Overview and Comparison Genome Compression Algorithms

Tim Snyder
snyde479@morris.umn.edu
University of Minnesota Morris

December 2, 2012

What Are Genomes?

Your genome is your unique sequence of DNA
Made up of 4 nucleotides abbreviated as A, C, G and T
Genomes can be represented as long strings of those characters

Image from [4]
Human genome: 3000 MB

Complete works of William Shakespeare: 5 MB

What Do We Know?

DNA is made up of non-coding and coding portions

Non-coding portions are often random

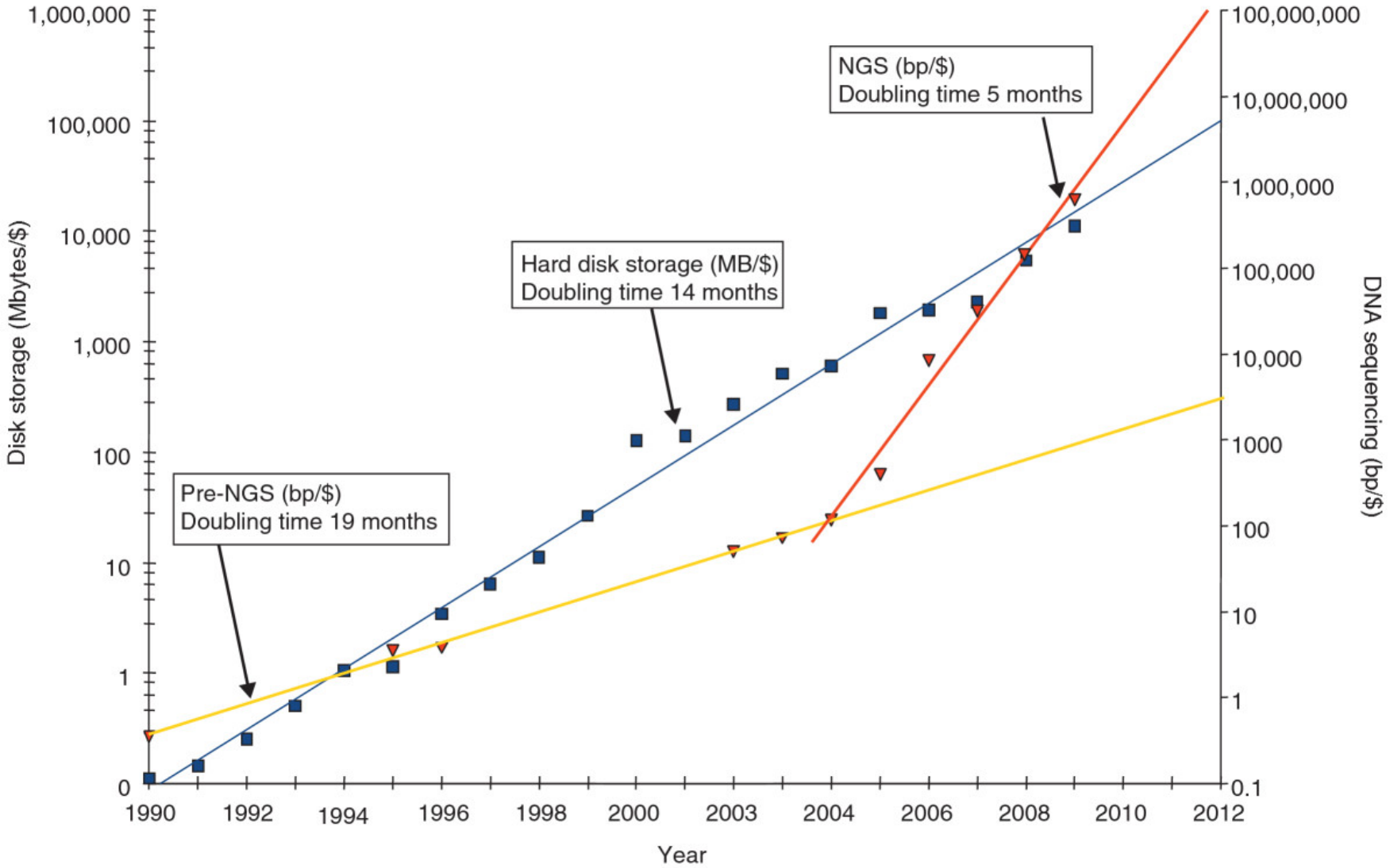
Coding portions are made of trios of nucleotides called codons

Why Do We Care?

More genomes leads to:

- Better understanding of biology
- Better medical treatments
- Better food

The Problem



From [2]

What is a Compression Algorithm?

Take a set of data

Reversibly modify it

Store it and hopefully have smaller file sizes

Why Do We Need Compression Algorithms?

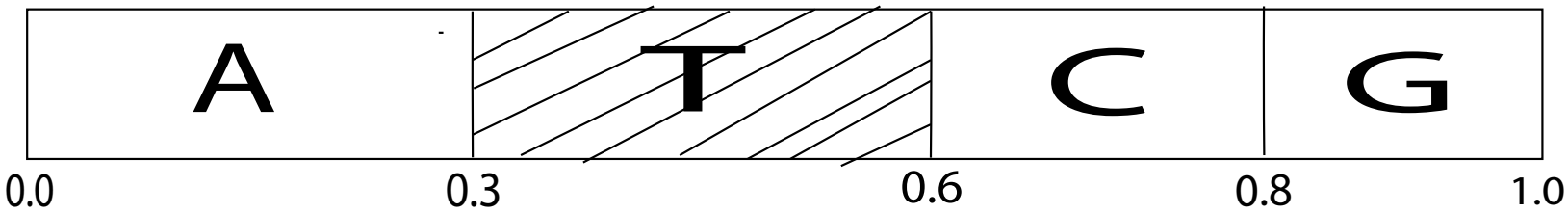
Make files take up less space on computers

Make file transfers faster

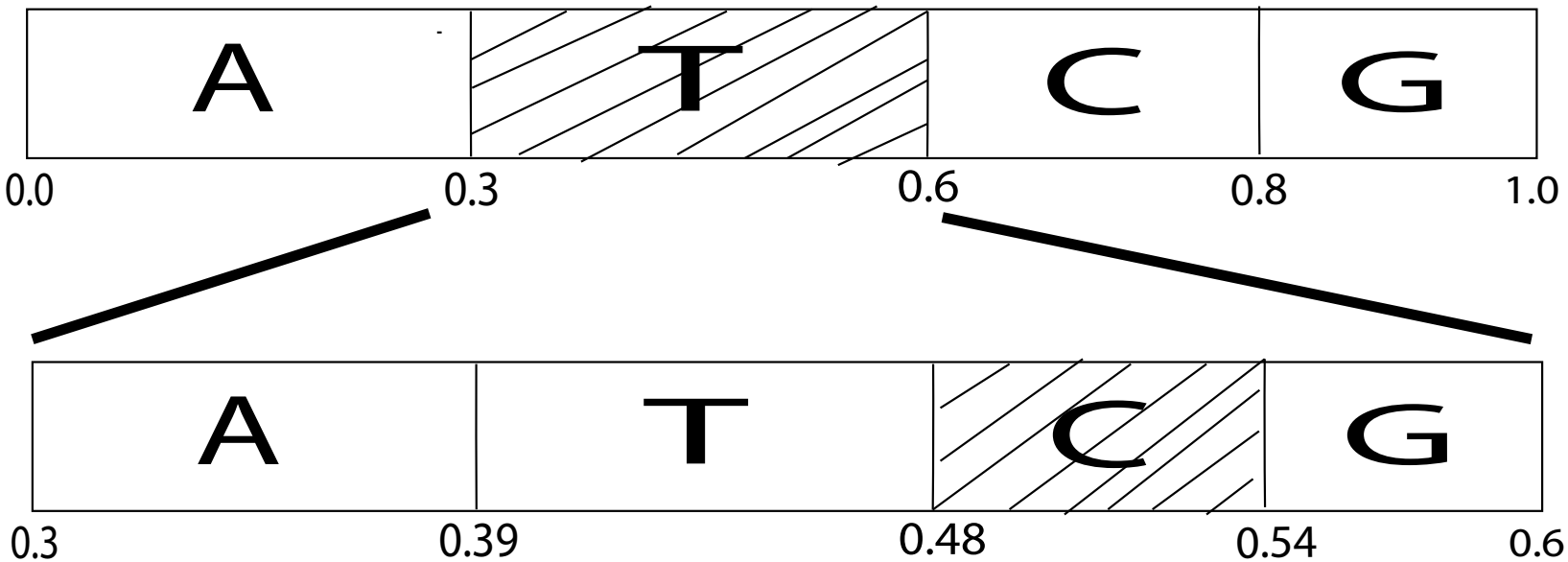
Arithmetic Coding

Takes in probabilities for characters
TCAGTGACTA
A=0.3 C=0.2 G=0.2 T=0.3
Creates a range for each of them
Finds a decimal to represent the string based on the ranges

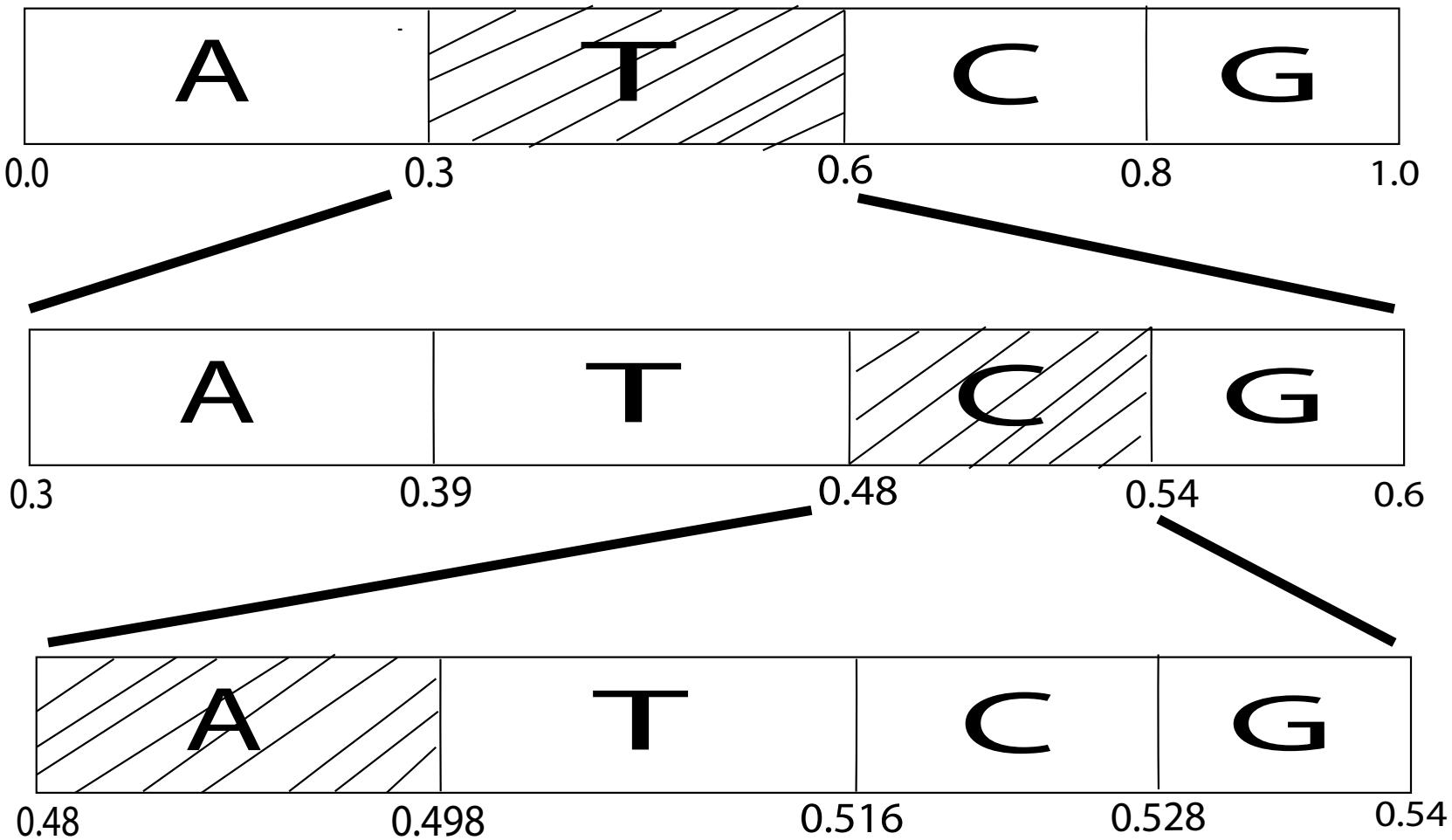
Compressing TCA



Compressing TCA



Compressing TCA



Arithmetic Conclusions

2 bits per character

Genomes still increasingly expensive to store

Is a baseline

Tabus and Korodi First Steps

Splits genome into substrings of length k

Then uses 3 different encoding methods

Best encoding method is used

Tabus and Korodi Encoding Methods

- 2 bits encoding method with arithmetic coding
- Order-1 Markov model
- Compressing based off of a previous substring

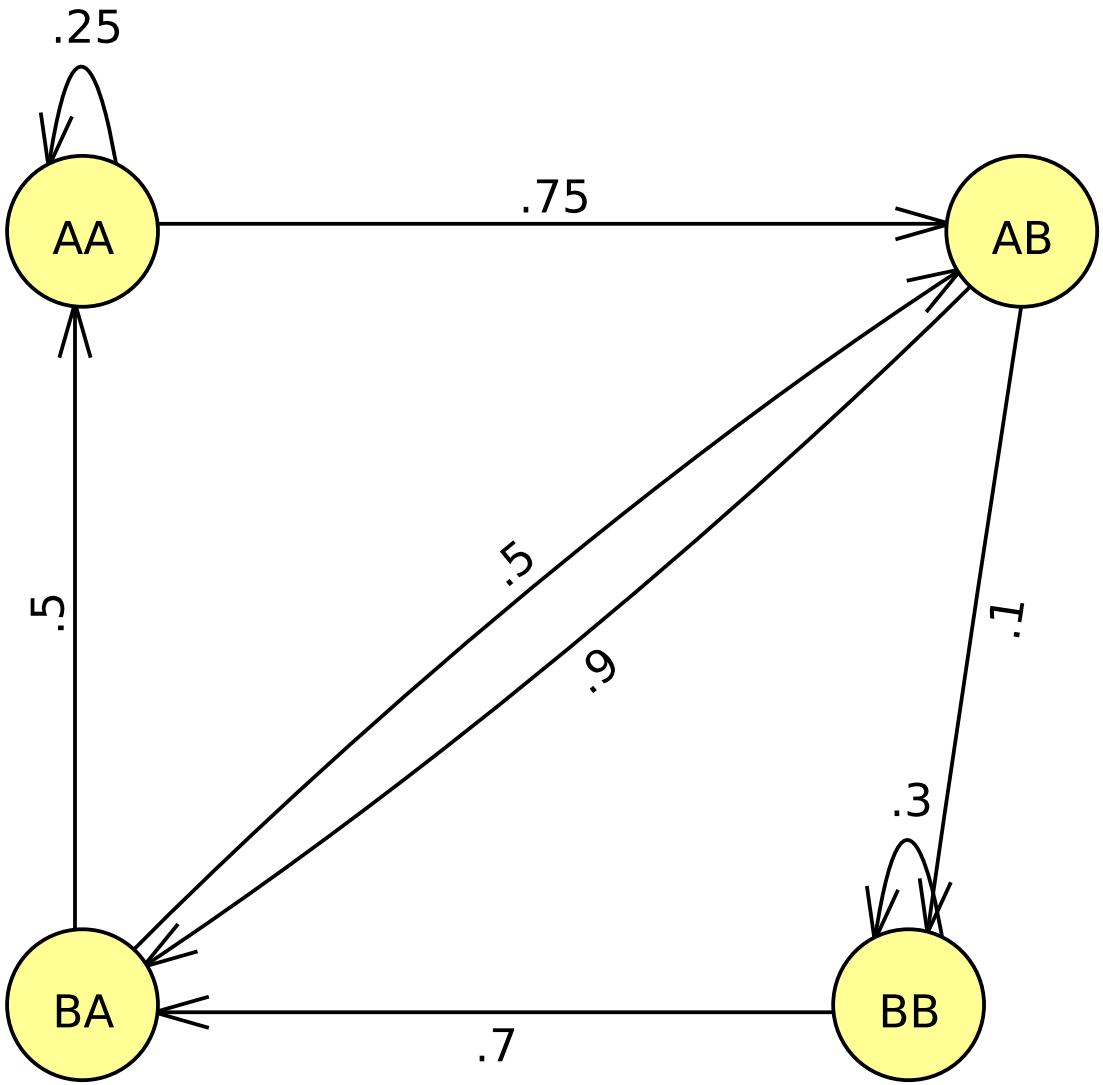
Second Encoding Method

Order 1 Markov model

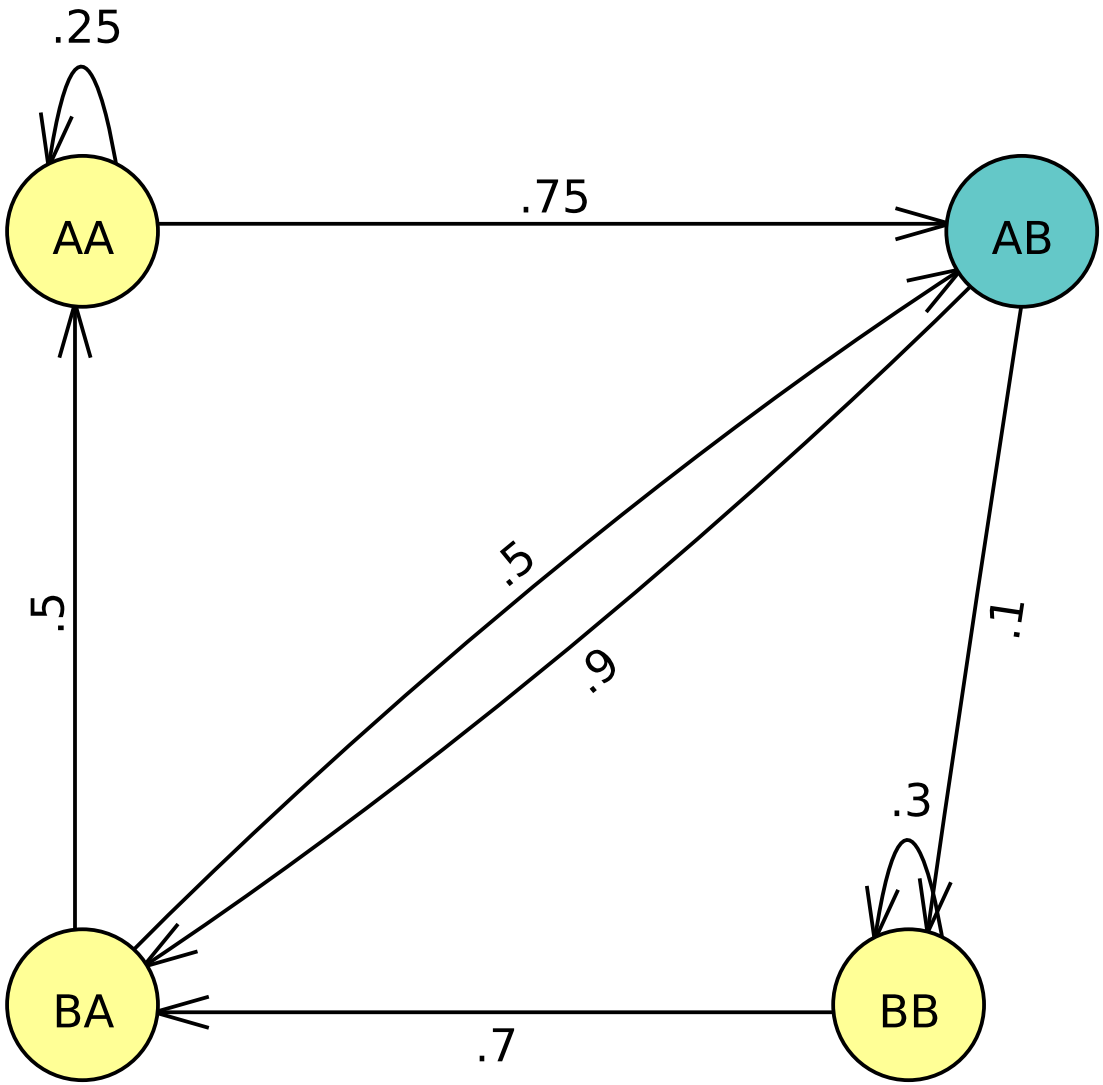
Uses last 2 characters to predict next

Uses arithmetic coding to compress

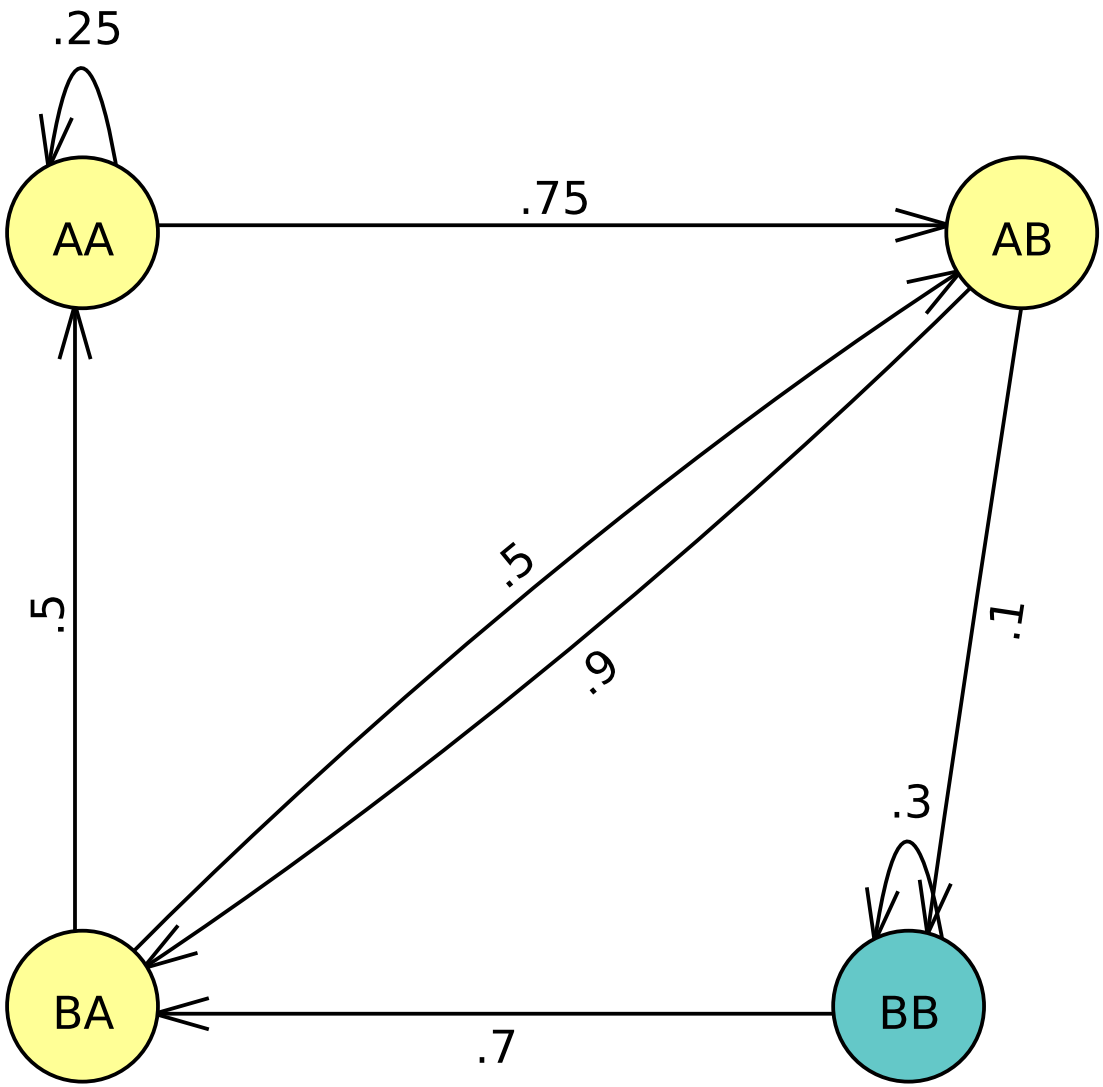
Second Encoding Method



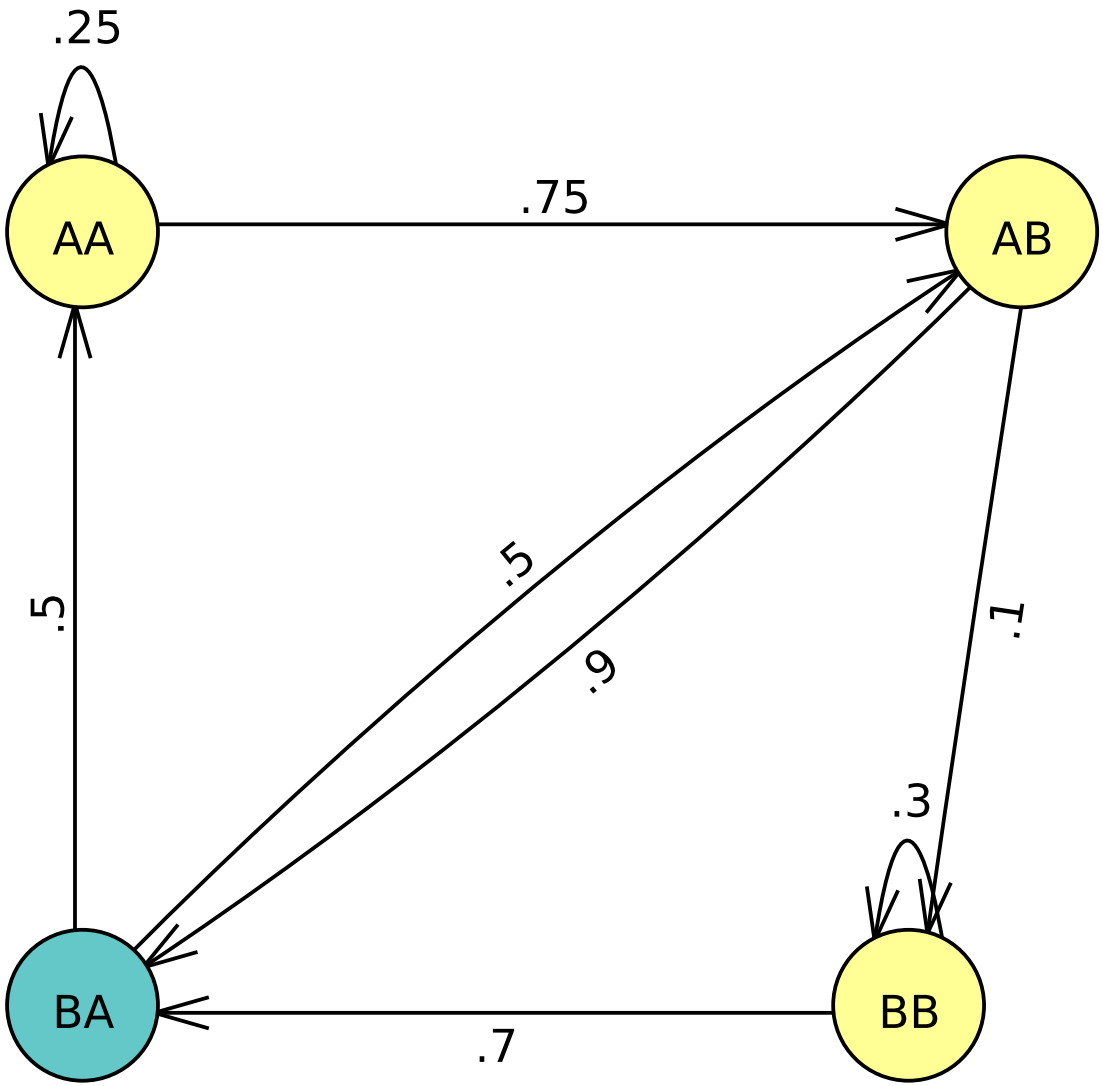
String ABBA



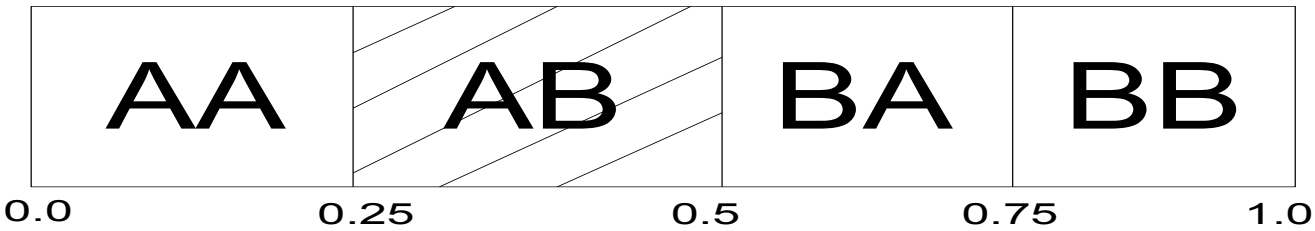
String ABBA



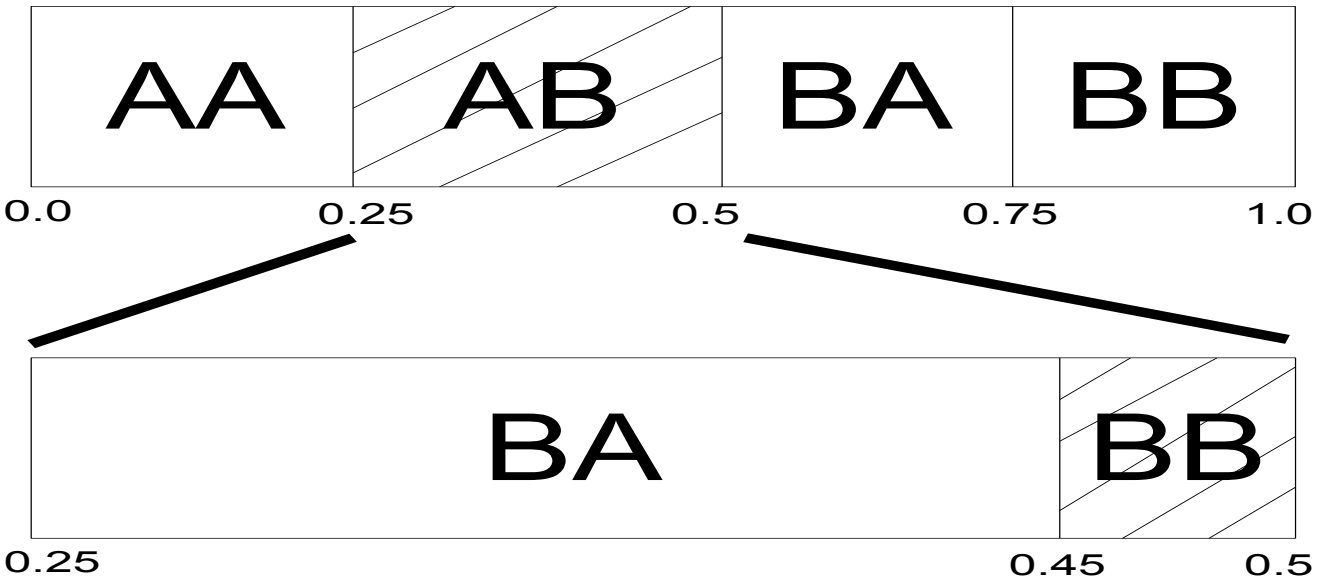
String ABBA



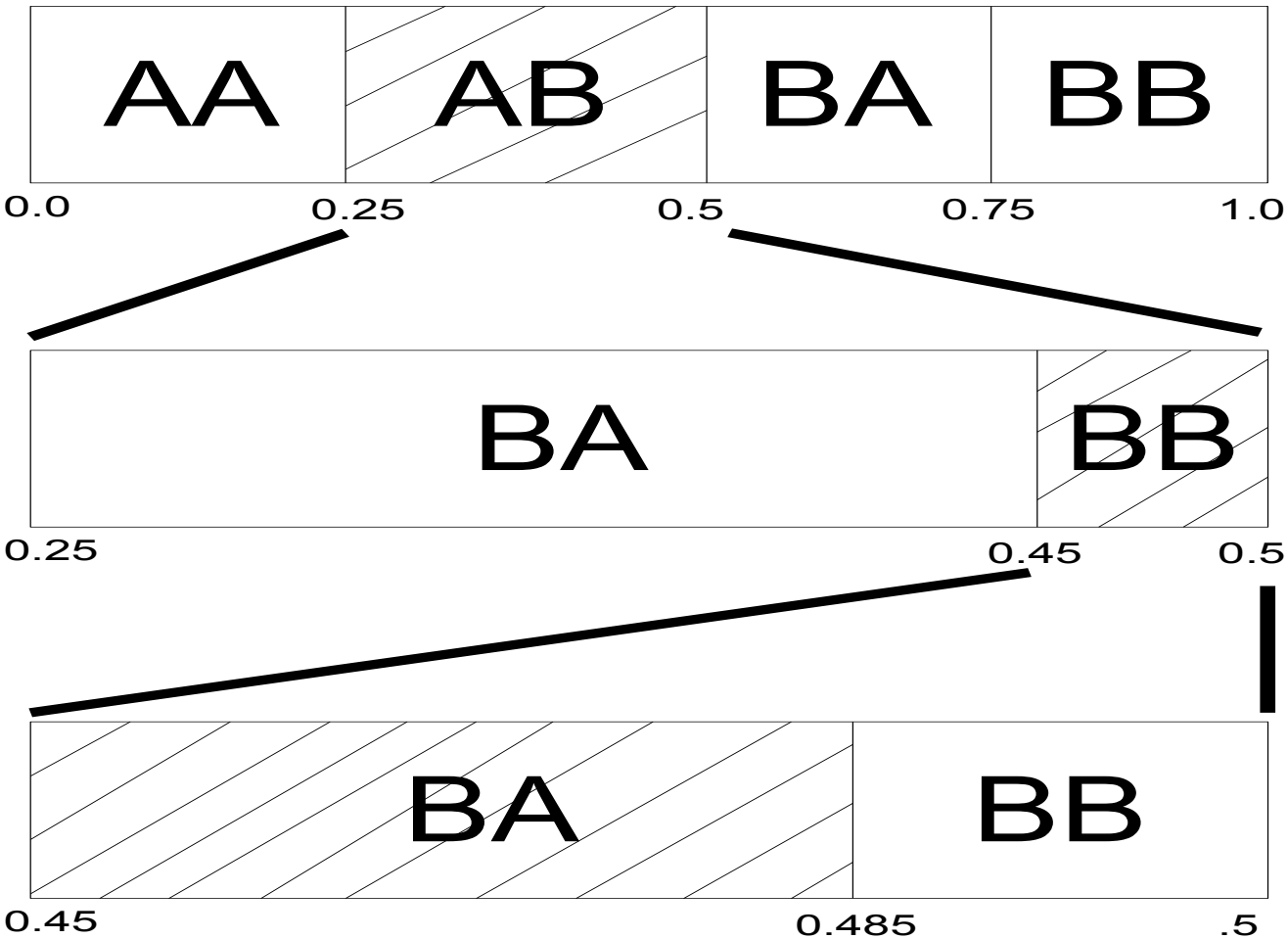
Markov Models Compress ABBA



Markov Models Compress ABBA



Markov Models Compress ABBA



Third Encoding Method

Current substring:AACTGT

Finds the most similar previous substring and stores its index

Previous:AATGGT

Creates a binary string of where the two substring differ

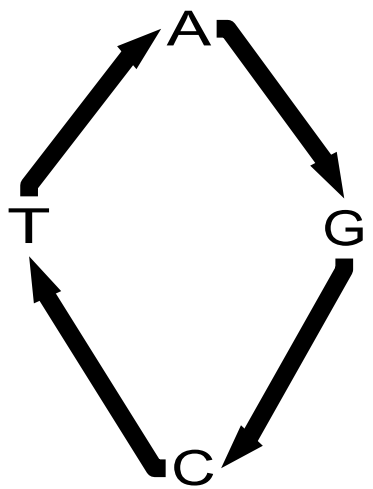
Binary: 0 0 1 1 0 0

Third Encoding Method

Previous: AATGGT
Current substring: AACTGT

Stores the distance from the characters in previous substring to current one

Distances: 3 2



Why Three Methods?

They perform differently on different portions

- Arithmetic works well on non-coding DNA
- Markov works well on coding portions
- Difference compression works well on repeated sections

Results

Chromosome	bits/char
chr1	1.641
chr2	1.662
chrX	1.548
chrY	1.149
Average	1.616

Efficiency in bits per base

Based on data from [3]

Database Compression Algorithm

Database compression algorithms compress entire sets of data

Having more data means more information for compression

Can lead to better compression rates

COMRAD

Step 1: count all substrings of length n
Step 2: replace the most common substring with a character
Step 3: repeat 1 and 2 until there are not enough substrings to replace
Step 4: store final string and replacements to get there

COMRAD in Action

Input

aabcbcaabcabc

Step 1

aa:2 ab:3 bc:4 cb:1 ca:2

...

Step 3

bc $\rightarrow$ A

aaAAaaAaA

...

CACB

Based on figure in [1]

COMRAD results

Dataset	COMRAD
Influenza	0.43
Hemoglobin	1.16
Bacteria	2.26
H. sapiens	1.44
Average	1.10

Table based on data from [1]

Conclusions

We can beat the 2 bits per character

Simple database compression works better than more complicated single genome compression

More work still needs to be done to compress genomes further

Future Work

Database seems to be the way to go

More data means better compression

More complicated database algorithms might be needed

Any questions?


S. Kuruppu, B. Beresford-Smith, T. Conway, and J. Zobel.
Iterative dictionary construction for compression of large DNA data sets.
IEEE/ACM Trans. Comput. Biol. Bioinformatics, 9(1):137–149, Jan. 2012.


L. Stein.
The case for cloud computing in genome informatics.
Genome Biology, 11(5):207, 2010.


I. Tabus and G. Korodi.
Genome compression using normalized maximum likelihood models for constrained markov sources.
In *Information Theory Workshop, 2008. ITW '08. IEEE*, pages 261 –265, may 2008.


Wikipedia.
Gene — wikipedia, the free encyclopedia, 2012.
[Online; accessed 27-November-2012].