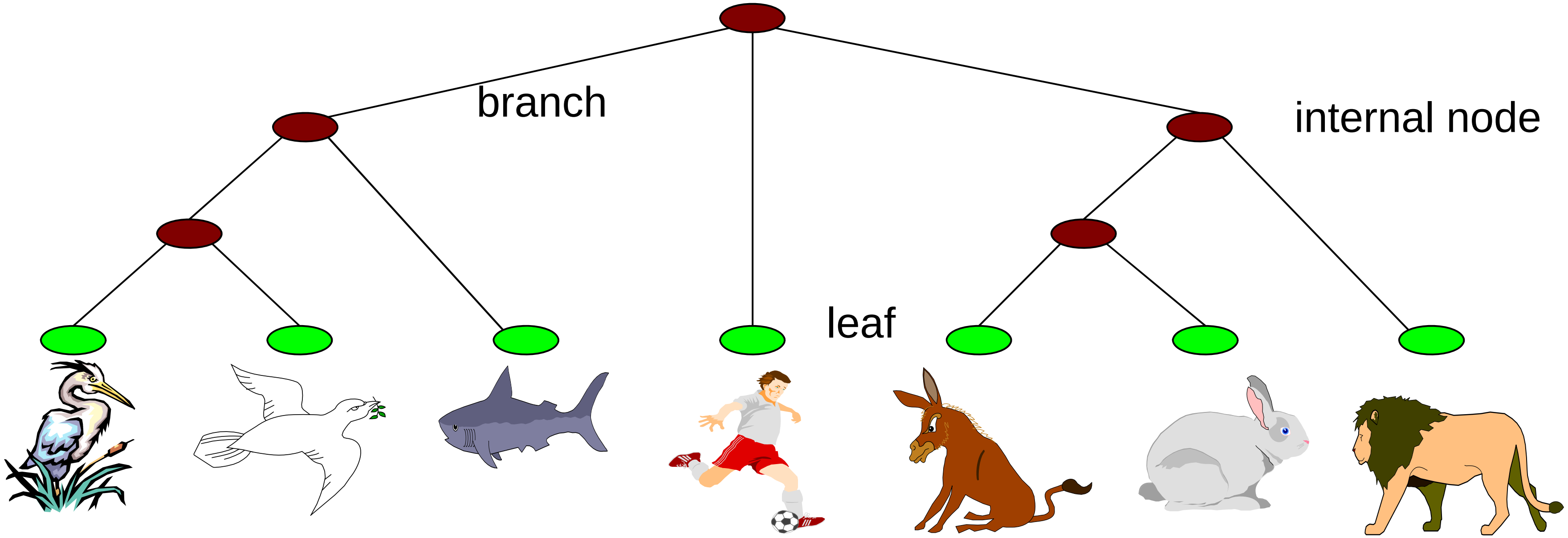
第4章：进化树构建的概率方法

- 问题介绍
- 进化树构建方法的概率方法

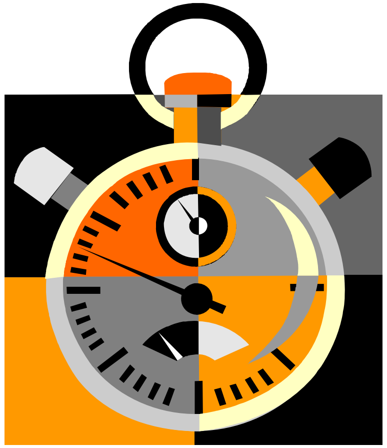
部分Slides修改自University of Basel的Michael Springmann课程“CS302 Seminar Life Science Informatics”的讲义

Phylogenetic Tree



- Topology: bifurcating
  - Leaves - *1…N*
  - Internal nodes *N+1…2N-2*
- Branch length

Molecular Clock Hypothesis



- Amount of genetic difference between sequences is a function of time since separation.
- Rate of molecular change is constant (enough) to predict times of divergence

Likelihood of a Tree

- Given:
  - n aligned sequences M= $X_1,\ldots,X_n$
  - A tree T, leaves labeled with $X_1,\ldots,X_n$
- Reconstruction t*:
  - Labeling of internal nodes
  - Branch lengths

Goal: Find optimal reconstruction t* : One maximizing the likelihood P(M|T, t*)

Probabilistic Methods

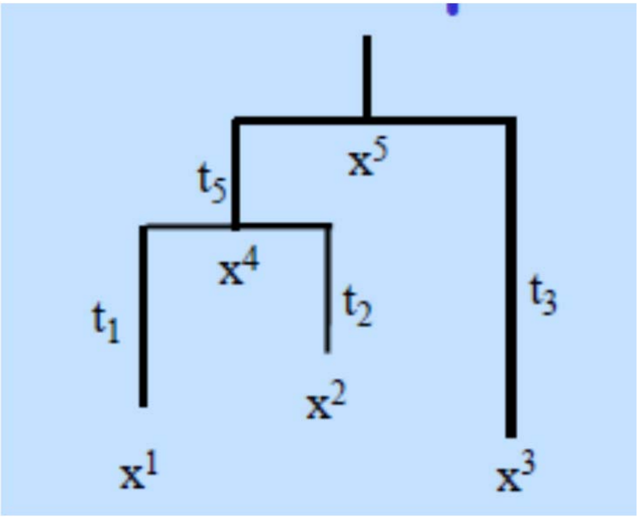
- The phylogenetic tree represents a generative probabilistic model (like HMMs) for the observed sequences.
- Background probabilities: q(a)
- Mutation probabilities: P(a|b, t)
- Models for evolutionary mutations
  - Jukes Cantor
  - Kimura 2-parameter model
- Such models are used to derive the probabilities

Probabilistic Model

- Assumptions:
  - Each character is independent
  - The branching is a Markov process: The probability that a node x has a specific label is only a function of the parent node y and the branch length t between them
  - The probabilities P(x|y,t) are known

Example

- Given then tree



$$P(x_1, x_2, x_3, x_4, x_5 | T, t^*)$$
$$= P(x_1 | x_4, t_1) P(x_2 | x_4, t_2) P(x_3 | x_5, t_3) P(x_4 | x_5, t_5)$$

Molecular Evolution

Q: How can we model evolution on nucleotide level? (ignore gaps, focus on substitutions)

A: Consider what happens at a specific position for small time interval Δt

- $P(t)$ = vector of probabilities of {A,C,G,T} at time t
- μ_{AC} = rate of transition from A to C per unit time
- $\mu_A = \mu_{AC} + \mu_{AG} + \mu_{AT}$ rate of transition out of A
- $p_A(t+\Delta t) = p_A(t) - p_A(t)\ \mu_A\ \Delta t + p_C(t)\ \mu_{CA}\ \Delta t + \ldots$

Molecular Evolution

In matrix/vector notation, we get

$$P(t + \Delta t) = P(t) + QP(t)\Delta t$$

where Q is the substitution rate matrix

$$Q = \begin{pmatrix} -\mu_A & \mu_{AG} & \mu_{AC} & \mu_{AT} \\ \mu_{GA} & -\mu_G & \mu_{GC} & \mu_{GT} \\ \mu_{CA} & \mu_{CG} & -\mu_C & \mu_{CT} \\ \mu_{TA} & \mu_{TG} & \mu_{TC} & -\mu_T \end{pmatrix}$$

Molecular Evolution

- This is a differential equation:

$$P'(t) = Q\,P(t)$$

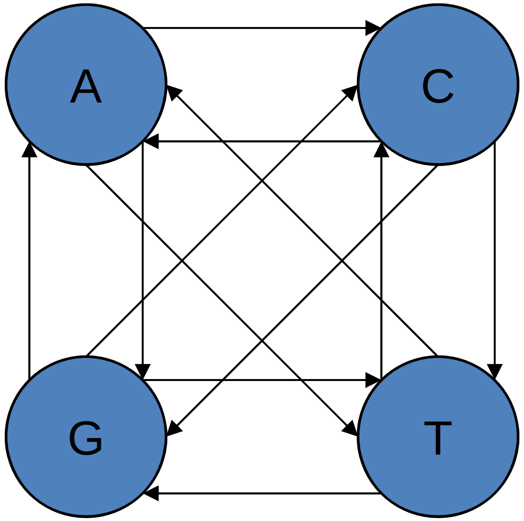
- A substitution rate matrix Q implies a probability distribution over {A,C,G,T} at each position, including stationary (equilibrium) frequencies π_A, π_C, π_G, π_T
- Each Q is an evolutionary model (some work better than others)

Mutation Probabilities

P(t) satisfy the following two property:

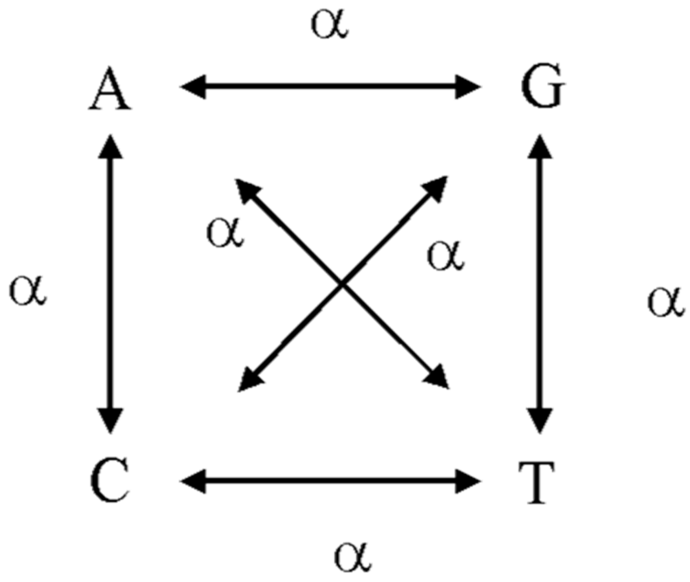
- **Lack of memory**:
  - $P_{a\to c}(t+t') = \sum_b P_{a\to b}(t) P_{b\to c}(t')$
- **Reversibility:**
  - Exist stationary probabilities $\{P_a\}$ s.t.

$$P_a P_{a\to b}(t) = P_b P_{b\to a}(t)$$



Jukes Cantor model

- Mutation occurs at a constant rate
- Each nucleotide is equally likely to mutate into any other nucleotide with rate a.



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

Substitution Matrix

- 由对称性，可设

$$P(t) = \begin{pmatrix} \gamma(t) & s(t) & s(t) & s(t) \\ s(t) & \gamma(t) & s(t) & s(t) \\ s(t) & s(t) & \gamma(t) & s(t) \\ s(t) & s(t) & s(t) & \gamma(t) \end{pmatrix}$$

- 又由其满足的微分方程

$$\frac{dP(t)}{d(t)} = QP(t)$$

Substitution Matrix

- 可得方程

$$\begin{cases} \dfrac{d\gamma(t)}{dt} = -3\alpha\gamma(t) + 3\alpha s(t) \\ \dfrac{ds(t)}{dt} = -\alpha s(t) + \alpha\gamma(t) \end{cases}$$

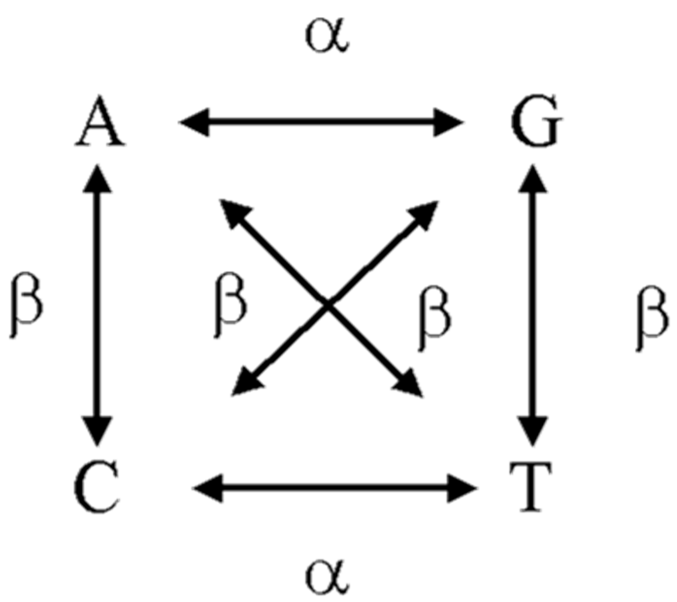
- 容易求得

$$\gamma(t) = \frac{1}{4}(1 + 3e^{-4\alpha t})$$

$$s(t) = \frac{1}{4}(1 - e^{-4\alpha t})$$

Kimura 2-parameter Model

- Allows a different rate for transitions and transversions.



$$Q = \begin{pmatrix} -\alpha - 2\beta & \beta & \alpha & \beta \\ \beta & -\alpha - 2\beta & \beta & \alpha \\ \alpha & \beta & -\alpha - 2\beta & \beta \\ \beta & \alpha & \beta & -\alpha - 2\beta \end{pmatrix}$$

Substitution Matrix

- 由对称性，可设

$$P(t) = \begin{pmatrix} \gamma(t) & s(t) & u(t) & s(t) \\ s(t) & \gamma(t) & s(t) & u(t) \\ u(t) & s(t) & \gamma(t) & s(t) \\ s(t) & u(t) & s(t) & \gamma(t) \end{pmatrix}$$

- 又由其满足的微分方程

$$\frac{dP(t)}{d(t)} = QP(t)$$

Substitution Matrix

- 可得方程

$$\begin{cases} \dfrac{d\gamma(t)}{dt} = -(2\beta+\alpha)\gamma(t) + 2\beta s(t) + \alpha u(t) \\ \dfrac{ds(t)}{dt} = -2\beta s(t) + \beta\gamma(t) + \beta u(t) \\ \dfrac{du(t)}{dt} = -(2\beta+\alpha)u(t) + 2\beta s(t) + \alpha\gamma(t) \end{cases}$$

- 容易求得

$$\begin{cases} s(t) = \dfrac{1}{4}(1 - e^{-4\beta t}) \\ s(t) = \dfrac{1}{4}(1 + e^{-4\beta t} - 2e^{-2(\alpha+\beta)t}) \\ \gamma(t) = 1 - 2s(t) - u(t) \end{cases}$$

Substitution Matrix: General Case

- 对于对称矩阵Q可以对角化, 即存在正交矩阵U, 和特征值 $\lambda_1 \geq \cdots \geq \lambda_n$, 使得

$$Q = U^T diag\{\lambda_1, \cdots, \lambda_n\} U$$

- 于是

$$P(t) = U^T diag\{e^{\lambda_1 t}, \cdots, e^{\lambda_n t}\} U$$

PAM矩阵

- Point accepted mutation (Dayhoff et al 1978)
- Given an tree of protein family, the frequence matrix A_{ab} counting the occurrence of an “a” in the ancestral sequence was replaced by a “b” in the descendant.
- Estimate the conditional probability p(b|a)

$$P(b|a) = B_{a,b} = \frac{A_{ab}}{\sum_c A_{ac}}$$

PAM矩阵

- Scaling B

$$C_{ab} = \sigma B_{ab}, C_{aa} = \sigma B_{aa} + (1 - \sigma)$$

- Such that the expected number of substitution is 1%, i.e.

$$\sum_{ab} q_a q_b C_{ab} = 0.01$$

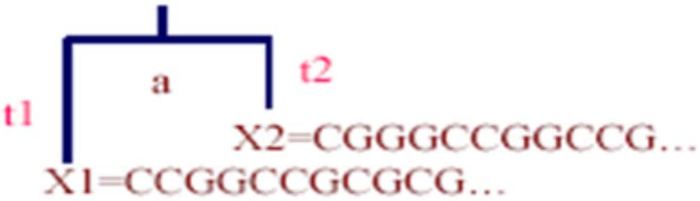
- Then the PAM(1) matrix is given by

$$S(1) = (C_{ab})$$

Calculating the Likelihood for Ungapped Alignments

$$P(X^1, X^2|T, t_1, t_2) = \prod_{u=1}^{N} P(X_u^1, X_u^2|T, t_1, t_2)$$

$$P(X_u^1, X_u^2|T, t_1, t_2) = \sum_{a} q_a P(X_u^1, |a, t_1) P(X_u^2|a, t_2)$$



- 假设突变符合JC model, 等初始概率 $q_A = q_C = q_G = q_T = \frac{1}{4}$

$$\begin{aligned} P(C, C|T, t_1, t_2) &= q_c \gamma(t_1)\gamma(t_2) + q_G s(t_1)s(t_2) + q_a s(t_1)s(t_2) + q_T s(t_1)s(t_2) \\ &= \frac{1}{3}(r(t_1)r(t_1) + 3S(t_1)S(t_2)) \end{aligned}$$

$$P(C, G|T, t_1, t_2) = P(G, C|T, t) = \frac{1}{4}(\gamma(t_1)s(t_1) + s(t_1)\gamma(t_2) + 2s(t_1)s(t_2))$$

$$P(X^1, X^2|T, t_1, t_2) = 16^{-(n_1+n_2)}(1 + 3e^{-4\alpha(t_1+t_2)})^{n_1}(1 - e^{-4\alpha(t_1+t_2)})^{n_2}$$

其中n1是匹配数，n2是不匹配数目.

Calculating the Likelihood for Ungapped Alignments

- n sequences of length N, site u=1...N
- Given a rooted tree contains *2n - 1* nodes*, 1... n* being the leaf nodes, n+1 ... 2n-1 non-leaf, tree lengths t1, …, t_{2n-1}.
- Let a(i) denote the ancestor of node a^i

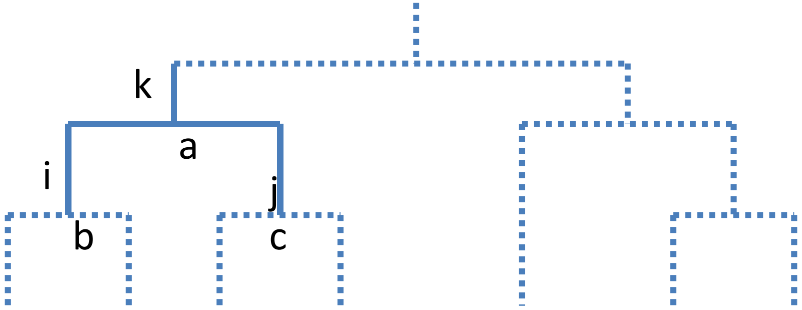
$$P(x^1, \cdots, x^n | T, t) = \prod_{u=1}^{N} P(x_u^1, \cdots, x_u^n | T, t)$$

$$P(x_u^1, \cdots, x_u^n | T, t) = \sum_{a^{n+1}, \cdots, a^{2n-1}} q_{a^{2n-1}} \prod_{i=n+1}^{2n-2} P(a^i | a^{\alpha(i)}, t_i)$$

$$\times \prod_{i=1}^{n} P(x_u^i | a^{\alpha(i)}, t_i)$$

Felsenstein's Recursive Algorithm

- Let $P(L_k|a)$ denote the probability of all the leafs below node k given that the residue at k is a.
- Then we compute $P(L_k|a)$ from the probabilities $P(L_i | b)$ and $P(L_j |c)$ for all b and c, where i and j are the daughter nodes of k.



Felsenstein's Recursive Algorithm

- Initialization: set k=2n-1
- Recursion: Compute P(L_k | a) for all a as follows:
  - – If k is leaf node: P(L_k|a)=1 only if $a = x_u^k.$
  - – If k is not a leaf node:
    - Compute P(L_i|a), P(L_j|a) for all a at the daughter nodes i,j , and set $P(L_k|a) = \sum_{bc} P(b|a,t_i)P(L_i|b)P(c|a,t_j)P(L_j|c)$
- Temination: Likelihood at site u,

$$P(x_u|T,t) = \sum_{a} P(L_{2n-1}|a)q_a$$

Reversibility & Independence of Root Position

- The score of the optimal tree is independent of the root position if and only if:

  - the substitution matrix is **multiplicative**

  - the substitution matrix is **reversible**

- A substitititution matrix is reversible if for all a,b and t:

$$P(b|a,t)q_a = P(a|b,t)q_b$$

Maximum Likelihood (ML)

- Score each tree by
  - Assumption of independent positions “m”
- Branch lengths *t* can be optimized
  - Gradient Ascent
  - EM
- We look for the highest scoring tree
  - Exhaustive
  - Sampling methods (Metropolis)

Computational Problem

- Such procedures are computationally expensive!
- Computation of optimal parameters, per candidate, requires non-trivial optimization step.
- Spend non-negligible computation on a candidate, even if it is a low scoring one.
- In practice, such learning procedures can only consider small sets of candidate structures

参考文献

- S. Durbin, S. Eddy, A. Krogh and G. Mitchison. Biological Sequence Analysis—Probabilistic Models of Proteins and Nucleic Acids. 1998, Cambridge University Press.