Computational Systems Biology

. . . Biology X – Lecture 5 . . .

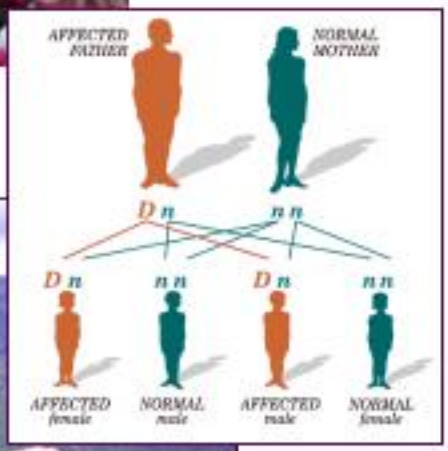
Bud Mishra

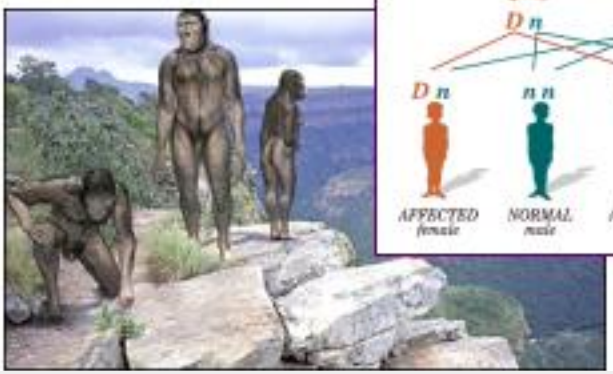
Professor of Computer Science, Mathematics, & Cell Biology

Molecular Evolution

Polymorphisms in Population

- Why do we care about variations?
  - Underlie phenotypic differences
  - Cause inherited diseases
  - Allow tracking ancestral human history



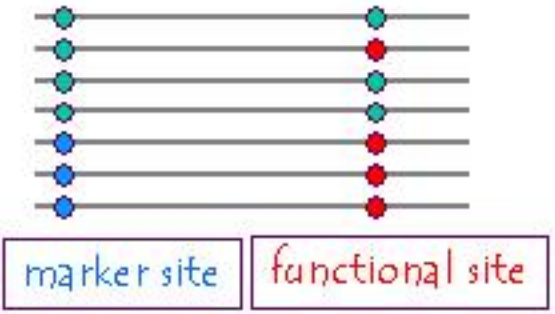


How do we find sequence variations?

- Look at multiple sequences from the same genome region
- Use base quality values to decide if mismatches are true polymorphisms or sequencing errors
- Distinguish variation derived from father vs. that from mother: *Haplotypes*

Allelic association

- It is the non-random assortment between alleles
  - It measures how well knowledge of the allele state at one site permits prediction at another
  - Significant allelic association between a marker and a functional site permits localization (mapping) even without having the functional site in our collection
- Strength of allelic association
  - Pair-wise and multi-locus measures of association.



Motivation

- Disease association studies
  - identify genetic variation that contributes to a particular disease
- Drug Design
  - design drugs tailored to specific populations
- Population Genetics Inference
  - the extent of linkage disequilibrium can tell you about the patterns of recombination, or about demographic events (like recent bottlenecks).

Inferring Population Genetics

- The limited diversity in the European population as compared to the African population
  - It may be indicative of the founder effect.
  - It supports the out-of-Africa theory.
- IBM-National Geographic project:
  - GENOGRAPHIC
  - https://www9.nationalgeographic.com/genographic/index.html

Genographic Project

- What is expected:
- Public database of anthropological genetic information
- Virtual museum of human history
  - Online at nationalgeographic.com/genographic,
  - Information about genetics, migration, linguistics, indigenous populations and the threats facing them, anthropology, archaeology, and more.
  - Public participation
- New information on genetic anthropology
- Improved global awareness of indigenous

Mitochondria and Phylogeny

- **Mitochondrial DNA (mtDNA):** Extra-nuclear DNA, transmitted through maternal lineage. Mitochondria are inherited in a growing mammalian zygote only from the egg.
- 16.5 Kb, contains genes: coding for 13 proteins, 22 tRNA genes, 2 rRNA genes.
- mtDNA has a pointwise mutation substitution rate 10 times faster than nuclear DNA.
- Phylogeny based on human mtDNA can give us molecular (hence accurate!) information about human evolution.

Rate of Evolutionary Changes

- Taxa of nucleotide or amino acid sequences.
- Given two taxa s_i and s_j, measure their distance
  - Distance(s_i, s_j), d_{ij} = Edit distance based on pairwise sequence alignment.
- Assumptions about the Molecular Clock (governing rate of evolutionary change):
  - Only independent substitutions
  - No back or parallel mutations
  - Neglect selection pressure.

Distance Based Approaches

- **Given**:

  An $n \times n$ nonnegative-valued distance matrix $M \in R_+^{n \times n}$, where M_{ij} is the distance between objects i and j:

- **Construct**:

  An edge-weighted tree such that the distances between leaves i and j are "close" to M_{ij}

Average Linkage Clustering

- UPGMA
  - (Unweighted Pair-Group Method using an Arithmetic Average).
- Distance between clusters (disjoint sets of taxa) C_i and C_j is
- Distance(C_i, C_j) = d_{ij}

$$= (1/|C_i| \cdot |C_j|) \sum_{p \in C_i,\, q \in C_j} d_{pq}.$$

- This is the average distance between pairs of taxa from each cluster.

UPGMA

- Assign each taxon to is own cluster.
- Define one leaf for each taxon—
  - Place it at height 0.
- While more than two clusters exist
  - Determine two clusters i and j with smallest d_{ij}
  - Define a new cluster $C_k = C_i \cup C_j$
  - Define a node k with children i and j—
    - Place it at height $d_{ij}/2$.
  - Replace clusters C_i and C_j with C_k
- Join the last two clusters (i and j) by root at height $d_{ij}/2$. □

Nucleotide Sequences

- Synonymous or Neutral Substitutions:
  = Nucleotide substitutions with no effect on expressed amino acid sequences
  - RECALL: Genetic code is redundant—Most substitutions to 3rd positions are synonymous.
  - Often a single non-synonymous nucleotide substitution is likely to change one amino acid into a related amino acid (e.g., both hydrophobic).
- Molecular clock is modeled based on non-synonymous substitution rate.

Variability of Nucleotide Mutation Rate

- **Transitional Mutations:**
  - purine-purine, i.e. A ↔ G
  - pyrimidine-pyrimidine, i.e. C ↔ T
- **Transversal Mutations:**
  - purine-pyrimidine: A ↔ T, A ↔ C, G ↔ C, G ↔ T
- Usually transitional mutations are more likely. Mutation into A is more likely.

DNA repair

- Effect of DNA repair mechanism

λ for ...	#per site per year
higher primate	$\approx 1.3 \times 10^{-9}$ /site/yr
sea urchins & rodents	$\approx 6.6 \times 10^{-9}$/site/yr
mammalian mtDNA	$\approx 10^{-8}$/site/yr
plant cpDNA	$\approx 1.1 \times 10^{-9}$/site/yr

Markov Process Model of Mutation

- Evolution is modeled by a stochastic process, $X(t)$ with real-valued time parameter $t \geqq 0$
- A time-homogeneous Markov process
- $(Q, \pi, \mathbf{P}(t))$
- $Q = \{A, C, G, T\}$ = States
- $\pi = \{\pi_A, \pi_C, \pi_G, \pi_T\}$ = Initial Distribution
- $\mathbf{P}(t) =$

$$\begin{pmatrix} p_{A,A}(t) & p_{A,C}(t) & p_{A,G}(t) & p_{A,T}(t) \\ p_{C,A}(t) & p_{C,C}(t) & p_{C,G}(t) & p_{C,T}(t) \\ p_{G,A}(t) & p_{G,C}(t) & p_{G,G}(t) & p_{G,T}(t) \\ p_{T,A}(t) & p_{T,C}(t) & p_{T,G}(t) & p_{T,T}(t) \end{pmatrix}$$

Markov Process (Contd.)

- $P_{\sigma,\tau}(t)$
  $= Pr[\sigma \mid \tau, t] = Pr[X(t) = \sigma \mid X(0) = \tau]$
  = Probability that a nucleotide with a value τ at time 0 mutates to a σ by time t
- $P(t+s) = P(t)P(s)$
- $p_i(t) = Pr[X(t) = i]$
  $= \sum_{k \in \{A,C,G,T\}} \pi_k p_{k,i}(t)$
- $\pi^* = \{\pi_A^*, \pi_C^*, \pi_G^*, \pi_T^*\}$ is a stationary distribution for $P(t)$

$$\forall\, t \quad \pi^* P(t) = \pi^*$$

Markov Process (Contd.)

- $P'(t)$
  $= P(t) \lim_{\Delta t \to 0} [P(\Delta t) - P(0)]/[\Delta t]$
  $= P(t)\, \Lambda$
- Solution to the differential equation:
  $P(t) = \exp(\Lambda t) = \sum_{n=0}^{\infty} \Lambda^n t^n / n!$
- Row-sum for Λ is 0:
  $\sum_j \lambda_{i,j} = \lim_{\Delta t \to 0} [\sum p_{i,j} - 1]/[\Delta t] = 0.$

Juke-Cantor Model

- $(\pi_A, \pi_T, \pi_C, \pi_G) = (1/4, 1/4, 1/4, 1/4)$

- $\Lambda = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix}$

- $\pi = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix}$

Juke-Cantor Model (Contd.)

- $\Lambda = -4\alpha(1-\pi)$
- $P(t) = e^{-4\alpha(1-\pi)t}$

  $= I\,[\sum_{n=0}^{\infty}(-4\alpha\, t)^n/n!]\;\{\sum_{n=0}^{\infty}\pi^n(4\alpha\, t)^n/n!\}$

  $= I\, e^{-4\alpha t}\{I + \pi(e^{4\alpha t} - 1)\}$

  $= e^{-4\alpha t} I + \pi(1 - e^{-4\alpha t})$
- $p_{i,i}(t) = 1/4(1 + 3e^{-4\alpha t})$
- $p_{i,j}(t) = 1/4(1 - 4e^{-4\alpha t}),\quad i \neq j.$

Example

- (Based on mtDNA Sequences)
- Let q = the proportions of nucleotides that is same in two mtDNA sequences.
- K =6 α t expected number of substitutions

$$q = 1/4(1 + 3\, e^{-3/2K});$$

$$K = (2/3)\ \ln\ (3/(4q-1))$$

- Juke-Cantor distance between a pair of mtDNA sequences is given by

$$K' = (2/3)\ \ln\ (3/(4q-1))$$

Example (Contd.)

- Differences in mtDNA sequences

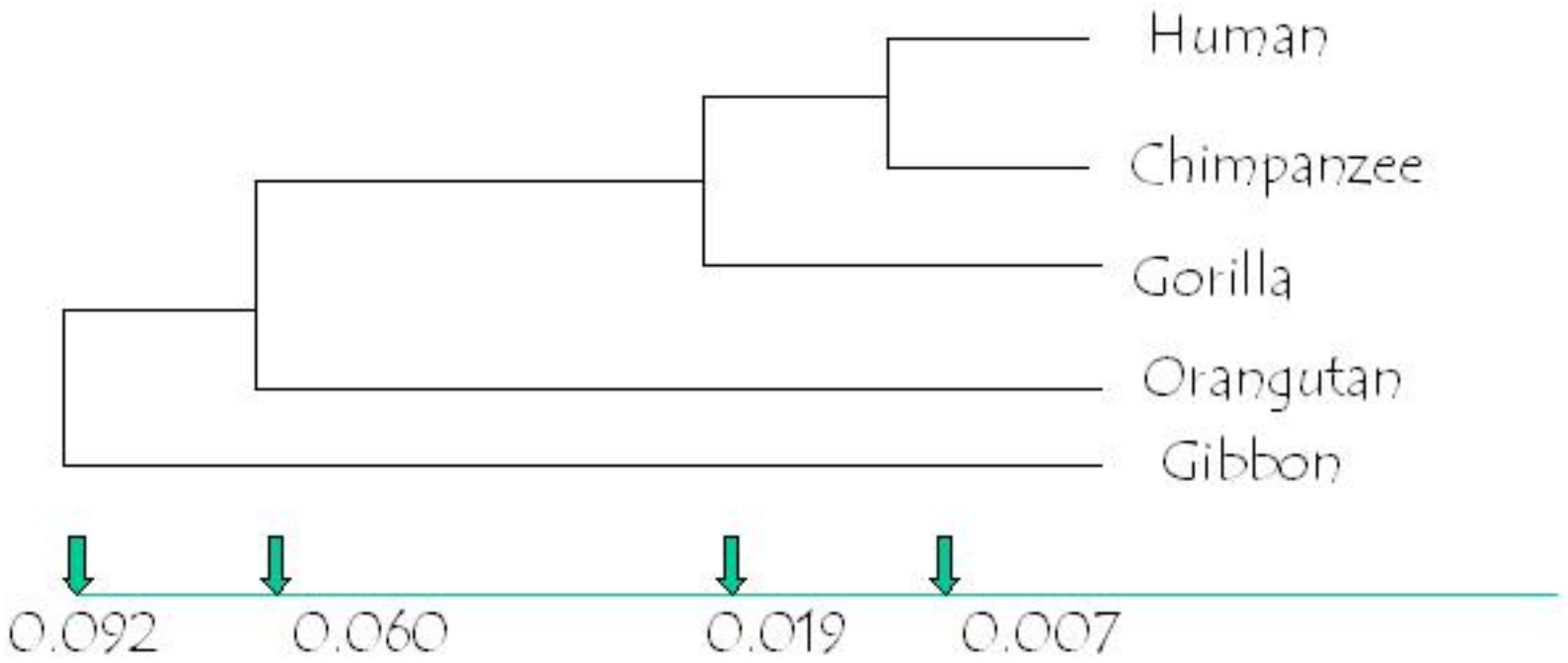
	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	-	1	3	9	12
Chimpanzee		-	2	8	11
Gorilla			-	6	11
Orangutan				-	11
Gibbon					-

Example (Contd.)

- Juke-Cantor distances between primates

	Human	Chimpanzee	Gorilla	Orangutan	Gibbon
Human	-	0.015	0.045	0.143	0.198
Chimpanzee		-	0.030	0.126	0.179
Gorilla			-	0.092	0.179
Orangutan				-	0.179
Gibbon					-

UPGMA Phylogeny



Kimura's Model

- More realistic than Juke-Cantor
- Kimura (1980) proposed a two parameter model

$\Lambda =$

$$\begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & G & C & T \end{array} \\ \begin{pmatrix} -\alpha-2\beta & \alpha & \beta & \beta \\ \alpha & -\alpha-2\beta & \beta & \beta \\ \beta & \beta & -\alpha-2\beta & \alpha \\ \beta & \beta & \alpha & -\alpha-2\beta \end{pmatrix} \end{array}$$

- Thus, $p'_{i,j}(t) = \sum_k p_{i,k}(t)\, \lambda_{k,j}$

Kolmogorov's b.w. & f.w. eqns.

- The transition probability of Kimura's two parameter model is

$$p_{AA}(t) = (1/4) + (1/4)\, e^{-4\beta t} + (1/2) e^{-2(\alpha+\beta)t}$$

$$p_{AG}(t) = (1/4) + (1/4)\, e^{-4\beta t} - (1/2) e^{-2(\alpha+\beta)t}$$

$$p_{AC}(t) = (1/4) - (1/4)\, e^{-4\beta t}$$

$$p_{AT}(t) = (1/4) - (1/4)\, e^{-4\beta t}$$

- All $p_{i,j}(t) \to 1/4$ as $t \to \infty$ &
- $p_{\cdot,\cdot}(t) > 1/4 > p_{\cdot,\cdot}(t)$, $p_{AT}(t)$

Measuring Distance with Kimura's model

- Frequencies of transitional and transversal changes between two sequences= q, & r, respectively. $\mapsto$ Q & R, are their expected values:
- $Q = p_{AG}(2t) = 1/4 + 1/4\, e^{-8\beta t} - 1/2\, e^{-4(\alpha+\beta)t}$
- $R = p_{AC}(2t) + p_{AT}(2t) = 1/2 - 1/2\, e^{-8\beta t}$
- $1-2Q-R = e^{-4(\alpha+\beta)t}$
- $1-2R = e^{-8\beta t}$
- Expected number of substitutions: $(\alpha+2\beta)t$...estimated as
- $K \propto -1/2 \ln(1-2q-r) - 1/4 \ln(1-2r)$

Kimura's 6 parameter model

- Kimura's 2 parameter model differentiates between transitions and transversions... but in equilibrium all four bases have equal frequencies.
- However in the heavystrand of human mtDNA, the frequencies are unequal!

A	T	C	G
0.247	0.313	0.302	0.139

- Kimura's 6 parameter model addresses it!

Kimura's 6-par model

$\Lambda =$

$$\begin{array}{c} \\ A \\ G \\ C \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & G & C & T \end{array} \\ \begin{pmatrix} -2\alpha-\gamma_1 & \gamma_1 & \alpha & \alpha \\ \delta_1 & -2\alpha-\delta_1 & \alpha & \alpha \\ \beta & \beta & -2\beta-\gamma_2 & \gamma_2 \\ \beta & \beta & \delta_2 & -2\beta-\delta_2 \end{pmatrix} \end{array}$$

UPGMA & The Molecular Clock

- Assumes a *constant molecular clock*:
  - Divergence of sequences is assumed to occur at the same rate at all points in the tree.
- This assumption is in general false
  - Selection pressures vary across time periods, organisms, genes within an organism, regions within a gene.

Ultrametric Trees

- Distance function ρ satisfies the axioms:

  $\rho(i, j) \geq 0$ with equality iff $i = j$;

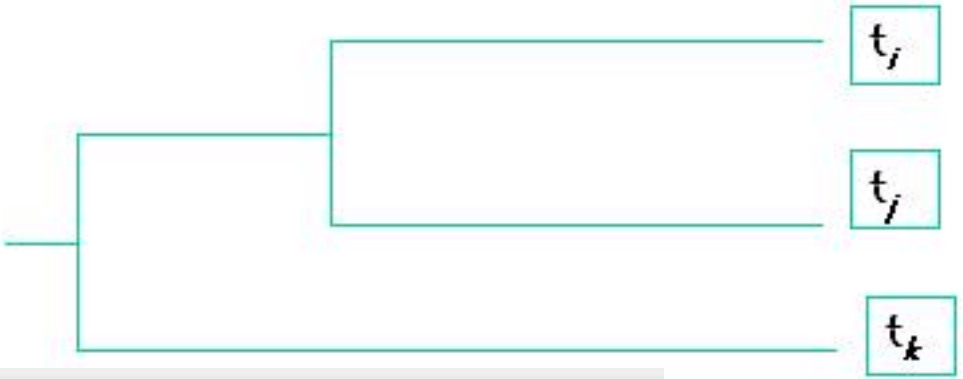
  $\rho(i,j) = \rho(j,i)$ (symmetry);

  $\rho(i,k) \leq \rho(i,j) + \rho(j,k)$ (triangle inequality).

- Path length between i, j of T = Sum of edge weights along the path connecting i and j.
- If $\forall$ i and j, $\rho_{i,j}$ = path length between i, j of T, then ρ is called an additive tree metric.
- If the path length from the root to every leaf is identical then ρ is called an ultrametric.

UPGMA & Ultrametric Data

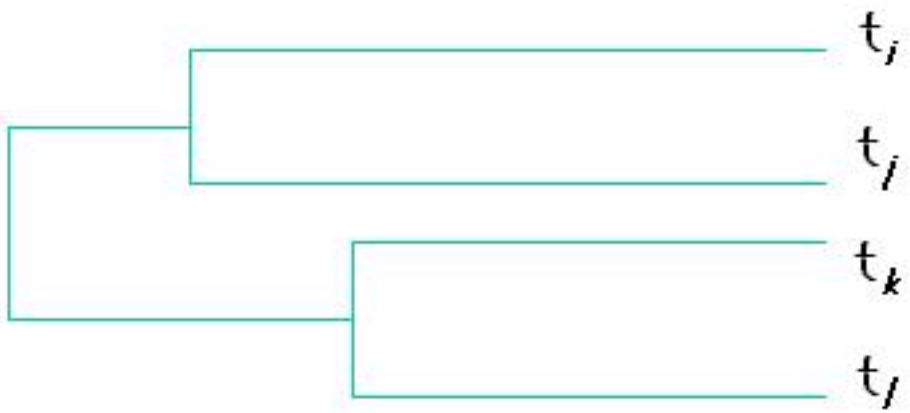
- If the rates of evolution among different lineages are exactly the same, then the data is **ultrametric**.
- Definition (3-Point Condition): For any triplet of sequences, $i \neq j \neq k$, of the three distances d_{ij}, d_{jk}, d_{jk} r two are equal and not less than the third.



$$d_{i,k} = d_{j,k} \geq d_{i,j}$$

Additive Metric

- Every additive metric satisfies the 4-point condition:
- $\forall\ i, j, k, l$, of the three sums $S_1 = d_{i,j} + d_{k,l}$, $S_2 = d_{i,k} + d_{j,l}$ and $S_3 = d_{i,l} + d_{k,j}$ two are equal and not less than the third. E.g. $S_1 \leq S_2 = S_3$.



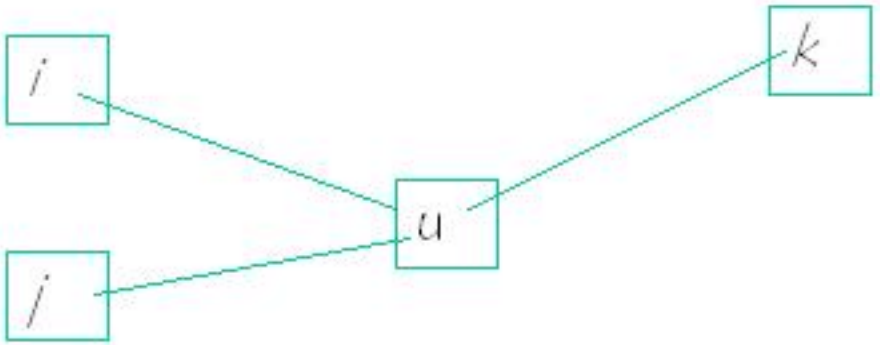
Neighbor Joining

- Like UPGMA constructs a tree by sequentially joining subtrees
- Unlike UPGMA
  - Does not make molecular clock assumption
  - Produces unrooted tree
- Does assume additivity: Distance between a pair of taxa is the path length in the tree.

Distances in Neighbor Joining

- Given a new internal node u, the distance to another node k is given by

$$s_{u,k} = (d_{i,k} + d_{j,k} - d_{i,j})/2$$

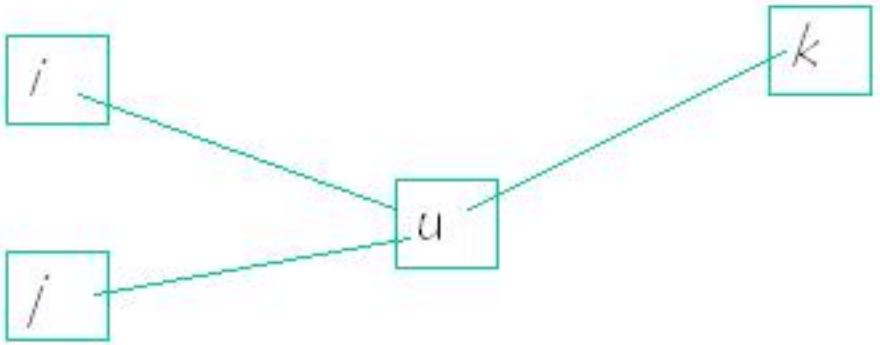


Distances in Neighbor Joining

- Calculate the distance from a leaf to its parent node similarly:

$$s_{i,u} = (d_{i,j} + d_{i,k} - d_{j,k})/2 = d_{i,j}/2 + (d_{i,k} - d_{j,k})/2$$

$$s_{j,u} = d_{i,j} - s_{i,u}$$

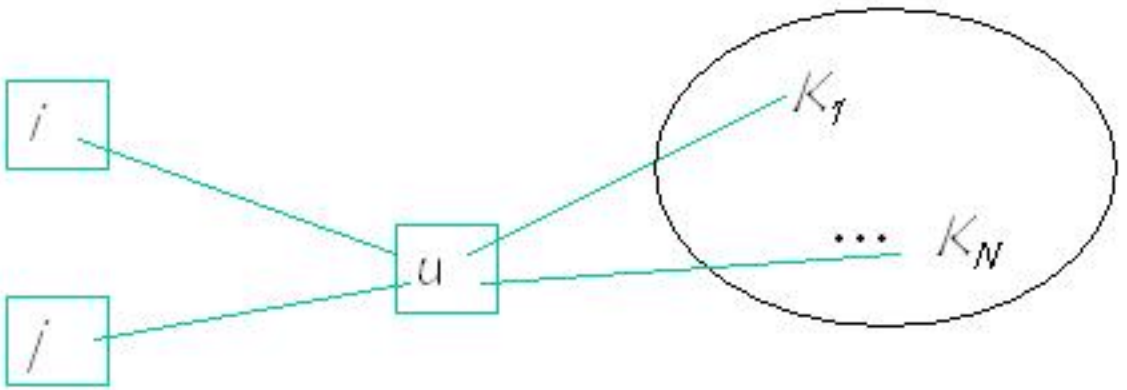


Generalizing the Scheme

- (To more than Three Leaves)
- Define $r_i = \sum_{k=1}^{N} d_{i,k}$
- Rate corrected distance between taxa i and j:

$$m_{i,j} = d_{i,j} - (r_i + r_j)/(N-2)$$

is used to choose the "nearest neighbors" to be joined.

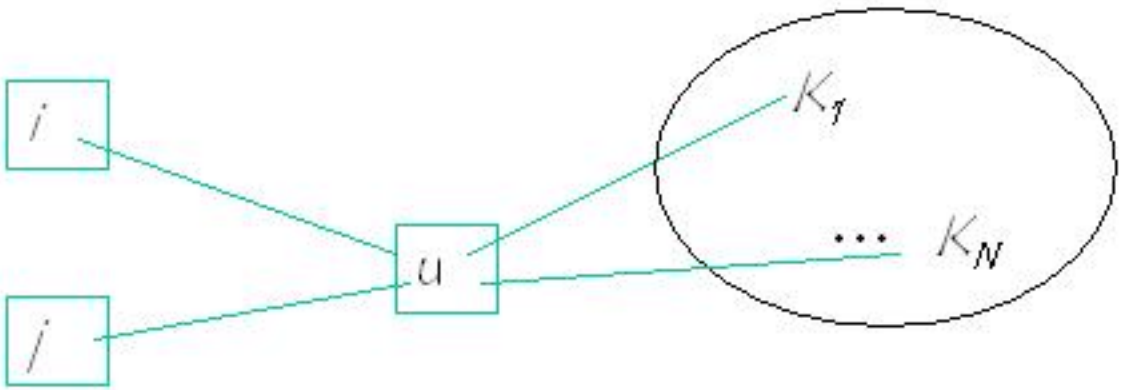


Generalizing Distances in NJ

- Calculate the distance from a leaf to its parent node similarly:

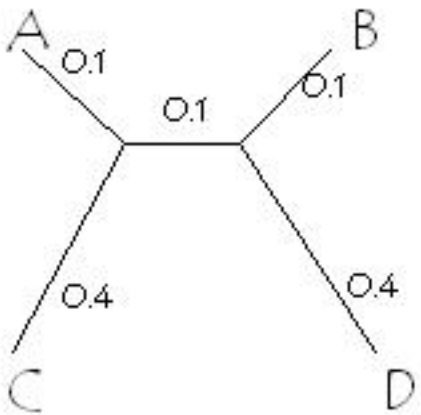
$$s_{i,u} = d_{i,j}/2 + (r_i - r_j)/(2(N-2))$$

$$s_{j,u} = d_{i,j} - s_{i,u}$$



Picking a Pair of Nodes to Join

- At each step, pick a pair of "nearest neighbor" nodes to join- Nearest neighbor is not determined by minimal $d_{i,j}$ but $m_{i,j}$



$d_{A,B} = 0.3$

$d_{A,C} = 0.5$

$r_A = 1.4,\ r_B = 1.4,\ r_C = 2.0$

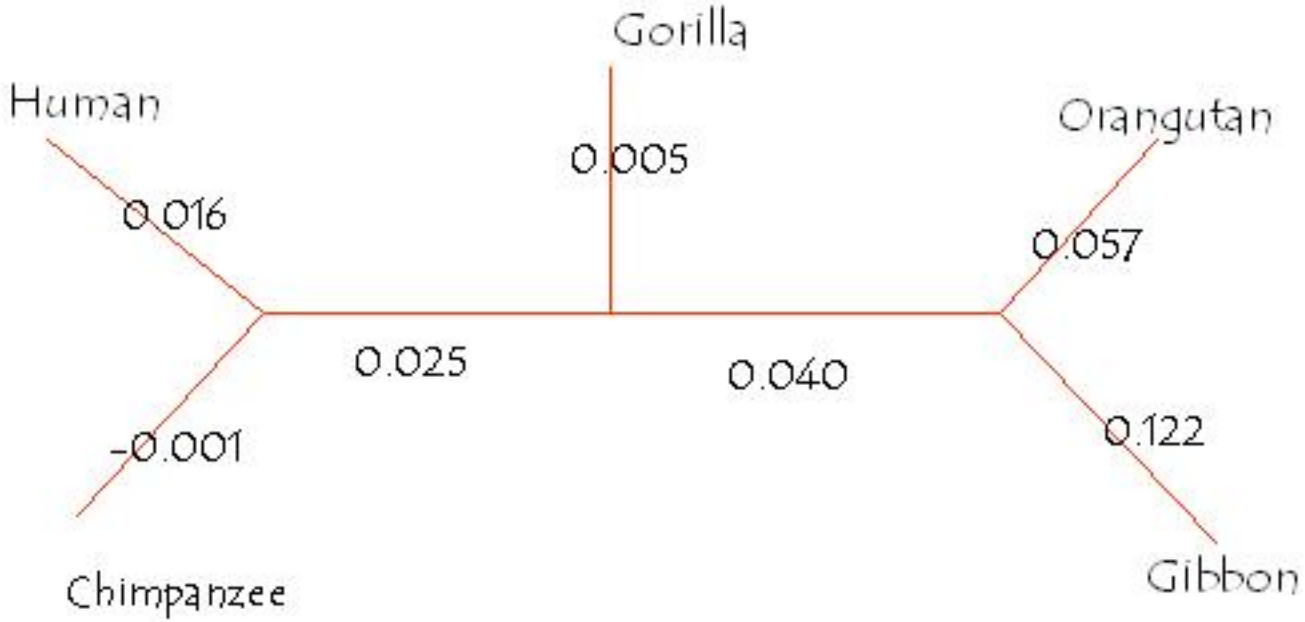
$m_{A,B} = d_{A,B} - (r_A + r_B)/2 = -1.1$

$m_{A,C} = d_{A,C} - (r_A + r_C)/2 = -1.2$

Neighbor Joining Algorithm

- T= Set of leaf nodes
- While more than two subtrees in T
  - Pick a pair i, j in T with minimal m_{ij}
  - Define a new node u joining i and j
  - Remove i and j from T and insert u
- Join the last two remaining subtrees

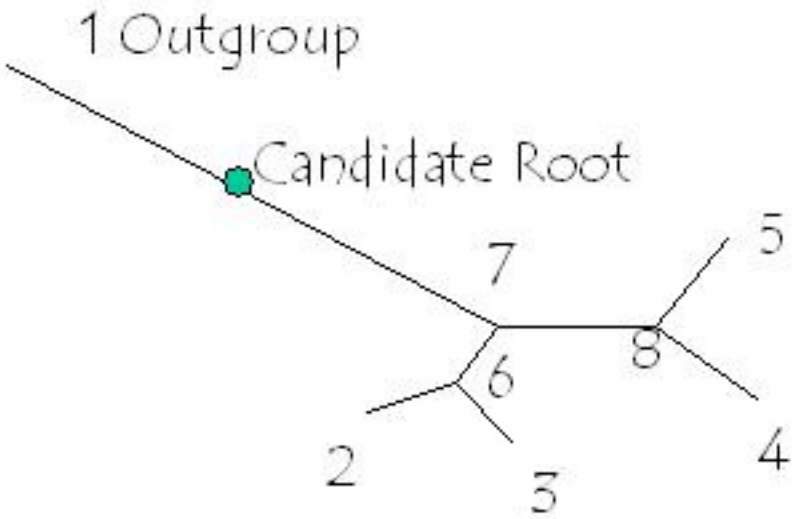
Example



Rooting Trees

- Neighbor joining method creates an unrooted phylogenetic tree.
- A root is assigned to an unrooted tree by finding an *outgroup*.
  - An outgroup is a species known to be more distantly related to remaining species than they are to each other.
  - Point where the outgroup joins the rest of the tree is best candidate for root position.

Rooting Trees



Other Distance Matrix Methods

- Phylogenetic Trees are constructed using:
  - Clustering Method: Identifies groups of close taxa. E.g. UPGMA or Average Linkage Clustering Methods.
    - Sequential
    - Agglomerative
    - Hierarchical
    - Nonoverlapping
  - Pairwise Method: Pairs a taxon (or a group of taxa) with its nearest neighbor. E.g. Additive trees constructed with Fitch-Margolish Algorithm.

Matching and Alignment

Inexact Matching

- Example: Edit Distance Problem:
  - Edit distance between two biological sequences
  - May correspond to:
    - Evolutionary Distance
    - Functional Distance
    - Structural Distance

Edit Distance

- Simplest distance function corresponds to:

  EDIT DISTANCE

- Atomic Edit Functions:
  - Insertion AATCGG $\mapsto$ AATACGG
  - Deletion AATACGG $\mapsto$ AATCGG
  - Substitution AATCGG $\mapsto$ AATAGG
- A composite edit function

  $\simeq$ Function Composition of Atomic Edit Functions

Cost of a Composite Edit Function

- (Based on the cost or distance for Atomic Edit Functions)
- Given: Two strings S_1 and S_2

$$\text{Distance}(S_1, S_2) = \min \{ \text{cost}(E) \mid E(S_1) = S_2 \}$$

Where

E = composite edit function mapping S_1 to S_2.

Some Properties of Distance Function

- Assume:

  $(\forall\ e$= Atomic Edit Function$)$ cost(e) = cost(e^{-1})

  - Distance(S_1, S_2) = Distance(S_2, S_1) *Symmetric*
  - Distance$(S_1, S_1) = 0$
  - Distance(S_1, S_2) + Distance$(S_2, S_3) \geq$ Distance(S_1, S_3)

    Triangle Inequality

- Simplest Cost Function:

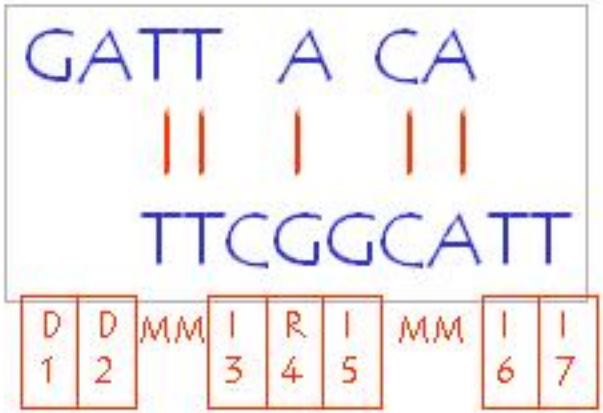
  - *Each atomic edit function is of unit cost.*

Edit Operations

- I: Insertion of a character into the first string S_1
- D: Deletion of a character from the first string S_1
- R: Replacement (or Substitution) of a character in the first string S_1 with a character in the second string S_2
- M: Matching (Identity)

Edit Transcript

Example:



- The complete edit function is described by an "edit transcript"
- EDIT TRANSCRIPT

  $= \sigma \in \{D, M, R, I\}^*$
- Example (in left):
  - Edit transcript = DDMIRIMMII
  - Edit Distance = 1+1+0+1+1+1+0+0+1+1=7

Levenshtein (or Edit) Distance

- Edit Distance between two strings S_1 and S_2 is defined as the minimum number of atomic edit operations – insertions, deletions (indels), and substitutions – needed to transform the first string into the second
- Optimal Transcript = An edit transcript corresponding to the minimum number of atomic edit operations of unit cost.

EDP

The Edit Distance Problem

is to compute

- the edit distance between two given strings, along with
- an optimal edit transcript that describes the

ment)

Dynamic Programming Calculation of Edit Distance

- Define:

$D(i,j) \equiv$ Min number of atomic edit operations needed to transform the first i characters of S_1 into the first j characters of S_2

$\equiv$ EditDistance($S_1[1..i]$, $S_2[1..j]$)

- $|S_1| = n \quad |S_2| = m$

$$\text{Distance}(S_1, S_2) = D(n,m)$$

- Dynamic Programming: 3 components:
  - Recurrence Relation
  - Tabular Computation
  - Traceback

Recurrence

- Base Relation:
  - D(0,0) = 0
  - EditDistance(λ,λ) = 0
- Recurrence Relations:
  - <u>In 1 coordinate:</u>

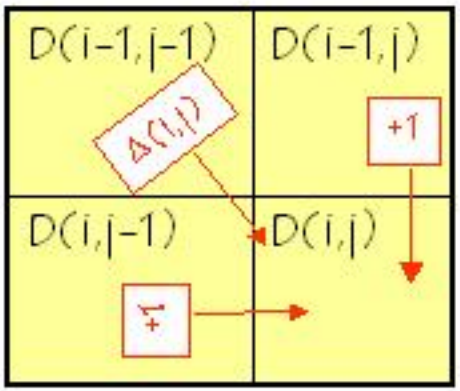
    D(i,0) = D(i-1,0)+1 (S_1[i] deleted)

    D(0,j) = D(0,j-1)+1 (S_2[j] inserted)
  - EditDistance(S_1[1..i], λ) =i (i deletions)
  - EditDistance(λ, S_2[1..j]) =j (j insertions)
  - <u>In both coordiates:</u>

    D(i,j) = min {D(i-1,j) +1, (S_1[i] deleted)

    {D(i,j-1)+1, (S_2[j] inserted)

    {D(i-1,j-1)+ Δ(i,j)} (substn or match)
  - Δ(i,j) = 1, if S_1[i] ≠ S_2[j]; 0 otherwise.



Efficient Tabular Computation of Edit Distance

- Recursive Implementation $\mapsto 2^{O(n+m)}$-time computation
- Bottom-up computation

  $(n+1) \times (m+1)$ distinct values for $D(i,j)$ to be computed
- Dynamic Programming Table of size $(n+1) \times (m+1)$
  - String S_1 corresponds to the rows (Vertical Axis)
  - String S_2 corresponds to the columns (Horizontal Axis)
- Fill out $D(i,0) \leftarrow$ First Column
- Fill out $D(0,j) \leftarrow$ First Row
- Fill out rows $D(i,j) \leftarrow$ Left-to-Right (increasing i)

The Algorithm

- for i=0 to n do
  D(i,0) ← i;
  for j=0 to m do
  D(0,j) ← j;
  for i=1 to n do
  for j=1 to m do
  D(i,j) ← min[D(i-1,j)+1,
  D(i,j-1)+1,
  D(i-1,j-1) + Δ(i,j)]

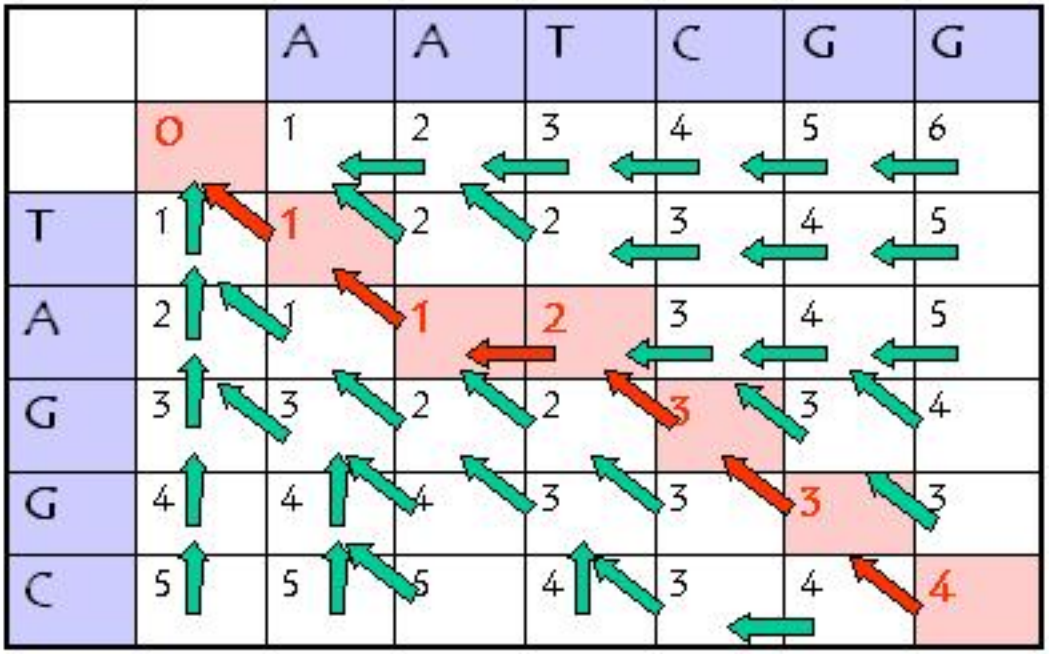
□

- Time complexity = O(nm)

Example

$S_2 \rightarrow$

$S_1 \downarrow$



Solutions with minimal edit distance =4

(sol.1) TA-GGC
AATCGG
1 23 4

(Edit Script= RMIRMR)

(sol.2) - -TAGGC
AATCG G -
1 2 3 4

(Edit Script = IIMRMMD)

Trace Back

- Extracting Optimal Edit Transcript:
- Set a pointer from:
  - Cell(i,j) → Cell(i,j-1), if D(i,j) = D(i,j-1)+1
    Horizontal Edge ⇒ I, Insertion
  - Cell(i,j) → Cell(i-1,j), if D(i,j) = D(i-1,j)+1
    Vertical Edge ⇒ D, Deletion
  - Cell(i,j) → Cell(i-1,j-1), if D(i,j) = D(i-1,j-1)+Δ(i,j)
    Diagonal Edge ⇒ R, Substitution, if Δ(i,j)=1
    M. Match, if Δ(i,j) = 0.
- Optimal Edit Transcript can be computed in O(n+m) additional time.

GAPS: The Scoring Model

- Basic operations:
  - Sequencing Errors or Evolutionary processes of Mutations and Selections
  - Substitution: Changes one base to another.
  - Gaps: Insertions or Deletions:

    Adds or removes a base.
- Total Score Assigned to an Alignment=
  - Sum of terms for each aligned pair of bases plus terms for each gap.

Total Score of an Alignment with Gaps

- Total Score Assigned to an Alignment
  - Corresponds to log of the
  - Relative likelihood that the two sequences are related compared to being unrelated.
- Assumptions:
  - Mutations or Sequencing Errors at different sites in a sequence occur independently.

Substitution Matrices

- Notation: x and y ≡ Pairs of sequences,

$$|x| = n \text{ and } |y| = m.$$

$$x, y \in (A+G+C+T)^*$$

  - x_i = i^{th} symbol in x
  - y_j = j^{th} symbol in y
- Random Model, R:

$$P(x,y|R) = \prod q_{xi} \prod q_{yj}$$

  - q_a = probability that the letter "a" occurs independently at a given site.

Random Model vs. Alternative Model

- Alternative Model, M:
- $P(x,y \mid M) = \prod p_{xi,\, yj}$
  - p_{ab} = Probability that the letters "a" and "b" have each been derived independently from some common letter.
- Log-Odds Ratio (LOD):

$$s(a,b) = \ln (p_{ab}/q_a\, q_b)$$

- $P(x,y|M)/P(x,y|R) = \prod (p_{xi\, yi}/\, q_{xi}\, q_{yi})$

$$= \prod \exp[s(x_i,y_i)] = \exp[\sum s(x_i,\, y_i)]$$

Score

- Score = ln [P(x,y|M)/P(x,y|R)]

  $= \sum s(x_i, y_i) = s(x,y)$

- Score Matrix or Substitution Matrix:

	A	T	C	G
A	2	-1	-1	-1
T	-1	2	-1	-1
C	-1	-1	2	-1
G	-1	-1	-1	2

- Blossum 50
- PAM
  - (Point Accepted Mutation)

1-PAM Matrix

- Let M be a probability transition matrix. M_{ab} = Pr(a ⇔ b),
  - a, b = chracters
- p_a = Pr("a" occurs in a string)
- f_{ab} = The number of times the mutation a ⇔ b was observed to occur.
- $f_a = \sum_{a \neq b} f_{ab}$ & $f = \sum f_a$.
- K = 1-PAM Evolutionary distance
  - "The amount of evolution that will change 1 in K characters on average."

1-PAM Matrix

- $m_a = f_a / (Kf\, p_a)$,
  $M_{aa} = 1 - m_a$, $M_{ab} = f_{ab} / (Kf\, p_a) = (f_{ab}/f_a)\, m_a$
- α-PAM Matrix = M^α
- $M^* = \lim_{\alpha \leftarrow \infty} M^\alpha$
- $\text{Score}_\alpha(a,b) = 10 \log_{10} M^\alpha{}_{ab}/p_b$
- Sequence comparison with 40 PAM, 120 PAM & 250 PAM score functions…

Gaps:

- g = length of a gap,

  exponentially distributed $\sim$ Exp(λ)

  $$f(g) = \lambda e^{-\lambda g}$$

- $P(g) = f(g) \prod q_{xi}$

$\ln P(g) = -\lambda g + \ln \lambda + \text{sum } \ln q_{xi}$

$= -d - (g-1)\, e$ (Affine Score Model)

- d = Gap-open Penalty
- e = Gap-Extension Penalty

Multiple Sequence Alignment

- Defn: Given strings $S_1, S_2, \ldots S_k$ a multiple (global) alignment maps them to strings $S'_1, S'_2, \ldots, S'_k$ (by inserting chosen spaces) such that

1. $|S'_1| = |S'_2| = \cdots = |S'_k|$, and
2. Removal of spaces from S'_i contracts it to S_i, for $1 \leq i \leq k$.

Value of a Multiple Global Alignment

- The sum of pairs (SP) value for a multiple global alignment $\mathcal{A}$ of k strings is the sum of the values of all $C_{k,2}$ pairwise alignments induced by $\mathcal{A}$.
- Given: Two strings S_1 and S_2. The expanded strings S'_1 and S'_2 correspond to a pairwise alignment.
- $\delta(x, y)$ = distance between two characters x and y

  = 1, if $x \neq y$ and 0, if $x = y$.
- $\delta(x,-) = \delta(-,y) = 1$.
- Distance$(S'_1, S'_2) = \sum_{i=1}^{l} \delta(S'_1[i], S'_2[i])$,

  where $l = |S'_1| = |S'_2|$.

Optimal Global Alignment

- An optimal SP(global) alignment of strings $S_1, S_2, \ldots, S_k$ is an alignment that has a minimum possible sum-of-pairs value for these strings among all possible multiple sequence alignments.

Generalization of DP

- Assume $|S_1| = |S_2| = \cdots = |S_k| = n$.
- The generalized k-dimensional DP table has $(n+1)^k$ entries.
- **Each entry depends on $2^k - 1$ adjacent entries.**
  - $D(i_1, 0, \ldots, 0) = i_1$
  - $D(0, i_2, \ldots, 0) = i_2$

    $\vdots$
  - $D(0, 0, \ldots, i_k) = i_k$
  - $D(i_1, i_2, \ldots i_k) = \min_{\emptyset \neq S \subseteq \{1..k\}} [$
    $D[\ldots, i_j - 1, \ldots]_{j \in S}$
    $+ \sum_{l \neq m \in S} \delta(i_l, i_m) + |S| \times (n - |S|)$
    $]$

Complexity

- The time and space complexity of the generalized DP solution of the multiple alignment problem is = $O((2n)^k)$
- Theorem: *The optimal SP alignment problem is NP-complete.*
- In the worst-case, one cannot expect to do much better unless *P=NP.*

P-Time Heuristics

- A Polynomial Time Approximate Algorithm for Multiple String Alignment:
- Assumption about the distance function:
  - Triangle Inequality:

    $\forall_{chars,\, x,y,z}\ \delta(x,z) \leq \delta(x,y) + \delta(y,z)$
  - $\forall_{char,\, x}\ \delta(x, x) = 0$
- $D(S_1, S_2)$

  $\triangleq$ Value of the min. global alignment of S_1 & S_2.

Algorithm

- Input: $\mathcal{T} = \{S_1, S_2, \ldots, S_k\}$
- Step 1: Find $S_1 \in \mathcal{T}$ that minimizes

$$\sum_{S \in \mathcal{T} \setminus \{S1\}} D(S_1, S)$$

  - Time Complexity = $O(k^2 n^2)$
    - $\mapsto C_{k,2}$ DP each taking $O(n^2)$ time
  - Call the remaining strings $S_2, \ldots, S_k$

The i^{th} Step

- Step i: Assume $S_1, \ldots S_{i-1}$ have been aligned as $S'_1, \ldots, S'_{i-1}$
- Add S_i: Run DP to align S'_1 & $S_i \mapsto S''_i$ and S'_i
  - Adjust $S'_2, \ldots, S'_{i-1}$ by adding spaces where spaces were added in S'_1
- $S_1, S_2, \ldots, S_i \Rightarrow_{\text{aligned}} S'_1, S'_2, \ldots, S'_i$.
  - Length(S'_1) in step i $\leq$ i n.
  - DP(S'_1, S_i) takes $O(i\, n^2)$ time
- Total time Complexity

  $= O(k^2 n^2) + \sum_{i=1}^{k} O(i\, n^2) = O(k^2 n^2)$

Competitiveness

- $\mathcal{M}$ = Alignment induced by the algorithm
- $d(i,j)$ = Distance $\mathcal{M}$ induces on pair S_i, S_j
- $\mathcal{M}^*$ = Optimal alignment
- $2\,SP(\mathcal{M}) = \sum_{i=1}^{k} \sum_{j=1,\, j \neq i}^{k} d(i,j)$

  $\leq \sum_{i=1}^{k} \sum_{j=1,\, j \neq i}^{k} d(i,1) + d(1,j)$

  (Triangle Inequality)

  $= \sum_{i=1}^{k} \sum_{j=1,\, j \neq i}^{k} d(1,i) + \sum_{i=1}^{k} \sum_{j=1,\, j \neq i}^{k} d(1,j)$

  (Symmetry)

  $= \sum_{i=2}^{k} (k-1)\, d(1,i) + \sum_{j=2}^{k} (k-1)\, d(1,j)$

  $= 2(k-1) \sum_{i=2}^{k} d(1,i)$

Competitiveness

- $2SP(\mathcal{M}') \geq \sum_{i=1}^{k} \sum_{j=1,\, j\neq i}^{k} D(S_i, S_j)$

  $= \sum_{j=2}^{k} D(S_1, S_j) + \sum_{j=1,\, j\neq 2}^{k} D(S_2, S_j)$

  $+ \cdots + \sum_{j=1}^{k-1} D(S_k, S_j)$

  $\geq k \sum_{i=2}^{k} d(1,i)$

- $SP(\mathcal{M}') \geq (k/2) \sum_{i=2}^{k} d(1,i)$

  $\geq (k/2)\, [SP(\mathcal{M})/(k-1)]$

- $SP(\mathcal{M}) \leq 2\,(1 - 1/k)\, SP(\mathcal{M}^*)$

To be continued…

. . .