Larry barcodes model

MB

February 2023

Overview

We have data from an in vitro time-course lineage tracing experiment involving 3-4 time points. At time 0, barcoding is induced in some of the HSCs and their immediate progeny. The barcodes are unique and inheritable. They are neither accumulated nor diluted.

The following type of data is available per time point:

- total number of cells in the dish, and fraction of cells that is recultivated
- single-cell transcriptomics of a known fraction of the cells in the culture
- whether a cell is barcoded or not, and if so, what is the barcode.

We have two datasets on which to test the model:

- Human haematopoiesis, in adult and fetal liver, produced by a Cambridge group. This dataset may probably be published before we are done, but we can use the data to benchmark our model.
- Mouse adult haematopoiesis, completely new and ready to be analysed.

We can also benchmark the model on published datasets, like the Cospar one.

We want to learn:

- what trajectories does each clones undergo
- what are the kinetic differences among the clones.

Strategy:

1. Since there are thousands of clones, we cluster the clones according to their kinetics. More precisely, according to the time-dependent fraction of barcoded cells per time point. This reduces the clones to a few tenths.
2. We then model the total number of cells over time per cluster per cell population. Here we have to introduce priors on what populations can transition into what populations, and on what the differences among clusters shall be. To do so, we rely on similarity matrixes. We then solve a systems of ODEs.

3. To evaluate our model, we minimise a cost function including 3 terms: the model output (number of cells over time) compared to the data, the sparsity constraint, the coherence constraint.

Details

We now expand the 3 points of the strategy above.

Point 1

For the human dataset, this has already been addressed by my student upon using hierarchical clustering algorithm. For the new mouse dataset, I was thinking of using the R clustering function clusterExperiment, which is used to cluster gene trajectories within the Tradeseq R package.

Point 2

The ODEs are straightforward to write, so here the trick is mainly about reducing the parameters. Let's define:

- p (constant): the number of populations in our landscape. This is usually the result of leiden clustering plus manual annotation based on gene markers
- m (constant): the number of clone clusters
- c (constant): the number of cells
- tp (constant): the number of time points
- b_0 (constant): the number of populations that are initially barcoded. In the data we have so far, $b_0 = 3$, because the procedure labels HSCs and 2 other progeny populations.
- $N_o(m, t, i)$: the observed number of cells in cluster i, clone m, time t.
- $N_g(m, t, i)$: the modelled number of cells in cluster i, clone m, time t.
- **Im** (c $\times$ m constant matrix): Im_{ij} is 1 if cell i is barcoded with a barcode belonging to cluster j, 0 otherwise. Note that, for $m = 1$, we use all the cells together, included the non barcoded ones, to create a reference.
- **Ip** (c $\times$ p constant matrix): Ip_{ij} is 1 if cell i belongs to population j, 0 otherwise.
- **Sc** (c $\times$ c constant matrix): similarity matrix based on the euclidean or cosine product metric.
- **Sp**(m) (p $\times$ p constant matrix): similarity matrix among populations, obtained as $Ip^T \times (Sc \odot Ic(:, m)) \times Ip$

- $\mathbf{Sm}(p)$ (m $\times$ m constant matrix): similarity matrix among clones for population p, obtained as $Im^T \times (Sc \odot Ip(:, p))$
- $\boldsymbol{K(m)}$ (p $\times$ p matrix of parameters for clone m): $K_{ij}(m)$ is the differentiation rate from population i to j, if $i \neq j$, or the net proliferation rate if $i = j$. Note that these matrixes contain at the same time the kinetic and the topological information.
- $\boldsymbol{\Delta}(m)$ (i $\times$ j matrix of parameters): $\Delta_{ij}(m)$ is the differences between the $K_{ij}(1)$ and $K_{ij}(m)$ rates. It depends on clone similarity. If $Sm(i)_{m,1}$ is above a certain threshold, then $\Delta_{ii}(m) = 0$. If $Sm(i)_{m,n}$ is above a certain threshold, then $\Delta_{ii}(m) = \Delta_{ii}(n)$. If $Sm(i)_{m,1}$ and $Sm(j)_{m,1}$ are above a certain threshold, then $\Delta_{ij}(m) = 0$, and so on.

So far we have $p^2 \times m$ parameters (we are assuming for the moment that the parameters are constant over time, but this can be changed later). Furthermore, we should add $b_0 \times$ m initial conditions, one per clone per each of the b_0 populations that are initially barcoded.The number of equations is p $\times$ m, and the number of data points is p $\times$ m $\times$ tp. Since tp $<$ p, the number of data is less than the number of parameters, and this means we need more constrains to be able to estimate the parameters.

In order to reduce the number of parameters, we adopt this strategy:

- Topology: in principle each population can transition into each cell type, but this is unlikely. We can set transitions to 0 in a clone specific manner, based on similarities. We thus compute the clone dependent cell population similarity matrix Sp(m), and threshold it (alternative, use PAGA). Note that establishing whether 2 populations are connected doesn't clarify the direction of the transition. This can be fixed using pseudotime, or the 2 differentiation directions can be kept as parameters.
- Across clones: clones that are kinetically similar should have the same prior. For example, the average value of a parameter among all cells belonging to type i is $K_{ij}(1)$. Then $K_{ij}(m) = K_{ij}(1) + \Delta_{ij}(m)$.

  The ODE system then reads:

$$\dot{N}_{\mathrm{g}}(i, m, t) = \Sigma_j(K_{j,i}(m)N_{\mathrm{g}}(j, m, t)) + (\Sigma_j K_{i,j}(m))N_{\mathrm{g}}(i, m, t)$$

Point 3

For the cost function, we use a similar idea to the one shown in the Cospar supplement:

$$C = \alpha\|\boldsymbol{\Delta}\| + \beta\|\mathbf{L}\ \boldsymbol{K}\|_2 + \Sigma_{p,m,t}\|(N_{\mathrm{o}}(i, t, m) - N_{\mathrm{g}}(i, t, m))\|$$

Potentially, we can estimate an error on $N_{\mathrm{g}}(i, t, m)$ and add the respective log likelihood term.