Conrad Tibesar, Khoa Le, Justin Schneider
Bioinformatics
Final Project Paper
December 13th, 2022

Introduction:

The purpose of this research project is to simulate the various different evolutionary models that we discussed in class, namely the *Jukes-Cantor model* (*JC*), *Kimura 2-Parameter* model (*K2P*), *Hasegawa-Kishino-Yano (1985)* model (*HKY85*), and finally the *General Time-Reversible* model (*GTR*). By simulating these different models, we wanted to determine the difference in time scales the models fit well with. To do this, we simulated these models using various different genes across different species, measuring the genetic distance at each generation from the original gene. Once the genetic distance was close to the maximum genetic distance (0.75) for a given number of consecutive generations, the simulation ended, and we counted the number of generations it took for the simulation to reach that threshold. This tells us how quickly these different models change, and it gives us information about the limitations of the models in regards to the timescale at which they stop providing much information.

Methods:

Coding Process:

The program will take one sequence of DNA at a time to analyze. The sequence will be taken from the text file and undergo validation. If the sequence is valid, it will be plugged into four different evolutionary models: Jukes-Cantor, Kimura 2-Parameter, HKY85, and GTR respectively. Each simulation accepts the initial nucleotide sequence and will run until the average genetic distance of a user-defined number of consensus generations is greater than or equal to a threshold of maximum genetic distance (0.75). The number of consensus generations is the number of consecutive generations which must average at or above the genetic distance threshold. This lets the program run a bit past the first instance of exceeding the threshold, which is important as the genetic distance should fluctuate around the threshold. In this program, the consensus generations will be declared as 5 consecutive generations.

The genetic distance is calculated by dividing the differences between two DNA sequences by its length (the two sequences have the same length so either of them will work). However, we will not be using this value to compare to the user defined threshold. The average

genetic distance will be calculated by using the last *n* consensus sequences' genetic distances. This value will now be used to compare to the user defined threshold as an indicator whether we have enough data to stop the simulation and go to the next model.

During the simulation, each nucleotide will undergo mutation independently. Firstly, a copy of the original sequence will be made for later calculating the genetic distance. The built-in random function will generate a random number between 0 and 1 inclusive. The nucleotide will be used to find its mutation rate in the corresponding table (differs from models to models). The random number will now be used as the probability of the nucleotide mutating. Mutation can happen or not (mutating to itself – not mutating) depending on the probability. The mutated nucleotide will now replace the old one in the sequence and the mutating process continues until all the nucleotides have undergone mutation. The genetic distance between the new and old sequence will be stored in a list that's called distance_by_generation. As long as the average distance of the last *n* consensus generations does not exceed the threshold for maximum genetic distance, the newly mutated sequence will repeat the mutation process to produce the next mutated generation. All the distances by generation are calculated using the first generation and the newest mutated generation.

Mutation probabilities are user-defined variables. In this program, however, in order to simulate the evolutionary models within reasonable computation time, we must make some assumptions about the number of mutations over a period of generations. For some of the genes, the mutation rate between individual generations is too small, so we must speed up the process by adding in some determinism. To do this, we will assume a mutation rate that is greater over a greater number of generations, so each generation in the produced list will represent a certain number of generations that have passed.

For our research, a simulation for each model will be run 20 times using the input file, and each simulation will result in a scatter plot displaying the genetic distance as a function of generation, or normalized generation, and the number of normalized generations needed to attain the maximum genetic distance. Additionally, it will create a graph with each model's scatter plots overlaid to allow for simple model comparison throughout a simulation. The program will then display the gene's nucleotide frequency distribution and a dictionary listing the number of generations needed per model type for each of the 20 simulations to attain the maximum genetic

distance. Following this, the means, standard deviations, and confidence intervals of each of the models for that specific gene were determined using the provided data.

Research Methods:

Research was a much more arduous process than initially anticipated. Our goal for research was to gather general information about the mutation rates of genes. After finding a source with information on mutation rates, we needed to then find the genetic sequence of the gene whose research findings were being represented. This was a lot easier said than done, however, as sources often didn't publish concrete values by nature of how inconsistent mutation actually is. Research was often flooded with sources that either studied mutation rates too specific for the scope of our processes, or otherwise studied the effects of mutation, something our program does not look to address. Rates of mutation in most organisms are so low that even observation over decades is not necessarily conclusive. This is the primary reason *HIV* was the easiest to obtain information for, and one of the only sequences we were able to use, as the rate of mutation is great enough that individual rates per nucleotide are more readily available. The other sequence we were able to find information for relatively early was yeast. Despite yeast's incredibly low mutation rate, even as far as mutation goes, yeast is a very simple organism, and generations of yeast can be studied over a very short timeframe. For this reason, yeast is one of the most common organisms to conduct studies on, so we were able to find a *GTR* value matrix relatively quickly. However, research came to a grinding halt when considering other species, especially eukaryotic species. It seems that *GTR* values are infrequently, if at all represented through research of most eukaryotes. This seems in part because eukaryotic genes are much more complicated, and often have multiple functional roles. It is much simpler to perform studies on specific mutations and their effects, as opposed to giving generalized information about an entire chromosome and its mutation rates. Because of this, finding mutation frequencies in eukaryotic organisms was much more difficult, to the point that we even had to scrap the idea of including a human gene, as interesting as that research may have been. In addition, it appears that it is generally more effective for the purpose of studying to label eukaryotic mutations through the corresponding amino acid changes instead of single nucleotide mutations. This is likely because eukaryotic multicellular organisms can have mutations at either a cellular level, such as cancerous cells, or at a reproductive, hereditary level, such as the presence, or lack

thereof, lactose intolerance in humans. These factors compounded together and greatly obscured the research results for many of the eukaryotes we had intended to also study. The research conducted of these other eukaryotes was still able to lead us to a greater understanding of the limitations of our study. It thus informed us of means to improve and refine it if we wanted to increase the scope of our project.

Results:

1. *HIV* Gag Gene Simulation Results:

The *HIV Gag* gene provided had the following nucleotide distribution, which will be important when discussing and analyzing the results.

Number of nucleotides	% Adenine	% Cytosine	% Guanine	% Thymine
1500	36.8%	19.6%	24.5%	19.1%

(Table 1: *HIV Gag* nucleotide distribution)

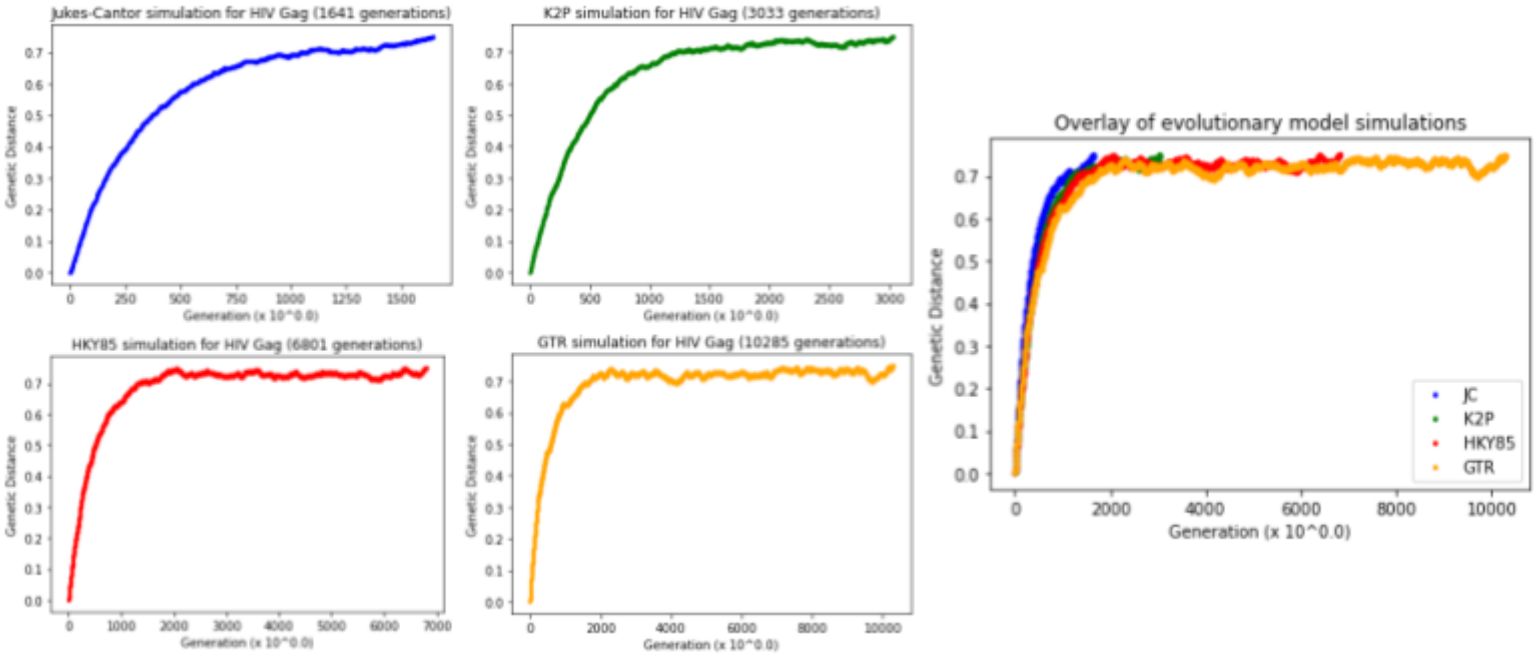
The research showed that the mutation rates are as follows:

Parameter	**Mutations (per nucleotide per 1 generation)**
General mutation rate (*JC_alpha*)	7.00×10^{-4}
Transition rate (*K2P_alpha* and *HKY85_alpha*)	5.38×10^{-4}
Transversion rate (*K2P_beta* and *HKY85_beta*)	1.62×10^{-4}
A ⟷ G (*alpha_AG*)	3.5×10^{-4}
C ⟷ T (*alpha_CT*)	1.88×10^{-4}
A ⟷ C (*beta_AC*)	0.672×10^{-4}
A ⟷ T (*beta_AT*)	0
C ⟷ G (*beta_CG*)	0

G ⟷ T (*beta_GT*)	0.938×10^{-4}

(Table 2: *HIV Gag* mutation rates)

With these mutation rates, we ran the simulations 20 times for each model. Each simulation was plotted, displaying the progression as the genetic distance moves towards the threshold of 0.749 for each different evolutionary model. One of the simulations produced these graphs:



(Graphs 1 - 5: *HIV Gag* simulation results)

Over the 20 simulations ran, the overall data is:

Data point	**Number of Generations to Reach Max. Genetic Distance**			
	Jukes-Cantor	**Kimura 2-Parameter**	**HKY85**	**General Time-Reversible**
Average # to reach threshold	1770	2850	4250	7200
Standard Deviation	407	601	1560	2890
Maximum	2890	4210	8250	12100
Minimum	1180	1960	1780	2610
90% confidence	1770 ± 150 (±8.47%)	2850 ± 221 (±7.77%)	4250 ± 572 (±13.5%)	7200 ± 1060(±14.8%)
95% confidence	1770 ± 178 (±10.1%)	2850 ± 263 (±9.26%)	4250 ± 682 (±16.0%)	7200 ± 1270(±17.6%)

(Table 3: *HIV Gag* simulation results)

2. Drosophila Melanogaster White Gene Simulation Results:

The *Drosophila Melanogaster White* gene provided had the following nucleotide distribution, which will be important when discussing and analyzing the results.

Number of nucleotides	% Adenine	% Cytosine	% Guanine	% Thymine
5,868	26.6%	23.1%	22.6%	27.8%

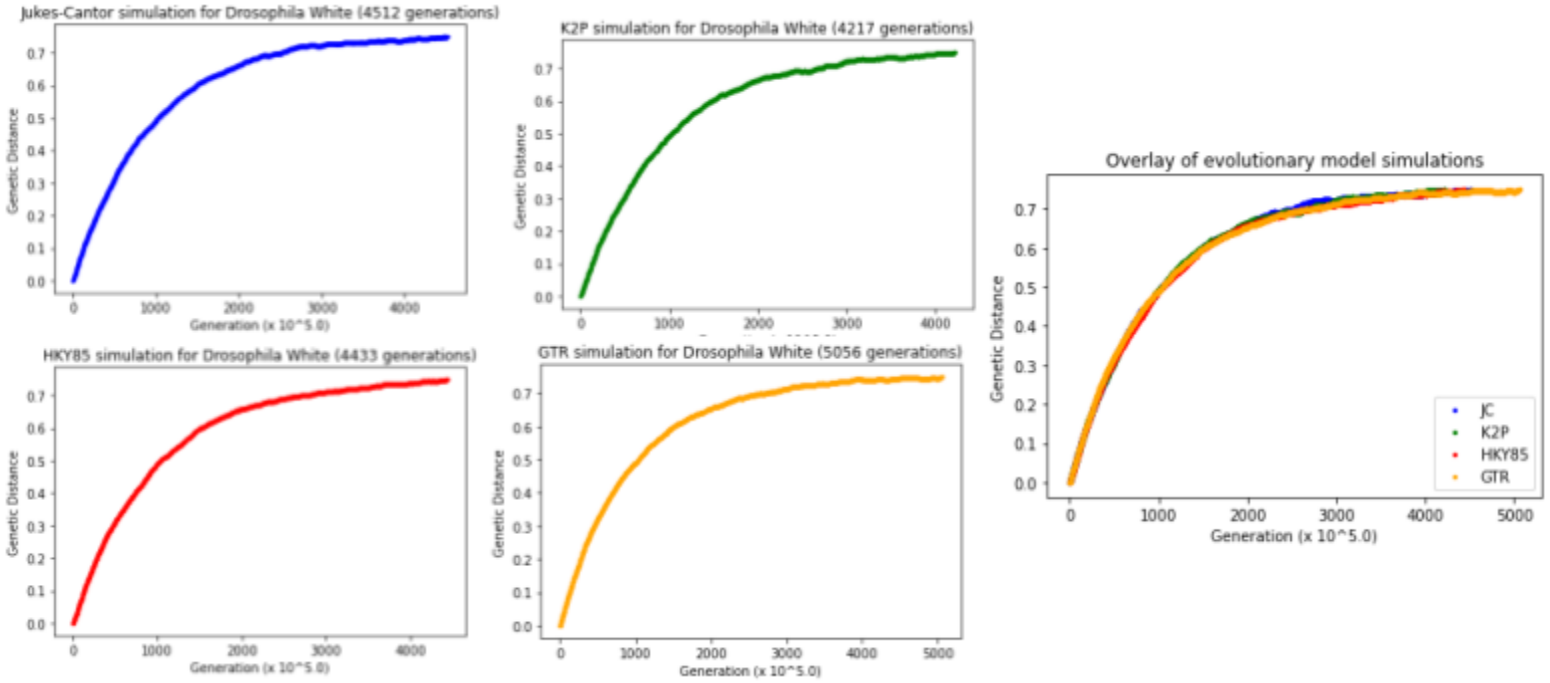
(Table 4: *Drosophila Melanogaster White* nucleotide distribution)

For the *Drosophila Melanogaster White* gene, the research showed that the mutation rates are as follows:

Parameter	**Mutations (per nucleotide per 10^5 generations)**
General mutation rate (*JC_alpha*)	2.8×10^{-4}
Transitions rate (*K2P_alpha* and *HKY85_alpha*)	1.51×10^{-4}
Transversion rate (*K2P_beta* and *HKY85_beta*)	1.29×10^{-4}
A ⟷ G (*alpha_AG*)	0.750×10^{-4}
C ⟷ T (*alpha_CT*)	0.759×10^{-4}
A ⟷ C (*beta_AC*)	0.339×10^{-4}
A ⟷ T (*beta_AT*)	0.356×10^{-4}
C ⟷ G (*beta_CG*)	0.258×10^{-4}
G ⟷ T (*beta_GT*)	0.339×10^{-4}

(Table 5: *Drosophila Melanogaster White* mutation rates)

With these mutation rates and a terminating threshold genetic distance of 0.749, one of the 20 simulations ran produced these graphs for each different evolutionary model:



(Graphs 6 - 10: *Drosophila Melanogaster White* simulation results)

Over the 20 simulations ran, the overall data is:

Data point	**Number of Generations ($\times 10^5$) to Reach Max. Genetic Distance**			
	Jukes-Cantor	**Kimura 2-Parameter**	**HKY85**	**General Time-Reversible**
Average # to reach threshold	4860	4685	5170	5240
Standard Deviation	676	679	841	1050
Maximum	6510	6390	7300	7510
Minimum	3858	3732	4110	3850
90% confidence	4860 ± 249(±5.11%)	4685 ± 250(±5.34%)	5170 ± 310 (±5.99%)	5240 ± 387 (±7.39%)
95% confidence	4860 ± 296 (±6.09%)	4685 ± 300 (±6.26%)	5170 ± 369 (±7.14%)	5240 ± 461 (±8.81%)

(Table 6: *Drosophila Melanogaster White* simulation results)

3. Yeast YML093W Gene Simulation Results:

The *Yeast YML093W* gene provided had the following nucleotide distribution, which will be important when discussing and analyzing the results.

Number of nucleotides	% Adenine	% Cytosine	% Guanine	% Thymine
2700	38.8%	15.4%	24.0%	21.8%

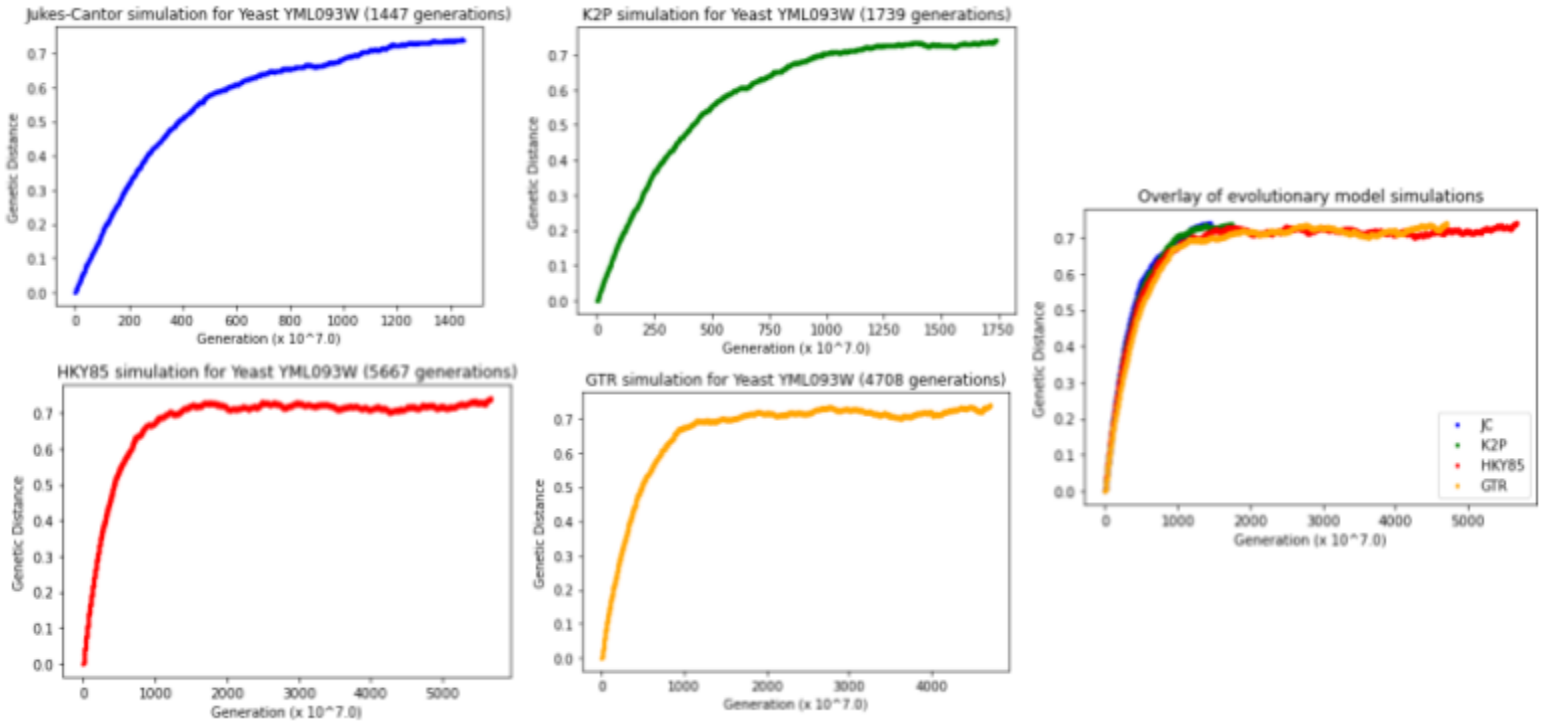
(Table 7: *Yeast YML093W* nucleotide distribution)

For the *Yeast YML093W* gene, the research showed that the mutation rates are as follows:

Parameter	**Mutations (per nucleotide per 10^7 generations)**
General mutation rate (*JC_alpha*)	6.7×10^{-4}
Transitions rate (*K2P_alpha* and *HKY85_alpha*)	3.31×10^{-4}
Transversion rate (*K2P_beta* and *HKY85_beta*)	3.39×10^{-4}
A ⟷ G (*alpha_AG*)	1.66×10^{-4}
C ⟷ T (*alpha_CT*)	1.65×10^{-4}
A ⟷ C (*beta_AC*)	0.978×10^{-4}
A ⟷ T (*beta_AT*)	0.442×10^{-4}
C ⟷ G (*beta_CG*)	1.02×10^{-4}
G ⟷ T (*beta_GT*)	0.978×10^{-4}

(Table 8: *Yeast YML093W* mutation rates)

With these mutation rates and a terminating threshold genetic distance of 0.749, one of the 20 simulations ran produced these graphs for each different evolutionary model:



(Graphs 11 - 15: *Yeast YML093W* simulation results)

Over the 20 simulations ran, the overall data is:

Data point	Number of Generations ($\times 10^7$) to Reach Max. Genetic Distance			
	Jukes-Cantor	**Kimura 2-Parameter**	**HKY85**	**General Time-Reversible**
Average # to reach threshold	1570	1530	13100	13,400
Standard Deviation	201	200	8860	11800
Maximum	1540	1970	44600	47700
Minimum	1450	1160	3020	2800
90% confidence	1570 ± 74.2 (±4.37%)	1530 ± 73.4 (±4.79%)	13100 ± 3260 (±24.8%)	13,400 ± 4340 (±32.3%)
95% confidence	1570 ± 88.3 (±5.63%)	1530 ± 87.5(±5.71%)	13100 ± 3880 (±29.6%)	13,400 ± 5180(±38.5%)

(Table 9: *Yeast YML093W* simulation results)

Analysis

1. ***HIV Gag* Analysis**

The simulations for the *HIV Gag* gene were unique from the studies on the two other genes for two specific reasons. Firstly, the general mutation rate for *HIV Gag* is much higher than the other two, and did not require us to normalize the data by multiplying each generation by a factor of 10 as we did with the other two genes. This means that the simulations had a higher degree of stochasticism, and were less deterministic. The second primary difference in the *HIV Gag* gene is the strong preference towards transitions versus transversions. The mutation rates in *HIV Gag* show that transitions are three times more likely to occur than transversions. This is starkly different from the other two, which are much closer to being equally likely. It is also worth noting nucleotide distribution in the *HIV Gag* gene. There is a significant weight towards purines in the sequence, accounting for 61.3% of the nucleotides versus only 38.7% for pyrimidines. Even within the purines, there is almost a $3:2$ ratio between adenine and guanine.

Simulating the models showed a steady increase in the average number of generations it took to reach the maximum genetic distance threshold. Based on the nucleotide distribution and mutation parameters, this makes sense. By taking into account the significantly higher transition rate, K2P increases the timescale of the simulation by almost double. By going even further and accounting for the imbalance of the nucleotide distribution within the gene in the *HKY85* simulations, we get an even greater timescale until the evolution reaches maximum genetic distance. We have another increase with the GTR simulations, which is due to the purine transition mutation rate accounting for almost half of all mutations. This, in combination with a large portion of purines, suggests that there are likely many purines mutating back to their original nucleotide ($A \rightarrow G \rightarrow A$, for instance). This would be something acting against the evolution, thus increasing the timescale of the model. It is worth noting that K2P produced the most consistent results, having the lowest relative standard deviation and best confidence levels.

2. *Drosophila White* Analysis

Compared to the *HIV Gag* gene, *Drosophila White* has a much more uniform nucleotide distribution, and an incredibly balanced distribution of purines, accounting for 49.2%, to pyrimidines, accounting for 50.8%. Again, unlike *HIV Gag*, there is much more balance between the transition and transversion mutation rates. When looking at the *GTR* mutation rates, we see that the transversion rates are similar, though $C \longleftrightarrow G$ is a bit lower than the others. The

transition rates are also quite balanced between purine transitions and pyrimidine transitions. We do however, have to make a deterministic assumption to allow these simulations to be computationally feasible. We do this by assuming the general mutation rate of 2.8×10^{-9} will cause a rate of 2.8×10^{-4} mutations per nucleotide per 10^5 generations. Although this is a large set of generations, this assumption allows us to not only run the simulations, but also allows us to normalize the mutation rates across genes.

When looking at the results of the simulations for the different models for *Drosophila White*, we see that the average number of generations to reach the maximum genetic distance stayed relatively constant when compared to the results of *HIV* Gag. This is due to the fact that the transition and transversion mutation rates are much more similar than in the previous gene, as well as the nucleotide distribution being much more uniform. *K2P* and *HKY85* thus come close to approximating the *JC* model, as they should when these 2 conditions are met. The *GTR* model is also very similar to the other three models, which also makes sense given that the purine transitions and pyrimidine transitions have about the same rate, and all four transversion types have similar rates. If the two types of transitions had the same mutation rates, and the four transversion rates were the same, then *GTR* reduces to *HKY85*. Thus it makes sense that it is not far off. Generally, *Drosophila White* provides a great example of the evolutionary models reducing when certain conditions are met.

It is worth noting that as degrees of freedom are added to the models, the more inconsistent the results of the models become, though slowly.. There seems to be a tradeoff between consistent results and timescale.

3. *Yeast YML093W* Analysis

The evolutionary simulations for *Yeast YML093W* were interesting because the parameters for the simulations were somewhat between those of the previous two gene simulations. The nucleotide distribution of the yeast gene was similar to that of the *HIV Gag* gene, with purines accounting for 62.8% of the nucleotides and pyrimidines accounting for 37.2%. The general mutation rate, when normalized for 10^7 generations, is quite close to the general mutation rate of *HIV Gag*. The mutation rate due to transitions is similar to the mutation

rate due to transversions, an aspect that is similar to the ratio of mutation rates in the *Drosophila White* gene.

In running the evolutionary model simulations for Yeast *YML093W*, we can see where certain models start to break down and become very inconsistent, as well as an example of *K2P* almost perfectly reducing to the *JC* model. The results of the simulations for *JC* and *K2P* are almost indistinguishable, which is not surprising given the near 1:1 ratio between transition and mutation rates. Interestingly, when adding in the nucleotide frequencies in the *HKY85* model, there is not much consistency in the results between simulations. Of any of the genes studied, *YML093W* has the greatest imbalance of nucleotide frequencies, which we would have anticipated providing the model an advantage. Though on average it takes longer to reach maximum genetic distance, it is wildly inconsistent. Some simulations took over 40,000 normalized ($\times\ 10^7$) generations to reach maximum genetic distance, while others took under 4,000. This resulted in a very high standard deviation and poor confidence intervals when compared to *JC* and *K2P*. *GTR* had very similar results to *HKY85*, which is interesting because like Drosophila White, the two types of transitions have similar rates and the four types of transversion types have similar rates.

4. General Analysis

Overall the simulations provided a lot of interesting information about how these genetic evolutionary models work, and how they are affected by different mutation rates and nucleotide distributions. For instance, we see how *HKY85* produces strikingly similar results to *K2P* when the nucleotide distribution is relatively uniform, which the math supports, as shown by the *Drosophila White* simulations. Comparatively, it is shown how taking into account biological aspects like transition versus transversion rates can increase the timescale while also increasing consistency, such as in *HIV Gag*. We also see how with relatively similar normalized mutation rates, two gene's with similar general mutation rates, like *HIV Gag* and *YML093W*, have very similar *JC* results. Generally the *GTR* model has the longest average generations to reach maximum genetic distance, but this can come at the cost of consistent results, and may not provide much advantage to other models, especially when mutation rates do not differ much within the transitions and transversion types.

Discussion

1. What could have been done differently

Note that this research does not take into account the effect of natural selection on the models. This causes the models to suggest that a mutant allele can arise within a population and reach fixation by chance, rather by selective advantage. The results could've turned out differently if the mutation is based on the organism's ability to survive and reproduce rather than mutation and genetic drift.

The pool of sequences being used is relatively small. Insertions and deletions are not included since it will drastically increase the complexity of the procedure. Also, there are other common models that we did not take into consideration, such as:

- The *K3P* model has distinct rates for transitions and two distinct types of transversions. The two transversion types are those that conserve the weak/strong properties of the nucleotides (A ⟷ T and C ⟷ G) and those that conserve the amino/keto properties of the nucleotides (A ⟷ C and G ⟷ T)
- The *F81* model, which is an extension of the JC model in which base frequencies are allowed to vary from 0.25
- The *T92* model is a mathematical method developed to estimate the number of nucleotide substitutions per site between two DNA sequences, by extending the K2P method to the case where a G + C content bias exists
- The *TN93* model distinguishes between the two different types of transition (A ⟷ G) is allowed to have a different rate to (C ⟷ T). Transversions are all assumed to occur at the same rate, but that rate is allowed to be different from both of the rates for transitions

2. Limitations of models

The primary limitation of our model is in how infrequent mutations actually occur. The incredibly low order of magnitudes they sometimes occur in requires the process to be sped up by accounting for multiple generations in a single run of our code, increasing the mutation rate by a corresponding amount. One of the other limitations of our models is that they don't reasonably measure the impact of each mutation. For biological accuracy, we would need a model to account for their effects, and give them metaphorical weight to better our research. In doing so, our program would have to

anticipate if our mutation is occurring in an intron or an exon, and know that the expected outcome of such a mutation to be beneficial, detrimental, or neutral, which would require more context than our program can offer, and thus greatly increase the runtime of execution. Our models also do not account for mutations in specific locations being more frequent than others, although improving this would be practically impossible to fully accurately reflect biology, as the implications are such that each individual nucleotide would need its own *GTR* model, and that level of detail and strive for accuracy is unlikely to ever be attempted. If we were to also include indels in our mutation model, we would also need to scan for the new location of introns and exons, which would greatly increase the computational runtime for each indel that occurs.

3. **How other biological factors could have changed the models**

Mutations are not strictly limited to single nucleotides. Our models do not account for nucleotides surrounding the corresponding mutant, despite those often playing a role in the frequency of mutations in specific locations on the gene. If we would account for this, our model would need to adjust how mutations get allocated based on a buffer of prior nucleotides. Because multicellular eukaryotic organisms can have mutations occur at differing levels, as mentioned under the research section of this paper, we may want to have our model check if our user is attempting to find mutation at RNA or DNA level, and adjust parameters to match the user's request. For this, we would need to have a dictionary of amino acids, and all possible codons that account for said amino acids, in the case that our user, or even our own program wants amino acid data from a nucleotide sequence, as would be the case if we wanted to append our research with natural selection and have our program roughly calculate the impact of mutations as it executes.