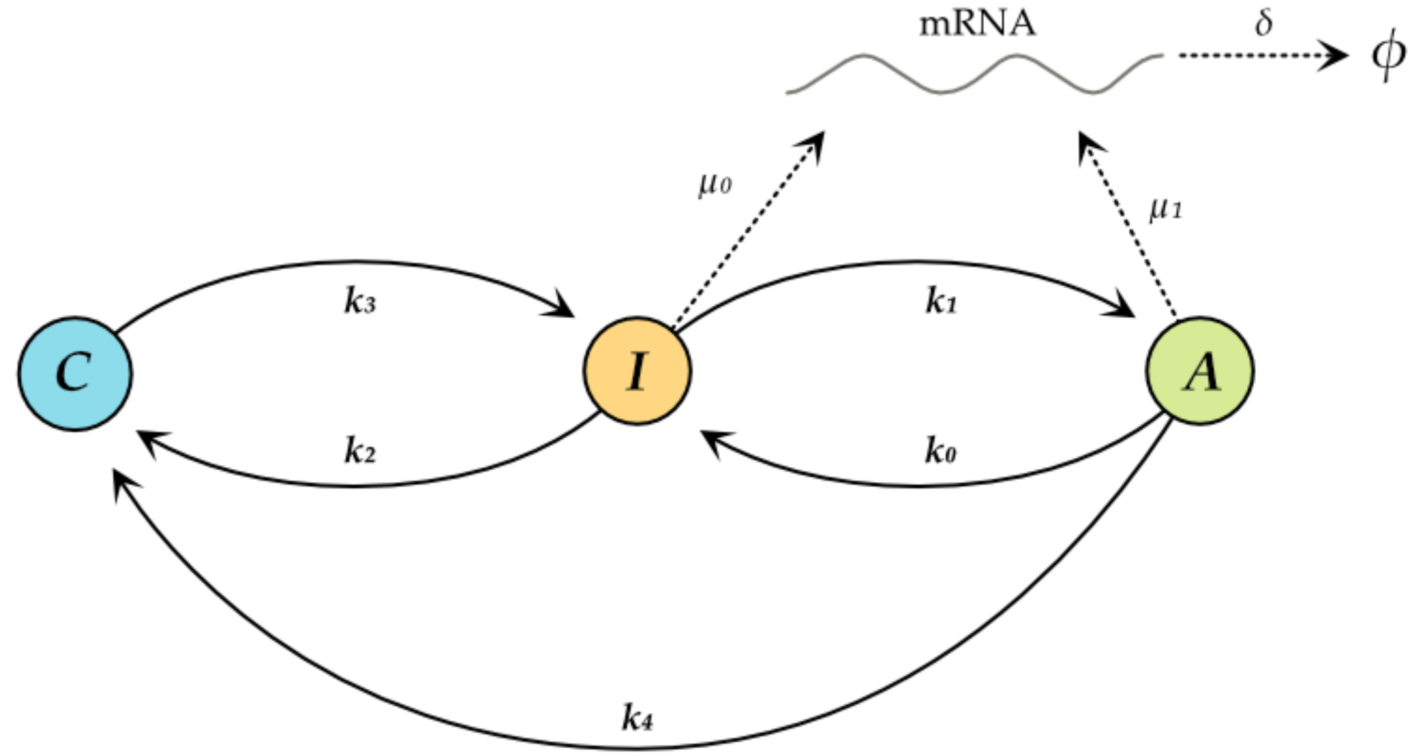
Three State Model

The functions in this file implement a three-state model for gene transcription that is summarised by the figure below



*The three-state model. Each locus can be in one of three states: Closed, Inactive or Active. Transcripts are produced from inactive loci at rate μ_0 per locus, from active loci at rate μ_1 and degrade at a rate δ per transcript.**

Computing probabilities

The main function of interest here is

```
matExpDistribs3( maxRNA, nLoci, mu, k, delta, times, eps )
```

whose arguments are:

`maxRNA`
: the maximal number of transcripts that we expect to see at the longest times. If this is too small the estimates of the distribution will be poor.

`nLoci`

the number of potential sites of transcription. Typically it's two, but other values are possible. The number states for a cell is `(nLoci + 2)*(nLoci + 1)/2`.

`mu`

a vector containing the rates of transcription in the form [μ_0, μ_1]. Both are measured in transcripts per minute per locus.

`k`

a vector containing the rates of locus-state transitions in the form [k_0, k_1, k_2, k_3, k_4]. All are measured in transitions per minute per locus.

`times`

a vector of times, which should be
* positive,
* measured in minutes,
* arranged in ascending order,

at which we want the distribution.

`eps`

an optional threshold for small probabilities whose default value, `eps = 0.000001` is set in the file `matExpDistribs3.m`, Probabilities less than `eps` are set to zero.

Format of the output

The function `matExpDistribs3()` computes a probability for each of the

```
nModelStates = (maxRNA + 1)*(nLoci + 2)*(nLoci+1)/2
```

possible states of the model. A state, which describes what is happening in a single cell, is specified by by four non-negative whole numbers `(nc, ni, na, m)` where

`nc` is the number of loci in the closed state;

`ni` is the number of loci in the inactive state;

`na` is the number of loci in the active state;

`m` is the number of transcripts.

Thus a cell with `nLoci` possible sites of transcription should always have `nc + ni + na = nLoci.`

The output of `matExpDistribs3()` is a matrix with one row for each state and one column for each time in the input argument `times`. The way the states get mapped to rows of this matrix is determined by the function `stateToIdx3( ni, na, nLoci, m)` and is illustrated in the table below

The mapping between states `(nc, ni, na, m)` and row numbers in the output of matExpDistribs3().

nc	ni	na	m	rowNum
2	0	0	0	1
1	1	0	0	2
0	2	0	0	3
1	0	1	0	4
0	1	1	0	5
0	0	2	0	6
2	0	0	1	7
1	1	0	1	8
0	2	0	1	9
1	0	1	1	10
0	1	1	1	11
0	0	2	1	12

Probability distribution over number of transcripts

All the states that involve the same number of transcripts are stored in consecutive rows of the output of `matExpDistribs3()`. This makes it easy to implement a convenience function, `marginalise()`, that computes a matrix of probabilities with one row for each of the `(maxRNA +1)` possible values of the number of transcripts and one column for each of the times. The `m, j` entry in this matrix is the probability of seeing `m-1` transcripts at time t_j. A typical application is

```
stateProbMat = matExpDistribs3( maxRNA, nLoci, mu, k, delta, times )
transcriptCountProbMat = marginalise( stateProbMat, nLoci )
```