^{M} \log p(\boldsymbol{y}|\boldsymbol{\theta}_i^{(m)}) \tag{12}$$

for each temperature threshold $1 \leq i \leq N$ using samples $\{\boldsymbol{\theta}_i^{(m)}\}_{m=1}^{\infty}$ obtained from $p_{\gamma_i}(\boldsymbol{\theta})$. However this strategy is likely to fail at low temperatures $i \approx N$ since the power posterior $p_{\gamma_i}(\boldsymbol{\theta})$ can become highly nontrivial (with multiple modes), precluding standard sampling schemes.

The solution which we entertain is known as population MCMC. Here a population of particles $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N\}$ are allowed to evolve (under a "transition kernel") in such a way as to generate sequences of values $\{\boldsymbol{\theta}_1^{(n)}, \ldots, \boldsymbol{\theta}_N^{(n)}\}$ with marginal stationary distributions $p_{\gamma_i}(\boldsymbol{\theta})$. However, in contrast to the independent estimation of Eqn. 12, population MCMC allows for the sharing of information across temperatures γ_i. Such information sharing has been demonstrated empirically to improve convergence of marginal distributions in this setting [6].

The choice of transition kernel influences the rate of convergence to the marginal likelihood. Within our population MCMC scheme, we distinguish between "local" (within temperature) updates and "global" (between temperature) updates. Our local updates were based on a Metropolis-within-Gibbs scheme [7], which may provide improved convergence compared to a naive Metropolis-Hastings approach. In addition to superior mixing, such a strategy is also computationally favourable because the Metropolis acceptance probabilities for σ_j and β_j contain only terms relevant to a single gene j. For the interested reader we briefly summarise the details of our local transition kernel. The parameter set is partitioned as

$$\theta = \{\sigma_1, \beta_1, \ldots, \sigma_n, \beta_n, w\}, \tag{13}$$

and this partition defines an order for the Gibbs sampler. For all proposal distributions (independently of temperature) we took

$$\log(\sigma_j^*)|\sigma_j, \mathbf{y} \sim N(\log(\sigma_j), \lambda_1^2 S_j^2) \tag{14}$$

$$\log(\beta_j^*)|\beta_j \sim N(\log(\beta_j), \lambda_2^2 \mathbf{I}_n) \tag{15}$$

$$\log(w^*)|w \sim N(\log(w), \lambda_3^2 \mathbf{I}_{n-1}), \tag{16}$$

where superscript * denotes proposed parameter value. Symmetry of the proposal distribution is computationally advantageous, since we do not need to evaluate proposal densities in the Metropolis acceptance ratio. The λ's are scale parameters that control the proposal; these were set so as to yield average acceptance rates of approximately 30% over all possible temperatures (specifically, we set $\lambda_1 = \lambda_2 = \lambda_3 = 1$).

For the global update, we followed [5] by selecting adjacent temperatures γ_i, γ_{i+1} and proposing an exchange $\boldsymbol{\theta}_i^{(m+1)} := \boldsymbol{\theta}_{i+1}^{(m)}$ and vice versa $\boldsymbol{\theta}_{i+1}^{(m+1)} := \boldsymbol{\theta}_i^{(m)}$, which is accepted or rejected according to a Metropolis-Hastings step. This information sharing between temperatures enables an unconstrained exploration of the state space at high temperatures $\gamma \approx 0$ to enter into the proposal mechanism for the (more important) low temperature sampler. We observed that this exchange, which is unparametrised, was accepted approximately 90% of the time at high temperatures and approximately 50% of the time at low temperatures.

We performed Bayesian model selection using the k_1 genes chosen by k-means clustering. Our full transition kernel alternated evenly between local and global updates, with 10,000 iterations of both update performed in total (2,000 were discarded as burn-in). Monte Carlo convergence was checked using multiple runs with different random seeds; error estimates shown in the main text represent this between-run variation.

7 Gene ontologies

We ranked genes for each state by the ratio of expression in that particular state over the mean expression across all states. For each state, we passed the names of the genes with a ratio greater than or equal to 2 to the Bingo software to obtain a over-representation p-value [8].

We discarded all genes with $p > 0.01$ (corresponding to a 99% significance level). We then calculated the pairwise sum of differences between the p-values of each state and discarded genes whose sum was below 0.001. This served to remove those terms which were over-represented in all states and therefore not state-specific.

In Fig. S4 we show an list of GO terms where we have further trimmed the list by discarding terms with fewer than 200 and greater than 800 descendants, to avoid overly specific or general terms. The rows of terms were then sorted into clusters by k-means.

8 Applications to other systems

8.1 Gene expression time-course data due to Mikkelsen *et al.*

Our main results were obtained from analysis of gene expression time-course data due to Samavarchi-Tehrani *et al.* [1] (Samavarchi-Tehrani system). We also fitted the models to data from a primary MEF-based reprogramming system due to Mikkelsen *et al.* [2] (referred to as Mikkelsen; see Fig. S8 for results). Fitting to core reprogramming genes (as listed in Table S1, excluding 18 genes not present in the Mikkelsen data) gave qualitatively similar results to those obtained from Samavarchi-Tehrani, suggesting that the hypothesis of a small number of intermediate states may hold also for this system. However, the Mikkelsen dataset has only 5 time points and further data will be needed to fully investigate this notion. Nonetheless, our results suggest that despite the fact that the two reprogramming systems differ in many details, some key aspects of the transitions we discuss may be common to these and possibly other systems. That is, while different systems have different starting points (or initial locations in gene expression space), it is possible that they must pass through similar transitions and intermediate states to reach the iPSC state.

8.2 Single-cell mRNA-seq data due to Tang *et al.*

We reasoned that some aspects of the cell-state transitions described in the Main Text might be general not only to reprogramming systems but to the attainment of pluripotency more generally. To explore the notion of common pathways towards pluripotency we also analyzed single-cell, genome-wide mRNA-seq data pertaining to embryonic stem cell (ESC) establishment [9]. These assays are carried out as the inner cell mass (ICM) is explanted and propagated *in vitro* to derive pluripotent ESCs.

The mRNA-seq data from [9] is an integer count of mRNA fragments. For each pair of genes, genome-wide, the number of states (off/off, on/off, off/on, on/on; we call a gene off if no mRNA fragments of that gene are found, otherwise we call the gene on) occupied by the individual cells is counted to give an indication of the number of discrete states in the population.

Since these data comprise single-cell expression levels they allow us to directly observe whether individual cells are in transcriptionally distinct states. For every gene pair (i, j) across the genome we counted the number of states $n_{i,j}$ observed in the corresponding single-cell ICM-ESC data. For each such gene pair, our reprogramming analysis yields a score $d_{i,j}$ that quantifies the ability of the pair to discriminate model states (see Methods of the Main Text

for details). We therefore sought to ask whether gene pairs predicted to be state-specific in reprogramming also showed evidence of multiple states in the ICM-ESC data. Fig. S9 shows number of states $n_{i,j}$, as observed in single-cell ICM-ESC data, as a function of discriminatory rank under scores $d_{i,j}$ from the Samavarchi-Tehrani reprogramming model. We see significant enrichment: gene pairs predicted to be state-specific during reprogramming are also likely to show multiple states during the ICM-ESC transition ($p < 10^{-6}$ under permutation test). While the biology of establishing ESCs from the ICM is clearly distinct from direct reprogramming, the ESC end state has a number of broad similarities with the iPSC state [10]. These results suggest that some aspects of the transitions in each case may rely on similar mechanisms.

8.3 Single-cell data due to Buganim *et al.*

Single-cell gene expression data were obtained from Buganim et al. [11], data were as described in the reference. Data were clustered using the R package `mclust` with default settings. Individual cells were assigned to reprogramming states on the basis of Euclidean distance between single-cell expression profiles and state signatures.

9 Gene Set Enrichment Analysis

Gene set analysis was carried out using the Gene Set Enrichment Analysis tool (`http://www.broadinstitute.org/gsea/msigdb/`). Key pluripotency-related gene lists that were significantly enriched (hypergeometric p-value $p \leq 1.35 \times e^{-6}$) in the top 200 genes under the S_3 switch ranking included (i) those constituting the PluriNet protein-protein network shared by pluripotent cells (gene set name "MUELLER_PLURINET") and (ii) a set of genes overexpressed in multiple profiling studies of human embryonic stem cells (gene set "BENPORATH_ES_1"). Gene lists (i) and (ii) were highly significant for the top 200 genes under the S_4 switch ranking ($p < e^{-15}$); additionally a gene set comprising a core ESC-like gene module (gene set "WONG_EMBRYONIC_STEM_CELL_CORE") was also highly significant ($p < e^{-15}$). Full details of gene lists are presented in Table S4.

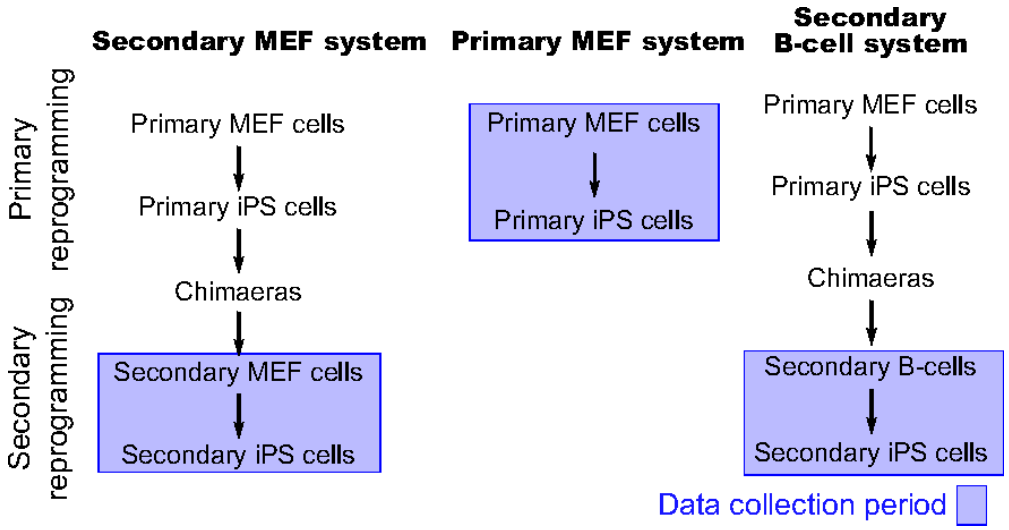

Figure S1: Three different reprogramming systems. Secondary and primary MEF systems are distinguished by the production of chimaera from which the starting colony is derived in the secondary system. B-cells are more homogeneous than MEFs systems, since all B-cells have gone through lineage-commitment. In the Main Text, we focus on time-course data from a secondary MEF system due to Samavarchi-Tehrani et al. [1], as well as data from a primary MEF system due to Mikkelsen et al. [2] for comparison. An example of a reprogramming system based on B-cells is discussed in Hanna et al. [12].

Bmi1	Dppa4	Zeb1	Ccnd1	Cdh1	Gdf3	Col5a2
Lifr	Cldn11	Inadl	Fut4	Nr0b1	Terf1	Rnf2
Cldn7	Suz12	Ocln	Myst4	Crb3	Epcam	Csnk2a1
Dnmt3b	Ctnnbl1	Sall4	Esrp1	Cdc20	Alpl	Cldn3
Cldn4	Nanog	Zfp248	Phc1	Fgf4	Nacc1	Tdgf1
Tgfb2	Prdm1	Esrrb	Cdh2	Nr6a1	Slc2a1	Cdkn2a
Kdm1	Mad2l1	Ezh2	Tcf3	Utf1	Ctcf	Prmt7
Zfp42	Ncam1	Lefty1	Nodal	Gata4	Snai2	Ccnf
Bub1	Nes	Tgfb1	Dnmt3l	Tcl1	Zeb2	Cdkn2b
Thy1	Stat3	Pecam1	Snail	Notch1	Eed	Tcf7
Klf5	Gsk3b	Dppa2	Dazl	Jag1	Lin28	Dppa5a
Rest	Fgf5	Tbx3	Dppa3	Trim28	Klf2	Dnmt1
Hprt1						

Table S1: The subset of core reprogramming genes considered in the Main Text. This includes genes discussed in [1], as well as a number of genes that have been discussed in the reprogramming literature.

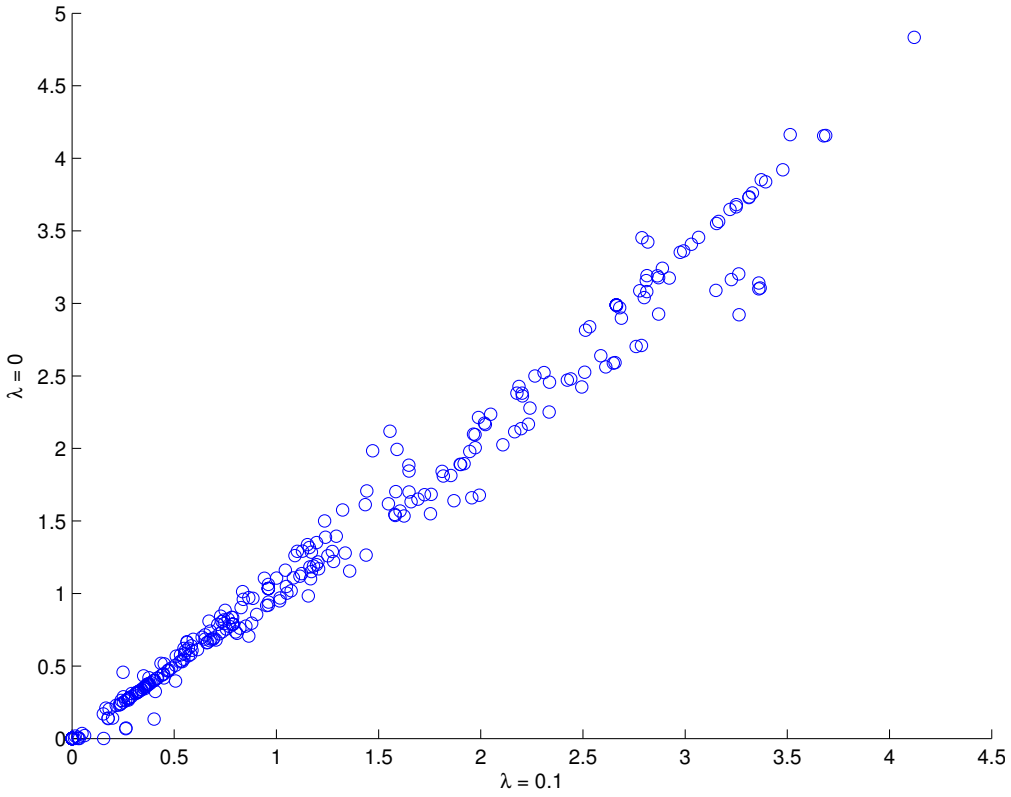

Figure S2: Entries of β matrix from fit with penalty $\lambda = 0$ plotted against those from fit with $\lambda = 0.1$. Estimates are in close agreement, indicating the penalty used in the Main Text ($\lambda = 0.1$) was not unduly influential.

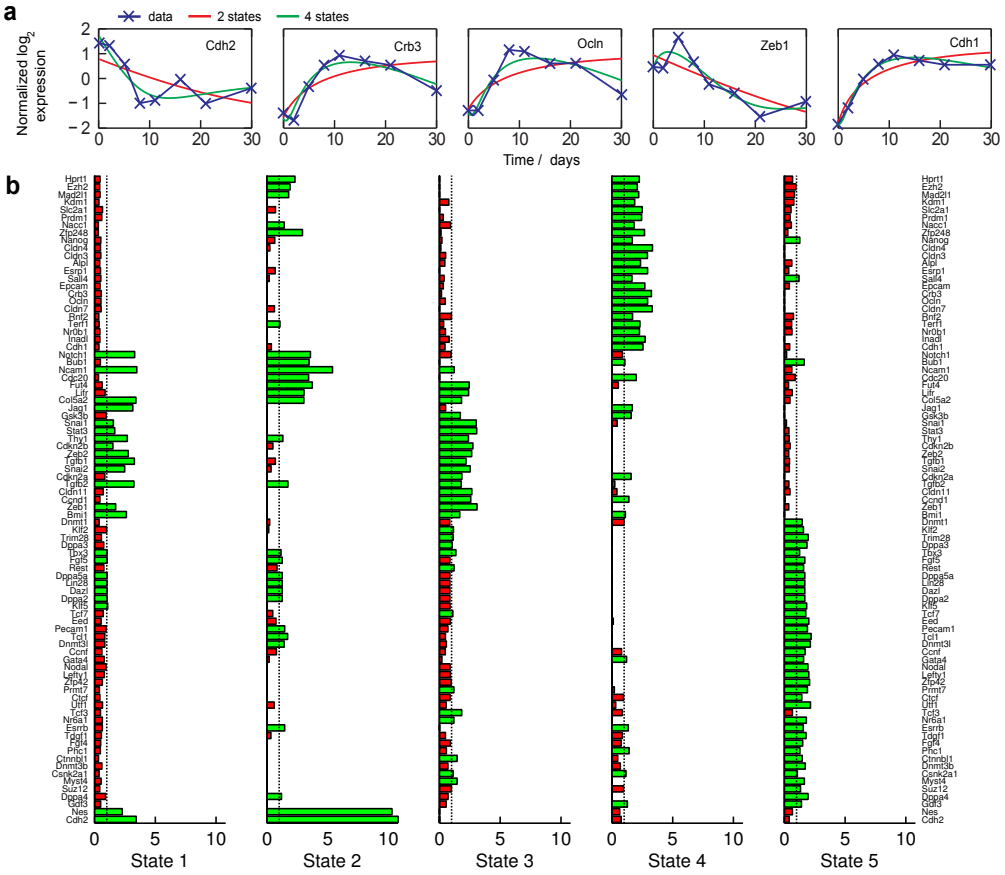

Figure S3: Comparison of fits of the stochastic model with different numbers of states to the Samavarchi-Tehrani et al. [1] dataset that we focus on in the Main Text. a, Comparison of fits between four-state model (green) and two-state model (red) to data (blue with crosses). The two-state model is not capable of following transient changes in expression and consequently is not able to fit the data well. b, Gene expression signatures for the five-state model. States 1 and 2 have overlapped up-regulated expression for a large number of genes. This indicates that the introduction of a 5th state is due simply to duplication of states. Nanog, an indicator of pluripotency, is highly expressed in state 4, but not in state 5. This suggests overfitting where a pluripotent final state is split into a state with pluripotentcy markers and another state with lower expression.

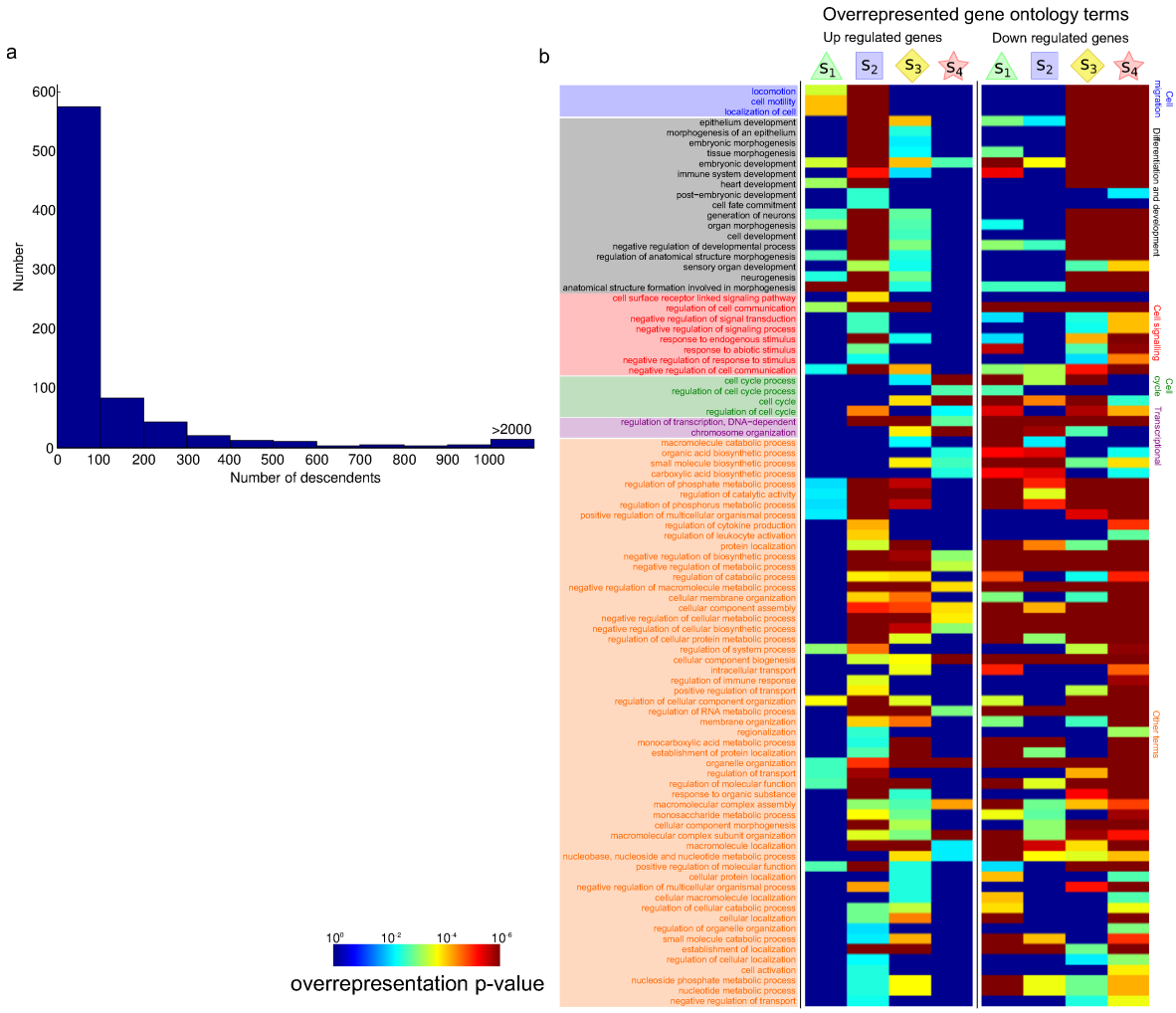

Figure S4: State-specific over-representation of unsupervised selection of gene ontology terms. Row show different terms and columns indicates the over-representation within each state. Color temperature indicates significance of over-representation of term (see p-value color bar). Terms are those remaining after discarding terms with p-value$>$ 0.01, terms with pairwise sum of p-value differences across states less than 0.001 and terms with fewer than 200 or greater than 800 descendants. Remaining rows were then ordered into clusters by k-means.

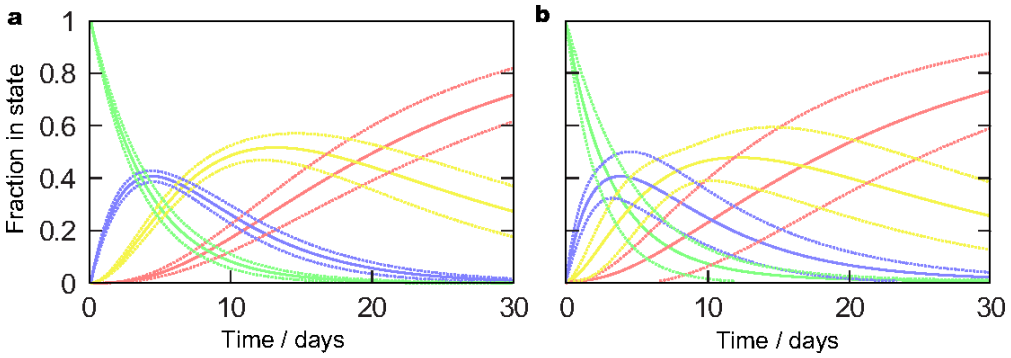

Figure S5: Robustness to data perturbation. Results obtained from data perturbed by a, additive Gaussian noise and b, deletion of an entire time point (mean over ten (a) and eight (b) iterations shown as solid line, ± standard deviation indicated by dotted lines; dataset from Samavarchi-Tehrani et al. [1]; see SI Text for details).

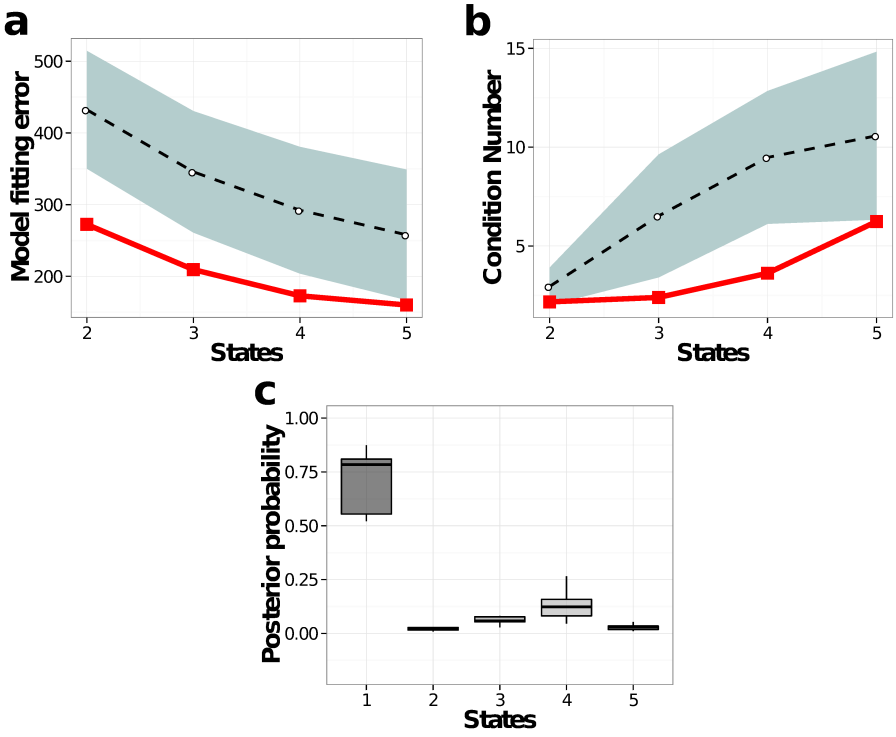

Figure S6: Random permutation of time points. a, model fitting error, b, reciprocal condition number and c, Bayesian posterior probability as a function of number of states. We generated a set of randomized data by random reordering of time indices and the model was re-fit for each of the permuted data. Curves (a,b) and box plots (c) shown are over ten samples; in (a,b) dotted lines indicate means and the shaded area standard deviations, whilst the corresponding result for the correct time ordering is shown in red. Both model fit and distinctness of state signatures are systematically worse under permutation of time indices. Bayesian model selection applied to the randomly permuted data show no evidence of intermediate states (c), in contrast with the original data (Fig. 2c, Main Text). [Dataset from Samavarchi-Tehrani et al. [1]; see SI Text for details.]

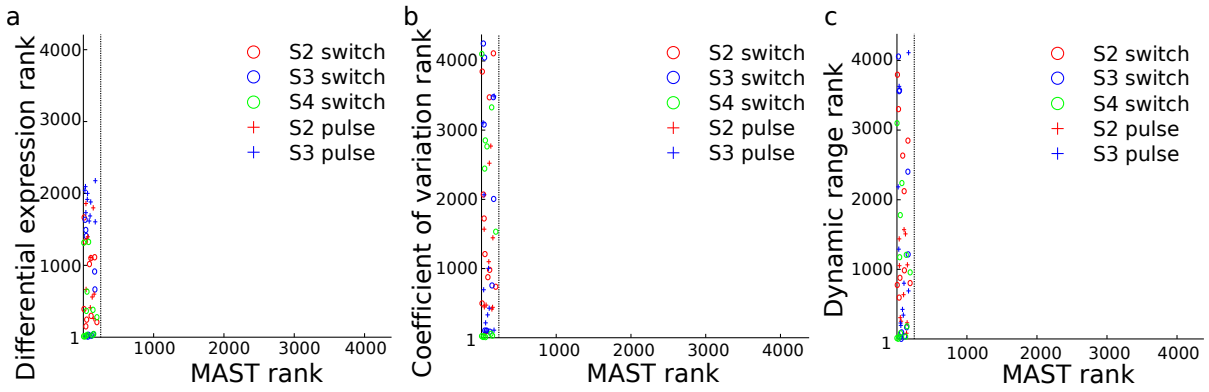

Figure S7: Comparison of STAMM ranking with other gene ranking analyses. For the pluripotency related genes shown in Main Text Fig. 4, ranking under STAMM is compared with (a) differential expression (between starting population and final time point after reprogramming) and two temporal change criteria, (b) coefficient of variation (calculated through time, per gene) and (c) dynamic range (calculated over time, per gene). Under STAMM ranking each gene is among the top 200 genome-wide (shaded in yellow) under the corresponding ranking, i.e., in the top 5% group (see Table S2 for full list and Main Text and Methods for explanation of switch and pulse profiles), whereas their ranks under (a), (b) or (c) span the entire range, down to the bottom 5%. Many of the STAMM genes shown here have been implicated in reprogramming, as discussed in Main Text. (For full description of STAMM, see Main Text and Methods; gene rankings carried out using data from Samavarchi-Tehrani *et al.* [1]; to ensure a fair comparison, genes were filtered in the same way as for STAMM; rank shown for differential expression is the highest of rank by up- and down-regulation.)

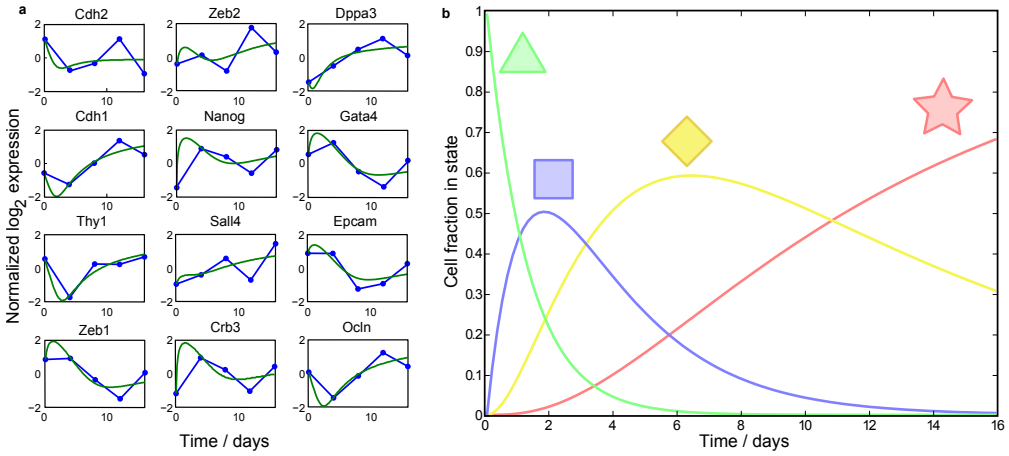

Figure S8: Fit of a four-state model to a time-course dataset from a primary MEF-based system due to Mikkelsen et al. [2]. The data comprise five time-points extending to 16 days. a, Fitting the four-state model discussed in Main Text to core reprogramming genes (as listed in Table S1, excluding 18 genes not represented in the Mikkelsen et al. data; genes shown here are those in Fig. 3d of the Main Text) gives an overall reasonable fit: root sum-of-squared error (per gene per time-point) is 0.025, compared with 0.017 for the fit to data from the secondary MEF-based system due to Samavarchi-Tehrani et al. (note that due to standardization, data standard deviation is equal to unity). The mean transition times were $w_{12} = 1.28$, $w_{23} = 2.63$, $w_{34} = 10.0$ days, close to those estimated from the Samavarchi-Tehrani et al. data (see Fig. 3b of Main Text). b, Estimated state transition dynamics are similar to those found using the secondary MEF data (see Fig. 3c of the Main Text). This suggests that the state transition dynamics in the two systems might share similar features.

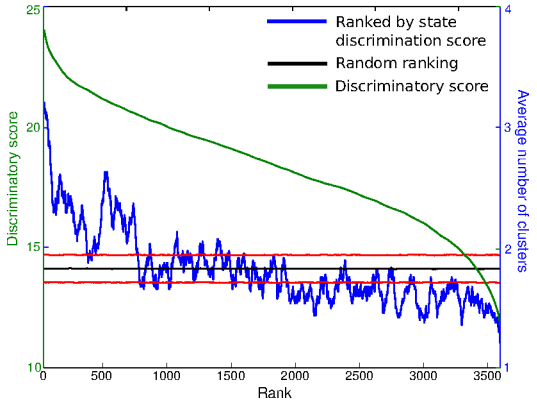

Figure S9: Gene pairs that characterize cell states in the reprogramming model indicate multiple states in single-cell data from an independent embryonic stem cell (ESC) system. For each gene pair (i, j), the number of states $n_{i,j}$ observed in single-cell mRNA-seq data, obtained during the derivation of ESCs from the inner cell mass (ICM; data due to [9]), is shown as a function of rank under scores $d_{i,j}$ from the reprogramming model. For each pair of genes, genome-wide, $n_{i,j}$ was obtained by counting the number of states (off/off, on/off, off/on, on/on) that are occupied by individual cells. Blue line shows average number of states (smoothed with a 50 value moving window) observed in the ICM-ESC data, as a function of rank under the reprogramming model; black line shows the corresponding mean number of states for 10000 random permutations of the ordering of gene pairs, with $\pm$ standard deviation indicated by the red lines. The green line (and left axis) shows scores $d_{i,j}$ (smoothed with a 50 value moving window). (Ranking based on state-specific signatures obtained from fitting four-state model to [1] data; ranking score $d_{i,j}$ equal to sum of pairwise differences in state-specific expression, for each gene across all pairings of states, as described in Methods; enrichment significant under permutation test, $p < 10^{-6}$.

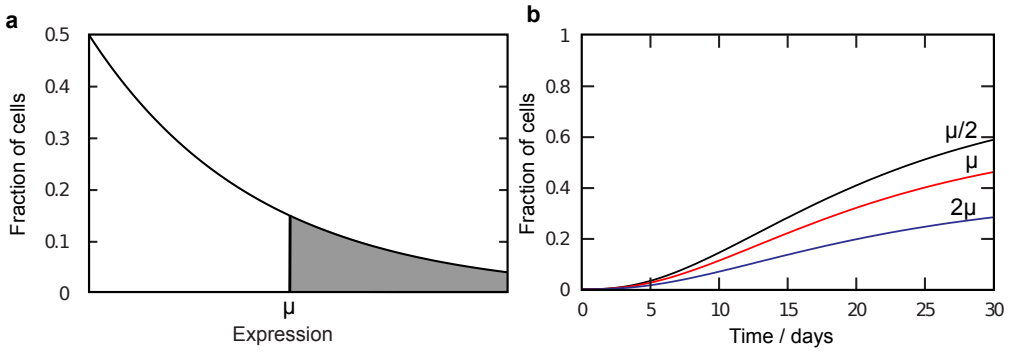

Figure S10: Effect of gene expression variability and detection thresholds on cell fraction measurements. a, An illustrative model for stochastic gene expression. An example exponential distribution with mean μ is shown for single-cell gene expression. The shaded region marks the cells that would be counted as positive for the gene under a single-cell assay (e.g. FACS-based) with detection threshold μ. The substantial fraction of cells in the illustration with expression lower than μ would not be detected as positive for the gene under such an assay. b, Using the illustrative model of stochastic gene expression in (a), and the dynamics of the four-state model discussed in Main Text, the fraction of cells that would test positive for a gene that is expressed only in the final state S_4 is plotted as a function of time. This is illustrated for detection thresholds of $\mu/2$ (black), μ (red), 2μ (blue), where μ is the mean expression signature for Nanog. Under such a scenario, the fraction of cells detected as positive in a single-cell assay at 30 days may be substantially lower than the corresponding probability to be in state S_4.

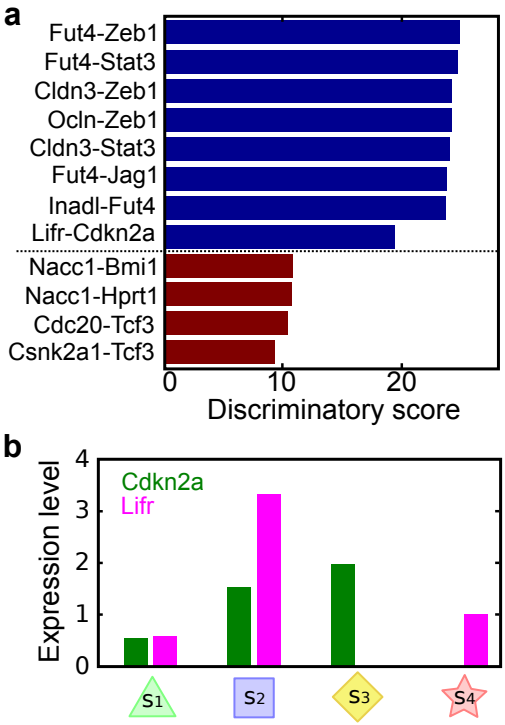

Figure S11: State-specific marker genes. The state-specific signatures obtained using our approach allow unbiased, genome-wide identification of marker genes by which to define individual states. While cellular states have often been characterised by single genes, pairs of genes taken together may represent more robust markers of cell state. a, State-specific transcriptional signatures from the reprogramming model yield scores $d_{i,j}$ that quantify the ability of gene pairs (i, j) to discriminate model states. Example gene pairs, with good discriminators in blue and poor discriminators in red. b, State-specific signatures for one example of a highly-ranked gene pair (Cdkn2a/Lifr); the pair are good discriminators since taken together they show markedly different levels in the four states: S_1 is $\text{Cdkn2a}^{\text{Lo}}/\text{Lifr}^{\text{Lo}}$ (i.e., both Cdkn2a and Lifr are low compared to the average); S_2 is $\text{Cdkn2a}^{\text{Hi}}/\text{Lifr}^{\text{Hi}}$ (both genes are high); S_3 is $\text{Cdkn2a}^{\text{Hi}}/\text{Lifr}^{\text{Lo}}$; and S_4 is $\text{Cdkn2a}^{\text{Lo}}/\text{Lifr}^{\text{Hi}}$. (As a resource for design of single-cell experiments a full list of gene pairs with state discriminatory scores and state-specific expression levels is provided in Table S3; see SI text for definition of discriminatory score $d_{i,j}$.)

References

[1] Samavarchi-Tehrani, P. *et al.* Functional genomics reveals a BMP-driven mesenchymal-to-epithelial transition in the initiation of somatic cell reprogramming. *Cell Stem Cell* **7**, 64–77 (2010).

[2] Mikkelsen, T. S. *et al.* Dissecting direct reprogramming through integrative genomic analysis. *Nature* **454**, 49–55 (2008).

[3] Claeskens, G. & Hjort, N. L. *Model selection and model averaging* (Cambridge University Press, Cambridge, 2008).

[4] Vyshemirsky, V. & Girolami, M. Bayesian ranking of biochemical system models. *Bioinformatics* **24**, 833–839 (2008).

[5] Calderhead, B. & Girolami, M. Estimating Bayes factors via thermodynamic integration and population MCMC. *Comput. Stat. Data Anal.* **53**, 4028–4045 (2009).

[6] Jasra, A., Stephens, D. A. & Holmes, C. C. On population-based simulation for static inference. *Stat. Comp.* **17**, 263–279 (2007).

[7] Roberts, G. O. & Rosenthal, J. S. Harris recurrence of Metropolis-within-Gibbs and trans-dimensional Markov chains. *Ann. Appl. Prob.* **16**, 2123–2139 (2006).

[8] Maere, S., Heymans, K. & Kuiper, M. BINGO: a Cytoscape plugin to assess overrepresentation of Gene Ontology categories in biological networks. *Bioinformatics* **21**, 3448–3449 (2005).

[9] Tang, F. *et al.* Tracing the derivation of embryonic stem cells from the inner cell mass by single-cell RNA-seq analysis. *Cell Stem Cell* **6**, 468–478 (2010).

[10] Bock, C. *et al.* Reference maps of human ES and IPS cell variation enable high-throughput characterization of pluripotent cell lines. *Cell* **144**, 439–452 (2011).

[11] Buganim, Y. *et al.* Single-cell expression analyses during cellular reprogramming reveal an early stochastic and a late hierarchic phase. *Cell* **150**, 1209–1222 (2012).

[12] Hanna, J. *et al.* Direct cell reprogramming is a stochastic process amenable to acceleration. *Nature* **462**, 595–601 (2009).