

Figure 3 intMEMOIR simulations and analysis

Folder structure:

Main scripts to reproduce the figures:

/Fig3/Fig3DandH/

Reconstruction code location:

/ReconstructionSoftware/

Data for estimation of parameters:

/Fig3/reconstructionParameters/

Description:

The ReconstructionSoftware folder contains all the relevant functions to

1. Reconstruct lineage trees from intMEMOIR array data
2. Perform *in silico* recording using intMEMOIR
3. Compute upper and lower bounds for the performance of intMEMOIR recording.

The reconstructionParameters folder contains the FISH readout files from which the array states are extracted. Particularly, `allBarcodes.csv` is the main input file for Figure 2, and contains the array states for all cells across all colonies. This file is used to compute edit rates and state frequencies.

Figure 3D

Accuracy on clonal classification using intMEMOIR

This script loads the list of experiments from `groundTruthNewick.csv` and creates a data frame that works as the main data structure throughout the script.

The function `process.all.files()` will read the array states for all colonies directly from the reconstructionParameters folder. Make sure that this folder is accessible and the location is correctly specified in the variable `file.path`

`memoirData = reconstruct.all.lineages(dataTree,...)` is the main line that calls the reconstruction function on all the trees specified in the data frame `dataTree`

From the `memoirData` data frame we can now call several functions:

1. The reconstruction of MEMO mode,
2. Simulations of ground truth (for upper bound on performance),
3. Negative control (randomized labels).
4. Accuracy of clonal classification

For clonal accuracy we run the function `clonal.score` from `MEMO_accuracy.R`

Figure 3H

Since we start by reconstructing all lineages, the first part of this script contains the same steps as in 3D.

From the `memoirData` data frame, however, we perform the `reconstructMembow()` method which collapses cells with identical barcodes into single leaves. The output contains the collapsed (MEMO) trees along with the accuracy score.

Upper bound

We start from the `memoirData` data frame, and use the `runAllMetrics()` function which will take the ground truth lineages (in MEMO mode if specified) and perform *in silico* recording using experimental parameters.

Negative control

From the `memoirData` data frame, we can use the function `randomControl()` to compute a lower bound on performance by randomizing the labels on the trees.

Supplementary Figures

Fig S5 Entropy calculation

Starts at line 119 Fig3H.R

From the `memoirData` data frame we simulate the all the lineages 10 times and compute the entropy for each simulation using the function `barcodeEntropy()`

We then compute normalized entropy using the edit rate.

Fig S6 Additional intMEMOIR arrays increase reconstruction accuracy and depth.

Starts at line 172 Fig3H.R

From the `memoirData` data frame, we can call the `runAllMetrics()` function specifying the number of array states that we want to use for the simulation. There is no limit on this parameter, and so this function can be used to explore how many recording units are necessary to achieve a given level of accuracy in reconstruction.