CSC3431: Coursework 2

Harry Hughes

Introduction

In this report, I detail the process of developing a simulator and running simulations for various chemical reaction networks implementing the approximate majority algorithm.

A solution to the approximate majority algorithm requires a mixture of reactants to reach majority (consensus), where the mixture is wholly one chemical (for example $n_x = n_{tot}$ would be x as the consensus). A 'correct' sample would reach consensus on the reactant with the larger initial volume (if initially $n_x > n_y$ then it should reach $n_x = n_{tot}$).

The system of simulation is based on Petri Nets, these represent a changing system with 'places' and 'transitions'. The places are locations to store items and the transitions are processes that can modify the contents of the places.

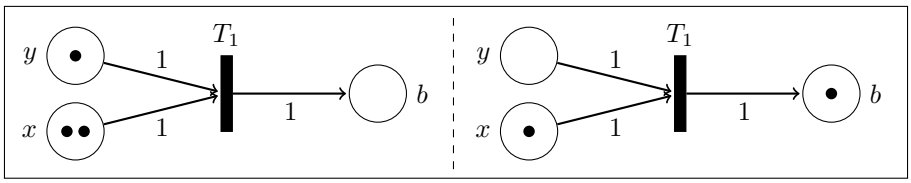


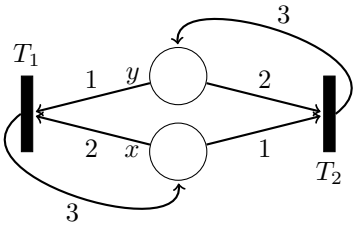
Figure 1: Example Petri Net before and after firing

For example in Figure~1, the reactants (contained in places) are x and y while b is the product, with T_1 as the transition between them. When fired T_1 takes 1 of each input (reactants, in this case x and y) and produces 1 of its output (product, in this case b).

These can be used to represent far more complex systems than this, for example one of the chemical reaction networks specified for solving the AM algorithm (Tri-Molecular) is shown in Figure~2.

$$X + X + Y \xrightarrow{1} X + X + X$$
$$X + Y + Y \xrightarrow{1} Y + Y + Y$$

(a) Equations



(b) Petri Net

Figure 2: Tri-Molecular CRN

In the case of Figure~2 the Petri Net representation includes everything but the **kinetic rate** k. This is the rate per second on average that the transaction will take place, for the Tri-Molecular CRN shown both transitions are of rate $k = 1$, this means they will occur with approximately equal likelihood at any instant. However, in an unbalanced mix the likelihood must be adjusted by the proportion of reactants available. This is done with the following calculations (see Condon et al. 2020, pg. 254) for propensity with x reactants of amounts $n_1 \cdots n_x$ needing quantities $r_1 \cdots r_x$ to fire:

$$a = \frac{c}{2}\binom{n_1}{r_1}\binom{n_2}{r_2}\cdots\binom{n_x}{r_x} = \frac{c}{2}\prod_{i=1}^{x}\binom{n_i}{r_i}$$

This is the number of ways it could choose the reactants from the mixture multiplied by the **stochastic rate** c divided by 2. The stochastic rate is defined as:

$$o = \sum_{j=1}^{x} r_j,\ c = \frac{2k}{n_{tot}^{o-1}}$$

Therefore this can be simplified to:

$$a = \frac{k}{n_{tot}^{o-1}} \prod_{i=1}^{x} \binom{n_i}{r_i}$$

The system of choosing the next reaction uses a random variation to distribute reactions on a Poisson Distribution ($U(a,b)$ indicates a uniform distribution between a and b):

$$r = U(0,1)$$
$$t_{next} = t_{now} - \frac{\ln r}{a}$$

This report is focused on assessing the findings of Condon et al. 2020, and particularly those found in Figure~3. These show that, with respect to a logarithmic volume scale (n), the end_t/final time should have a linear gradient according to their experiment. As noted by the original document, these samples were run with an initial gap of $|n_y - n_x| = \sqrt{n_{tot} \ln n_{tot}}$ thus meaning a correct sample would produce $n_x = n_{tot}$.

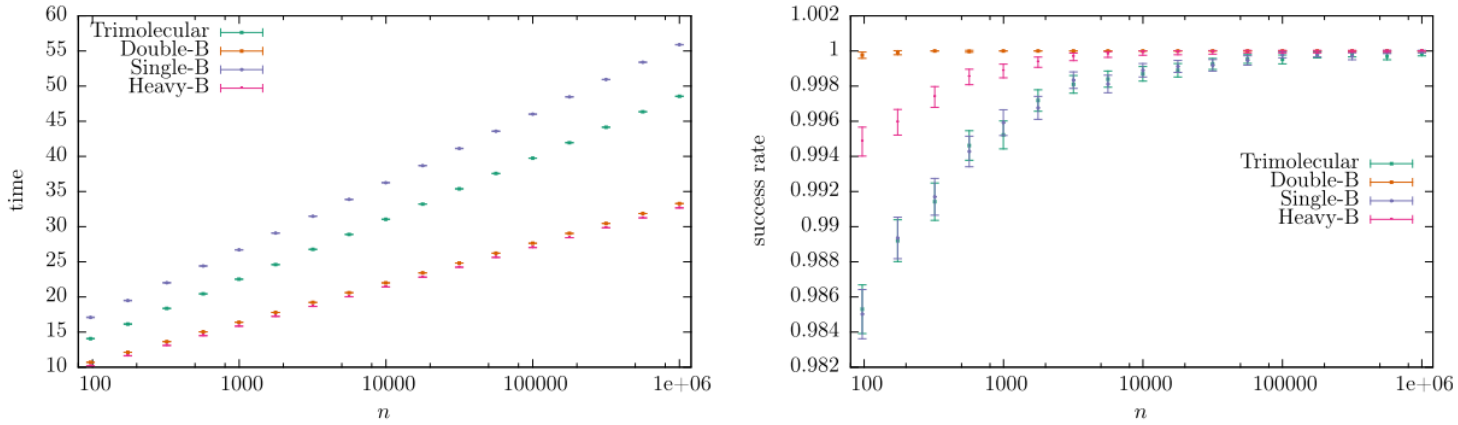


Figure 3: Condon et al. 2020 final time and correctness graphs

Additionally, Condon et al. 2020 posits that:

$$\text{firings} = \Omega(\gamma n \ln n)$$

Meaning that the number of times a reaction fires in a sample should be bounded by:

$$\gamma_1 n \ln n \leq \text{firings} \leq \gamma_2 n \ln n$$

for some constants γ_1, γ_2. This can be tested by graphing $n_{tot} \ln n_{tot}$ against the number of firings taken to achieve consensus. This should be a linear relationship.

Methodology

To run these simulations, I have written a Java library and CLI executable. The simpler solution would have been to run them in Python, but for the size of simulation being executed, efficiency was important.

This Java library (PetriNetCRN) uses the calculations above with specification JSON file. More information can be found in the README markdown file with the source code. This system uses multiple threads to improve performance however it should only use a maximum of 3 so at least a quad core processor is recommended. Additionally, it is very high in memory cost so beware running out of memory.

To compare with the results of Condon et al. 2020 I ran each of the Petri Nets over a logarithmic range from 100 - 10000, repeating each sample 500 times. Which gave me relatively clear results however more repeats could be beneficial. Unfortunately the sheer size of data produced for this number of repeats would have required a redesign of my graphing system.

The simulation system implements the Gillespie method, this uses Monte Carlo sampling. For more information see Gillespie 1977.

Like Condon et al. 2020 I have used the initial gap of $|n_y - n_x| = \sqrt{n_{tot} \ln n_{tot}}$ to make our results comparable.

Below are the CRNs being tested:

Tri-Molecular

$$\begin{aligned} X + X + Y &\xrightarrow{1} X + X + X \\ X + Y + Y &\xrightarrow{1} Y + Y + Y \end{aligned}$$

Double-B

$$\begin{aligned} X + Y &\xrightarrow{1} B + B \\ X + B &\xrightarrow{1} X + X \\ Y + B &\xrightarrow{1} Y + Y \end{aligned}$$

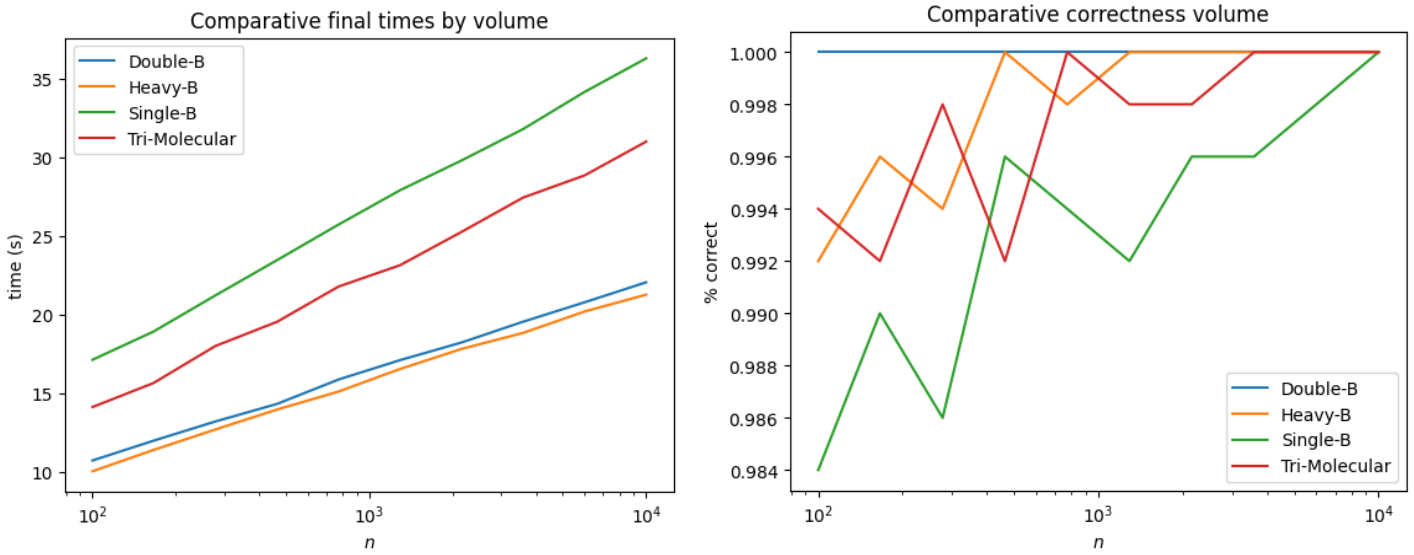
Single-B

$$\begin{aligned} X + Y &\xrightarrow{\frac{1}{2}} X + B \\ X + Y &\xrightarrow{\frac{1}{2}} Y + B \\ X + B &\xrightarrow{1} X + X \\ Y + B &\xrightarrow{1} Y + Y \end{aligned}$$

Heavy-B

$$\begin{aligned} X + Y &\xrightarrow{1} B_2 \\ X + B_2 &\xrightarrow{1} X + X + X \\ Y + B_2 &\xrightarrow{1} Y + Y + Y \end{aligned}$$

Results



(a) Final Times

(b) Success Rate

Figure 4: Equivalent graphs to Figure~3 for my data

Figure~4a shows the final time, t_{end}, against n (volume) on a logarithmic x scale. This result correlates closely with those from Condon et al. 2020, in that all show a linear relationship. It even shows the same order and grouping with Double and Heavy-B both very close with similar gradients, and then Tri-Molecular and Single-B with higher displacement but also with similar gradients.

Figure~4b shows the correctness/success ratio by volume for each CRN, again on a logarithmic x scale. This is less clear and evidently suffers from the low number of repeats I was able to run in comparison to Condon et al. 2020. As with their results, Double-B remains at 100% across all volumes with no variation, Heavy-B has a slightly lower start but reaching 100% quickly (at around $n = 1,000$).

By contrast, Tri-Molecular has wide variation but tends towards 100% above $n = 1,000$ and Single-B has a much lower starting correctness and requires a very large volume to be consistent.

To check whether increasing the repeats did produce better results, I tried generating data for Tri-Molecular and Single-B with $10^2 \leq n \leq 10^3$ with 10,000 repeats. The original report ran 50,000 repeats however running that many generates more data than I can load from a CSV.

These results (seen in Figure~5) show at least that for $10^2 \leq n \leq 10^3$ the correctness is increasing as n increases.

Overall these results support the conclusion that in this configuration the final time increases logarithmically with n and that the correctness asymptotically approaches 100% as n increases. These same conclusions are made by Condon et al. 2020 on pg. 267 under section 9.

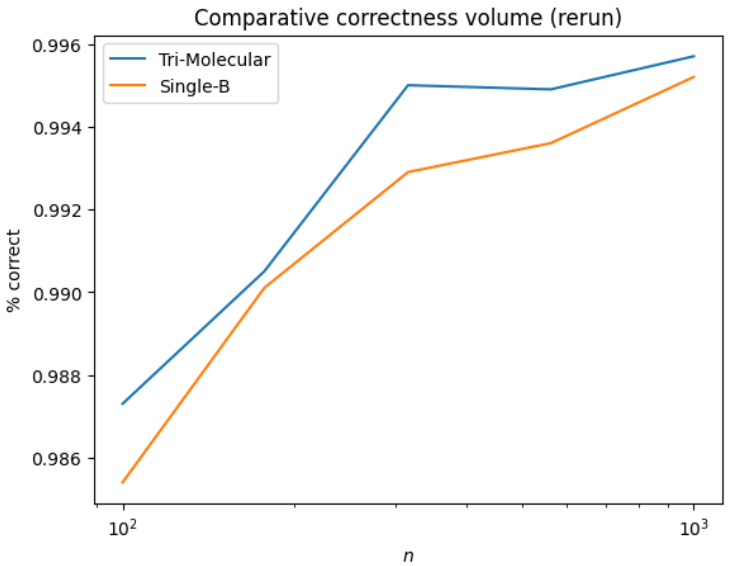
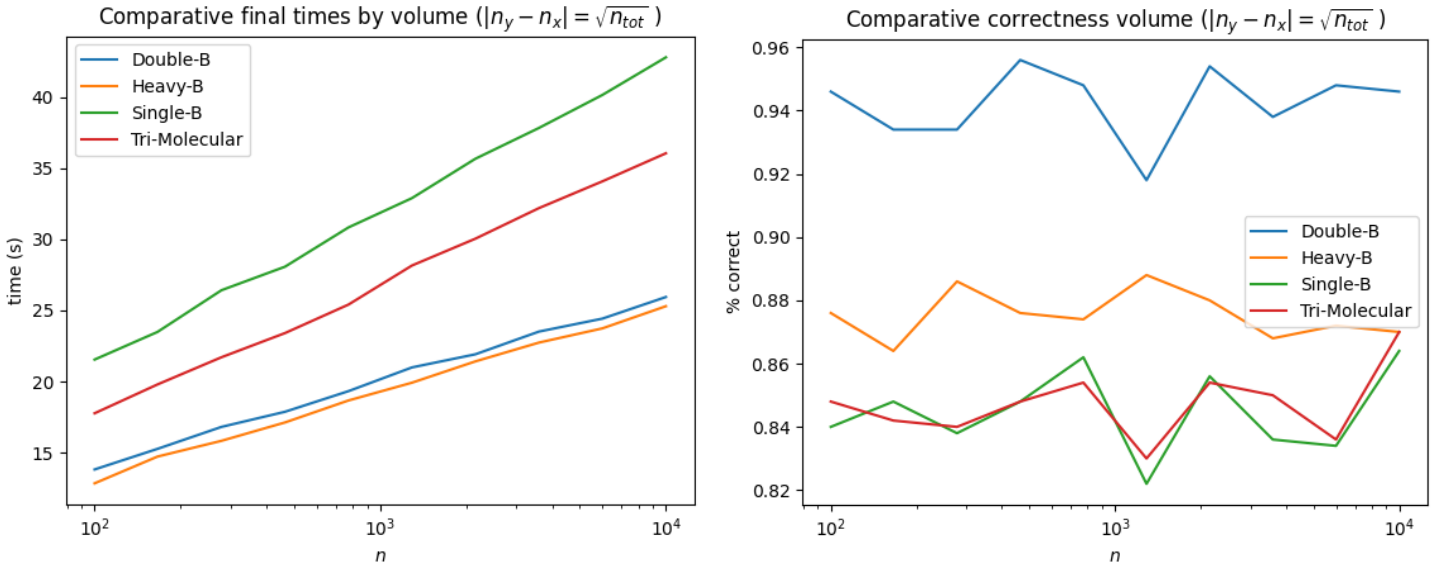


Figure 5: Single-B and Tri-Molecular Rerun for Correctness



(a) Final times for $\Delta_0 = \sqrt{n_{tot}}$

(b) Correctness for $\Delta_0 = \sqrt{n_{tot}}$

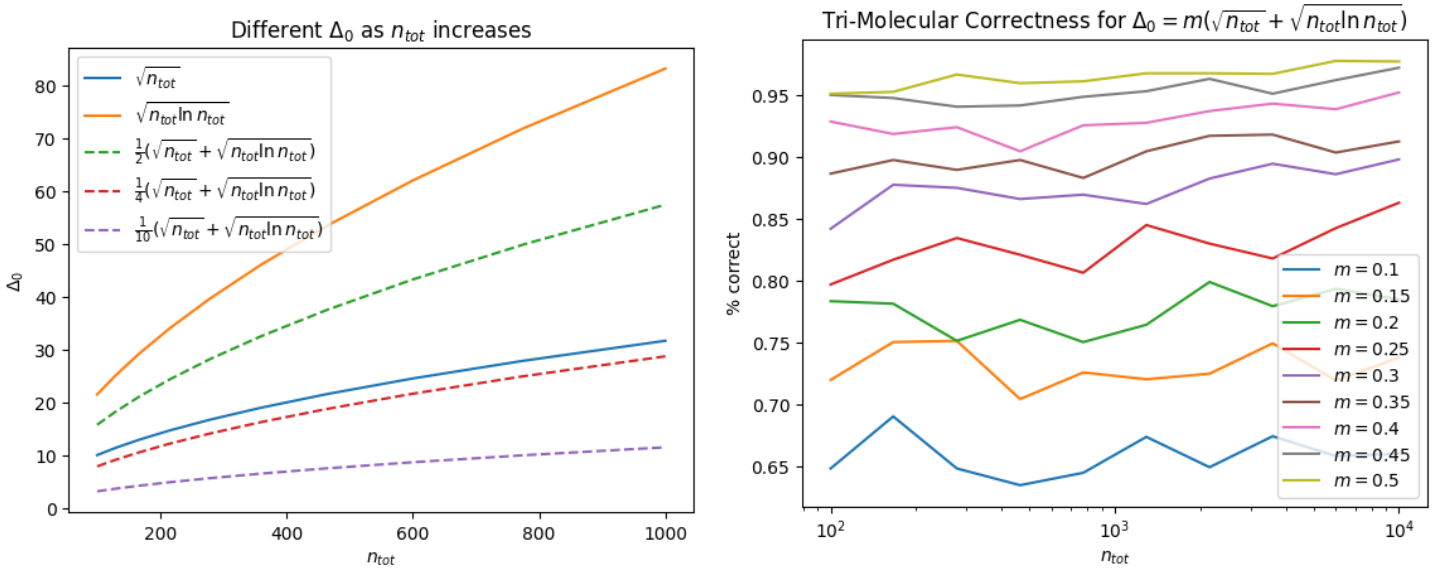
Figure 6: Equivalent graphs to Figure~4 for $\Delta_0 = \sqrt{n_{tot}}$

As in Condon et al. 2020, I ran samples for a gap of $\sqrt{n_{tot}}$. Like them, I reached the conclusion that in this case the correctness remains mostly constant as n_{tot} increases. In general, the correctness being correlated to the initial gap can be logically predicted, if one reactant has a higher initial volume then it obviously has an advantage at being the consensus.

In Figure~7a, the initial gap given by these different formula can be seen. Obviously $\sqrt{n_{tot}}$ rises much more slowly and produces a smaller gap, so this is less likely to produce a high level of correctness.

Given that the results from Condon et al. 2020 produce a very consistent constant correctness for this initial gap, I thought perhaps that any $\Delta_0 < \sqrt{n_{tot} \ln n_{tot}}$ would have constant correctness, therefore I ran samples for $\Delta_0 = m(\sqrt{n_{tot}} + \sqrt{n_{tot} \ln n_{tot}})$.

Figure~7b, this seems to support this prediction: $\Delta_0 \geq \sqrt{n_{tot} \ln n_{tot}}$ could be the requirement for a sample to gain correctness with higher system volume. Figure~7a shows the values of the original Δ_0s and, in dotted lines, the values of this new initial gap function.



(a) Plot for different initial gaps

(b) Plot of correctness for $\Delta_0 = m(\sqrt{n_{tot}} + \sqrt{n_{tot} \ln n_{tot}})$

Figure 7

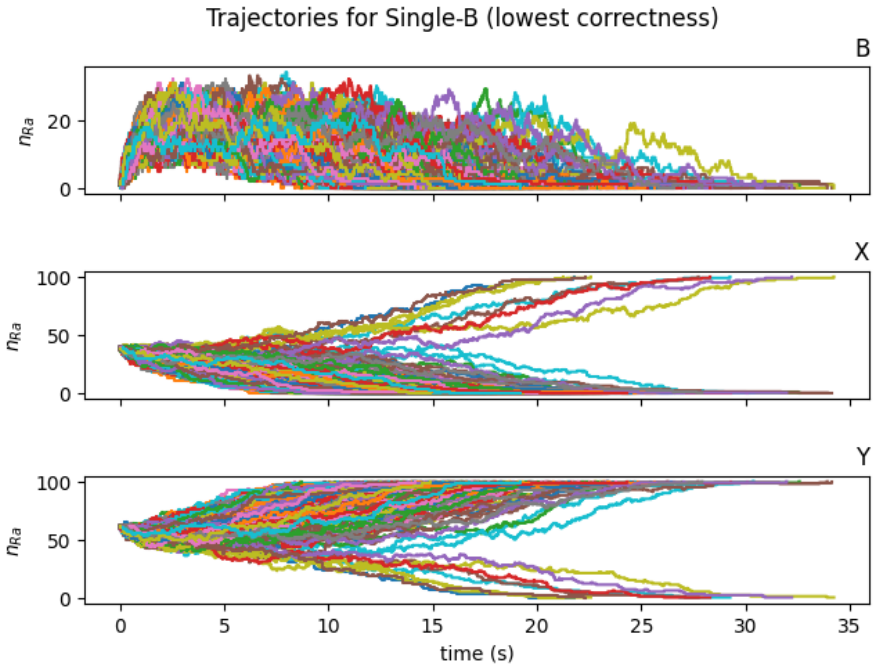


Figure 8: Trajectory of each reactant for the lowest correctness run (Single-B CRN)

I plotted trajectories for the lowest correctness volume in each experiment for each repeat, these can be seen for Single-B in Figure~8. These show that X and Y are always complementary and inverse in value whilst B, which essentially acts as a temporary store for X or Y, rises then falls as the mixture acquires consensus.

This is true for all CRNs however in the case of Double-B in the $\Delta_0 = \sqrt{n_{tot} \ln n_{tot}}$ samples there are no instances with any correctness below 100%.

As can be seen in Figure~9, when $\Delta_0 = \sqrt{n_{tot}}$ the success rate drops significantly as previously mentioned. This is observable in the trajectory plot as the number of traces seems almost equal between reaching X consensus and Y consensus.

We can also see in both of these plots that for Single-B the quantities of both X and Y initially drop as B rises, then recover as B reduces.

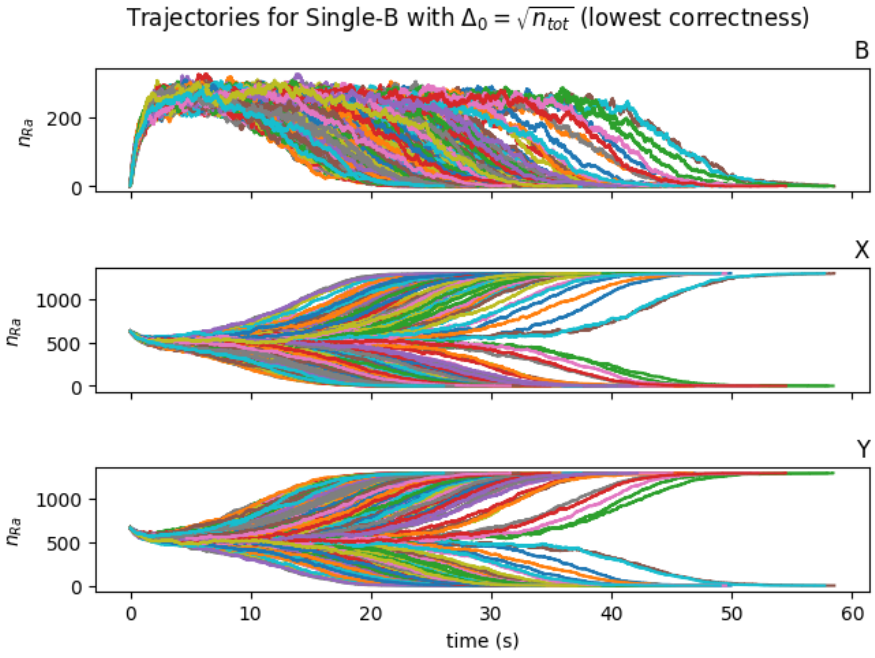
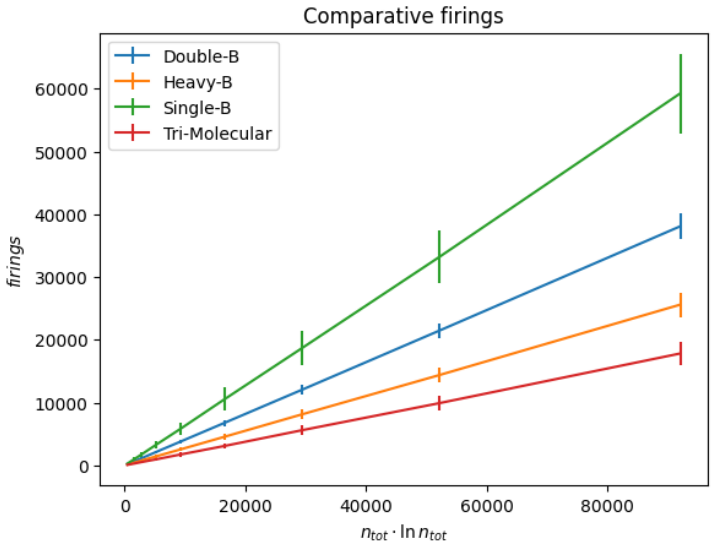
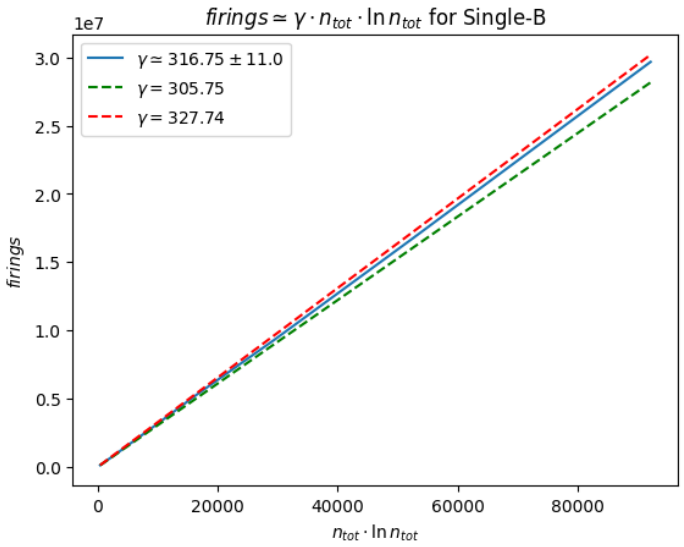


Figure 9: Equivalent to Figure~8 but with $\Delta_0 = \sqrt{n_{tot}}$



(a) For all CRNs



(b) For Single-B (with standard deviation bounds)

Figure 10: Graphs of number of transitions fired against $n_{tot} \ln n_{tot}$

To test the statement that firings $= \Omega(\gamma n \ln n)$ for varying n, I graphed the equivalent for my data (see Figure~10a). These lines show that this assertion holds for my data, as the relationship is linear. It therefore that the number of firings is linearly proportional to $(\Delta_0)^2$ - in this case, at least ($\Delta_0 = |n_x - n_y|$ which in this case means $\Delta_0 = \sqrt{n_{tot} \ln n_{tot}}$).

However this is not true for linear initial gaps as can be seen in Figure~11, the relationship is almost polynomial. Based on this interpretation, I also plotted for Δ_0 against the number of firings. For the Tri-Molecular CRN at least the number of firings does seem to be linearly proportional to Δ_0.

This makes sense, because as $\Delta_0 = \sqrt{n \ln n}$ produces a logarithmically decreasing percentage gap as n increases, whereas a simple percentage will be constant.

I have included Figure~10b to show the values for γ and the standard deviation. In this case the standard deviation is very low so we can summise that $\gamma = 316.75$ is a reasonable predictor for the number of firings Single-B will take to reach consensus with this initial gap.

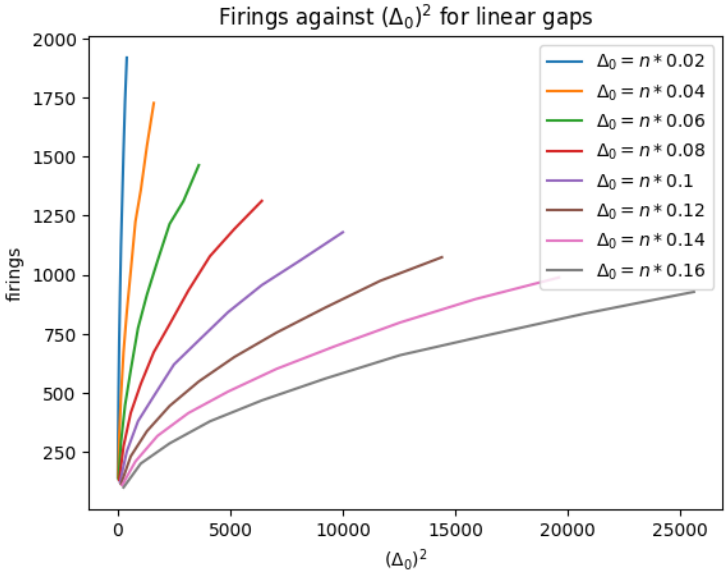


Figure 11: Plot for $(\Delta_0)^2$ (Tri-Molecular CRN)

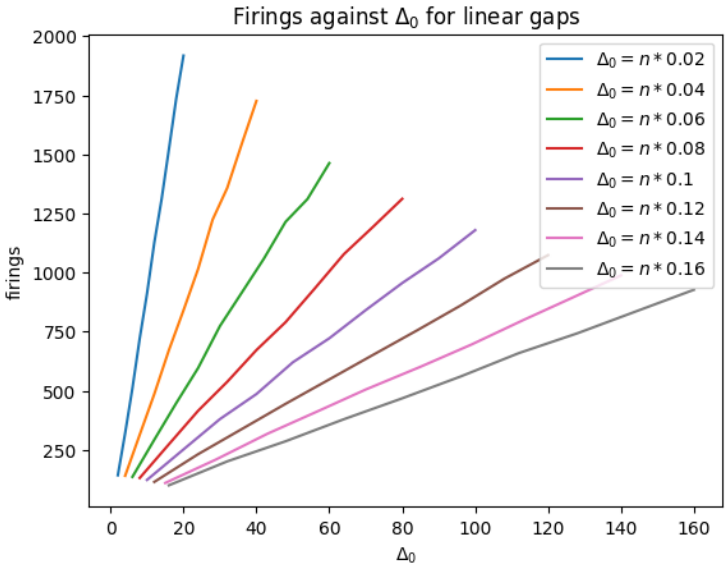


Figure 12: Plot for Δ_0 (Tri-Molecular CRN)

Conclusion

Results and support

Overall my results across the board support the conclusions of Condon et al. 2020, although my samples are universally lower in accuracy due to a lower number of repeats.

Both mine and the results of the original report support the conclusion that Double-B produces the highest success rate overall. With a minimum initial gap of $\Delta_0 = \sqrt{n_{tot} \ln n_{tot}}$ it seems to only produce correct results. This comes at the cost of more firings and longer simulation execution time until consensus is reached, but even so the simulated time for Double-B is one of the lowest.

This is also true for the comparative inefficacy of Single-B, having one of the lowest success rates with the highest firings and simulation time.

Additionally, my findings suggest that changing the initial gap Δ_0 either to $\sqrt{n}$ or $m(\sqrt{n_{tot}} + \sqrt{n_{tot} \ln n_{tot}})$ for some $0 < m \leq 0.5$ produces linear correctness, supporting the choice of $\sqrt{n_{tot} \ln n_{tot}}$ as a minimum initial gap to achieve higher success rates for larger n_{tot}.

I have found no evidence that contradicts Figure~3 (from the original report), this seems to be a valid comparison of the performance of each CRN and the conclusions drawn from it. The following is the linear regression of $\ln n_{tot}$ against t_{end} for the lines drawn in Figure~4a (all have rvalues more than 0.99):

CRN	Gradient (Condon et al. 2020)	Gradient (my results)
Double-B	$2.4 \ln n_{tot}$	$2.44 \ln n_{tot}$
Heavy-B	$2.4 \ln n_{tot}$	$2.44 \ln n_{tot}$
Single-B	$4.2 \ln n_{tot}$	$4.2 \ln n_{tot}$
Tri-Molecular	$3.7 \ln n_{tot}$	$3.65 \ln n_{tot}$

Table 1: Relative gradients for my data vs Condon et al. 2020

The strongest support I have found is for the claim of

$$\text{Number of firings} = \Omega(\gamma n \ln n)$$

holds for $\Delta_0 = \sqrt{n_{tot} \ln n_{tot}}$.

Figure~10a shows a very strong linear correlation, indicating that for some γ and n_{tot} the number of firings can be accurately predicted.

Reflection and potential improvements

While my data supports the conclusions of Condon et al. 2020, there is reasonable doubt in some cases as due to both time and space limitations my results have fewer repeats. This could be solved by using a more scalable system for sampling and then storing the data, for example an easy method would be separating data by volume into separate CSVs and zipping them with a JSON file to map them. An even better method would be to use some kind of database to manage this.

As previously mentioned, I have not covered some of the more complex simulations mentioned in Condon et al. 2020 such as CRNs with infection mentioned in that report. Some of these would require improvements to my simulation system, for example using sampler implementations other than Gillespie's.

Additionally, my code may be an imperfect representation of the problem, it certainly does not allow for the variations in the original report that are out of scope for this one. Were I to repeat this I would start by either finding or producing a consistent and well tested open-source system for running Petri Net style simulations. This would allow understanding of specific details of implementation and not only the mathematical formulae involved.

Bibliography

Condon, Anne et al. (2020). “Approximate majority analyses using tri-molecular chemical reaction networks”. In: *Natural Computing* 19.1.

Gillespie, Daniel T (1977). “Exact stochastic simulation of coupled chemical reactions”. In: *The journal of physical chemistry* 81.25, pp. 2340–2361.