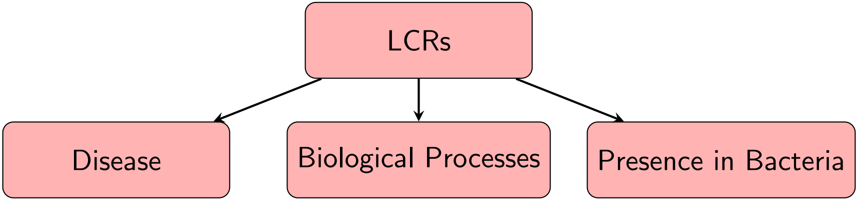
Creating an Evolution Simulator for Protein Low Complexity Regions and Attempting to Utilize it for an Approximate Bayesian Computation

Alexander Turco

April 13, 2023

McMaster University
SCIENCE
Department of Biology

Why Care about Low Complexity Regions and their Evolution?



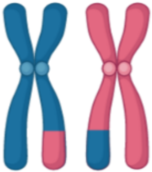
LCRs

Disease

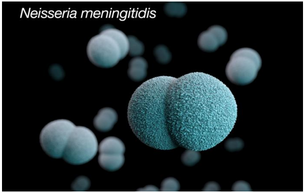
Biological Processes

Presence in Bacteria

Huntington's Disease



Genetic Recombination



Neisseria meningitidis

What are Low Complexity Regions?

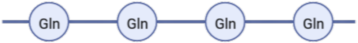
Saccharomyces cerevisiae SRP40 Protein LCRs

```
>CAA82171.1(25-125) complexity=0.92 (15/1.90/2.20)
ssssssssssssssssssssssssgessssssssssssssdssdssdsesssssssssss
ssssssdsesssesdssssgsssssssssdesssesesede

>CAA82171.1(149-282) complexity=1.33 (15/1.90/2.20)
esssssessssgsssssesesgsesdsdsssssssssdsesdsesdsqssssssssdsss
dsdssssdsssdsdssssssssssdsdsdsdsssdsdssgssdsssssdsssdestssds
sdsdsdsdsgssse

>CAA82171.1(298-316) complexity=2.18 (15/1.90/2.20)
tpassnestpsasssssan
```

LCRs Present in Unique Ways

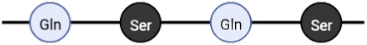
Homorepeats

Consecutive iterations of a single residue



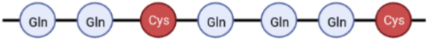
Direpeats

Consecutive iterations of two ordered, different residues

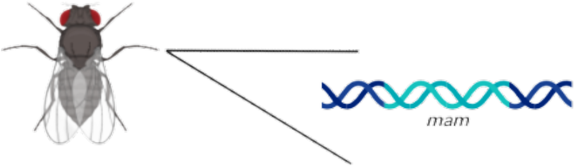


Imperfect Repeats

Regions in which the repeat units are not the same



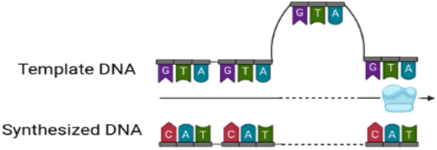
LCRs are Hypermutable



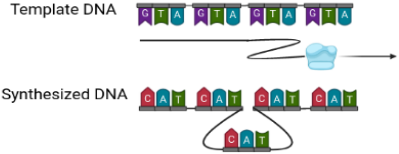
mam domain	Size (bp)	Amino Acid Substitutions	Amino Acid/ Total Substitutions
Unique	933	26	0.15
Repetitive	810	47	0.42

Proposed Mechanisms of LCR Evolution

1. *Polymerase Slippage/Slipped Strand Mispairing*



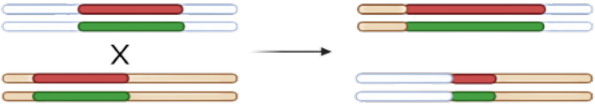
Polymerase Slips Forward



Polymerase Slips Backwards

Proposed Mechanisms of LCR Evolution

2. *Unequal Recombination*



What we Did in this Study

- Utilized C++ to build an evolution simulator which altered protein sequences via point mutations, insertions, and deletions
- Tested the simulator with various insertion/deletion rates and mutation rates
- Attempted to program an approximate bayesian computation markov chain monte carlo algorithm (ABC-MCMC) in C++ using the evolution simulator as an important step in the algorithm, in order to estimate parameters like mutation and indel rates.

LCR Simulator Overall Process

1. mutation rate = 0.14 indel rate = 0.14
   Random Protein Sequence
   GGAGGGAQ

LCR Simulator Overall Process

1. mutation rate = 0.14 indel rate = 0.14
   Random Protein Sequence
   GGAGGGAQ

2. Assign Exponential Deviates
   mutation deviates = (0.83, 1.82, 2.35, 0.54, 0.98, 0.76, 1.53, 2.34)
   indel deviates = (0.21, 1.21, 1.49, 0.86, 0.97, 1.13, 0.53, 0.35)
   exp(β), where β = mutation rate
   exp(β), where β = length of repeat * indel rate

LCR Simulator Overall Process

1. mutation rate $=$ 0.14 indel rate $=$ 0.14
   Random Protein Sequence
   GGAGGGAQ

2. Assign Exponential Deviates
   mutation deviates $=$ (0.83, 1.82, 2.35, 0.54, 0.98, 0.76, 1.53, 2.34)
   indel deviates $=$ (0.21, 1.21, 1.49, 0.86, 0.97, 1.13, 0.53, 0.35)
   exp(β), where β $=$ mutation rate
   exp(β), where β $=$ length of repeat * indel rate

3. Point Mutation, Insertion, or Deletion
   Lowest value deviate $=$ Residue that mutates fastest

LCR Simulator Overall Process

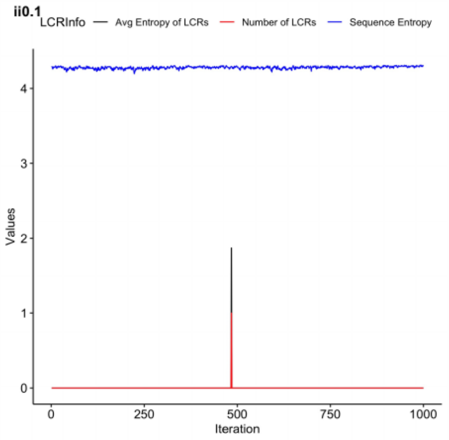
1. mutation rate = 0.14 indel rate = 0.14
   Random Protein Sequence
   GGAGGGAQ

2. Assign Exponential Deviates
   mutation deviates = (0.83, 1.82, 2.35, 0.54, 0.98, 0.76, 1.53, 2.34)
   indel deviates = (0.21, 1.21, 1.49, 0.86, 0.97, 1.13, 0.53, 0.35)
   exp(β), where β = mutation rate
   exp(β), where β = length of repeat * indel rate

3. Point Mutation, Insertion, or Deletion
   Lowest value deviate = Residue that mutates fastest

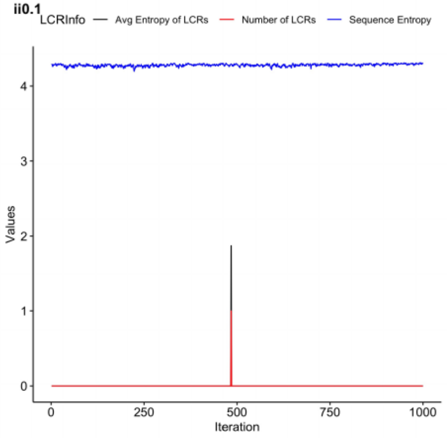
4. Upon point mutation, insertion, or deletion, scan the sequence again to see if the landscape of the sequence was affected, assign new deviates to affected amino acids only.

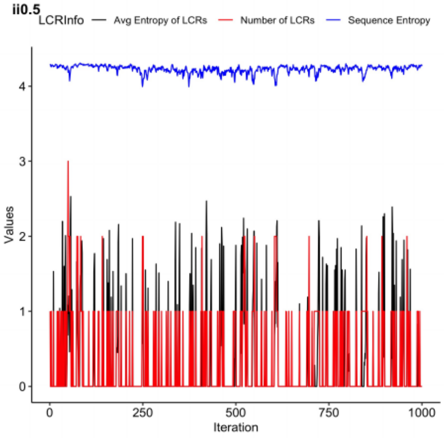
Lower Indel Rates Produce Less LCRs



Indel Rate = 0.1 Mutation Rate = 1

Lower Indel Rates Produce Less LCRs

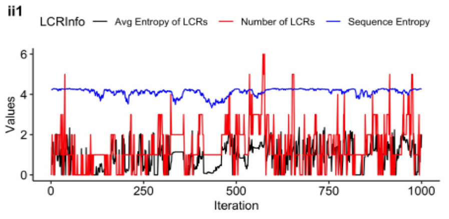




Indel Rate = 0.1 Mutation Rate = 1

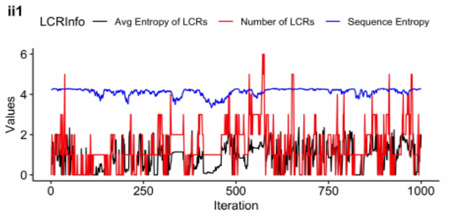
Indel Rate = 0.5 Mutation Rate = 1

Higher Indel Rates Produce More LCRs

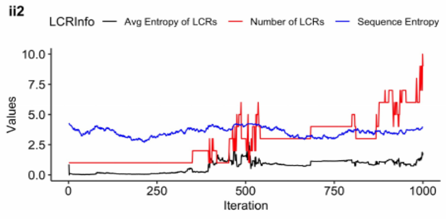


Indel Rate = 1 Mutation Rate = 1

Higher Indel Rates Produce More LCRs

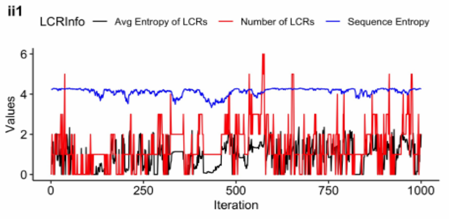


Indel Rate = 1 Mutation Rate = 1

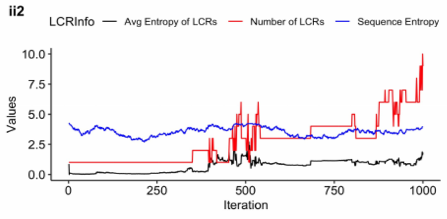


Indel Rate = 2 Mutation Rate = 1

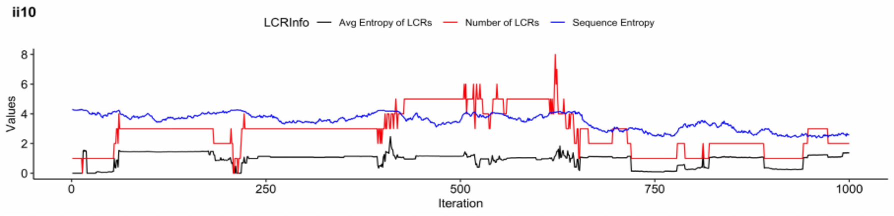
Higher Indel Rates Produce More LCRs



Indel Rate = 1 Mutation Rate = 1

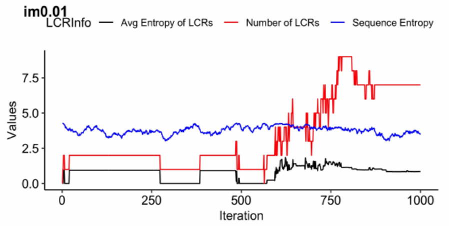


Indel Rate = 2 Mutation Rate = 1



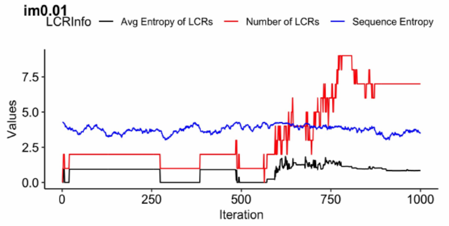
Indel Rate = 10 Mutation Rate = 1

Lower Mutation Rates Produce More LCRs

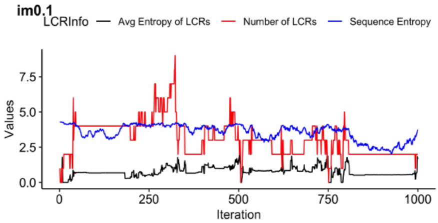


Indel Rate = 1 Mutation Rate = 0.01

Lower Mutation Rates Produce More LCRs

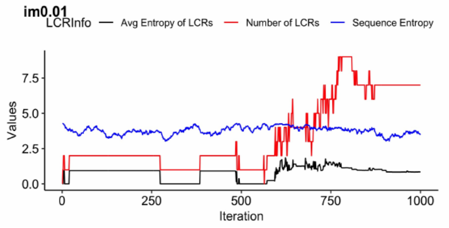


Indel Rate = 1 Mutation Rate = 0.01

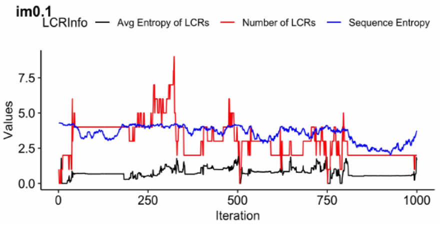


Indel Rate = 1 Mutation Rate = 0.1

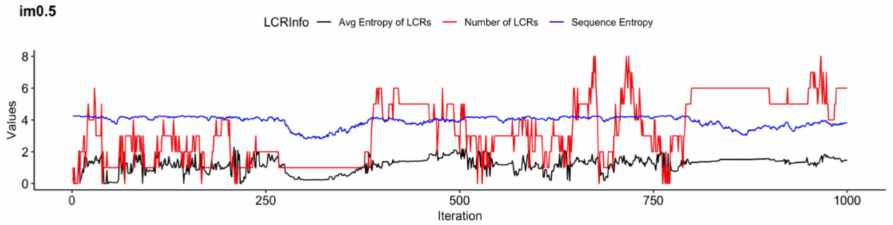
Lower Mutation Rates Produce More LCRs



Indel Rate = 1 Mutation Rate = 0.01

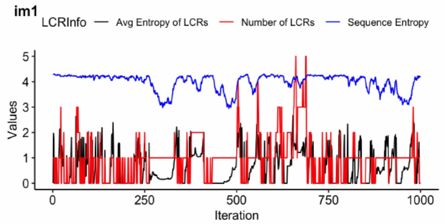


Indel Rate = 1 Mutation Rate = 0.1



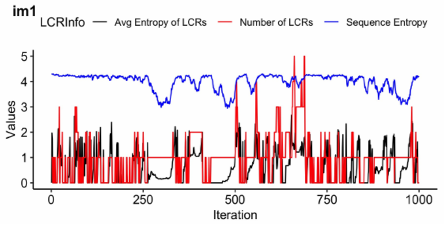
Indel Rate = 1 Mutation Rate = 0.5

Large Mutation Rates Prevent the Formation of LCRs

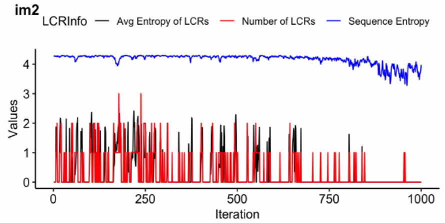


Indel Rate = 1 Mutation Rate = 1

Large Mutation Rates Prevent the Formation of LCRs

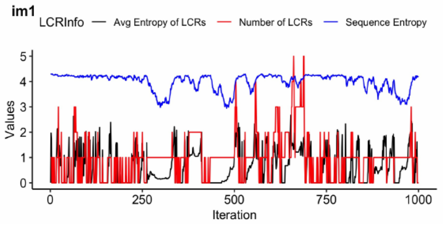


Indel Rate = 1 Mutation Rate = 1

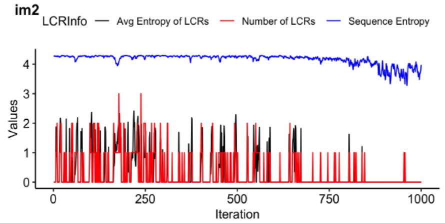


Indel Rate = 1 Mutation Rate = 2

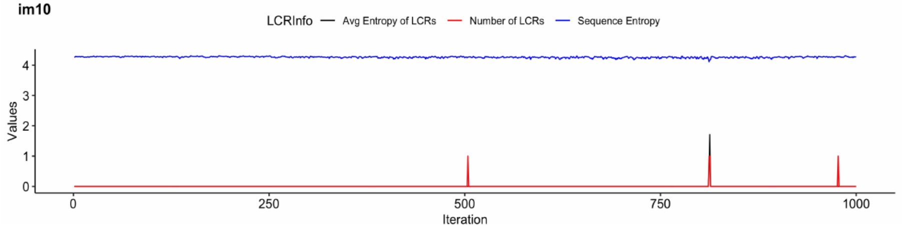
Large Mutation Rates Prevent the Formation of LCRs



Indel Rate = 1 Mutation Rate = 1



Indel Rate = 1 Mutation Rate = 2



Indel Rate = 1 Mutation Rate = 10

Bayesian Statistics Overview

Bayesian Statistics: Model-based statistical inference

$$p(D|\theta) \tag{1}$$

Likelihood

Bayesian Statistics Overview

Bayesian Statistics: Model-based statistical inference

$$p(D|\theta) \tag{1}$$

Likelihood

$$p(\theta|D) \tag{2}$$

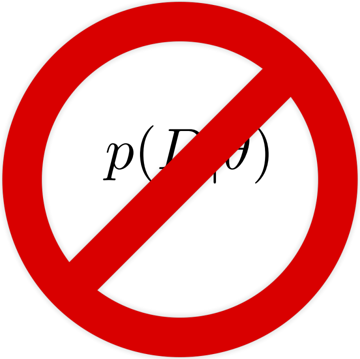
Posterior

Why use an ABC-MCMC

- The increasing complexity and magnitude of available data can make the likelihood difficult to calculate

$$p(D|\theta)$$

Why use an ABC-MCMC



- Calculation of the likelihood is replaced with a simulation step

MCMC for ABC

1. Propose a move from θ to θ' according to a transition kernel $q(\theta, \theta')$.
2. Generate simulated dataset D' using θ' and calculate S'.
3. If $\rho(S', S) \leq \epsilon$ continue to 4, otherwise remain at θ and go to 1.
4. Calculate

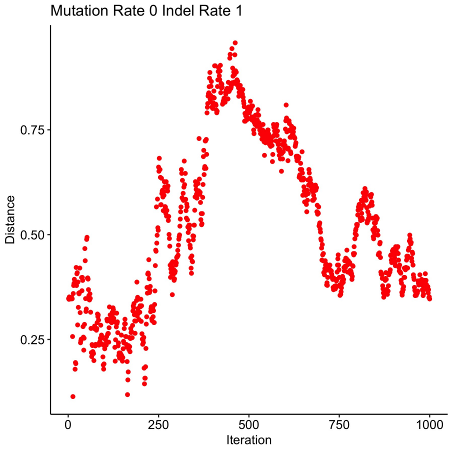
$$\alpha(\theta, \theta') = min(1, \tfrac{\pi(\theta')q(\theta',\theta)}{\pi(\theta)q(\theta,\theta')})$$

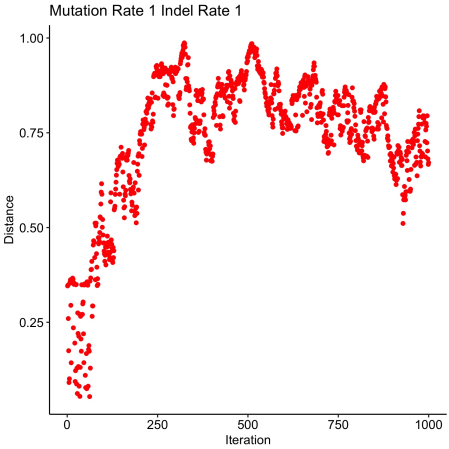
5. Accept θ' with probability α, otherwise stay at θ.
6. Return to 1.

MCMC for ABC: Modified Algorithm

1. Propose a move from θ to θ' according to the normal distribution $N(0.0, 1.0)$
2. Create and mutate a random protein sequence using θ' to generate the simulated Dataset D' - do this 1000 times per newly proposed parameter value.
3. Calculate summary statistics for simulated dataset D' (average of all 1000 vectors of summary statistics)
4. If $d(S', S) < previous\_distance$, go to next step, otherwise employ a one-sample t-test to assess the probability of accepting a larger distance. If the newly proposed distance is close to the previous distance, there is a higher chance we accept the value, otherwise we reject.
5. Accept θ'
6. Return to step 1

ABC MCMC Preliminary Results





Conclusions

- Created and tested a program to simulate the evolution of low complexity regions based off of two evolutionary parameters, mutation and indel rates
- The simulation program is compatible in a program written for an ABC-MCMC
- Struggling with creating posterior distribution, potentially due to selection of poor summary statistics or a lack of weight placed on each statistic

Acknowledgements

- Dr. Brian Golding
- Sam Long
- Zachery Dickson
- Johanna Enright