##########BUS APRIL 2023 SCRIPT/WHAT TO SAY##############

######1######

Hello Everyone, my name is Alexander Turco and I am a fourth year student in Brians lab.

Im going to be talking about how I attempted to create a program that could simulate the evolution of Low Complexity Regions in Proteins, and further how we could use this program in an Approximate Bayesian Computation

######2######

Before we begin heres just a brief framework of some things I'm gonna touch on today.

######3######

First things first what is a low complexity region?

Low complexity regions are segments of protein or DNA sequences that are biased in their composition.

- More formally LCRs are defined by their low information content and low entropy

- You can see a value called complexity and this value is calculated using Shannon's Entropy Equation which measures the disorder or randomness of a system

- The higher the entropy of a sequence, the more complex/random that sequence is! (equal proportions of each nucleotide)

######4######

- It is also important to mention that although LCRs share an overall low diversity of residues, there are many ways they present this low diversity

- sometimes they follow ordered patterns like homorepeats and direpeats, and sometimes they follow no patterns at all, but just exist as compositionally bias sequences

######5######

- another characteristic of Low complexity regions is that they have been found to be hypermutable - meaning over evolutionary time, it is thought that they frequently gained and lost repeats

- Just to provide a little evidence for this I have a Drosophila Study to share (I know we have a lot of Drosophila lovers in the room so I thought this would be fitting lol).

- Researchers performed an interspecific comparison between a gene in Drosophila Melanogaster and Drosophila Virilis which are believed to have separated approximately 60 million years ago (enough time for nonessential sequences to diverge).

- Part of the gene they looked at, the mastermind gene, is highly conserved but extensive variation is noted in homopolymer domains which flank the conserved region.

- The researchers characterized these into unique and repetitive regions and discovered that repetitive regions are characterized by almost a threefold higher rate of amino acid replacement.

######6######

- There are two mechanisms that have been proposed which try to explain the evolution of Low complexity regions/repetitive regions.

- The first being polymerase Slippage

- During replication, what can sometimes happen is that the DNA polymerase and the newly synthesized DNA strand complex can temporarily dissociate from the template DNA and depending on when this complex rejoins and begins transcribing again, we can get insertions and deletions of repetitive segments.

- Slipping forward means the polymerase skips one of the template repeats resulting in the deletion of the repeat in the newly synthesized strand.

- Slipping backwards means the polymerase copies an extra repeat into the newly synthesized DNA strand

######7######

- The second Proposed Mechanism of LCR Evolution is unequal recombination

- On the slide is an illustration of unequal crossing over between repetitive runs on homologous chromosomes during meiosis.

- When unequal crossing over occurs it can result in the reciprocal appearance of expanded and contracted repeats.

######8######

- But why should we care about LCRs and the way they evolve anyways?

- LCRs are implicated in a wide variety of diseases as well as Biological processes

- LCRs can appear as trinucleotide repeats which form repeated units of 3 nucleotides and are found to be associated with several human neeurodegenerative diseases including Huntington's disease, fragile X syndrome and muscular dystrophy.

- LCRs have also been found to be associated with important biological processes such as genetic recombination and this is actually evident through studying genomes of bacteria like neisseria meningitidis which causes meningococcal disease.

- This bacteria is abundant in LCRs and the LCRs are thought to drive phase variation which can give the bacteria the ability to change their adherance patterns to host cells (helps with survival of pathogens)

######9######

- Just to give a brief overview of some of the things we did in the study

- We used c++ to build an evolution simulator that would alter protein sequences via point mutations, insertions and deletions

- We tested this simulator with various insertion/deletion rates and mutation rates

- Attempted to program an ABC-MCMC in c++ using the evolution simulator to estimate evolutionary parameters like mutation and indel rates.

######10######

- Okay so first things first, were gonna run through how the simulator works

- First, we set our mutation and indel rates

- Then a random protein sequence of specified length is generated to begin the process

######11######

- In the next step we iterate over each amino acid in the protein sequence and assign exponential deviates to each residue based on mutation and indel rates

- Basically we store two vectors of exponential deviates, one for deviates generated using mutation rate as the scale parameter for the exponential distribution, and one for deviates generated using the indel rate as the scale parameter for the exponential distribution

- In the case of using the indel rate, the program scans around each residue and counts how many repeats of the same amino acid are present, and multiplies the length of the repeat the amino acid is contained in, by the indel rate.

- For example the first 2 G’s in the sequence would have their deviates generated using indel rate \* 2 because they are in a two-unit repeat.

######12######

- Now in order to determine if a mutation, insertion or deletion takes place, we select the amino acid with the lowest value exponential deviate (since we used the exponential distribution this means the amino acid that would be altered the quickest)

- If the lowest value deviate was found in the vector where deviates were generated using mutation rates, then a point mutation occurs, if it was found in the vector where deviates were generated using indel rates, then an insertion/deletion occurs. (GET HOW THIS INSERTION/DELETION OCCURS, PROBABILITY 0.5 OR SOMETHING)

######13######

- Finally, once we’ve made the appropriate change, the program then checks to see if the change that was made affected the landscape of the sequence – So if a mutation interrupted a repeat, or formed a repeat or something like that, the amino acids that were affected by the change get new deviates generated and the process goes again – FOR HOW LONG (iter <0.5 ask brian)

######14######

- To show the results from the simulation, I have plotted the sequence entropy (BLUE), Avg entropy of the LCRs(BLACK), and the number of LCRs (RED) over 1000 iterations of the simulation.

- In these first two graphs, the mutation rate is held constant at 1, but the indel rate is changed from 0.1 to 0.5.

- When the indel rate is extremely low (smaller than mutation rate), we see few LCR’s being formed, and the average sequence entropy does not decrease, although it does fluctuate more when the indel rate is 0.5

######15######

- We can compare those results to these, where we set the indel rate to higher values, 1, 2, and 10, and still keep the mutation rate constant at 1

- When the indel rate becomes larger (larger than mutation rate), we see a greater number of LCR’s being formed, and the overall sequence entropy decreases more due to the formation of more repetitive regions. – THIS IS WHAT WE WANT TO SEE, LOWER INDEL RATE GIVES LESS INDELS, HIGHER INDEL RATE GIVES MORE INDELS

######16######

- I also explored what would happen if we held the indel rate constant at 1, and changed the mutation rate.

- We kind of get the opposite pattern in this case where for lower mutation rates, we seem to have a larger number of LCR’s being formed and the sequence entropy seems to be affected.

######17######

- You can see that as we get to mutation rates of 2, and 10, we start to form less and less LCRs, and the sequence entropy remains more or less steady, indicating that mutation is not allowing for the creation of LCRs, and when they are formed they are destroyed by mutation as well.

######18######

- Before we move on to how we can apply this to an Approximate Bayesian Computation, it is important to understand the main principles of general model based inference first

- Any model based statistical inference generally requires DATA, A MODEL, and A SET OF MODEL PARAMETERS and the task often revolves around calculating the likelihood function

- It represents the probability of the observed data under a chosen model. It basically quantifies how well the data supports the parameter values and the model

######19######

- This likelihood is then used as a step towards calculating the posterior distribution in the Bayesian Paradigm.

- Posterior distributions are important because they will be what provides the interval estimates for parameter values which describe the uncertainty about the parameter.

######20######

- But we are at a point in time where both the complexity and magnitude of available data are increasing

- Because of this increase, many current model-based analyses have become intractable because calculation of the likelihood is too difficult

- In the case of my study, insertions and deletions alter the landscape of a sequence and make the calculation of the likelihood much too difficult

######21######

- Thats why in this Study we are going to use an Approximate Bayesian Computation.

- An approximate bayesian computation is an analysis rooted in Bayesian Statistics (so what I've mentioned before applies to this analysis as well), but what differs is that the likelihood calculation is replaced by a simulation step which estimates the likelihood function

- ABC methods rely on the comparison between real, observed data and simulated data in order to make this possible.

######22######

- So shown here is the general method/algorithm for an ABC MCMC which is the type of ABC we are utilizing. (taken from Marjoram 2003 where he first called the paper an MCMC method without the use of likelihoods)

- It starts with a selected parameter value and proposes a move to a new parameter value based on a proposal distribution

- Then using this new parameter value, we generate a simulated dataset and calculate summary statistics

- Next we assess how similar the summary statistics between the observed data and simulated data are. If the summary statistics are very similar, we accept the proposed parameter value, otherwise we reject it.

- the hastings ratio is then calculated and the proposed parameter can be accepted with a probability equal to the value of the hastings ratio. THen we return to step 1 and continue proposing and rejecting parameters for a desired number of times

######23######

So now I'm gonna go through the same algorithm, but this time explaining it in the context of my study.

- First things first we propose a move from one parameter value to another according to the normal distribution

- Step 2 (WHERE SIMULATOR COMES IN) - Next we generate a random protein sequence and use the new parameter value to simulate the mutation of the random protein sequence over a number of generations (1000).

- Once we mutate the simulated protein, we then calculate our summary statistics - right now we have a vector of 3 summary statistics - they are length of the protein, number of LCRs, and the average entropy of the LCRs

- I am also going to mention that the same summary statistics are used for the observed protein

- Using the summary statistics we calculated for the simulated and observed data, we calculate the euclidean distance between the vectors of summary statistics and if this distance is smaller than the previous calculated distance, we accept the newly proposed parameter value and move on. If this new distance is larger than the previous, we employ a one sample t-test to assess the likelihood of accepting that larger distance.

- When we accept the proposed parameter, it essentially means we use the parameter in calculating the posterior distribution

- We then return to step 1 and iterate through this for a desired number of simulations.

######24######

- ABC Prelim results – distance plots

- Although the overall program for the ABC is still in beta,

- I did want to show these graphs just to highlight why were using 1000 iterations of the simulator for each newly proposed parameter value

- These graphs are showing distances between the mutated protein sequence and the observed protein sequence (SRP40) over each iteration of the evolution simulator. – basically for each newly proposed parameter value, we do this (mutate under same parameters 1000 times).

- When mutation rate is 0 and indel rate is 1, the distance begins to level off at about 0.5, and when the mutation rate and indel rate are both 1, the distance begins to level off at about 0.8

- with 1000 iterations, we can be sure that we are getting summary statistics that are actually representative of the mutation simulation process

######25######

- Conclusions, summarize ur shit

- So just to conclude, in this study we created and tested a program to simulate the evolution of low complexity regions based off of two evolutionary parameters, mutation and indel rates.

- This simulation program is compatible within the program I wrote for an ABC-MCMC

- I have been struggling to get the ABC fully working.

- This could be due in part to selection of poor summary statistics or a lack of weightings being applied to these summary statistics which I have not done

######26######

- Future Work, Don’t know if I have time for this, took this one out

######27######

- Acknowledgements – thank you to all these people.

- Collaboration and mentorship have been critical throughout this process so far and so I would just like to thank these individuals because I could not have done anything without them.

EXTRA NOTES IM NOT PUTTING ON SLIDES

* The SRP40 protein is serine rich and plays a role in preribosome assembly and transport and we will be mentioning it later as well.
* ######Shannon's Entropy######
* - L is the number of elements, pi is the probability of occurrence, and H is the units which are measured in bits (bits because log base 2)
* - This equation measures the disorder or randomness in a system and is commonly used to search for patterns and complexity in both DNA and protein sequences
* - Just for better understanding, I show here two DNA sequences. One consists of all the same nucleotide (it is of minimal complexity) and once consists of each nucleotide being represented equally (maximal complexity)
* - The sequence made up of all the same nucleotide has an entropy of 0 (minimum for DNA sequence) and the sequence made up of equal frequencies of each nucleotide has an entropy of 2 (maximum entropy for DNA sequence)
* - All the examples I am showing in the next few slides are different forms of LCRs based on amino acid periodicity but it is important to understand that not all LCRs contain patterns like these but rather they deviate from a random composition
* - The number of repeated segments within a microsatellite sequence differs among individuals which makes these sequences interesting. This difference in the number of repeated segments makes microsatellites useful as polymorphic markers for studying inheritence patterns in families, or for creating DNA fingerprints from crime scene samples.
* The simplest way to put it, we want a randomly generated protein sequence to be mutated over and over again under certain parameter values. Then we want to compare the simulated protein to a protein which we know the sequence for (in my case I chose SRP40 - I think I mentioned this above, has lots of repeats) and assess the difference in values of their summary statistics.
* - To a Byaesian statistician, the posterior disribution is the complete answer the question, what is the value of the parameter?
* - Replacing the likelihood function with a simulation step is beneficial in areas such as the inference of genetic networks because as the complexity of the network grows, we are still able to perform a simulation where we would normally struggle to calculate the likelihood as complexity increased.
* PHASE VARIATION = (phenotypic changes resulting from genetic alterations)
* - the summary statistics make it possible to quantitatively compare differences between the dataset we simulated and the observed dataset (real data)