

Cell division

Model implementation. Cell division is modeled by the generalized `divide_cell` function and the *E. coli*-specific `ChromosomeReplication` process and `CellDivision` listener in the model. A Helmstetter-Cooper type model of chromosome replication initiation is coupled to cell division, inspired by work from Wallden *et al.* [1]. Chromosome replication initiation occurs at a fixed mass per origin of replication. Each initiation event is coupled to a cell division event after a constant period of time consisting of one round of chromosome replication (C period) and cytokinesis (D period). When a round of chromosome replication is completed, the `ChromosomeReplication` process adds the length of the D period to the current time and pushes this time to a queue. The `CellDivision` listener checks this queue and the current time at every timestep, and triggers cell division when the current time passes the earliest time in this queue.

Cell division itself done by the `divide_cell` function and is modeled as a binomial process where each daughter cell has an equal probability of inheriting the contents of the mother cell. The exception to this is if two chromosomes are present before cell division—each daughter is guaranteed to get one. Because the *E. coli* model is not yet gene complete, certain mechanistic details of cell division (eg. cytokinesis, septation, and chromosome segregation) are not yet modeled explicitly.

Due to the coupled nature of chromosome replication and this division model as mentioned above, Algorithm ?? (previous section) is used to create the doubled ended queue referenced in the cell division implementation provided in Algorithm 1 below.

Algorithm 1: Cell division

Input : d_{queue} a double ended queue data structure that stores the time(s) cell division should be triggered

Input : c_i counts of all molecules in simulation at cell division where $i = 1$ to $n_{species}$

Input : p binomial partition coefficient

Input : n_{chrom} number of chromosome molecules

Input : **rand()** returns a random number from a uniform distribution between 0 and 1

Input : **randint()** returns a random integer either 0 or 1

if $t > d_{queue}.peek()$ **then**

1. Trigger division and remove division time.
 $d_{queue}.pop()$
2. Divide bulk contents of cell binomially. Number partitioned into daughter one is stored in $n_{daughter,1}$ and to daughter two in $n_{daughter,2}$.
for $i = 1$ **to** $n_{species}$ **do**
 $n_{daughter,1} = 0$
 for $j = 1$ **to** c_i **do**
 if **rand()** $> p$ **then**
 $n_{daughter,1} = n_{daughter,1} + 1$
 $n_{daughter,2} = c_i - n_{daughter,1}$
3. Divide chromosome in binary manner. All replication forks and origins of replication associated with a chromosome molecule are partitioned as well. Number of chromosome molecules partitioned into daughter one is stored in $n_{chrom,daughter,1}$ and to daughter two in $n_{chrom,daughter,2}$.
if $\text{mod}(n_{chrom}, 2)$ **then**
 $n_{chrom,daughter,1} = \frac{n_{chrom}}{2}$
else
 $n_{chrom,daughter,1} = \text{floor}(\frac{n_{chrom}}{2}) + \text{randint}()$
 $n_{chrom,daughter,2} = n_{chrom} - n_{chrom,daughter,1}$

Result: Cell division is triggered at C+D time after DNA replication initiation. Contents of mother cell is divided between two daughter cells conserving mass.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	chromosome_replication.py	process
wcEcoli/wholecell/sim	divide_cell.py	function
wcEcoli/models/ecoli/listeners	cell_division.py	listener

Table 1: Table of files for transcription regulation.

References

- [1] Mats Wallden, David Fange, Ebba Gregorsson Lundius, Özden Baltekin, and Johan Elf. The synchronization of replication and division cycles in individual e. coli cells. *Cell*, 166(3):729–739, 2016.