Efficient adaptive uniformisation for the analysis of biochemical reaction networks

Casper Beentjes

beentjes@maths.ox.ac.uk http://people.maths.ox.ac.uk/beentjes/

Wolfson Centre for Mathematical Biology, University of Oxford

12 July 2018

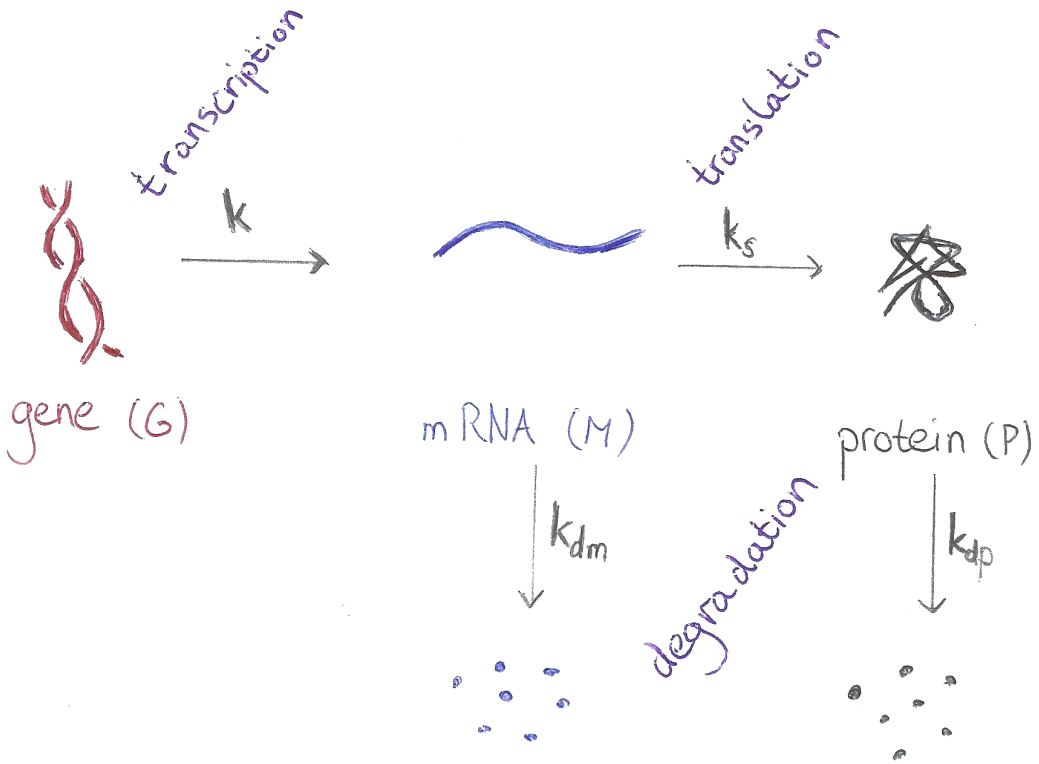
Joint work with Ruth Baker

Outline

Outline

Motivation for stochastic models of biochemical reactions

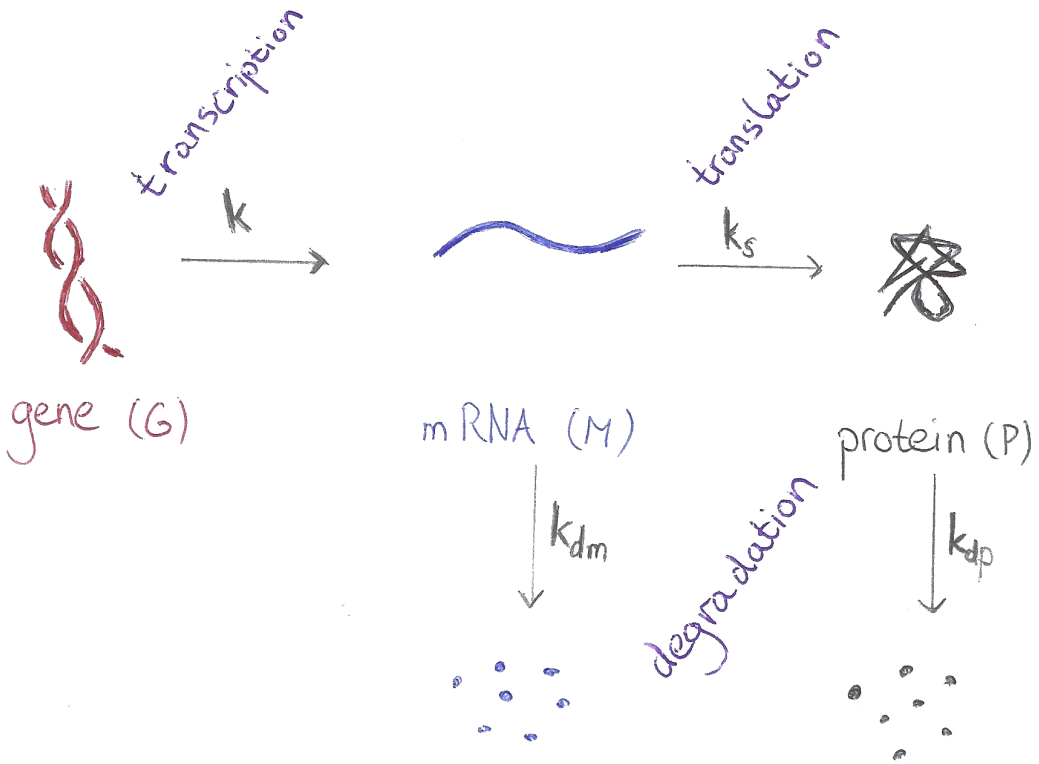
Example for today; a standard gene-expression model



Traditionally modelled with ODE systems

Motivation for stochastic models of biochemical reactions

Example for today; a standard gene-expression model

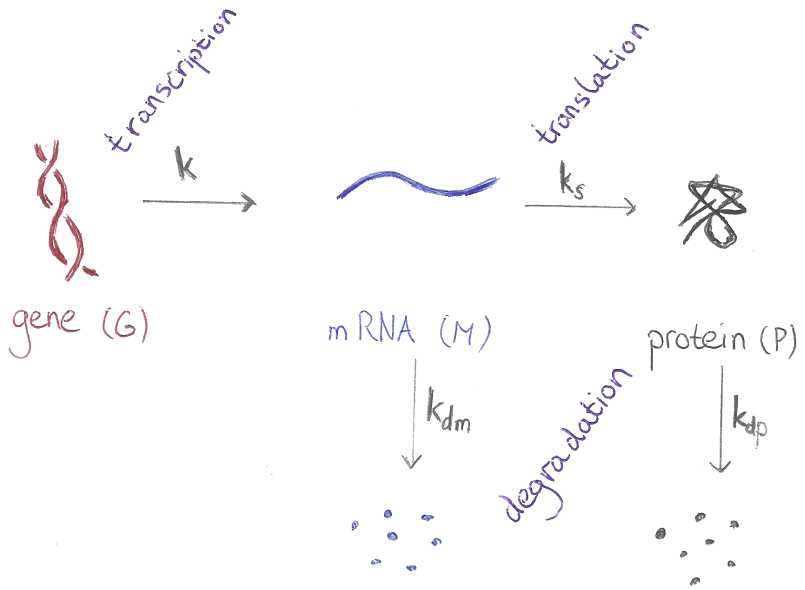


Traditionally modelled with ODE systems, but inherently is a (discrete) stochastic process due to

- **low copy numbers** of involved molecules,
- reactions can be rare and take place at **random times**.

Motivation for stochastic models of biochemical reactions

For the remainder of the talk



$\Downarrow$ notation

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset,$$

How can we simulate this?

Simulation with Gillespie SSA

Algorithm Gillespie's Direct Method

Input: Initial data for G, M and P

Input: Final time T

1: $t \leftarrow 0$
2: **while** $t < T$ **do**
3: Generate τ, the time until the next reaction.
4: Choose which of the reactions has to fire.
5: Update G,M and P according to the firing reaction.
6: $t \leftarrow t + \tau$

Simulation with Gillespie SSA

Algorithm Gillespie's Direct Method

Input: Initial data for G, M and P
Input: Final time T

1: $t \leftarrow 0$
2: **while** $t < T$ **do**
3: Generate τ, the time until the next reaction.
4: Choose which of the reactions has to fire.
5: Update G,M and P according to the firing reaction.
6: $t \leftarrow t + \tau$

3: Next reaction times τ are exponentially distributed with parameter equal to the total propensity of a reaction happening.

Simulation with Gillespie SSA

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset.$$

To sample the next reaction time τ;

1. define the **total propensity**,
   e.g. $a_0(t) = kG(t) + k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t)$,
2. solve for τ in

$$\underbrace{\text{Exp}(1)}_{\text{Standard exponential random variable with mean 1}} = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u.$$

Simulation with Gillespie SSA

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset.$$

To sample the next reaction time τ;

1. define the **total propensity**,
   e.g. $a_0(t) = kG(t) + k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t)$,
2. solve for τ in

$$\underbrace{\text{Exp}(1)}_{\text{Standard exponential random variable with mean 1}} = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u.$$

Simulation with Gillespie SSA

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset.$$

To sample the next reaction time τ;

1. define the **total propensity**,
   e.g. $a_0(t) = kG(t) + k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t)$,
2. solve for τ in

$$\underbrace{\text{Exp}(1)}_{\text{Standard exponential random variable with mean 1}} = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u.$$

Luckily, between reactions G, M and P are constant and thus

$$\text{Exp}(1) = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u = a_0(t)\tau.$$

Easy to solve for τ, happy days. 👍

Outline

Uniformisation example

Let's alter the system marginally

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u} \emptyset.$$

Behaviour of G, M and P is unchanged, but why would we do this?

Uniformisation example

Let's alter the system marginally

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Behaviour of G,M and P is unchanged, this seems even worse, stop it!

Uniformisation example

Let's alter the system marginally

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Behaviour of G,M and P is unchanged, this seems even worse, stop it!

However, there is good news, we are **free to choose** $k_u(t)$ to be whatever we want!

Uniformisation example

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Take $k_u(t) = \bar{a} - a_0(t)$, might seem quite complicated at first...

Uniformisation example

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Take $k_u(t) = \bar{a} - a_0(t)$, might seem quite complicated at first...

Good news; the total propensity is now equal to $\bar{a}$ & **independent of** t!

Uniformisation example

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Take $k_u(t) = \bar{a} - a_0(t)$, might seem quite complicated at first...

Good news; the total propensity is now equal to $\bar{a}$ & **independent of** t!

Thus we get the next reaction time via

$$\text{Exp}(1) = \bar{a}\tau,$$

happy days 👍.

Uniformisation example

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Take $k_u(t) = \bar{a} - a_0(t)$, might seem quite complicated at first...

Good news; the total propensity is now equal to $\bar{a}$ & **independent of** t!

All reactions times follow from

$$\text{Exp}(1) = \bar{a}\tau,$$

even better 👍👍.

Efficient simulation of uniformised reaction networks

Take our uniformised gene-expression system

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset,$$

and suppose we are interested in our system at some final time T.

Efficient simulation of uniformised reaction networks

Take our uniformised gene-expression system

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset,$$

and suppose we are interested in our system at some final time T.

If we can find a uniformisation rate $\bar{a}$ for $0 \leq t \leq T$, then we note that

$$\mathbb{P}(K \text{ reactions fire in } [0, T]) = \frac{(\bar{a}T)^K}{K!} e^{-\bar{a}T},$$

i.e. the **number of reactions is Poisson distributed** with rate $\bar{a}T$.

Efficient simulation of uniformised reaction networks

Algorithm Gillespie's uniformised method

Input: Initial data for G, M and P
Input: Final time T
Input: Uniformisation rate $\bar{a}$

1: $K \leftarrow$ Poisson random number with rate $\bar{a}T$
2: **for** $k = 1, \ldots, K$ **do**
3: Choose which of the reactions has to fire.
4: Update the G, M and P according to the firing reaction.

No need to generate the random reaction times.
Faster than Gillespie's SSA for the uniformised system + standard 'tricks' for Gillespie's SSA can be carried over.

Efficient simulation of uniformised reaction networks

Algorithm Gillespie's uniformised method

Input: Initial data for G, M and P
Input: Final time T
Input: Uniformisation rate $\bar{a}$

1: $K \leftarrow$ Poisson random number with rate $\bar{a}T$
2: **for** $k = 1, \ldots, K$ **do**
3: Choose which of the reactions has to fire.
4: Update the G, M and P according to the firing reaction.

No need to generate the random reaction times.
Faster than Gillespie's SSA for the uniformised system + standard 'tricks' for Gillespie's SSA can be carried over.

Potential issues:

1. Have to fire non-reactions, $\emptyset \xrightarrow{k_u(t)} \emptyset$, a waste of computational effort.
2. What happens if $a_0(t) > \bar{a}$?

Dealing with empty-reactions

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Let a_0 again be the propensity for a non-empty reaction firing.

Which reaction will we fire next?

$$\mathbb{P}\left(\text{a non-empty reaction fires first}\right) = \frac{a_0}{\bar{a}} = \alpha,$$

and

$$\mathbb{P}\left(\emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires first } \right) = 1 - \alpha.$$

Dealing with empty-reactions

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Let a_0 again be the propensity for a non-empty reaction firing.

Which reaction will we fire next?

$$\mathbb{P}\left(\text{a non-empty reaction fires first}\right) = \frac{a_0}{\bar{a}} = \alpha,$$

and

$$\mathbb{P}\left(\emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires first } \right) = 1 - \alpha.$$

But firing $\emptyset \xrightarrow{k_u(t)} \emptyset$ does not change the propensity a_0 and probability α of the non-empty reactions, so we can start again.

Dealing with empty-reactions

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Let a_0 again be the propensity for a non-empty reaction firing.

Which reaction will we fire next?

$$\mathbb{P}\left(\text{a non-empty reaction fires first}\right) = \frac{a_0}{\bar{a}} = \alpha,$$

and

$$\mathbb{P}\left(\text{a non-empty reaction fires second, after } \emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires}\right) = (1-\alpha)\cdot\alpha,$$

and

$$\mathbb{P}\left(\emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires twice}\right) = (1-\alpha)^2.$$

Dealing with empty-reactions

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Let a_0 again be the propensity for a non-empty reaction firing.

Which reaction will we fire next?

$$\mathbb{P}\left(\text{a non-empty reaction fires third, after } \emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires twice}\right) = (1-\alpha)^2 \cdot \alpha,$$

and

$$\mathbb{P}\left(\emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires three times}\right) = (1-\alpha)^3.$$

Dealing with empty-reactions

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Let a_0 again be the propensity for a non-empty reaction firing.

Which reaction will we fire next?

In general we have that

$$\mathbb{P}\left(\text{a non-empty reaction fires, after } \emptyset \xrightarrow{k_u(t)} \emptyset \text{ fires } m \text{ times}\right) = (1-\alpha)^m \cdot \alpha,$$

i.e. the number of empty-reactions firing consecutively follows a **geometric distribution** with parameter $\alpha = a_0/\bar{a}$.

Easy to sample! 👍

Efficient uniformisation simulation

Algorithm Gillespie's uniformised method (improved)

Input: Initial data for G,M and P
Input: Final time T
Input: Uniformisation rate $\bar{a}$

1: $K \leftarrow$ Poisson random number with rate $\bar{a}T$
2: $k \leftarrow 0$
3: **while** $k < K$ **do**
4: Choose which of the non-empty reactions has to fire.
5: Sample the number of empty-reactions, k_{empty}, firing
6: $k \leftarrow k + k_{\text{empty}}$ ▷ If $k \geq K$ after update break.
7: Update the G, M and P according to the firing reaction.
8: $k \leftarrow k + 1$

Only fire actual reactions, so can be made **at least as fast as Gillespie's SSA for the original system** and independent of $\bar{a}$.
Standard 'tricks' for Gillespie's SSA can be carried over.

Adapting the uniformisation rate

In theory with the previous approach we can take our uniformisation rate $\bar{a}$ as large as we want.

However, if for some reason $a_0 > \bar{a}$ occurs, can we use that sample path, without introducing a bias?

Adapting the uniformisation rate

In theory with the previous approach we can take our uniformisation rate $\bar{a}$ as large as we want.

However, if for some reason $a_0 > \bar{a}$ occurs, can we use that sample path, without introducing a bias?

Yes, because we can sample the time t^* at which $a_0(t^*) = \bar{a}$. Then we can restart the simulation (Markov property) from t^* with a new uniformisation rate.

Adapting the uniformisation rate

Suppose we sample K reactions in $[0, T]$, but after K^* reactions we see that $a_0 \geq \bar{a}$.

This corresponds to a time $t^* \in [0, T]$ which follows

$$\frac{t^*}{T} \sim \text{Beta}\left(K^*, K - K^* + 1\right).$$

Again, easy to sample.

Adapting the uniformisation rate

Suppose we sample K reactions in $[0, T]$, but after K^* reactions we see that $a_0 \geq \bar{a}$.

This corresponds to a time $t^* \in [0, T]$ which follows

$$\frac{t^*}{T} \sim \text{Beta}\left(K^*, K - K^* + 1\right).$$

Again, easy to sample.

For the remaining time, $[t^*, T]$, pick a $\bar{a}_{\text{new}}$, and generate the remaining number of reactions like before.

Adapting the uniformisation rate

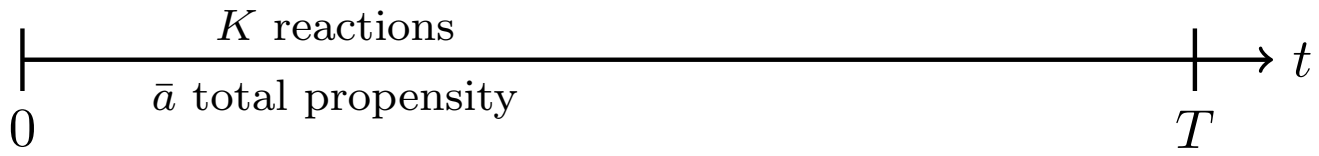
Suppose we sample K reactions in $[0, T]$, but after K^* reactions we see that $a_0 \geq \bar{a}$.

This corresponds to a time $t^* \in [0, T]$ which follows

$$\frac{t^*}{T} \sim \text{Beta}\left(K^*, K - K^* + 1\right).$$

Again, easy to sample.

For the remaining time, $[t^*, T]$, pick a $\bar{a}_{\text{new}}$, and generate the remaining number of reactions like before.



Adapting the uniformisation rate

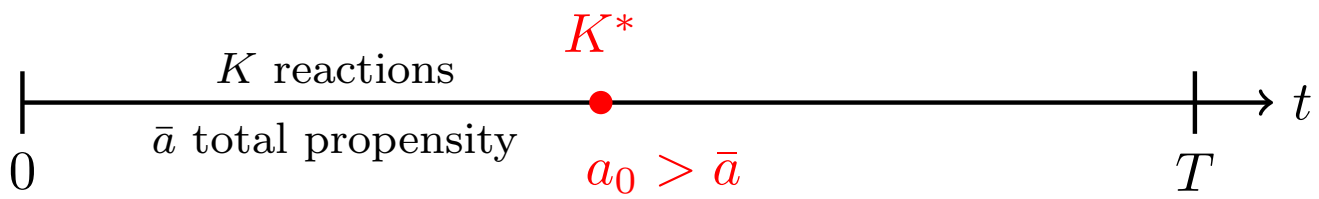
Suppose we sample K reactions in $[0, T]$, but after K^* reactions we see that $a_0 \geq \bar{a}$.

This corresponds to a time $t^* \in [0, T]$ which follows

$$\frac{t^*}{T} \sim \text{Beta}\left(K^*, K - K^* + 1\right).$$

Again, easy to sample.

For the remaining time, $[t^*, T]$, pick a $\bar{a}_{\text{new}}$, and generate the remaining number of reactions like before.



Adapting the uniformisation rate

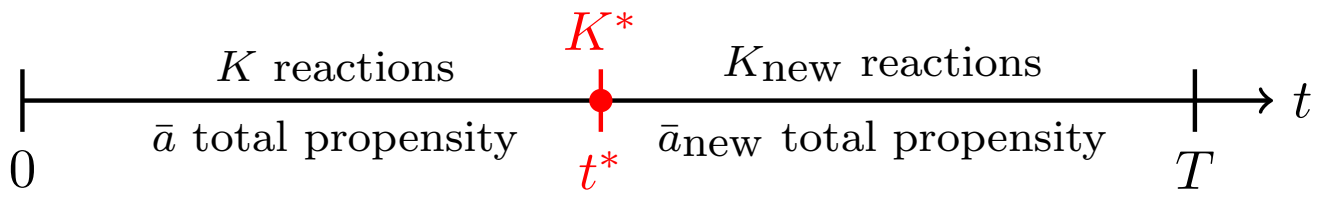
Suppose we sample K reactions in $[0, T]$, but after K^* reactions we see that $a_0 \geq \bar{a}$.

This corresponds to a time $t^* \in [0, T]$ which follows

$$\frac{t^*}{T} \sim \text{Beta}\left(K^*, K - K^* + 1\right).$$

Again, easy to sample.

For the remaining time, $[t^*, T]$, pick a $\bar{a}_{\text{new}}$, and generate the remaining number of reactions like before.



Outline

Stratification for biochemical reaction networks

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Suppose we are interested in the protein level P at some final time T.

Stratification for biochemical reaction networks

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Suppose we are interested in the protein level P at some final time T.

Uniformise with rate $\bar{a}$ for $0 \leq t \leq T$ means the number of reactions is Poisson distributed with rate $\bar{a}T$.

Stratification for biochemical reaction networks

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Suppose we are interested in the protein level P at some final time T.

Uniformise with rate $\bar{a}$ for $0 \leq t \leq T$ means the number of reactions is Poisson distributed with rate $\bar{a}T$.

So we can **stratify with respect to the number of reactions** that have happened in $[0, T]$.

Stratification for biochemical reaction networks

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Suppose we are interested in the protein level P at some final time T.

Uniformise with rate $\bar{a}$ for $0 \leq t \leq T$ means the number of reactions is Poisson distributed with rate $\bar{a}T$.

So we can **stratify with respect to the number of reactions** that have happened in $[0, T]$.

$$\mathbb{E}[P(T)] = \sum_{K=0}^{\infty} \mathbb{P}(K \text{ reactions fire in } [0, T]) \mathbb{E}[P | K \text{ reactions fire in } [0, T]]$$

Stratification for biochemical reaction networks

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

Suppose we are interested in the protein level P at some final time T.

Uniformise with rate $\bar{a}$ for $0 \leq t \leq T$ means the number of reactions is Poisson distributed with rate $\bar{a}T$.

So we can **stratify with respect to the number of reactions** that have happened in $[0, T]$.

Note that we can choose to truncate the infinite sum

$$\mathbb{E}[P(T)] = \sum_{K=K_l}^{K_u} \mathbb{P}(K \text{ reactions fire in } [0,T])\mathbb{E}[P|K \text{ reactions fire in } [0,T]]$$

Stratification for biochemical reaction networks

Define the variance reduction factor

$$\beta = \frac{\mathrm{Var}_{\text{stratified}}\left[P(T)\right]}{\mathrm{Var}_{\text{SSA}}\left[P(T)\right]}$$

Example 1;

$$P \xrightarrow{0.1} \emptyset, \qquad \emptyset \xrightarrow{1} P.$$

start with $P(0) = 10$.

T	β
1	0.93
10	0.94
100	0.97

Stratification for biochemical reaction networks

Define the variance reduction factor

$$\beta = \frac{\mathrm{Var}_{\text{stratified}}\left[P(T)\right]}{\mathrm{Var}_{\text{SSA}}\left[P(T)\right]}$$

Example 2;

$$G \xrightarrow{50} G + M, \qquad M \xrightarrow{2} M + P, \qquad M \xrightarrow{0.5} \emptyset, \qquad P \xrightarrow{2} \emptyset$$

start with $G(0) = 1$, $M(0) = 0$ and $P(0) = 0$.

T	β
1	0.98
10	0.99

Marginal gains.

Transient information

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

From uniformisation with rate $\bar{a}$ we can compute and store

$$\hat{\mu}_K \approx \mathbb{E}\left[P \middle| K \text{ reactions fired}\right]$$

Then in theory for any time t we know that

$$\mathbb{E}\left[P(t)\right] \approx \hat{\mu}(t) = \sum_{K=0}^{\infty} \frac{(\bar{a}t)^K}{K!} e^{-(\bar{a}t)} \hat{\mu}_K.$$

Transient information

$$G \xrightarrow{k} G+M, \qquad M \xrightarrow{k_s} M+P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset, \qquad \emptyset \xrightarrow{k_u(t)} \emptyset.$$

From uniformisation with rate $\bar{a}$ we can compute and store

$$\hat{\mu}_K \approx \mathbb{E}\left[P \middle| K \text{ reactions fired}\right]$$

Then in theory for any time t we know that

$$\mathbb{E}\left[P(t)\right] \approx \hat{\mu}(t) = \sum_{K=0}^{\infty} \frac{(\bar{a}t)^K}{K!} e^{-(\bar{a}t)} \hat{\mu}_K.$$

Note; the calculation of this estimator does **not require any new simulations**, we can just use our simulation results $\hat{\mu}_k$ and appropriately (re-)weight them.

Summary

Uniformisation ...

- is an intuitive technique which does not require much change in your existing knowledge/simulations.
- can be made **at least as fast as Gillespie's SSA**.
- can be used to create a **variance reduction** method by stratifying with respect to the number of fired reactions.
- can be used to get **transient information** over a whole time interval $[0, T)$ at no extra simulation cost (just post-processing).

Summary

Uniformisation ...

- is an intuitive technique which does not require much change in your existing knowledge/simulations.
- can be made **at least as fast as Gillespie's SSA**.
- can be used to create a **variance reduction** method by stratifying with respect to the number of fired reactions.
- can be used to get **transient information** over a whole time interval $[0, T)$ at no extra simulation cost (just post-processing).

Summary

Uniformisation ...

- is an intuitive technique which does not require much change in your existing knowledge/simulations.
- can be made **at least as fast as Gillespie's SSA**.
- can be used to create a **variance reduction** method by stratifying with respect to the number of fired reactions.
- can be used to get **transient information** over a whole time interval $[0, T)$ at no extra simulation cost (just post-processing).

Summary

Uniformisation ...

- is an intuitive technique which does not require much change in your existing knowledge/simulations.
- can be made **at least as fast as Gillespie's SSA**.
- can be used to create a **variance reduction** method by stratifying with respect to the number of fired reactions.
- can be used to get **transient information** over a whole time interval $[0, T)$ at no extra simulation cost (just post-processing).

Summary

Uniformisation ...

- is an intuitive technique which does not require much change in your existing knowledge/simulations.
- can be made **at least as fast as Gillespie's SSA**.
- can be used to create a **variance reduction** method by stratifying with respect to the number of fired reactions.
- can be used to get **transient information** over a whole time interval $[0, T)$ at no extra simulation cost (just post-processing).

Thank you for your attention.

Variance reduction via stratification

Standard Monte Carlo estimator:
Generate $x_i \sim p$ samples for $i = 1, \dots, N$ and calculate

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} f(x_i)$$

Stratified Monte Carlo estimator:
Generate $x_{i,j} \sim p_j$ for $j = 1, \dots J$ and calculate

$$\hat{\mu}_{\text{strat}} = \sum_{j=1}^{J} \frac{\omega_j}{n_j} \sum_{i=1}^{n_j} f(x_{i,j})$$

where $n_1 + \cdots + n_J = N$ so we have the same amount of samples.

Variance reduction via stratification

What's the use of the stratification?

Standard Monte Carlo estimator:

$$\text{Var}(\hat{\mu}) = \frac{\sigma^2}{N}$$

Stratified Monte Carlo estimator:

$$\text{Var}(\hat{\mu}_{\text{strat}}) = \sum_{j=1}^{J} \frac{\omega_j^2 \sigma_j^2}{n_j}$$

where σ_j is the variance *within* the j-th stratum. Again we take $n_1 + \cdots + n_J = N$ so we have the same amount of samples.

Note that

$$\sigma^2 = \sum_{j=1}^{J} \omega_j \sigma_j^2 + \sum_{j=1}^{J} \omega_j (\mu_j - \mu)^2$$

Variance reduction via stratification

What's the use of the stratification?

Suppose we take $n_j = N\omega_j$, i.e. proportional allocation of our samples.

Standard Monte Carlo estimator:

$$\text{Var}(\hat{\mu}) = \frac{\sigma^2}{N}$$

Stratified Monte Carlo estimator:

$$\text{Var}(\hat{\mu}_{\text{strat}}) = \frac{1}{N}\sum_{j=1}^{J}\omega_j\sigma_j^2 \leq \frac{1}{N}\sum_{j=1}^{J}\omega_j\sigma_j^2 + \frac{1}{N}\sum_{j=1}^{J}\omega_j(\mu_j-\mu)^2 = \frac{\sigma^2}{N} = \text{Var}(\hat{\mu})$$

So we have a reduced variance estimator!

Simulation with Gillespie SSA

For our standard model gene-transcription model in a volume V

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset,$$

the total propensity $a_0(t) = kG(t) + k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t)$.

Suppose from step 3 we know that a reaction takes place at $t^* = t + \tau$. Which reaction?

For example consider the first reaction:
$A_1 = \{G \xrightarrow{k} G + M \text{ fires in } [t^*, t^* + \mathrm{d}t)\}$,
$B = \{a \text{ reaction fires in } [t^*, t^* + \mathrm{d}t)\}$

$$\text{Prob}(A_1|B) = \frac{\text{Prob}(B|A_1)\text{Prob}(A_1)}{\text{Prob}(B)} = \frac{1 \cdot kG(t^*)\,\mathrm{d}t}{a_0(t^*)\,\mathrm{d}t} = \frac{kG(t^*)}{a_0(t^*)}$$

Similar relative propensities evaluated at t^* for the other propensities. Easy to sample, happy days. 👍

Extrinsic noise example

Let's return to our standard gene-expression model

$$G \xrightarrow{k} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset,$$

for the transcription of mRNA (M) from genes (G), the translation of mRNA into protein (P) and the degradation of both the mRNA and protein.

Extrinsic noise example

Let's return to our standard gene-expression model

$$G \xrightarrow{k(t)} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset,$$

for the transcription of mRNA (M) from genes (G), the translation of mRNA into protein (P) and the degradation of both the mRNA and protein.

But maybe now the transcription of mRNA follows a 24 hour day-night cycle so the rate of transcription is not constant in time.

For example $k(t) = c(1 + \sin(2\pi f t))$, where $f^{-1} = 24h$.

How can we simulate this?

Simulation with Gillespie SSA

Algorithm Gillespie's Direct Method

Input: Initial data for G, M and P

Input: Final time T

1: $t \leftarrow 0$
2: **while** $t < T$ **do**
3: Generate τ, the time until the next reaction.
4: Choose which of the reactions has to fire.
5: Update G,M and P according to the firing reaction.
6: $t \leftarrow t + \tau$

3: Next reaction times τ are exponentially distributed with parameter equal to the total propensity of a reaction happening.

Simulation with Gillespie SSA

For our standard gene-expression model

$$G \xrightarrow{k(t)} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset,$$

the total propensity $a_0(t) = k(t)G(t) + k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t)$.

To find the next reaction time we solve

$$\underbrace{\text{Exp}(1)}_{\substack{\text{Standard exponential random variable with mean 1}}} = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u.$$

Simulation with Gillespie SSA

For our standard gene-expression model

$$G \xrightarrow{k(t)} G + M, \qquad M \xrightarrow{k_s} M + P, \qquad M \xrightarrow{k_{\text{dm}}} \emptyset, \qquad P \xrightarrow{k_{\text{dp}}} \emptyset,$$

the total propensity $a_0(t) = k(t)G(t) + k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t)$.

To find the next reaction time we solve

$$\underbrace{\text{Exp}(1)}_{\text{Standard exponential random variable with mean 1}} = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u.$$

Again, between reactions G, M and P are constant, however

$$\text{Exp}(1) = \int_t^{t+\tau} a_0(u)\,\mathrm{d}u = (k_s M(t) + k_{\text{dm}} M(t) + k_{\text{dp}} P(t))\tau + \int_t^{t+\tau} k(u)\,\mathrm{d}u G(t).$$

Not so easy to solve for τ generally... 👎