

Code Breaking with MCMC



Shlok Mishra

Under the guidance of
Prof Dootika Vats

AGENDA

Cryptography

Substitution Cipher

Markov Chain

Monte Carlo

Markov Chain Monte Carlo

Text Analysis

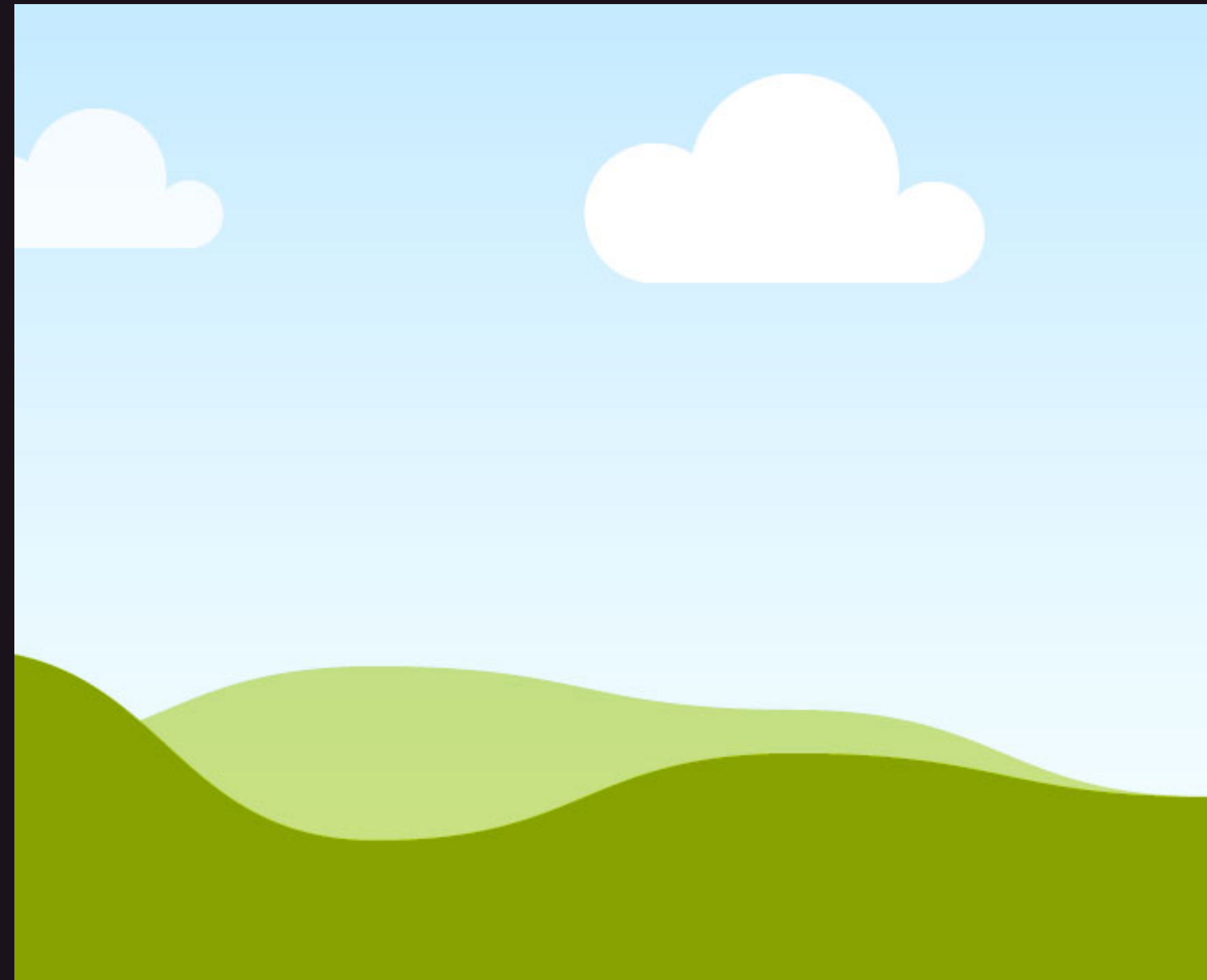
Metropolis Hastings Algorithm

Disclaimer

I have not come up with some cutting edge tool able to solve real-world problems

Just the fact that we can build something using **cryptography**, **text analysis**, and **MCMC** is itself of interest

Let's explore how!



What is Cryptography?

The practice and study of techniques for secure communication in the presence of adversarial behavior

ENCRYPTION

The process by which a readable message is converted to an unreadable form to prevent unauthorized parties from reading it

DECRYPTION

The process of converting an encrypted message back to its original (readable) format.

ENCRYPTION

PLAIN TEXT MESSAGE



CIPHER TEXT MESSAGE



"the anti-ragging policy at iitk has been enforced
pretty well"

ENCRYPTION

"ifm uliz-puvvzlv abrzqj ui zzin fus wmml mlebpqmh
apmiiij dmrr"

DECRYPTION

CIPHER TEXT MESSAGE



PLAIN TEXT MESSAGE



"ifm uliz-puvvzlv abrzqj ui zzin fus wmml mlebpqmh
apmijj dmrr"

DECRYPTION

"the anti-ragging policy at iitk has been enforced
pretty well"

What is a Cipher?

an algorithm for performing encryption or decryption
a series of well defined steps that can be followed as a procedure

SUBSTITUTION CIPHER

Substitution of single letters
separately.
Just simple substitution!

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
e	u	f	k	r	t	c	b	g	v	s	h	d	x	z	q	p	j	o	a	w	l	y	n	m	i

MCMC



MARKOV

CHAIN

MONTE

CARLO

What is a Markov Chain?

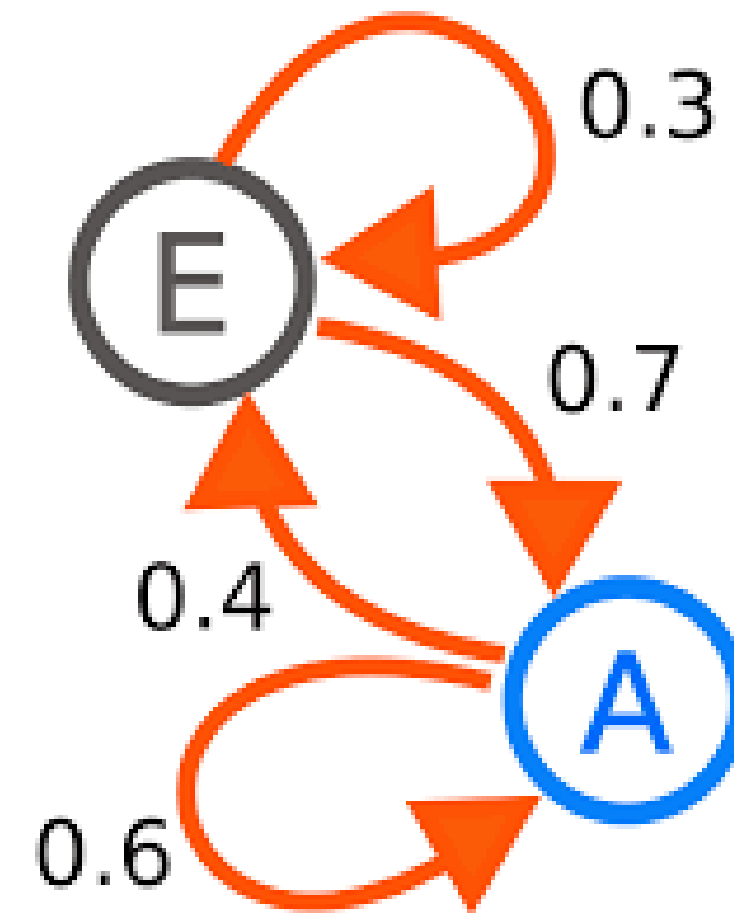
A stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event

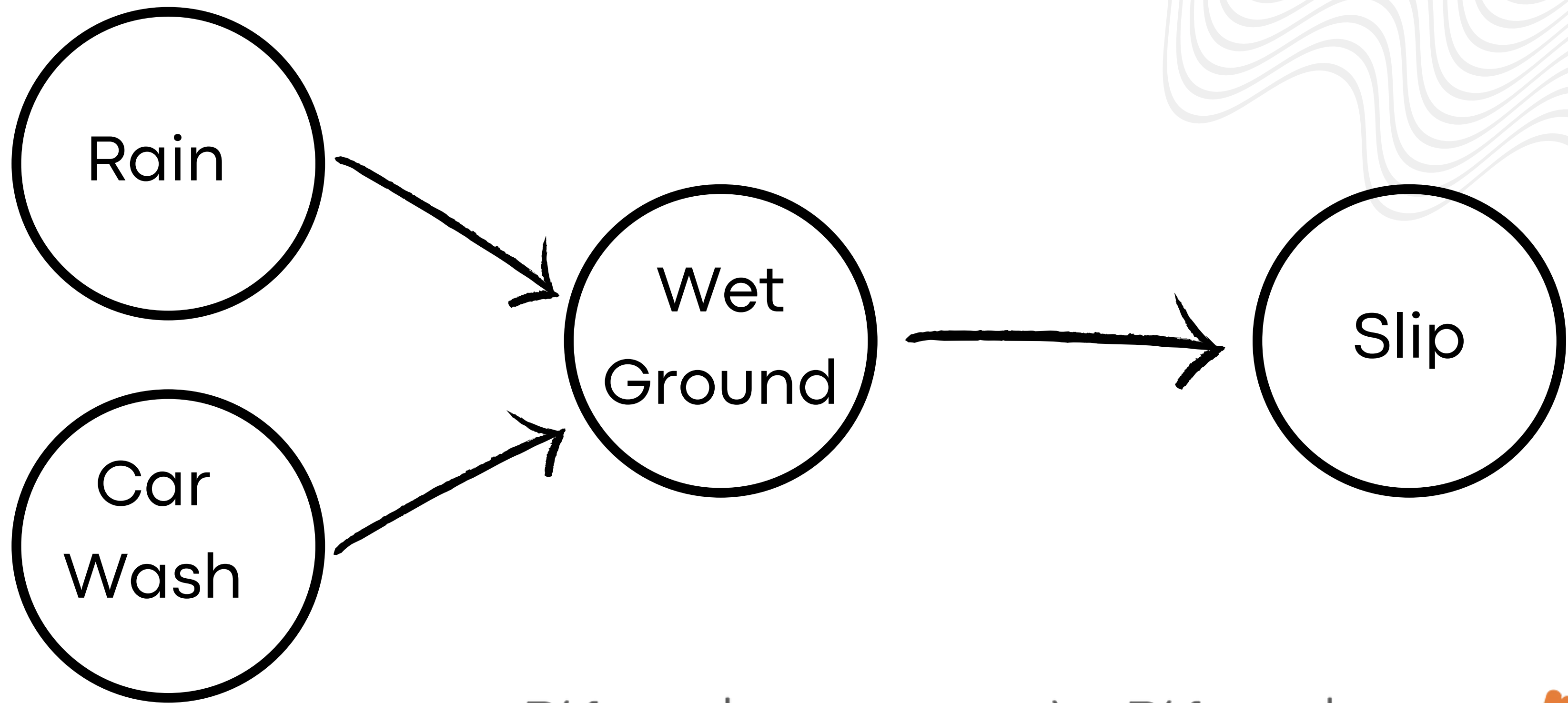
In Simpler Words:

"What happens next depends only on the state of affairs now"

The probability of jumping from one state to the next state depends only on the current state and not on the sequence of previous states that lead to this current state

A simple two-state Markov Process





$$P(\text{future} \mid \text{present, past}) = P(\text{future} \mid \text{present, ~~past~~})$$

Markov property 

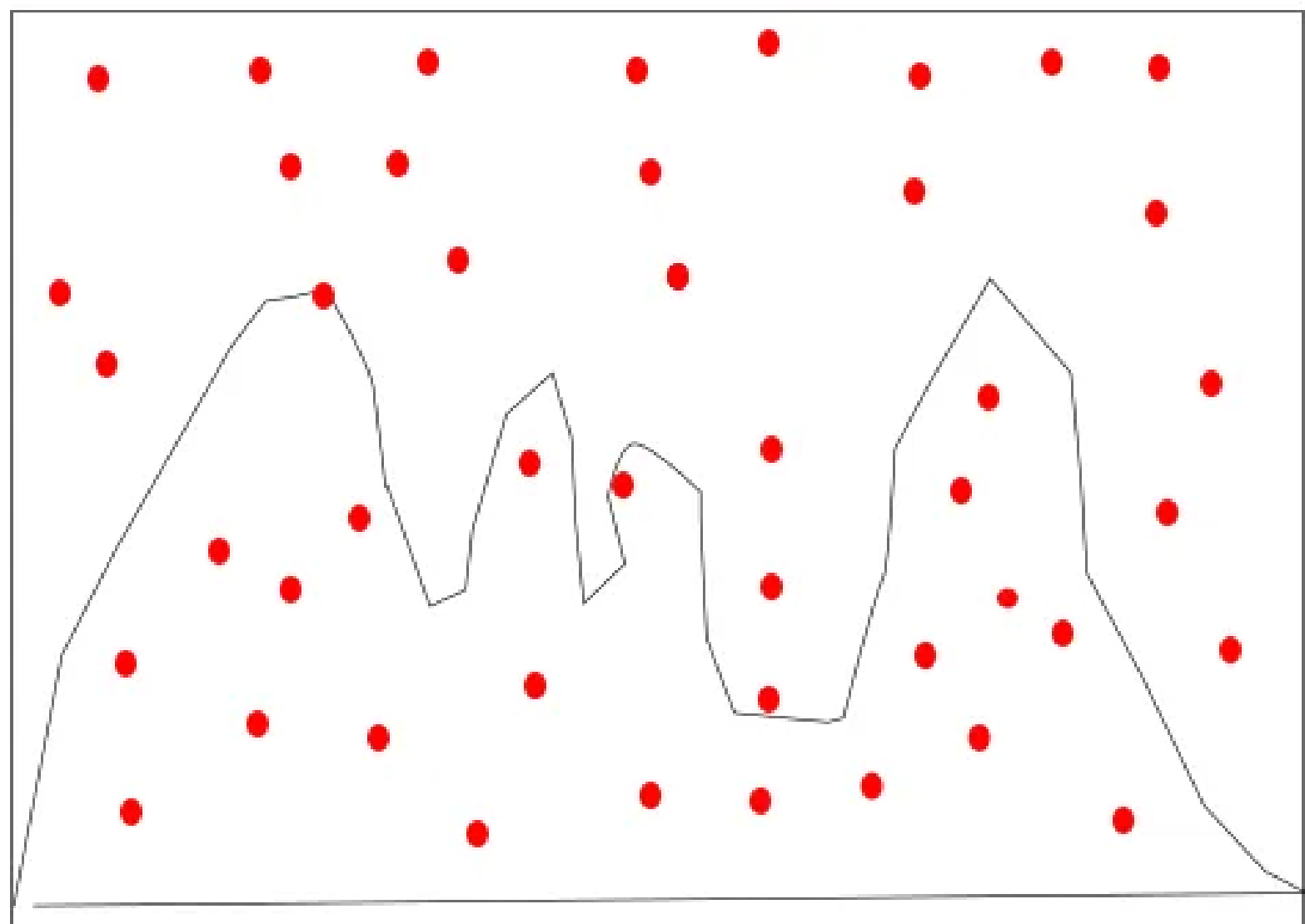
What are Monte Carlo methods?

These are techniques for sampling from a probability distribution and using those samples to approximate desired quantity.

In Simpler Words:

In other words, they use randomness to estimate some deterministic quantity of interest.

Basically, if calculating some quantity has a complex analytical structure, we can simply perform a simulation to generate lots of samples and use them to approximate the quantity.



Area under the curve?

Requires calculating a really
complex integral

Under Monte Carlo methods, find the ratio of red
dots falling under the curve

What is Markov Chain Monte Carlo?

MCMC methods comprise a class of algorithms for sampling from a high dimensional probability distribution

In Simpler Words:

The General Idea for the algorithm is to start with some random probability distribution and gradually move towards desired probability distribution

What exactly is the problem?

Given an input string of an encrypted text we need an algorithm to figure out what it's decrypted form in English actually is.

Trying to develop a decryption algorithm for substitution ciphers.

Why not use Brute Force?

Suppose we intercept the ciphertext, but we don't know the true cipher.

A brute force idea would be to try all possible ciphers until we find the correct one

There are $26!$ ciphers since each cipher is some combination of the alphabets

Let's do some math to check the feasibility of the brute force method.

Time taken by Brute Force?

Let us assume that we have access to the fastest super-computer in the world.

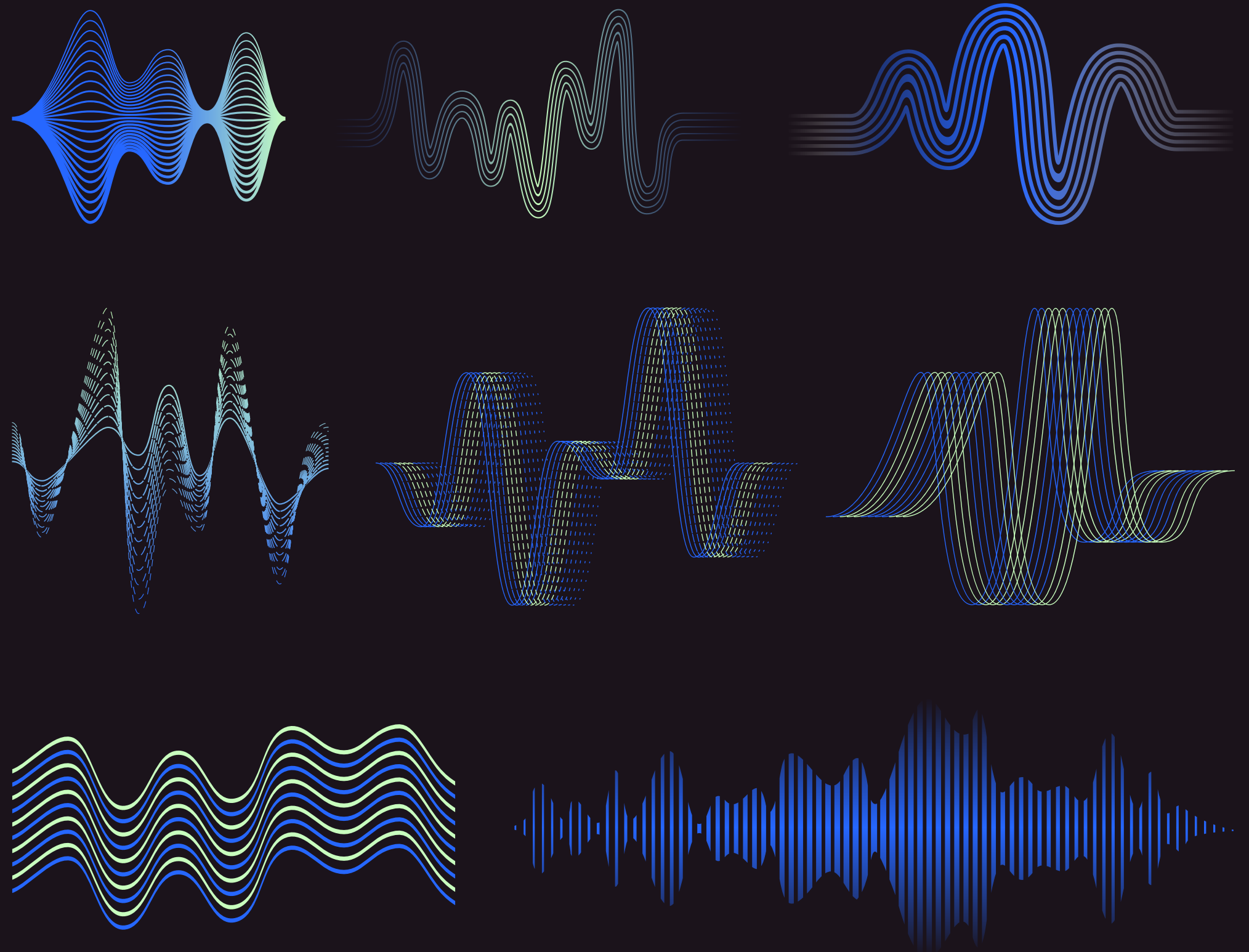
It is supposed to compute **1.1 quintillion** operations per second.

Assuming that each cipher check would take only one operation.

$$4.03 \times 10^{26} \text{ ciphers} \times \frac{1 \text{ second}}{1.1 \times 10^{18} \text{ ciphers}} \approx 366 \text{ million seconds} \approx 11.6 \text{ years}$$

Goal

To be able to come up with a probability distribution in order to deploy an MCMC algorithm



TEXT ANALYSIS

INPUT

Use a cipher to decrypt

OUTPUT

Some kind of score?

Defining an English Similarity Score

"xu rp uo zux xu rp xqwx ke xqp ifpexkuz vqpxqpo xke zurhpo kz
xqp bkzn xu efttpo xqp ehkzae wzn woouve ut ufxowapufe
tuoxfzp uo xu xwlp wobe wawkzex w epw ut xoufrhpe"

Even a non-native English speaker could likely tell that this
doesn't look like English. For example, most words don't have
any vowels.

English *Similarity* Score

How close is any given piece of text to English?



English *Similarity* Score

How close is any given piece of text to English?

"hello there"

"he" "el" "ll" "lo" "o "
" t" "th" "he" "er" "re"

$\text{Freq}(\text{he}) \times \text{Freq}(\text{el}) \times$
 $\dots \times \text{Freq}(\text{re})$

THAT'S OUR SCORE!

How to compute $\text{Freq}(x)$?

An approximation is made by using a very long English text and use the frequencies within it.

I have used War and Peace by Leo Tolstoy, a very long book by most standards. The text file of the book is available in the public domain.

We break the book into two character chunks and construct a probability table.

War and Peace

2 Characters

e	0.033159052
t	0.026464778
he	0.024673473
th	0.024416444
d	0.022819309
a	0.020971399

Since the all the probabilities are going to be really small, their product would easily go beyond $1e-50$.

Hence for numerical stability, we would take logarithm of the frequencies and then add them up.

METROPOLIS ALGORITHM

An MCMC algorithm which is used to generate Markov Chains that converge to a desirable stationary distribution.


In the framework of our problem, the states our Markov Chain is traveling between are the $26!$ possible ciphers

We want the Markov chain to travel to the ciphers that are more “likely to be correct” and stay away from the ciphers that are “unlikely to be correct”

Neighbour of a cipher

Defining the neighbour of any given cipher as
the set of all the ciphers obtained from
swapping any two alphabets

Every cipher will have $\binom{26}{2}$
neighbours



e u f k r t c b g v a h d x z q p j o s w l y n m i

e u f k r t c b g v q h d x z a p j o s w l y n m i

The diagram illustrates a swap between the letters 'a' and 'q' in the alphabet sequence. In the top sequence, 'a' is at index 10 and 'q' is at index 15. In the bottom sequence, 'q' has moved to index 10 and 'a' has moved to index 15. White arrows indicate the movement of each letter to its new position.

METROPOLIS ALGORITHM

PROPOSE A CIPHER

Compute the quantity $\lambda = \frac{\text{sim}(\text{proposal cipher})}{\text{sim}(\text{current cipher})}$

IF $\lambda > 1$, SET CURRENT CIPHER TO BE THE
PROPOSED CIPHER

Else accept the proposal with
probability λ
and reject it with probability $1 - \lambda$

How do we Propose a cipher?

Taking different proposal probability distributions into considerations will lead to different accuracy and computation times

I have implemented multiple variations of the MH algorithm and I'd be discussing the following:

- Random Walk
- Barker's Informed Proposal
- Locally Informed and Threshold (LIT) Proposal

RANDOM WALK

METROPOLIS HASTINGS

Swap any two letters at random in the current cipher

e u f k r t c b g v a h d x z q p j o s w l y n m i

e u f k r t c b g v q h d x z a p j o s w l y n m i

A diagram illustrating a swap operation. Two white arrows originate from the letters 'a' and 'q' in the top string. The arrow from 'a' points down to the position of 'q' in the bottom string, and the arrow from 'q' points down to the position of 'a' in the bottom string. This indicates that the letters 'a' and 'q' have been swapped.

BARKER'S INFORMED PROPOSAL

Here, we consider $\text{Proposal density} \propto g\left(\frac{\text{sim}(\text{proposal cipher})}{\text{sim}(\text{current cipher})}\right) = g(\lambda)$

Where $g(t)$ is a balancing function with its optimal value being $g(t) = \frac{t}{1+t}$

LOCALLY INFORMED AND THRESHOLD PROPOSAL

Here, we consider thresholding of masses, i.e.,
for $\lambda \in [\min, \max]$

$$\text{Proposal density} \propto \frac{\text{sim}(\text{proposal cipher})}{\text{sim}(\text{current cipher})} = \lambda$$

For $\lambda < \min$

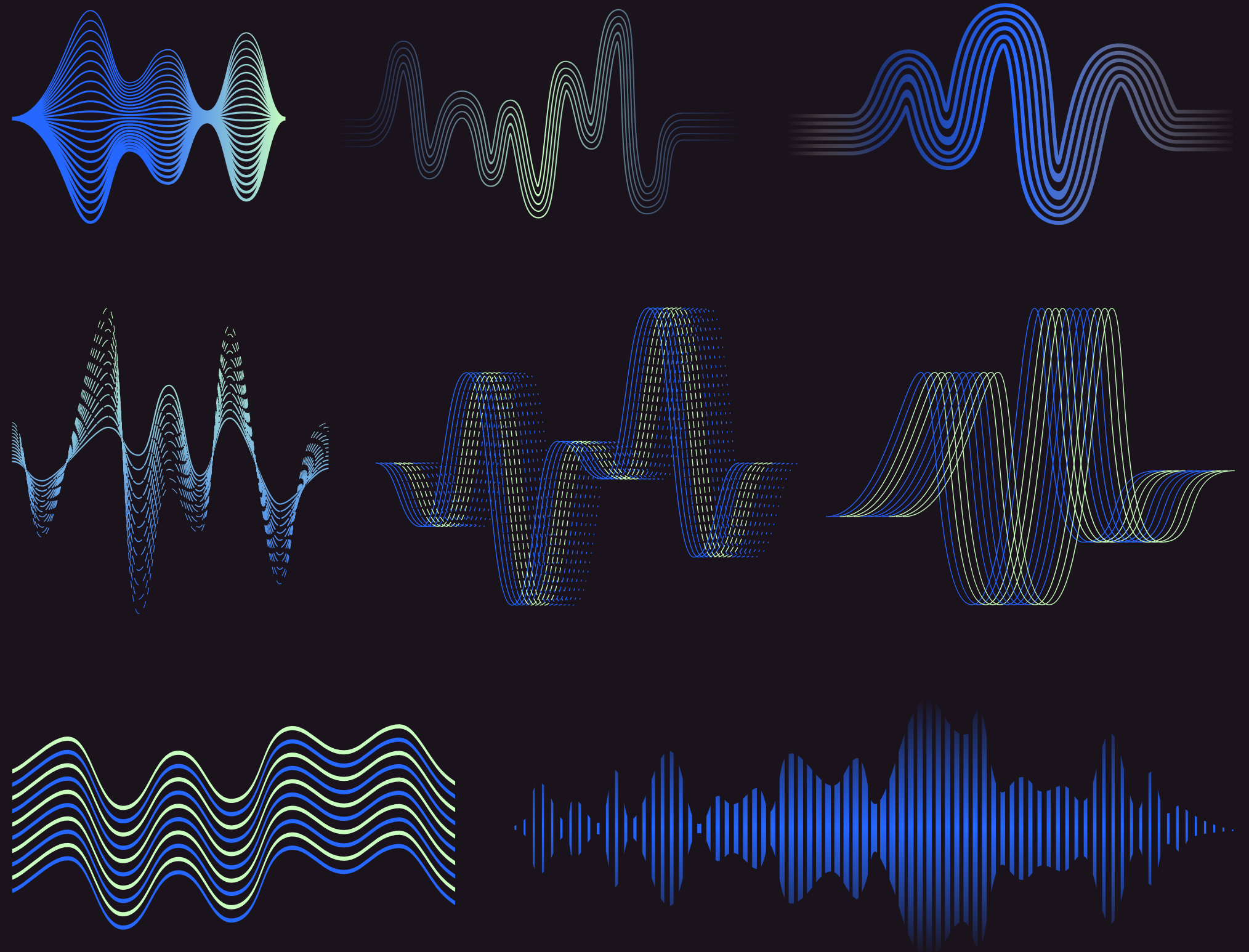
Proposal density $\propto \min$

For $\lambda > \max$

Proposal density $\propto \max$

Results

Let us run these different variations of the algorithm of a few iterations and look at their outputs



"prince wished to obtain this post for his son but others were trying through dowager"

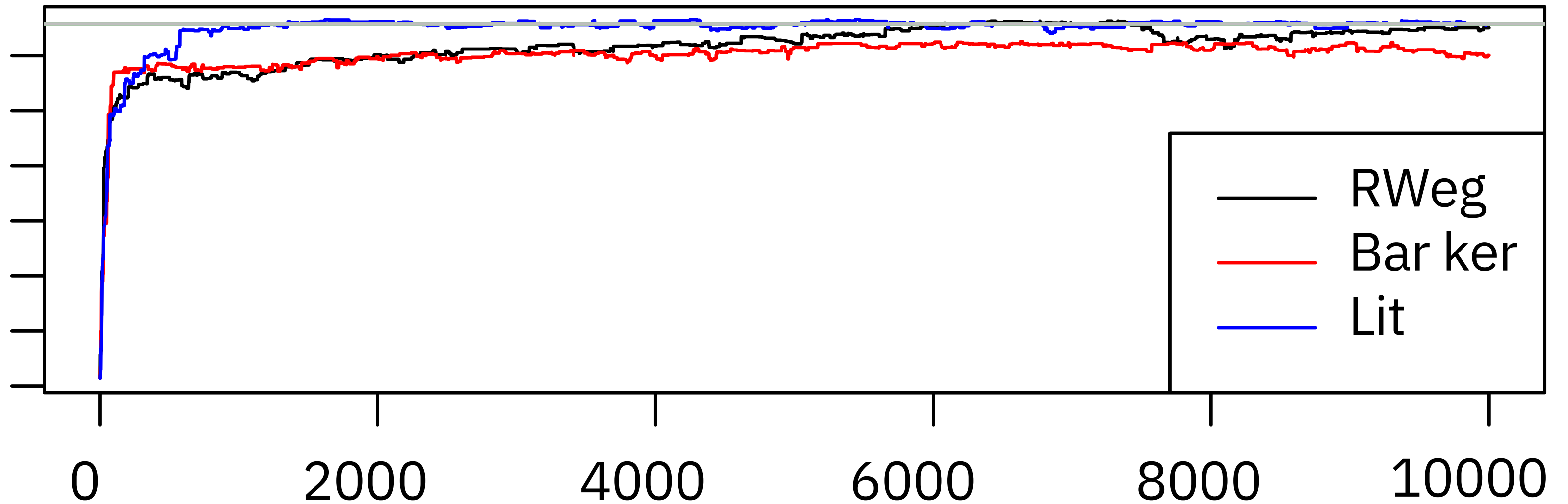
Input Text:

"rltkvf btsdfw qu uoqptk qdts rusq hul dts suk oaq uqdfis bflf qljtkm qdluamd wubpmfl"

Comparing different MH algorithms

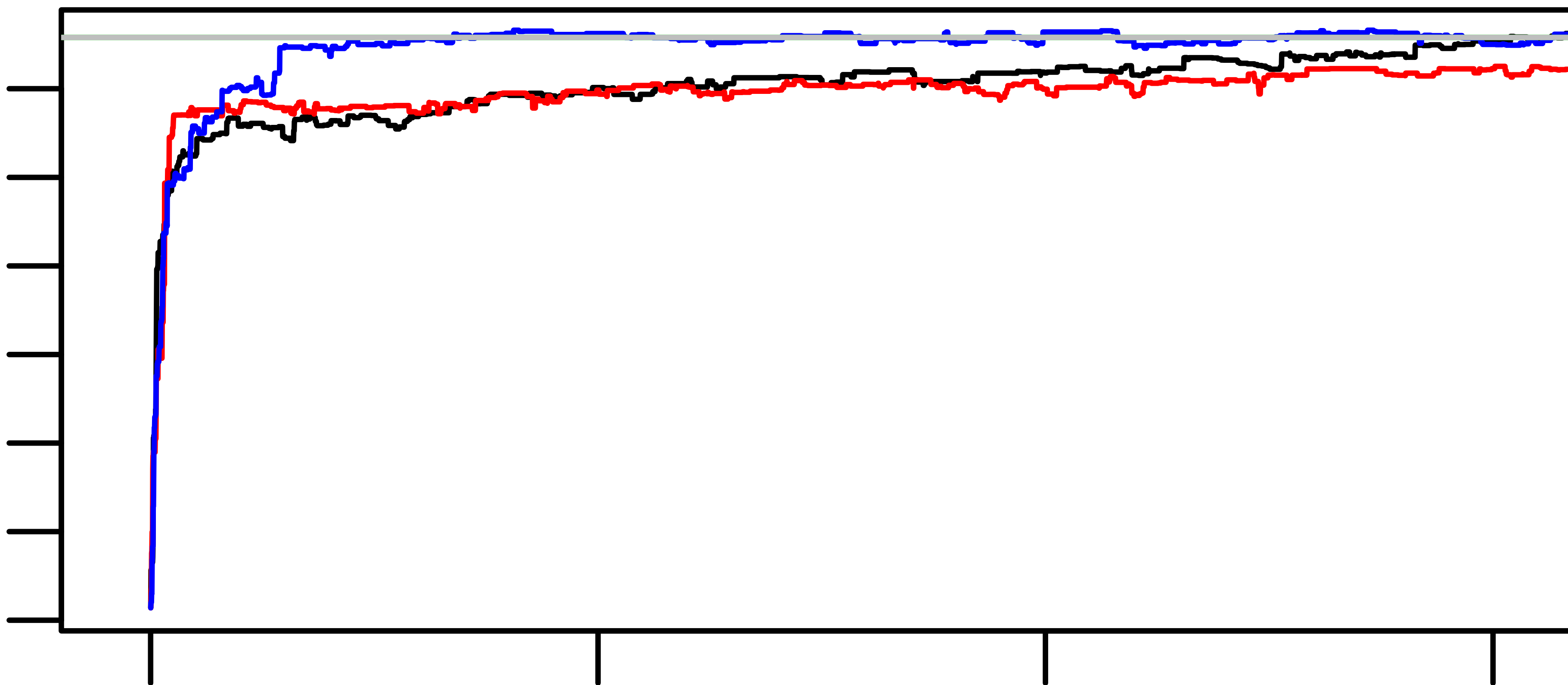
Number of Iterations

-750 -550



Log-Likelihoods

Comparing different MH algorithms



Output Text: "prince wished to obtain this post for his son but others were trying through dowager"

- Trying to decrypt longer sentences makes it easy for the algorithm to converge.
- Running the algorithm for large number of iteration helps.

Thank You!

