# Comp. Genomics - Lectures 4 & 5

- Various problems on strings

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions
- Solving above string problems with suffix trees

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions
- Solving above string problems with suffix trees
- Compression (Lempel-Ziv) using suffix trees

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions
- Solving above string problems with suffix trees
- Compression (Lempel-Ziv) using suffix trees
- An accelerated introduction to entropy and relative entropy

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions
- Solving above string problems with suffix trees
- Compression (Lempel-Ziv) using suffix trees
- An accelerated introduction to entropy and relative entropy
- Bio applications of suffix trees and suffix arrays

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions
- Solving above string problems with suffix trees
- Compression (Lempel-Ziv) using suffix trees
- An accelerated introduction to entropy and relative entropy
- Bio applications of suffix trees and suffix arrays

# Comp. Genomics - Lectures 4 & 5

- Various problems on strings
- Cost of naive solutions
- Suffix trees – definition. Inefficient construction
- Suffix trees – known efficient constructions
- Solving above string problems with suffix trees
- Compression (Lempel-Ziv) using suffix trees
- An accelerated introduction to entropy and relative entropy
- Bio applications of suffix trees and suffix arrays

- Gusfield's book, chapters 5, 6, 7

# Entropy of a Distribution $p$

Let $X = (x_1, \ldots, x_n)$ be elements in a finite probability space, and $p(x_1), \ldots, p(x_n)$ be their probabilities ($\sum_{i=1}^{n} p(x_i) = 1$).

# Entropy of a Distribution $p$

Let $X = (x_1, \ldots, x_n)$ be elements in a finite probability space, and $p(x_1), \ldots, p(x_n)$ be their probabilities ($\sum_{i=1}^{n} p(x_i) = 1$).

- The Shannon entropy of the distribution $p$ is defined as
  $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.

# Entropy of a Distribution $p$

Let $X = (x_1, \ldots, x_n)$ be elements in a finite probability space, and $p(x_1), \ldots, p(x_n)$ be their probabilities ($\sum_{i=1}^{n} p(x_i) = 1$).

- The Shannon entropy of the distribution $p$ is defined as
  $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.
- Informally, the entropy measures the amount of surprise (in bits) when observing a single occurrence of $X$.

# Entropy of a Distribution $p$

Let $X = (x_1, \ldots, x_n)$ be elements in a finite probability space, and $p(x_1), \ldots, p(x_n)$ be their probabilities ($\sum_{i=1}^{n} p(x_i) = 1$).

- The Shannon entropy of the distribution $p$ is defined as
  $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.
- Informally, the entropy measures the amount of surprise (in bits) when observing a single occurrence of $X$.
- Let $p, q$ be two probability distributions on the same probability space, $X$. If $H(p) > H(q)$ then $p$ is more surprising than $q$. We also say that there is more randomness in $p$.

# Entropy of a Distribution $p$

The Shannon entropy of the distribution $p$ is $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.

- The entropy is always non-negative.

# Entropy of a Distribution $p$

The Shannon entropy of the distribution $p$ is $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.

- The entropy is always non-negative.
- For a distribution over $n$ elements, the entropy is maximized when $p(x_i) = 1/n$ for all $i$.

# Entropy of a Distribution $p$

The Shannon entropy of the distribution $p$ is $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.

- The entropy is always non-negative.
- For a distribution over $n$ elements, the entropy is maximized when $p(x_i) = 1/n$ for all $i$.
- Entropy is a fundamental tool in Shannon's information theory.

# Entropy of a Distribution $p$

The Shannon entropy of the distribution $p$ is $H(p) = -\sum_{x \in X} p(x) \log p(x) = E_p(\log p(X))$.

- The entropy is always non-negative.
- For a distribution over $n$ elements, the entropy is maximized when $p(x_i) = 1/n$ for all $i$.
- Entropy is a fundamental tool in Shannon's information theory.
- Also essential in communication, coding, probability, combinatorics, ...

# Entropy in Boolean Case

Suppose $X$ has just two elements $x_0, x_1$. Denote $p = p(x_1)$, then the probability of $x_0$ is $1 - p$.

# Entropy in Boolean Case

Suppose $X$ has just two elements $x_0, x_1$. Denote $p = p(x_1)$, then the probability of $x_0$ is $1 - p$. In this case $H(p) = -p\log(p) - (1-p)\log(1-p)$, and is maximized at $p = 1/2$.

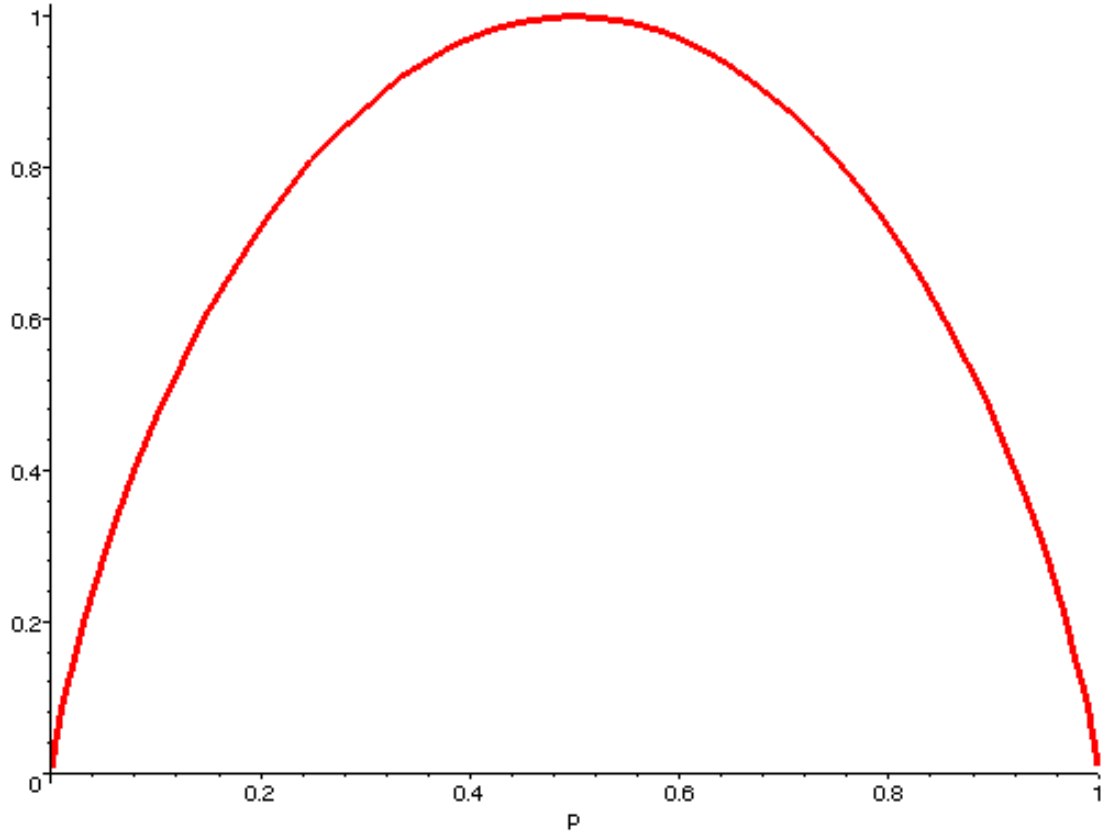
# Entropy in Boolean Case

Suppose $X$ has just two elements $x_0, x_1$. Denote $p = p(x_1)$, then the probability of $x_0$ is $1 - p$. In this case $H(p) = -p\log(p) - (1-p)\log(1-p)$, and is maximized at $p = 1/2$. In addition, $H$ is convex.

# Entropy in Boolean Case

Suppose $X$ has just two elements $x_0, x_1$. Denote $p = p(x_1)$, then the probability of $x_0$ is $1 - p$. In this case $H(p) = -p\log(p) - (1-p)\log(1-p)$, and is maximized at $p = 1/2$. In addition, $H$ is convex.

```
plot(-p*log[2](p)-(1-p)*log[2](1-p),p=0..1,thickness=3);
```



# Relative Entropy of Two Distributions

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over the same probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

# Relative Entropy of Two Distributions

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over the same probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- Intuitively, relative entropy measures how far apart two distributions are.

# Relative Entropy of Two Distributions

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over the same probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- Intuitively, relative entropy measures how far apart two distributions are.
- For example, if $p = q$ then clearly $D(p \,||\, q) = 0$.

# Relative Entropy of Two Distributions

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over the same probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- Intuitively, relative entropy measures how far apart two distributions are.
- For example, if $p = q$ then clearly $D(p \,||\, q) = 0$.
- Turns out the other direction holds too, *i.e.* $D(p \,||\, q) = 0 \Longrightarrow p = q$.

# Relative Entropy of Two Distributions (2)

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

# Relative Entropy of Two Distributions (2)

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- In general $D$ is *not* a distance metric.

# Relative Entropy of Two Distributions (2)

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- In general $D$ is *not* a distance metric.
- Usually, $D(p \,||\, q) \neq D(q \,||\, p)$.

# Relative Entropy of Two Distributions (2)

The Kullback-Leibler (KL) relative entropy of two distributions $p, q$ over probability space $X$ is defined as

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = E_p \left( \log \frac{p(X)}{q(X)} \right)$$

- In general $D$ is *not* a distance metric.
- Usually, $D(p \,||\, q) \neq D(q \,||\, p)$.
- In many cases, triangle inequality is violated: $D(p_1 \,||\, p_3) > D(p_1 \,||\, p_2) + D(p_2 \,||\, p_3)$ is possible.

# Examples

- $D(p\,||\,q) \neq D(q\,||\,p)$. Take $X = \{x_0, x_1\}$:

# Examples

- $D(p\,||\,q) \neq D(q\,||\,p)$. Take $X = \{x_0, x_1\}$:
- $p(x_0) = \frac{1}{2}, p(x_1) = \frac{1}{2}$, $q(x_0) = \frac{1}{4}, q(x_1) = \frac{3}{4}$,

# Examples

- $D(p\,||\,q) \neq D(q\,||\,p)$. Take $X = \{x_0, x_1\}$:
- $p(x_0) = \frac{1}{2}, p(x_1) = \frac{1}{2}$, $q(x_0) = \frac{1}{4}, q(x_1) = \frac{3}{4}$,

  Then $D(p\,||\,q) = \frac{1}{2}\log\left(\frac{1/2}{3/4}\right) + \frac{1}{2}\log\left(\frac{1/2}{1/4}\right)$,

  while $D(q\,||\,p) = \frac{3}{4}\log\left(\frac{3/4}{1/2}\right) + \frac{1}{4}\log\left(\frac{1/4}{1/2}\right)$.

# Examples

- $D(p\,||\,q) \neq D(q\,||\,p)$. Take $X = \{x_0, x_1\}$:
- $p(x_0) = \frac{1}{2}, p(x_1) = \frac{1}{2}$, $q(x_0) = \frac{1}{4}, q(x_1) = \frac{3}{4}$,

  Then $D(p\,||\,q) = \frac{1}{2}\log\left(\frac{1/2}{3/4}\right) + \frac{1}{2}\log\left(\frac{1/2}{1/4}\right)$,

  while $D(q\,||\,p) = \frac{3}{4}\log\left(\frac{3/4}{1/2}\right) + \frac{1}{4}\log\left(\frac{1/4}{1/2}\right)$.
- $D(p_1\,||\,p_3) > D(p_1\,||\,p_2) + D(p_2\,||\,p_3)$.
  Again, take $X = x_0, x_1$:

# Examples

- $D(p\,||\,q) \neq D(q\,||\,p)$. Take $X = \{x_0, x_1\}$:
- $p(x_0) = \frac{1}{2}, p(x_1) = \frac{1}{2}$, $q(x_0) = \frac{1}{4}, q(x_1) = \frac{3}{4}$,

  Then $D(p\,||\,q) = \frac{1}{2}\log\left(\frac{1/2}{3/4}\right) + \frac{1}{2}\log\left(\frac{1/2}{1/4}\right)$,

  while $D(q\,||\,p) = \frac{3}{4}\log\left(\frac{3/4}{1/2}\right) + \frac{1}{4}\log\left(\frac{1/4}{1/2}\right)$.
- $D(p_1\,||\,p_3) > D(p_1\,||\,p_2) + D(p_2\,||\,p_3)$.
  Again, take $X = x_0, x_1$:
- $p_1(x_0) = 1/2, p_1(x_1) = 1/2$,
  $p_2(x_0) = 1/4, p_2(x_1) = 3/4$,
  $p_3(x_0) = 1/3, p_3(x_1) = 2/3$.
  Calculation shows $\Delta$ inequality does not hold.

# Symmetric Relative Entropy

By taking the measure $d(p,q) = D(p\,||\,q) + D(q\,||\,p)$ we get a "symmetric" version of relative entropy. This version

# Symmetric Relative Entropy

By taking the measure $d(p,q) = D(p\,||\,q) + D(q\,||\,p)$ we get a "symmetric" version of relative entropy. This version

- is clearly symmetric, $d(p,q) = d(q,p)$,

# Symmetric Relative Entropy

By taking the measure $d(p,q) = D(p\,||\,q) + D(q\,||\,p)$ we get a "symmetric" version of relative entropy. This version

- is clearly symmetric, $d(p,q) = d(q,p)$,
- tends to satisfy triangle inequality more often (still, not always),

# Symmetric Relative Entropy

By taking the measure $d(p,q) = D(p\,||\,q) + D(q\,||\,p)$ we get a "symmetric" version of relative entropy. This version

- is clearly symmetric, $d(p,q) = d(q,p)$,
- tends to satisfy triangle inequality more often (still, not always),
- is more suitable as an "estimation" of distance between two distributions.

# Estimating Distances between Genomes

Given two genomes $g_1, g_2$ (each is a very long string over DNA alphabet, not necessarily of equal length), is there a way to define their distance?

- Motivation: Whole genome phylogenetic trees.

# Estimating Distances between Genomes

Given two genomes $g_1, g_2$ (each is a very long string over DNA alphabet, not necessarily of equal length), is there a way to define their distance?

- Motivation: Whole genome phylogenetic trees.
- Any suggestions?

# Estimating Distances between Genomes

Given two genomes $g_1, g_2$ (each is a very long string over DNA alphabet, not necessarily of equal length), is there a way to define their distance?

- Motivation: Whole genome phylogenetic trees.
- Any suggestions?
- Problem not well defined!

# Estimating Distances between Genomes

Given two genomes $g_1, g_2$ (each is a very long string over DNA alphabet, not necessarily of equal length), is there a way to define their distance?

- Motivation: Whole genome phylogenetic trees.
- Any suggestions?
- Problem not well defined!
- Can get clues from information theory, if think of genomes as product of two distributions $G_1, G_2$.

# Estimating Distances between Genomes

Given two genomes $g_1, g_2$ (each is a very long string over DNA alphabet, not necessarily of equal length), is there a way to define their distance?

- Motivation: Whole genome phylogenetic trees.
- Any suggestions?
- Problem not well defined!
- Can get clues from information theory, if think of genomes as product of two distributions $G_1, G_2$.
- For example, can take $d(G_1, G_2)$ as desired "distance".

# Distances between Genomes (2)

Problems with approach:

- Distributions $d(G_1, G_2)$ unknown.

# Distances between Genomes (2)

Problems with approach:

- Distributions $d(G_1, G_2)$ unknown.
- All we know is long samples of them, $g_1, g_2$.

# Distances between Genomes (2)

Problems with approach:

- Distributions $d(G_1, G_2)$ unknown.
- All we know is long samples of them, $g_1, g_2$.
- Should look for operators that approximate $d(G_1, G_2)$.

# Useful Notations

- Let $(g_1)_{i+1}^{i+k+1}$ denotes the substring of $g_1$ that starts in position $i+1$ and ends in position $i+k+1$.

# Useful Notations

- Let $(g_1)_{i+1}^{i+k+1}$ denotes the substring of $g_1$ that starts in position $i+1$ and ends in position $i+k+1$.
- Let $L_i(g_2, g_1) = \max_k((g_1)_{i+1}^{i+k+1} \subset g_2)$.

# Useful Notations

- Let $(g_1)_{i+1}^{i+k+1}$ denotes the substring of $g_1$ that starts in position $i+1$ and ends in position $i+k+1$.
- Let $L_i(g_2, g_1) = \max_k((g_1)_{i+1}^{i+k+1} \subset g_2)$.
- What is $L_i(g_2, g_1)$?

# Useful Notations

- Let $(g_1)_{i+1}^{i+k+1}$ denotes the substring of $g_1$ that starts in position $i+1$ and ends in position $i+k+1$.
- Let $L_i(g_2, g_1) = \max_k((g_1)_{i+1}^{i+k+1} \subset g_2)$.
- What is $L_i(g_2, g_1)$?
- This is the longest substring of $g_1$, starting at $i$, that is also a substring of $g_2$.

# Intuition

$L_i(g_2, g_1) = \max_k((g_1)_{i+1}^{i+k+1} \subset g_2)$.

- Strings $g_2, g_1$ that are close by will tend to have long common substrings.

# Intuition

$L_i(g_2, g_1) = \max_k((g_1)_{i+1}^{i+k+1} \subset g_2)$.

- Strings $g_2, g_1$ that are close by will tend to have long common substrings.
- Strings $g_2, g_1$ that are far away will tend to have short common substrings.

# Intuition

$L_i(g_2, g_1) = \max_k((g_1)_{i+1}^{i+k+1} \subset g_2)$.

- Strings $g_2, g_1$ that are close by will tend to have long common substrings.
- Strings $g_2, g_1$ that are far away will tend to have short common substrings.
- Maybe some average can lead to desired distance.

# The Distance Operator

Let $\bar{L}(g_2, g_1) = \frac{1}{|g_2|} \sum_i L_i(g_2, g_1)$ be the average over $i$'s of $L_i(g_2, g_1)$.

# The Distance Operator

Let $\bar{L}(g_2, g_1) = \frac{1}{|g_2|} \sum_i L_i(g_2, g_1)$ be the average over $i$'s of $L_i(g_2, g_1)$.

A theorem of Weiner states that as the length of $g_2, g_1$ increases,

$$\frac{\log |g_2|}{\bar{L}(g_2, g_1)} - \frac{\log |g_1|}{\bar{L}(g_1, g_1)}$$

converges to $D(G_1 \,||\, G_2)$.

# The Distance Operator

Let $\bar{L}(g_2, g_1) = \frac{1}{|g_2|} \sum_i L_i(g_2, g_1)$ be the average over $i$'s of $L_i(g_2, g_1)$.

A theorem of Weiner states that as the length of $g_2, g_1$ increases,

$$\frac{\log |g_2|}{\bar{L}(g_2, g_1)} - \frac{\log |g_1|}{\bar{L}(g_1, g_1)}$$

converges to $D(G_1 \,||\, G_2)$.

This gives a theoretical justification for using the average common string method.

# Average Common String Algorithm

Given two strings $g_2, g_1$

- Use a suffix tree to efficiently compute $\sum_i L_i(g_2, g_1)$, $\sum_j L_j(g_2, g_1)$.

# Average Common String Algorithm

Given two strings $g_2, g_1$

- Use a suffix tree to efficiently compute $\sum_i L_i(g_2, g_1)$, $\sum_j L_j(g_2, g_1)$.
- Average to estimate $d(g_1, g_2)$.

# Average Common String Algorithm

Given two strings $g_2, g_1$

- Use a suffix tree to efficiently compute $\sum_i L_i(g_2, g_1)$, $\sum_j L_j(g_2, g_1)$.
- Average to estimate $d(g_1, g_2)$.
- Repeat for any pair of genomes.

# Average Common String Algorithm

Given two strings $g_2, g_1$

- Use a suffix tree to efficiently compute $\sum_i L_i(g_2, g_1)$, $\sum_j L_j(g_2, g_1)$.
- Average to estimate $d(g_1, g_2)$.
- Repeat for any pair of genomes.
- Apply a distance based tree reconstruction to produce a “whole genome phylogeny”.

# Average Common String Algorithm

Given two strings $g_2, g_1$

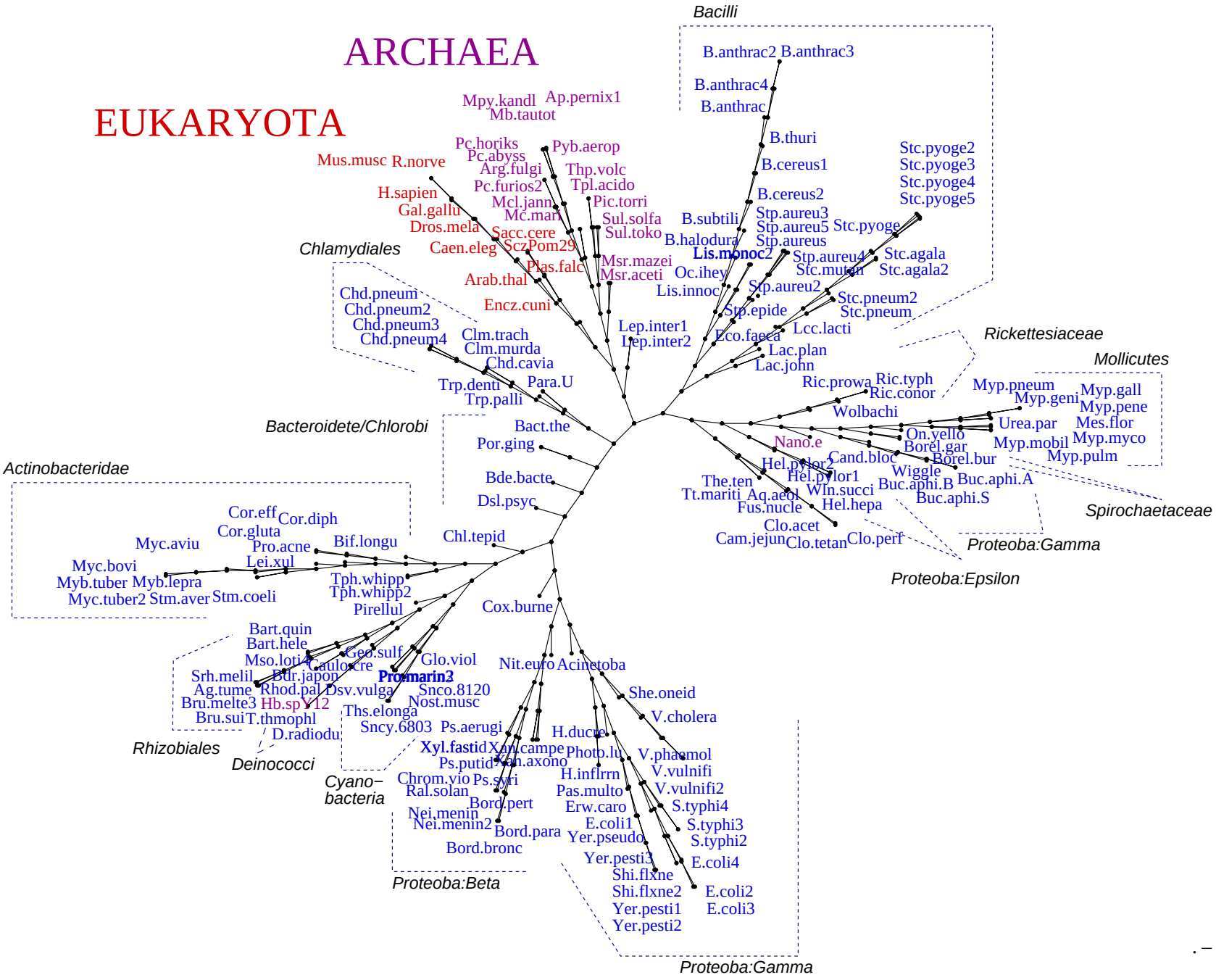
- Use a suffix tree to efficiently compute $\sum_i L_i(g_2, g_1)$, $\sum_j L_j(g_2, g_1)$.
- Average to estimate $d(g_1, g_2)$.
- Repeat for any pair of genomes.
- Apply a distance based tree reconstruction to produce a “whole genome phylogeny”.
- For real proteome and genome strings, triangle inequality satisfied for all pairs.

# Average Common String Algorithm

Given two strings $g_2, g_1$

- Use a suffix tree to efficiently compute $\sum_i L_i(g_2, g_1)$, $\sum_j L_j(g_2, g_1)$.
- Average to estimate $d(g_1, g_2)$.
- Repeat for any pair of genomes.
- Apply a distance based tree reconstruction to produce a "whole genome phylogeny".
- For real proteome and genome strings, triangle inequality satisfied for all pairs.
- Joint work with Burstein, Ulitsky, Tuller (2004).

# Prot. Tree (average $1M$ long), 191 Taxa



# Retroid Virus Tree

Part of 1837 virus forest. Average genome length 5K.