ABSTRACT

We describe an approach for simultaneous cell segmentation and cell tracking of bacteria held in narrow growth-lines (GL). Our model includes real-world knowledge like (i) a pixel intensity measure along the GL, (ii) the migration speed of cells, (iii) the continuity of size and growth rate, and (iv) volume conservation and symmetry of cell divisions. We find the MAP assignment by solving a minimization problem in the log-likelihood domain using an integer linear program solver.

1. Data and Data Processing

The data for which we describe this approach contains a time-series of dividing bacteria in narrow growth-lines (GL) that have been imprinted in a layer of PDMS. Each micrograph contains several such GLs, extending away from a central channel used to wash away cells that spilt out of GLs. (See Figure 1a.)

The raw data is preprocessed by (i) horizontal alignment of the central channel, (ii) background subtraction, and (vi) cutting the dataset into small sub-stacks, each containing all time points of a single GL. This leads to multiple datasets similar to the one shown in Figure 1b.

The blue plot in Figure 1c shows the image intensity values along the center of the GL (blue highlight in Fig. 1b). The data used for creating segmentation hypotheses is shown in green in Fig. 1c. It is deduced by computing the average intensity of the brightest line segment cutting through the GL along the GL center. The box-plot on top of Fig. 1c shows the extracted set of segmentation hypotheses. Each such hypothesis is a node in a component tree [1], computed on the data given by the green plot described above.

2. The Model

The model can be described by a large chain of factor graphs [2-3], similar to work described in [4]. Nodes within one time point represent segmentation hypotheses which are, via assignment factor nodes, connected to other segmentation hypothesis nodes in neighboring time points. In Fig. 2a we show two sets of segmentation hypothesis for time points $t$ and $t+1$. Between them we show three example assignments. An exit-assignment is shown in red, a mapping-assignment in blue, and a division-assignment in orange. All possible such assignments exist. For Fig. 2b, we show how segmentation constraints can be deduced from a hypothesis-tree. (Note: since cells cannot overlap with each other, at max one of the nodes along a root-leaf path in the tree can fit the data best.)

Next to segmentation constraints we have also to introduce ‘continuity constraints’ that ensure that each valid segmentation hypothesis is further explained in future time points. (See Figure 3, Equation (2).)

3. The Cost Functions

For each segmentation hypothesis $h$ and each assignment variable $a$ we compute a ‘price’ for them to be activated. These costs are proportional to the negative log-likelihood of the probability that the scenario described by the assignment best describes the given data. Fig. 4a shows the costs associated with segmentation variables, Fig. 4b (4c) visualizes the costs for a mapping (division) assignment. Exit assignments are ‘for free’, but they are constraint as described in Fig. 3, Eq. (3).

4. Computing the Most Probable Solution

We compute the exact maximum a-posteriori (MAP) by solving the integer linear program (ILP) shown in Fig. 3. For this we use a free academic license of Gurobi™. The MAP assignment is the model state with highest joint probability.

To boost performance we do usually remove assignments with overly high costs. A typical dataset containing 100 time points usually leads to an ILP containing between 3k and 4k variables and 15k and 20k constraints. To find the MAP in such an ILP usually takes about 5 seconds on a MacBook Pro (2.7 GHz Intel Core i7).

5. Dealing With User Input

The proposed model offers a relatively straightforward way to include a users knowledge for finding superior segmentation and tracking solutions. We use the ideas described below to allow a user to ‘proof-read’ the results found automatically. If a weak spot is found, the user can change the current solution by browsing all assignments and picking the one that best describes the data. This process takes only a couple of seconds. Internally this functionality is implemented by simply adding a constraint that requests the chosen assignment variable $a$ to have a value exactly equal to 1 (active).

The modified ILP is again solved using Gurobi™ and the newly found solution is the MAP assignment given data and user input.

6. Bibliography


