Mitochondrial DNA population dynamics are strongly affected by inherited mutation load

Gold CREST award report

Mairi MacIain

Mentored by Conor Lawless

Planning the project

Mitochondria are a type of organelle in eukaryotic cells which are involved in the process of respiration. Respiration uses fats, sugars, proteins and oxygen to make ATP which can then release energy for the cell to function. Mitochondria have other roles including being involved in apoptosis (cell death).

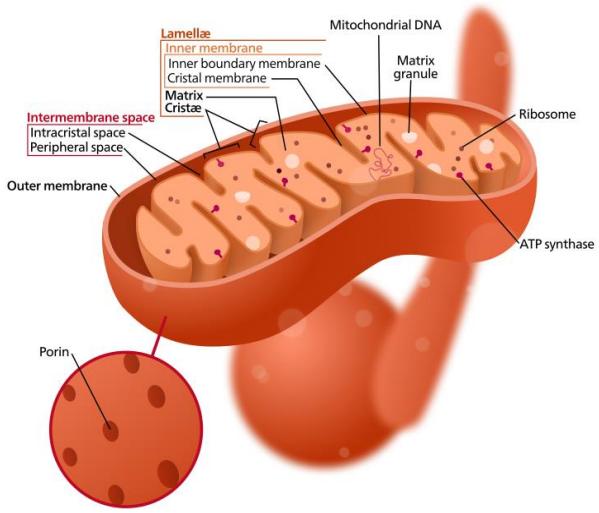

Figure 1: Diagram of a mitochondrion. Reproduced from Kelvinsong (2020)

Figure 2: Electron microscopy image of two mitochondria. Reproduced from Howard (2020)

Mitochondria have their own circular DNA (mtDNA), separate from nuclear DNA (nDNA). mtDNA codes for proteins specific to mitochondria. mtDNA is significantly smaller than nDNA however it is polyploidy so there are many copies in a single cell (Piotrowicz (2021)). Clonal expansion is when the concentration of mutated molecules in the mtDNA population increases. It is independent (has no effect on other cells) and stochastic (random). New mutations can arise in individual cells at low concentrations as humans age and those mutations can undergo clonal expansion. Disease-causing mutations can be inherited at high concentration and undergo clonal expansion.

Clonal expansion of mtDNA mutations to sufficiently high levels can cause disease (Lawless *et al*. (2020)). This is because mtDNA codes for many mitochondrial proteins, which when not produced accurately means that the body does not get enough energy to function properly. The result is called mitochondrial dysfunction. This can cause mitochondrial myopathies, a group of muscular diseases that can affect muscles everywhere in the body (including heart and lung muscles), or mitochondrial encephalomyopathies, which affect the muscles and the nervous system. Mitochondrial dysfunction can cause a number of other related syndromes (Lawless (2020) & NINDS(2021)).

Mitochondrial disease can have severe and untreatable consequences such as vision and hearing loss, difficulty walking and breathing, epilepsy, neurodegeneration and developmental defects. Mitochondrial dysfunction has also been connected to Alzheimer's, because of neurons not having enough energy to function properly.

Mitochondrial diseases are currently incurable and getting biological samples to look for a cure is a painful process. Sequential biopsies also cannot show the progression of the disease clearly because we cannot observe the same cell at multiple time points since measuring the mutation load destroys the cell. Therefore, computer simulations can be used to simulate a cell and the mitochondrial DNA within it and make the connection between time points.

The aim of this project is to create an agent-based model simulating mitochondrial DNA (mtDNA) population dynamics in a single cell. Such a model could be used to make the connection between observations of biological samples and the underlying mtDNA population dynamics.

Objectives

- Research the biology behind mitochondrial diseases to project to gain an understanding of mitochondria and mutations
- Decide on a computer programming language to create the model in
- Learn the language to know enough to complete the project
- Simulate a cell with mutant and wild-type mtDNA molecules
- Decide on a way to visualise the data and implement it
- Analyse the data
- Form a conclusion

I will have achieved my aim if I have a working simulation which outputs data and I can represent the output in a clear, visual form.

I considered the different ways I could plan my time. I decided against using a Gantt chart due to the flexible nature of my project and having to go back and forth learning more coding to be able to continue the project. I decided to allocate each of my objectives a period of time, to ensure I spent an appropriate amount of time on it, and created a spreadsheet to keep track of time. I also accounted for times for meetings and documentation which I did throughout the project. During the project I ensured I spent around the allocated time on each section while not going too much over so I kept on track.

The time allocations were as follows:

Objective	Time (hours)
Meetings with my mentor	20
Documentation	15
Researching background biology	6
Deciding and learning a language	10
Creating a simulation and a visual output	15
Analysing the results and forming a conclusion	4

There are many different programming languages I could have used for this project. My mentor and I considered using

- Julia: Fast, clear syntax, many scientific packages
- R: Emphasis on statistical computing
- Python: Clear syntax, many scientific packages

We also considered whether to use functional programming or object-oriented programming. Although Julia would be considered the best in the field, (Piotrowicz (2020)) we decided to complete the project in Python. Since I have experience in object-oriented programming in Python from Higher Computing, and python is often used for scientific computing, we sacrificed the speed at which the simulation would run, for ease of writing.

Using Python reduced the time it took to learn the programming language since I had a solid base in the language and was familiar with the syntax, however I had to learn how to use the Matplotlib package to be able to plot my data. We initially planned to create the model itself using the Mesa library, which is a Python package for creating agent based models. An agent based model simulates individual objects and their interactions with their environment and other objects. However, after learning how to use the Mesa library I realised that it would be more straightforward to code the simulation in pure Python.

To achieve the aim of creating an agent-based model simulating mitochondrial DNA (mtDNA) population dynamics in a single cell my mentor and I planned for my code to include:

- A system to contain agents, representing mtDNA molecules in a cell.
- Every agent to have the same replication and degradation rate attached to it.
- The rolling of random numbers to decide whether an agent (mtDNA molecule) would replicate, degrade or not change to simulate the stochasticity (randomness) of mtDNA population dynamics.
- Degraded molecule agents being removed from the simulation.
- Replicated molecule agents being removed and two daughter molecules added.
- A way to calculate the mutation load.
- A way to visually represent the dynamics of mutation load and total number of molecules

This model will not include random mutations, so will only focus on inherited mutations.

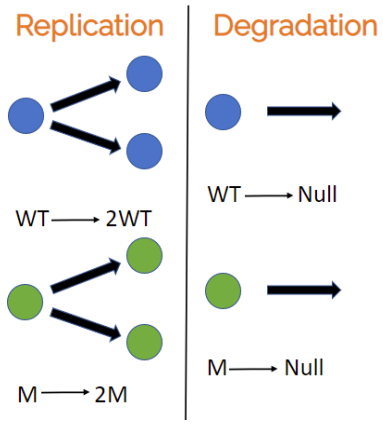

Figure 1: Biochemical reaction network underlying mtDNA population dynamics model

Once this base code was completed and functioned properly it produced a plot of the copy number (number of mtDNA molecules in the cell)/time and mutation load/time.

Although my 4th and 5th objective were technically completed to make the simulation more accurate and cell like I extended my plan and decided on some additions to the project:

- Copy number control: to simulate cellular homeostasis.
- Molecule ages: so a molecule that just came into existence can't immediately replicate / degrade.

I also made other additions to help with analysing and visualising my data:

- Batch runs.
- Layering batch runs on top of one another.
- Drawing percentile lines on the plots.
- Creating a plot with the median from each mutation load.

Throughout the project

There are many online resources I used to help learn how to use Matplotlib and the Mesa library. While I was learning Mesa I used the documentation written by the creators. I also worked through other online tutorials, one of which was an infection model which helped me think about how I could output my data. I used websites such as W3schools and stackoverflow to help me use Matplotlib. Throughout the project when I came to a problem I could not solve myself, my mentor and I would meet and he would help me work through the problem, and show me some more advanced computational techniques which I could use to solve any future problems.

Since my project was completed entirely on my computer and only involved programming and research there were no major safety concerns, however, throughout the project I did have to take into account the computational power of my computer. Running the simulation many times with many agents would have put a strain on my computer and possibly crashed it. Therefore, for most of the project and testing I ran the simulation with a low number of agents. Once I got to the analysing stage my mentor ran the simulation many times with many agents on his more powerful computer.

Results

I decided that the replication and degradation rates of the mutant and wild-type molecules would be the same. This means that my program simulated a cell with no bias towards either type of molecule. All changes in mutation load were because of random genetic drift (Kimura (1968)).

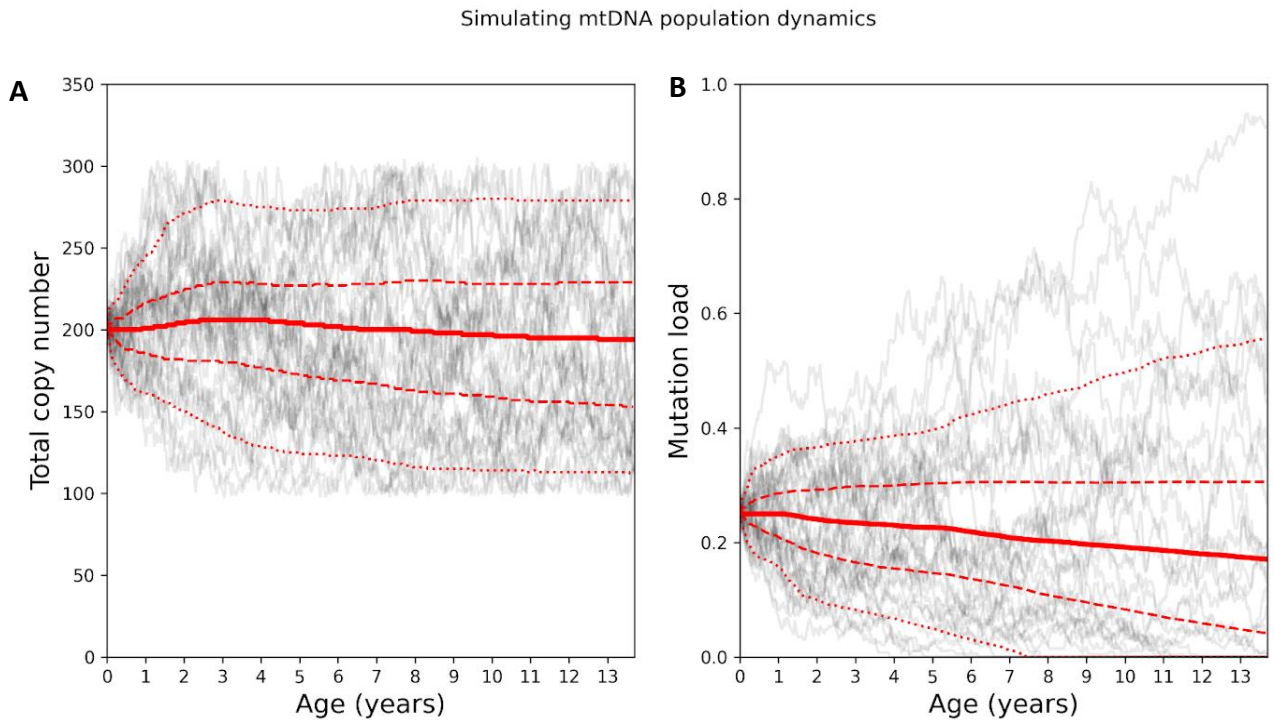

Figure 2: Summary of 35 stochastic simulations of mtDNA population dynamics with copy number control. Both panels summarise the same 35 simulations of 200 mtDNA molecules in a single cell. The solid red line is the median, and the dotted red lines are the 5th, 25th, 75th, and 95th percentile lines. ***(A)*** *Shows the number of agents in the cell at a point in time (initial condition: 200)* ***(B)*** *Concentration of mutant molecules in the cell (initial condition: 0.25).*

The copy number cannot go above 300 or below 100 because I implemented copy number control to better represent cellular homeostasis. Although the mutation load starts the same for each simulation the outcome can vary wildly, however the median decreases slightly. This was a trend for this mutation load when the simulations were repeated.

After I repeated these simulations many times I created a table recording the beginning and end position of the median of each plot. Then I created a visual comparison of the results (Figure 5).

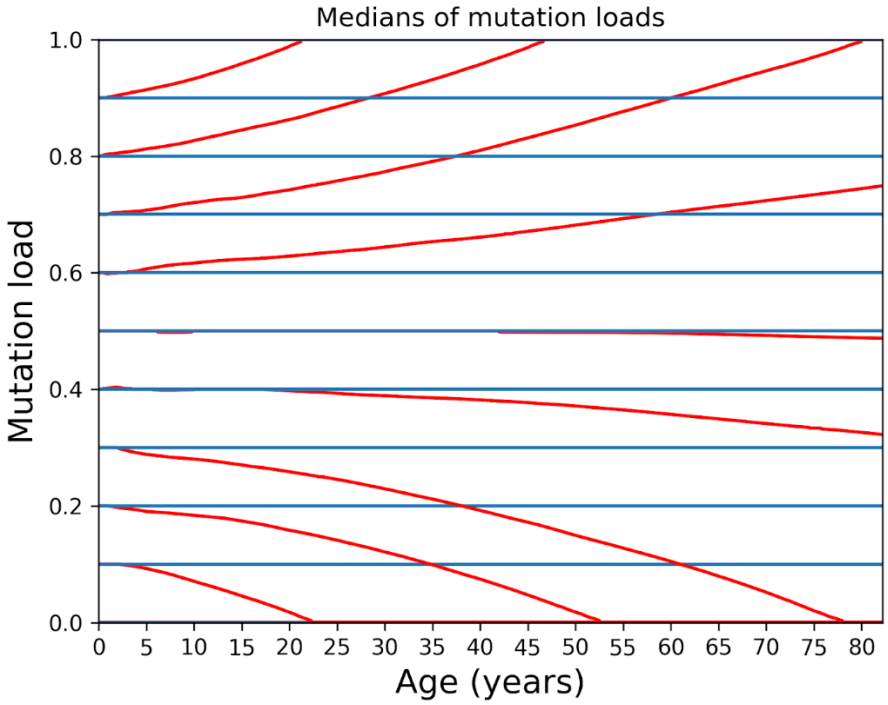

Figure 3: A plot of 11 medians each starting from a different mutation load. Each median is the result of 300 simulations.

The large amount of simulations did not completely reduce the statistical uncertainty since to 0.5 mutation load median is not a straight line. However, this plot shows that the mutation load does significantly and consistently change over time.

Conclusion

This work shows that if the initial mutation load is over 0.5, over a long enough time period the average mutation load can reach pathogenic levels, even with the cell having no bias towards either type of agent. This demonstrates that mitochondrial disease can progress under the assumption of random genetic drift. Conversely if the mutation load is less than 0.5 the average mutation load can fall to 0 given sufficient time. The initial condition is very important in mitochondrial DNA population dynamics

This simulation can then be compared with biological samples and if over many years the trend is the same then we can conclude that the cell has no bias towards or against mutant mtDNA molecules. This would give us a better understanding of clonal expansion and so would help find a cure for mitochondrial diseases.

Evaluation

The first main problem I came across was when I was using the Mesa library. Mesa would allow me to add each agent into the simulation one by one but not all at the same time. This meant that my cell could not start out with any mtDNA molecules but gradually gain them, which is not scientifically accurate. I tried different ways to work around this but ultimately I decided to use pure Python for the remainder of the project.

Another main problem was in my first stage of my plan, my simulation did not have copy number control. This meant that my copy number could become 0, because of the way Matplotlib works every time this

happened, which was often, an error message would appear and I would not be able to see any data from the simulation. I wrote a section of code which would check the copy number and work around it to ensure a plot was shown. This was very beneficial because now I could always see a plot of the simulation no matter how short it was, and see the effect on mutation load.

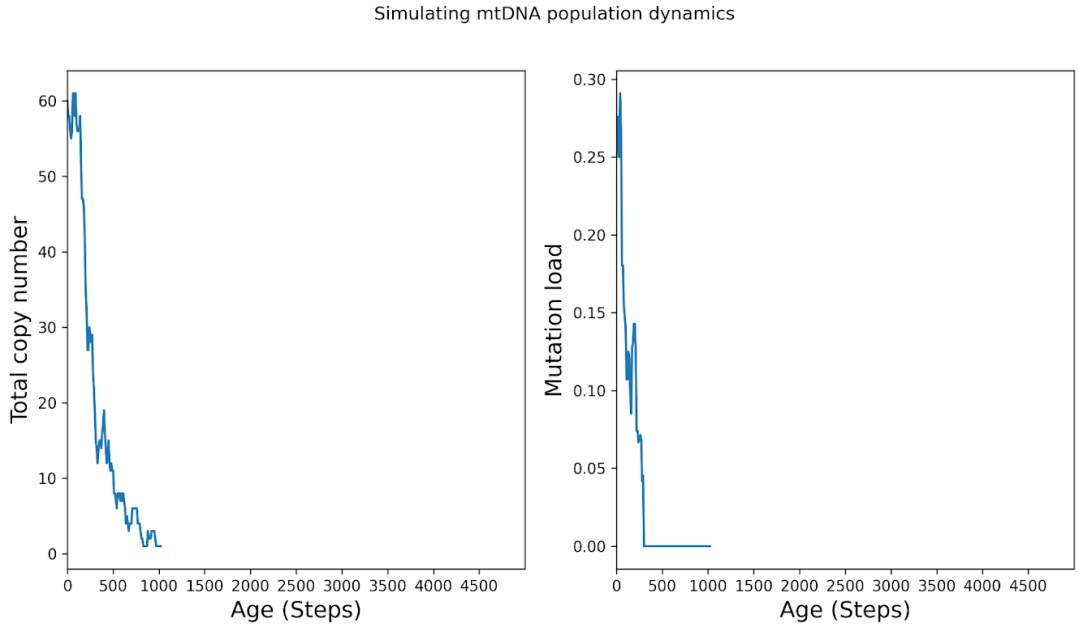

Figure 4: This is an example plot of a cell which died before the end of the simulation but still produced a plot. (Each step represents a day).

Throughout this project I have greatly increased my programming knowledge. I have built on my Python knowledge, including my mentor teaching me more efficient coding constructs, I am now capable of producing plots which clearly and attractively show data, and I have a good understanding on the benefits of scientific computing. Going into this project I had not studied biology to a senior level. This project gave me the opportunity to learn about a section of biology with a lot of ongoing research, very beneficial to society. This project has helped me build research skills, independence in my learning, and organisation.

To improve this project, I would have created a better way of having a timescale. Although my project was very flexible, I should have given myself some dates to stick to. I found myself not working on my project for periods of time and then doing a lot at once, and a timescale would have made my project more continuous. To improve my simulation, I could add spatial modelling and show the effect the cell and mtDNA molecules might have on one another. I could also do more research to find more attributes that would make my simulation more cell like.

References

Kelvinsong, Diagram of a mitochondrion, Wikipedia, 2020, *https://commons.wikimedia.org/w/index.php?curid=27715320* (accessed 01/12/2021)

Howard, L. Electron microscopy image of two mitochondrial cross-sections, Wikipedia, 2020, *https://commons.wikimedia.org/w/index.php?curid=1248089* (accessed 01/12/2021)

Piotrowicz, M. *Stochastic models of mitochondrial DNA population dynamics*, MRes dissertation, 2021

Lawless, C., Greaves, L, Reeve, A.K., Turnbull, D.M. and Vincent, A. E. *The rise and rise of mitochondrial DNA mutations*, Open Biology, 2020 https://doi.org/10.1098/rsob.200061

Lawless, C. *Mitochondria and clonal expansion*, 2020, *http://mito.ncl.ac.uk/clonexp/clonal_expansion/*

NINDS, *Mitochondrial myopathy fact sheet*, 2021, *https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Mitochondrial-Myopathy-Fact-Sheet*

Piotrowicz, M, *Comparing Simulation Speeds in Python*, Julia and R, 2020, http://mito.ncl.ac.uk/clonexp/speed_comparison/

Kimura, M. *Evolutionary Rate at the molecular level*, Nature, 1968 https://doi.org/10.1038/217624a0