SDTM, a regular grammatical inference method for pattern discovery in sets of proteins

Aurélien Leroux

aurelien.leroux@irisa.fr

IRISA/INRIA, Campus de beaulieu,35042, Rennes cedex, France

Plan

- Problem
- Background
- Motivation for our Research
- Adding Biological Information in the Inference
- Transition Merging Operation
- SDTM Heuristic
- Conclusions and Directions

Problem

Problem:

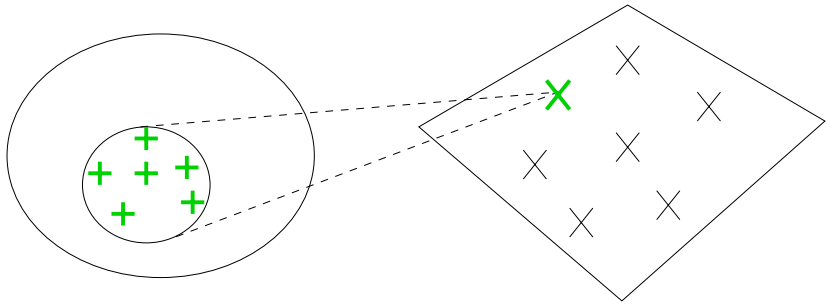
In regular grammatical inference, existing algorithms are not adapted for biological purposes.

We try to adapt these algorithms to the study of proteins.

Plan

- Problem
- Background
  - Grammatical Inference
  - Finite State Automaton (FSA)
  - State Merging Algorithm
  - Heuristic
- Motivation for our Research
- Adding Biological Information in the Inference
- Transition Merging and SDTM Heuristic
- Conclusions and Directions

Grammatical Inference

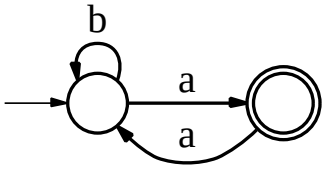


Instances — Concepts

Words — Automata

Regular Inference: *To learn a regular language from a sample of words belonging to the language*

$\{aaa, bba, baaa\}$ → Inference →



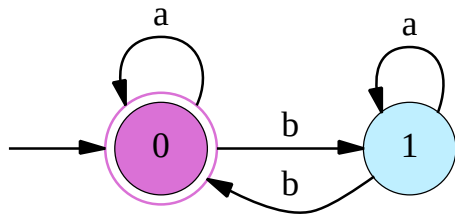
Finite State Automaton (FSA)

Definition: FSA is a tuple $\langle Q, I, F, \Sigma, \delta \rangle$, where

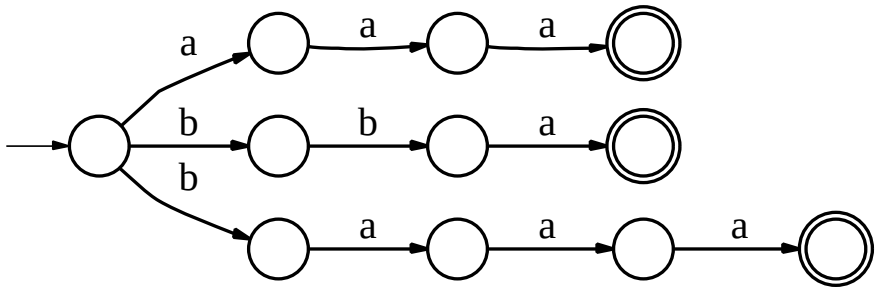
- Q – the finite set of states;
- $I \subseteq Q$ – the set of initial states;
- $F \subseteq Q$ – the set of final states;
- Σ – the alphabet;
- $\delta : Q \times \Sigma \mapsto 2^Q$ – the transition function.

Example:

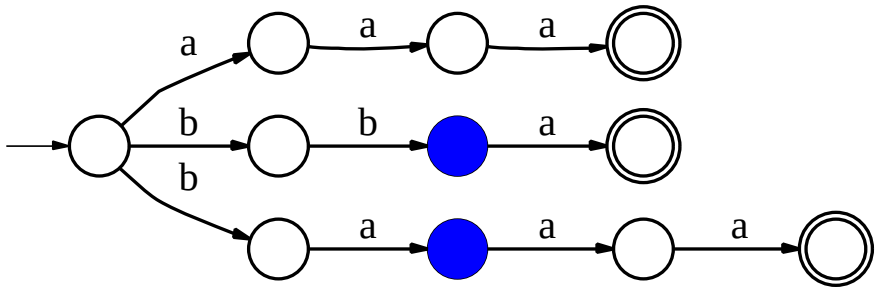
minimal deterministic FSA representing the language on the alphabet $\Sigma = \{a, b\}$, which consists of the words containing an even number of b.



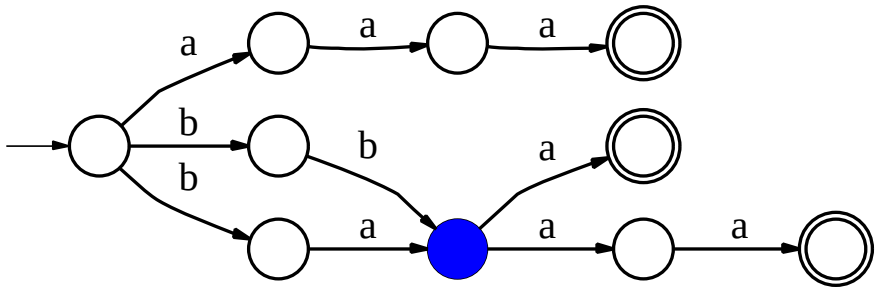
State Merging algorithm [TB73]



State Merging algorithm [TB73]



State Merging algorithm [TB73]



Heuristic

- RPNI [OG92]: breadth first order
- EDSM [LPP98]: maximizes the number of merging of states belonging to the same class

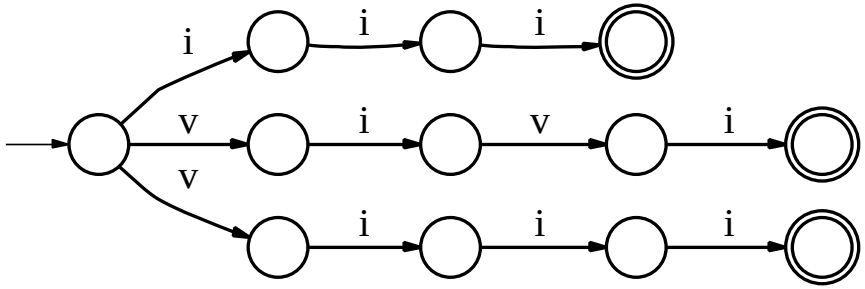
Plan

- Problem
- Background
- Motivation for our Research
- Adding Biological Information in the Inference
- Transition Merging Operation
- SDTM Heuristic
- Conclusions and Directions

Motivation for our Research

We try to adapt the algorithm based on state merging to the study of proteins by replacing state merging by transition merging.

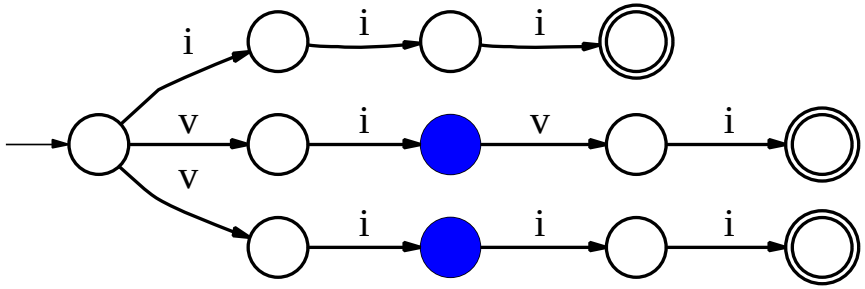
Reason:



Motivation for our Research

We try to adapt the algorithm based on state merging to the study of proteins by replacing state merging by transition merging.

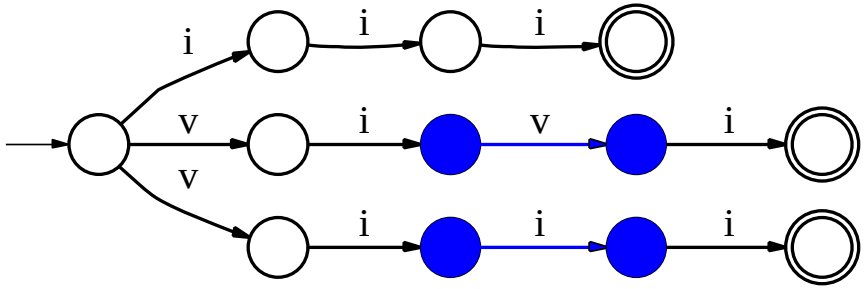
Reason:



Motivation for our Research

We try to adapt the algorithm based on state merging to the study of proteins by replacing state merging by transition merging.

Reason:



Plan

- Problem, Background, and Motivation
- Adding Biological Information in the Inference
  - Motivation
  - Taylor Diagram
  - From Taylor Diagram to Lattice
  - Lattice Based on the Taylor Diagram
  - Where do we use Biological Information?
- Transition Merging Operation
- SDTM Heuristic
- Conclusions and Directions

Motivation

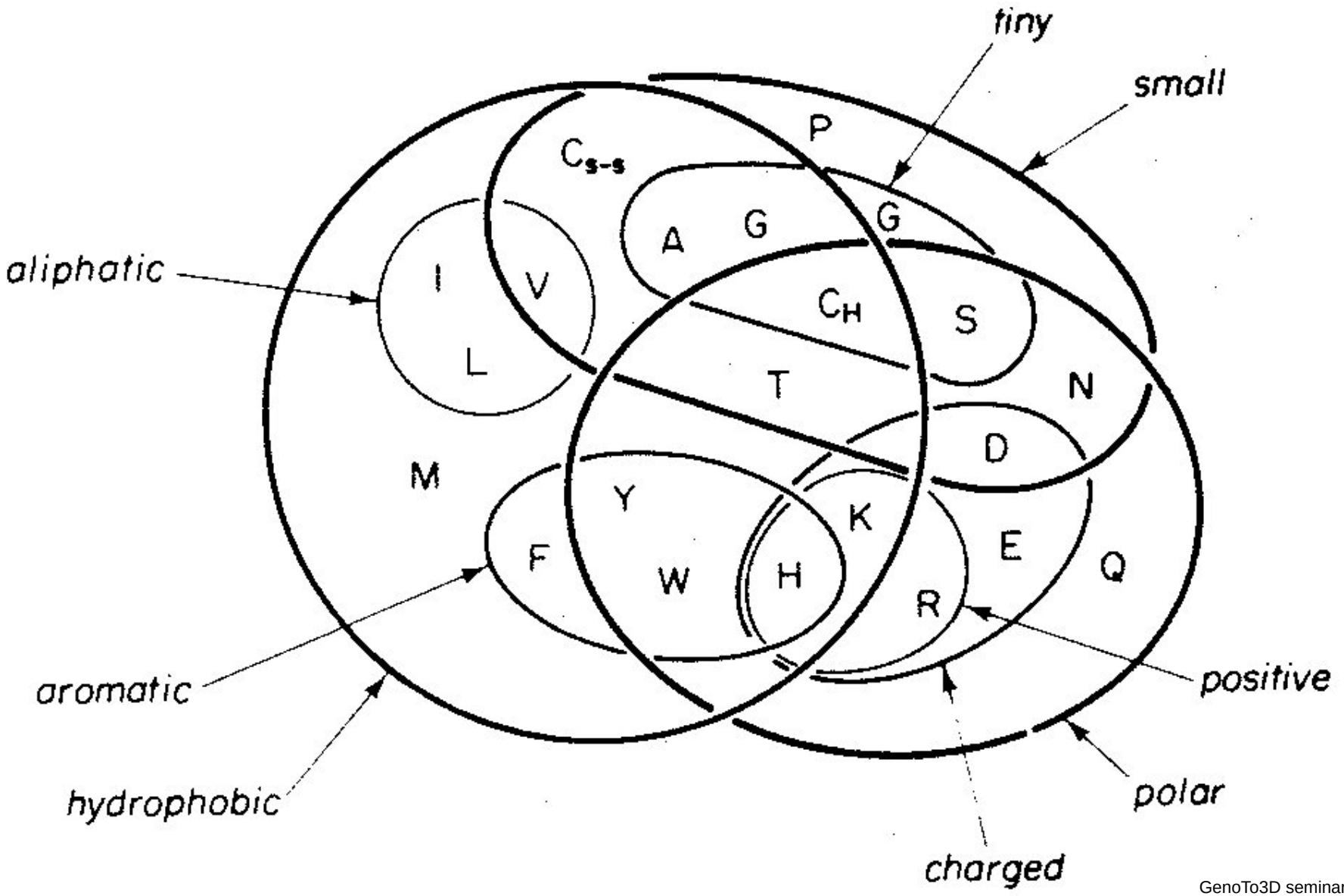
Problem:

- amino acids are not only letters (physico-chemical properties)
- existing algorithms compare only the letters

Possible Existing Solution:

- alignment algorithms use substitution matrices

Taylor Diagram [Tay86]



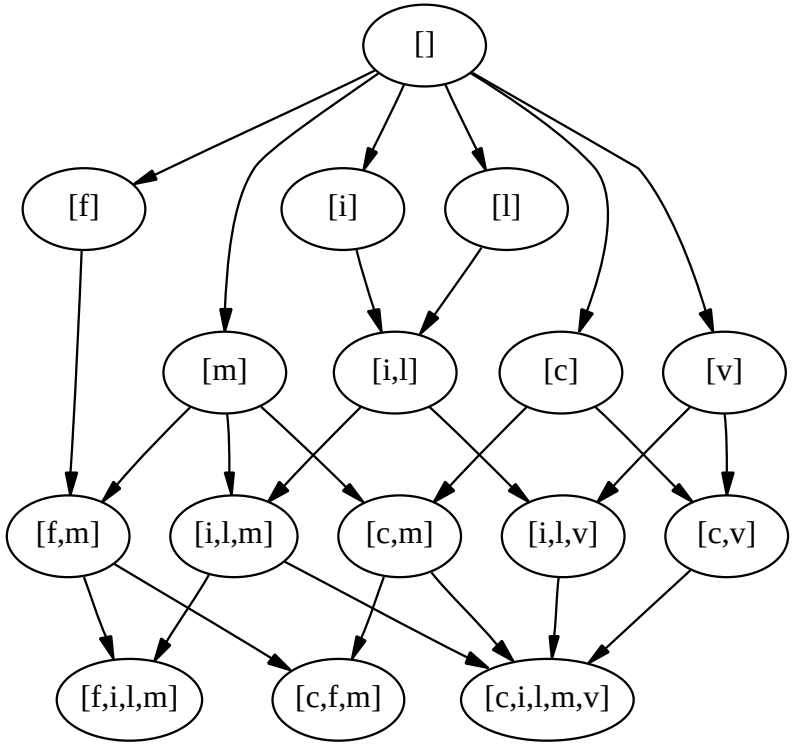
From Taylor Diagram to Lattice

- we start from the set $TD = \{A, B_1, \ldots, B_{70}, C_1, \ldots, C_n, \overline{C_1}, \ldots, \overline{C_n}\}$ containing:
  - the set of amino acids $A = \{a_1, \ldots, a_{|A|}\}$,
  - the 70 biologically significant classes $B_1, \ldots, B_{70}$,
  - all the classes of the diagram $C_1, \ldots, C_n$, and
  - all the complements of these classes $\overline{C_1}, \ldots, \overline{C_n}$

From Taylor Diagram to Lattice

- we start from the set $TD = \{A, B_1, \ldots, B_{70}, C_1, \ldots, C_n, \overline{C_1}, \ldots, \overline{C_n}\}$
- we recursively calculate the intersections between all the sets of TD, updating TD at each stage by adding the new intersections.
- we connect one set $L_1 \in TD$ to the other $L_2 \in TD$ with the edge $L_1 \longrightarrow L_2$ if $L_2 \subset L_1$, and we remove transitivity from the resulting graph

Lattice Based on the Taylor Diagram



On this lattice we define the least common ancestor relation.

Where do we use Biological Information?

- in the definition of transition merging and
- in the heuristic to choose transitions to merge

Plan

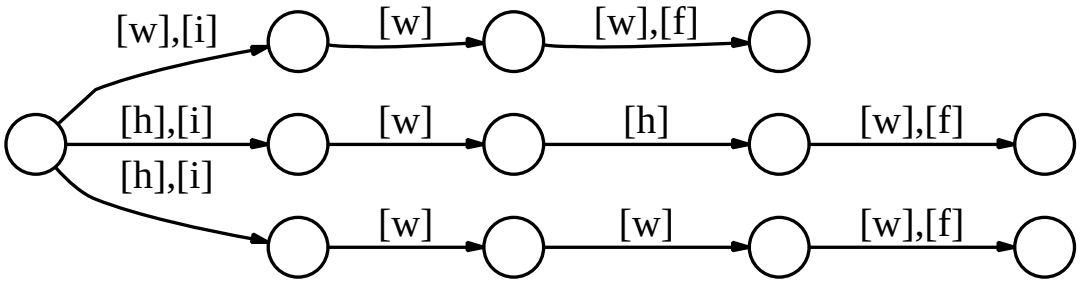
- Problem, Background, and Motivation
- Adding Biological Information in the Inference
- Transition Merging Operation
  - Transition Merging Operation
  - Example : Transition Merging
  - State Merging vs Transition Merging
- SDTM Heuristic
- Conclusions and Directions

Transition Merging Operation

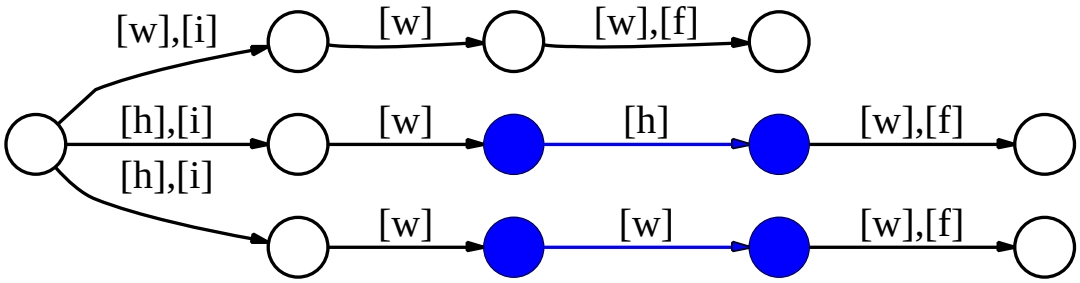
$\mathsf{merge\_trans}\left(\left(q_1, l_1, q_1'\right), \left(q_2, l_2, q_2'\right)\right) \longmapsto (q, l, q')$, where

- $q = \mathsf{merge\_state}\left(q_1, q_2\right)$,
- $q' = \mathsf{merge\_state}\left(q_1', q_2'\right)$,
- $l = \mathsf{lca}\left(l_1, l_2\right)$ in the lattice based on the Taylor Diagram.

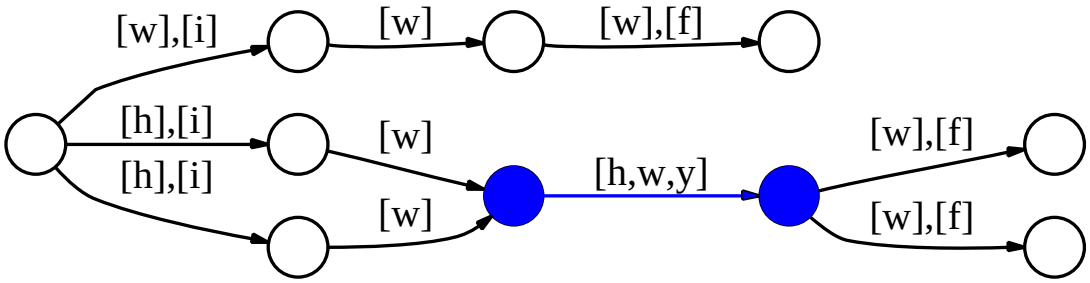
Example : Transition Merging



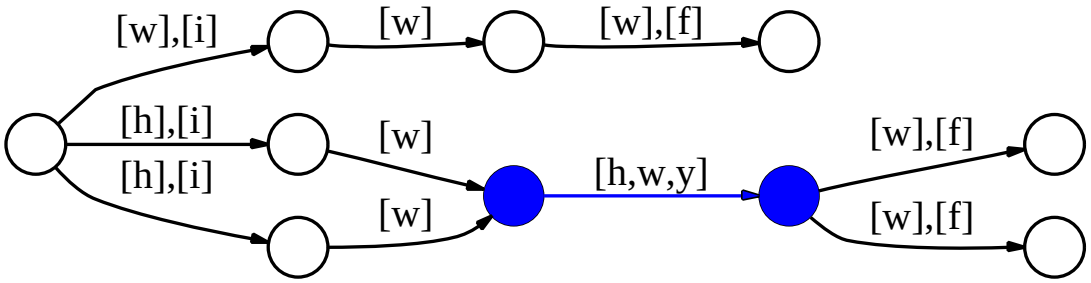
Example : Transition Merging

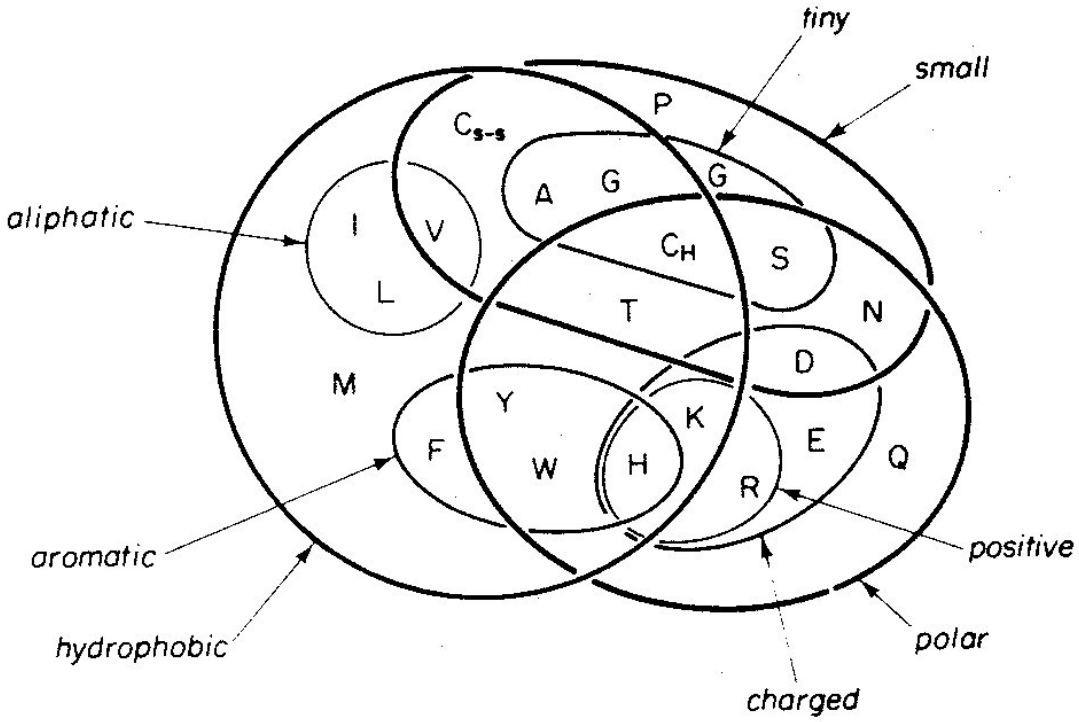


Example : Transition Merging



Example : Transition Merging





State Merging vs Transition Merging

Transition merging generalizes the automaton faster than state merging. Two reasons:

(1) 2 pairs of states are merged at a time during transition merging, and

(2) merging two transitions t_1 and t_2 with labels l_1 and l_2 produces a transition whose label is the least common ancestor of l_1 and l_2 and not only the union of l_1 and l_2.

This speedup preserves the validity of the merging choices due to the introduction of biological information.

Plan

- Problem, Background, and Motivation
- Adding Biological Information in the Inference
- Transition Merging Operation
- SDTM (Similarity Driven Transition Merging) Heuristic
  - Motivation
  - SDTM Heuristic
- Conclusions and Directions

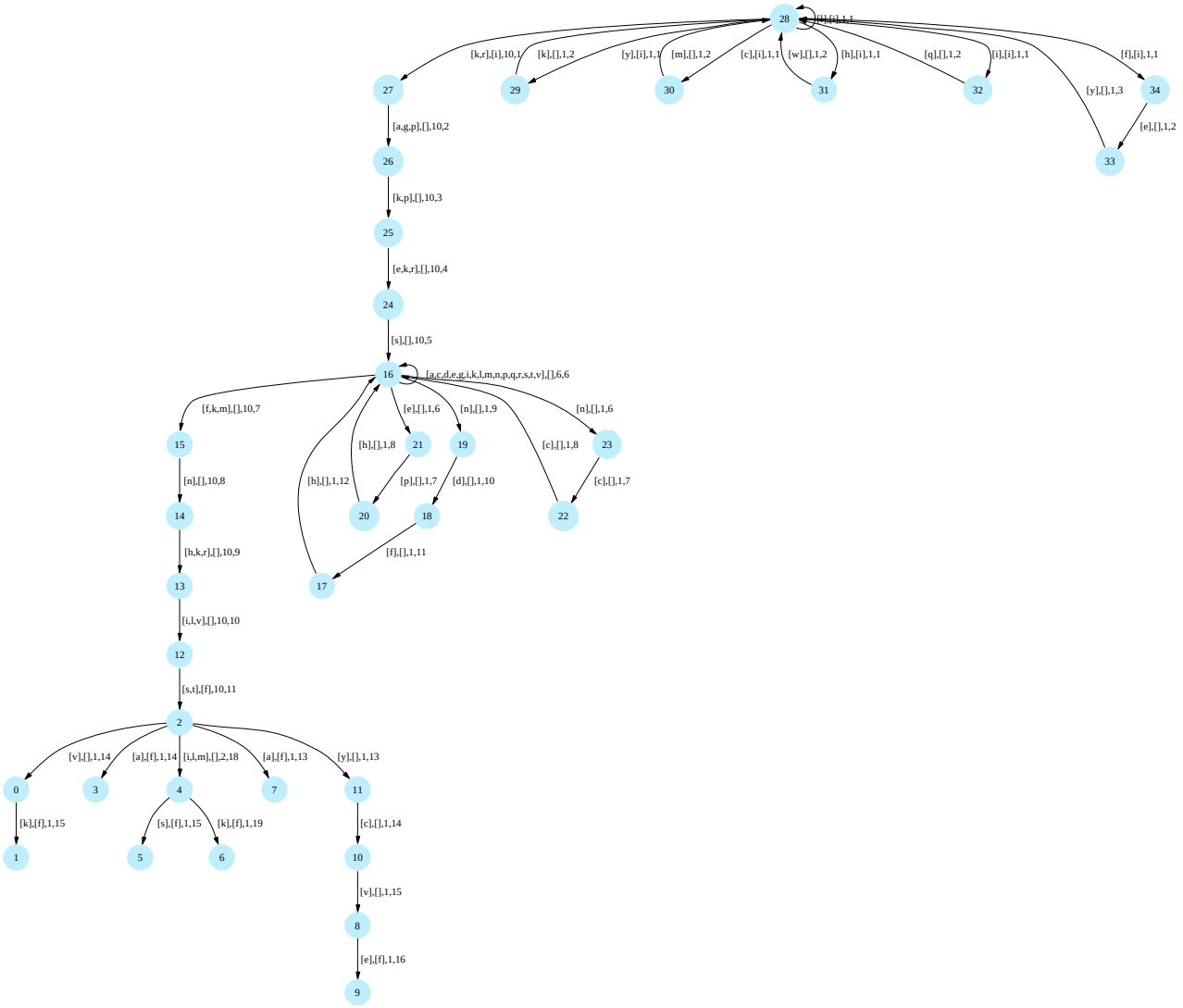
Motivation

- the Branch and Bound approach to merge transitions is NP-complete,
- thus, we must employ a greedy method, which means using a heuristic,
- we adapt the EDSM heuristic to the transition merging problem and to the study of proteins.

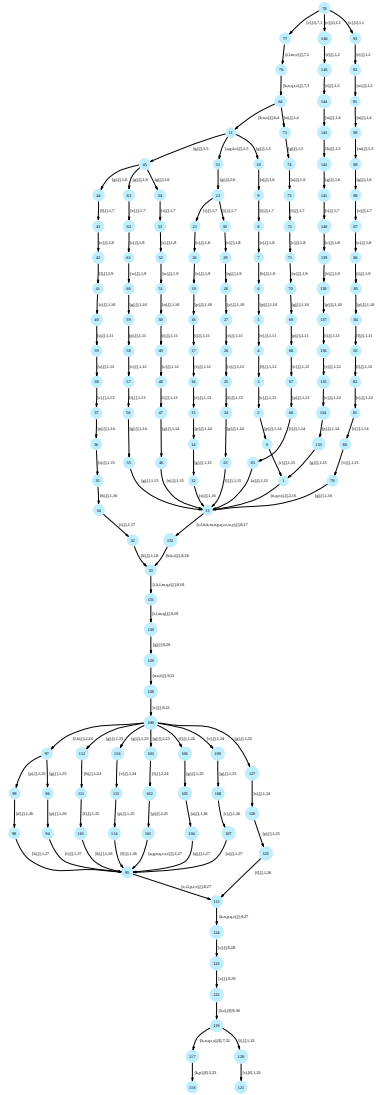
SDTM Heuristic

- the heuristic consists in choosing the best compatible set of local alignments over the sequences from the given sample set.
- on the automaton recognizing exactly the sequences from the given sample we merge the transitions corresponding to the similar amino acids of the alignments.

Resulting Examples



Resulting Examples



Conclusions and Directions

- the results are too specific. One way to change this is to execute recursively the algorithm changing at each step the frequencies according to the merging already processed,
- to improve the calculation of the score of merging,
- to test the algorithm and to compare it with the existing ones.

Bibliographie

References

[LPP98] Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. *Lecture Notes in Computer Science*, 1433:1–12, 1998.

[OG92] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. *Pattern Recognition and Image Analysis*, pages 49 – 61, 1992.

[Tay86] William Ramsy Taylor. The classification of amino acid conservation. *Journal of*