CLUSTERING OF BIOLOGICAL SEQUENCES

Petr Ryšavý

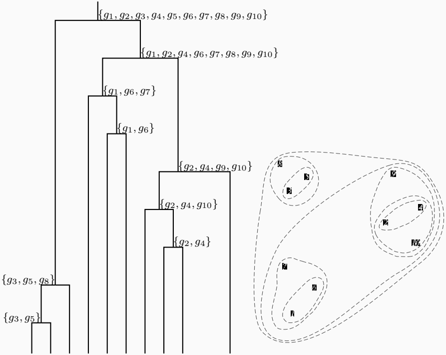
Thursday 20th October, 2016

IDA, Dept. of Computer Science, FEE, CTU

HIERARCHICAL CLUSTERING

Hierarchical clustering

- more informative than flat clustering
- agglomerative (bottom-up) or divisive (top-down)
- result of agglomerative hierarchical clustering usually in form of dendogram
- AHC runs usually in $\mathcal{O}(n^3)$, can be implemented in $\mathcal{O}(n^2 \log n)$



General algorithm

while There are more than one cluster **do**
    select two clusters and combine them into one cluster
end while

- Algorithm holds matrix of pairwise distances $\mathbf{D}$
- Two closest clusters are merged and $\mathbf{D}$ is updated

Lance-Williams formula [3]

Generic formula for updating the dissimilarity matrix $\mathbf{D}$.

while There are more than one cluster **do**

$(C_i, C_j) = \arg\min_{(C_l, C_m)} D(C_k, C_l)$

$C_{(ij)} = C_i \cup C_j$

for each Cluster C_k (where $k \neq i$, $k \neq j$) **do**

$D(C_{(ij)}, C_k) =$
$\alpha_i D(C_i, C_k) + \alpha_j D(C_j, C_k) + \beta D(C_i, C_j) + \gamma |D(C_i, C_k) - D(C_j, C_k)|.$

end for

remove clusters C_i, C_j and insert $C_{(ij)}$

end while

- Algorithms vary only in choice of $\alpha_i, \alpha_j, \beta, \gamma$

UPGMA [8], group average method

- unweighted pair group method using arithmetic averages
- Cluster distance is arithmetic average of all between-cluster values

$$D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(c_i, c_j)$$

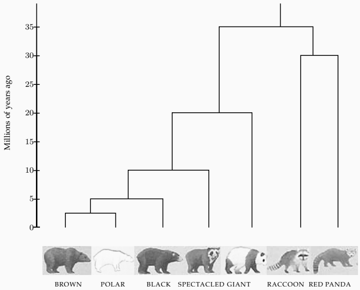
- $\alpha_i = \frac{|C_i|}{|C_i|+|C_j|}, \alpha_j = \frac{|C_j|}{|C_i|+|C_j|}, \beta = \gamma = 0$
- $D(C_{(ij)}, C_k) = \frac{|C_i|D(C_i, C_k) + |C_j|D(C_j, C_k)}{|C_i| + |C_j|}$

WPGMA [8], simple average method, McQuitty

- weighted pair group method using arithmetic averages
- smaller clusters receive larger weight, does not prefer same-size clusters
- $\alpha_i = \alpha_j = \frac{1}{2}, \beta = \gamma = 0$
- $D(C_{(ij)}, C_k) = \dfrac{1}{2}(D(C_i, C_k) + D(C_j, C_k))$

Molecular clock assumption [9]

- rate of evolutionary changes of DNA is approximately constant over time and branches of evolutionary tree
- evolutionary tree is ultrametric - distance from root to the leaves is constant
- let's measure edit distance between sequences
- for all triplets: pairwise distances are all same or two are same and one is less



Neighbor-joining [6]

- Reconstructs tree from additive matrix
- Matrix is additive if four point condition holds
- Does not make molecular clock assumption
- Merges clusters that are close to each other and far away from others
- Let $u(C) = \frac{1}{\text{num.ofclusters} - 1} \sum D(C, C')$
- Pick clusters minimizing $D(C_i, C_j) - u(C_1) - u(C_2)$
- New distance based on 3-leave formula ($\alpha_i = \alpha_j = \frac{1}{2}, \beta = -\frac{1}{2}, \gamma = 0$)

$$D(C_{(ij)}, C_k) = \frac{1}{2}\left(D(C_i, C_k) + D(C_j, C_k) - D(C_i, C_j)\right)$$

CHARACTER BASED TREE RECONSTRUCTION

Motivation

- alignment lost in distance matrix
- let's reconstruct tree directly from sequence alignment
- input: $n \times m$ matrix, n organisms m characters each
- parsimony approach : minimize number of mutations over evolutionary tree

Tree cost

- length of edge (u, v) is Hamming distance
- parsimony score for whole tree is sum of costs of all edges
- strings in internal vertices unknown
- find labeling of internal vertices that minimizes parsimony score

Small parsimony problem

- Find the most parsimonious labeling of the internal vertices in an evolutionary tree.

Fitch algorithm [1]

- dynamic programming algorithm
- assigns to each vertex a set of letters S_u so that
  - For any leaf u: S_u is label of the leaf.
  - for u with children v, w

$$S_u = \begin{cases} S_v \cap S_w, & \text{if } S_v \cap S_w \neq \emptyset, \\ S_v \cup S_w, & \text{otherwise.} \end{cases}$$

- in next pass label vertices
  - Assign root r any value from S_r.
  - for u with parent p

$$\mathrm{label}_u = \begin{cases} \mathrm{label}_p, & \mathrm{label}_p \in S_u, \\ \text{any element of } S_u, & \text{otherwise.} \end{cases}$$

Weighted small parsimony problem

- Find the minimal weighted parsimony score labeling of the internal vertices in an evolutionary tree.
- different character substitutions have different costs

Sankoff's algorithm [7]

- dynamic programming algorithm
- let $s_t(u)$ be parsimony score of tree with root u labeled by t
- for u with children v, w holds

$$s_t(u) = \min_i \{s_i(v) + \delta_{i,t}\} + \min_j \{s_j(w) + \delta_{j,t}\}.$$

- runs in $\mathcal{O}(|\Sigma|n)$

Large parsimony problem

- Find a tree with n leaves having the minimal parsimony score.
- NP-complete
- exhaustive search of tree topologies with heuristics and branch and bound

Thank you for your attention.
Time for questions!

Bibliography I

Walter M. Fitch.

Toward defining the course of evolution: Minimum change for a specific tree topology.

Systematic Zoology, 20(4):406–416, 1971.

Neil C Jones and Pavel Pevzner.

An introduction to bioinformatics algorithms.

MIT press, 2004.

G. N. Lance and W. T. Williams.

A general theory of classificatory sorting strategies: Ii. clustering systems.

The Computer Journal, 10(3):271–277, 1967.

Bibliography II

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman.
Mining of massive datasets.
Cambridge University Press, 2014.

Hannes Luz Martin Vingron, Jens Stoye.
Algorithms for phylogenetic reconstructions.
http://lectures.molgen.mpg.de/Algorithmische_Bioinformatik_WS0405/phylogeny_script.pdf.

Naruya Saitou and Masatoshi Nei.
The neighbor-joining method: a new method for reconstructing phylogenetic trees.
Molecular Biology and Evolution, 4(4):406–425, 1987.

Bibliography III

David Sankoff.

Minimal mutation trees of sequences.

SIAM Journal on Applied Mathematics, 28(1):35–42, 1975.

Robert Reuven Sokal and C. D. Michener.

A statistical method for evaluating systematic relationships.

University of Kansas Science Bulletin, 38:1409–1438, 1958.

Emile Zuckerkandl and Linus Pauling.

Molecular disease, evolution and genetic heterogeneity.

1962.

All images are taken from [2].