A Probabilistic Model for Mining Labeled Ordered Trees: Capturing Patterns in Carbohydrate Sugar Chains

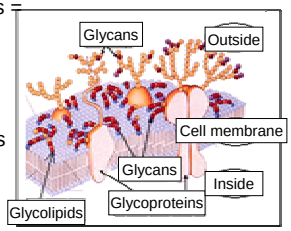
Hiroshi Mamitsuka
Bioinformatics Center
Kyoto University

Outline

- What is a Carbohydrate Sugar Chain (Glycan)?
- Glycan = Labeled Ordered Tree
- Databases on Glycans
- Probabilistic Models for Labeled Ordered Trees and their empirical experimental results
  - Probabilistic Sibling Dependent Tree Markov Model (PSTMM)
  - Profile PSTMM
  - Ordered Tree Markov Model
- Concluding Remarks

Glycans

- Carbohydrate Sugar Chains = ***Glycans***
- Third major class of biomolecules next to DNA and proteins
- Often found on cell surfaces
- Crucial to the development and function of multicellular organisms
- However, many still unknowns in glycobiology



Glycans: Third Major Class

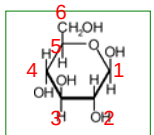
- DNA: Genome - Genomics
- Protein: Proteome - Proteomics
- Glycan: Glycome - Glycomics
  - The collective identity of the entirety of carbohydrates in an organism
  - The collective identity of the entirety of carbohydrates in a cell

Glycan Structure

- DNA: String of four letters
  - four kinds of nucleotides (A,G,C,T)
- Protein: String of twenty letters
  - twenty types of amino acids
- Glycan: Tree structure of *monosaccharides* (sugars) and *linkages*
  - More than 10,000 structures known
  - ~twelve main types of monosaccharides (i.e., Glucose [Glc], N-acetylglucosamine [GlcNAc], Mannose [Man])
  - 10-15 classes (i.e., N-Glycans, O-Glycans, GPI anchors, etc.)

Building Blocks of Glycans

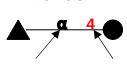
○	Galp	Galactose
□	GalpNAc	N-acetylgalactosamine
●	Glcp	Glucose
■	GlcpNAc	N-acetylglucosamine
●	Manp	Mannose
▲	Fucp	Fucose
☆	Xylp	Xylose
◆	NeupAc	N-acetylneuraminic acid
◇	NeupGc	N-glycolylneuraminic acid
◆	KDN	Ketodeoxynonulosonic acid



Glucose (Glc)

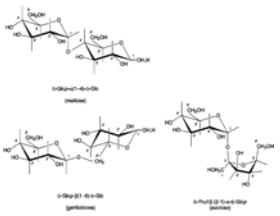
Sugars

Bonds



anomer Carbon #

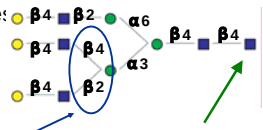
What Do Glycans Look Like?



- Variations in conformations between just two sugars (monosaccharides)
- Typical glycan structures contain 10-15 sugars!

What Do Glycans Look Like?

- IUPAC 2D Representation
- Tree structures of monosaccharides and linkages
- Nodes = sugars/monosaccharides
- Edges = bonds/linkages
- Features:
  1. Rooted tree
  2. Monosaccharides = Labels
  3. Ordered children



Glycan Structure

- Glycan is
  - Rooted Tree:
    - Tree with root
  - (Rooted) Labeled Tree:
    - Tree whose nodes have labels attached
      - Monosaccharide names
  - (Rooted) Labeled Ordered Tree
    - Labeled tree whose children are ordered

Outline

- What is a Glycan?
- Glycan = Labeled Ordered Tree
- Databases on Glycans
- Probabilistic Models for Labeled Ordered Trees and their empirical experimental results
  - Probabilistic Sibling Dependent Markov Model (PSTMM)
  - Profile PSTMM
  - Ordered Tree Markov Model
- Concluding Remarks

General Database Systems for Glycans

- CarbBank
- SWEET-DB / glycosciences.de
- KEGG GLYCAN
- Consortium for Functional Glycomics
- EuroCarbDB

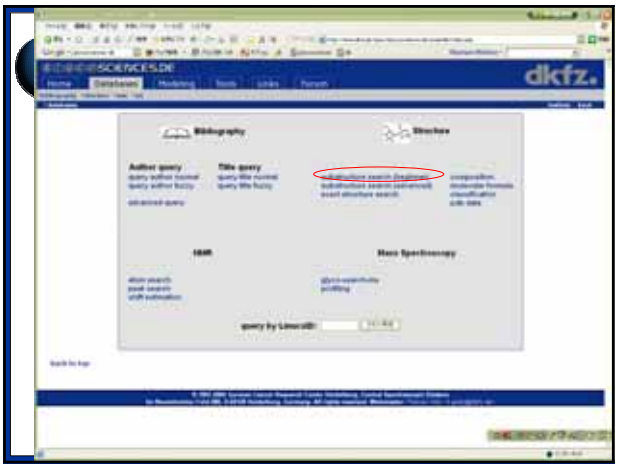
- Commercial databases:
  - GlycoSuite (Proteome Systems, Ltd.)
  - Glycomics DB (Glycominds, Ltd.)

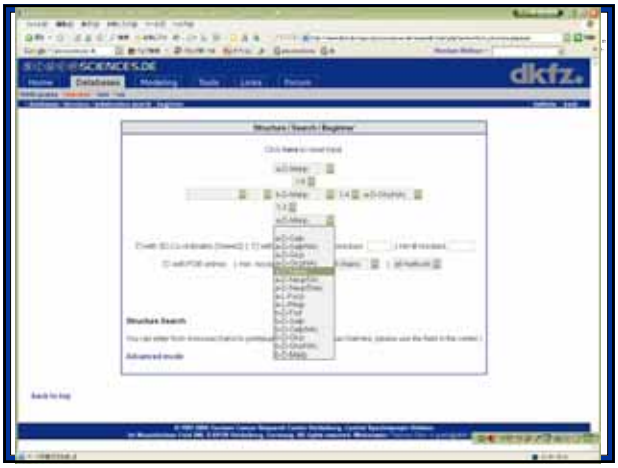
CarbBank

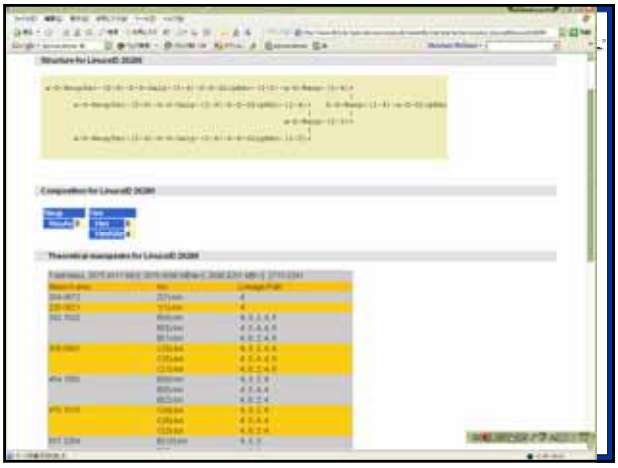
- Developed by Complex Carbohydrate Research Center, University of Georgia
- Community database of carbohydrates
- Project ended due to lack of funding in 1996
- Continued in Japan until around 2000

SWEET-DB

- A part of Glycoscience.de
- http://www.dkfz-heidelberg.de/spec/sweetdb/
- Combines CarbBank and Sugabase using a common web-based interface
- Provides searching by bibliography, structure, NMR and MS, as well as by LINUCS ID

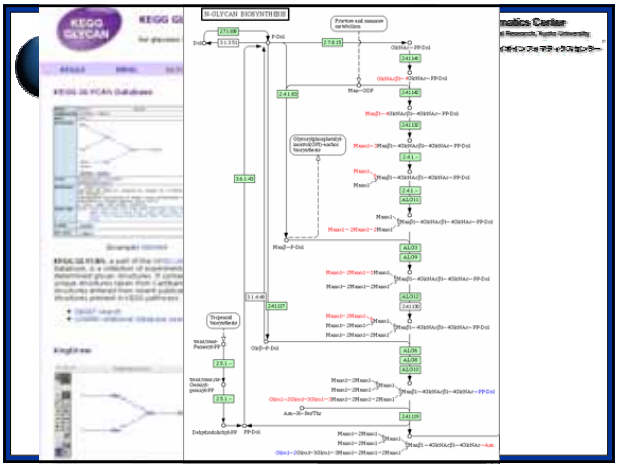






KEGG GLYCAN

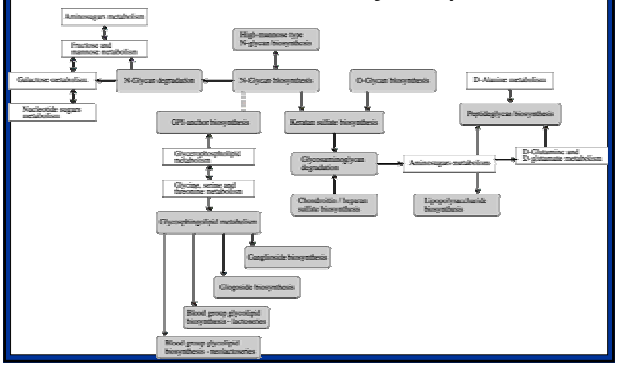
- http://www.genome.jp/kegg/glycan/
- Based on CarbBank as well as input from scientists
- All data is linked with KEGG's other resources: GENES, PATHWAY, KO (KEGG Ontology) and literary databases

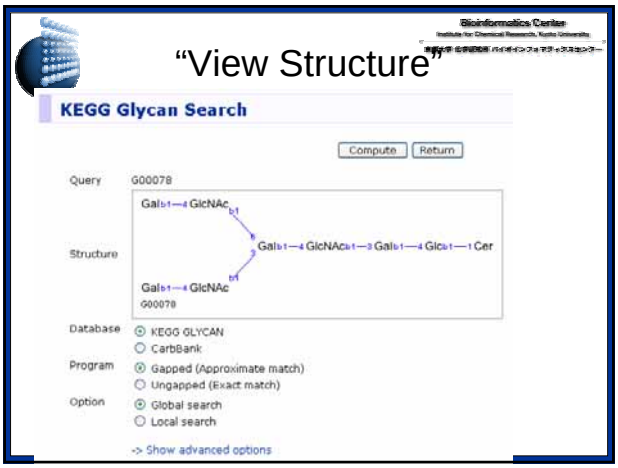


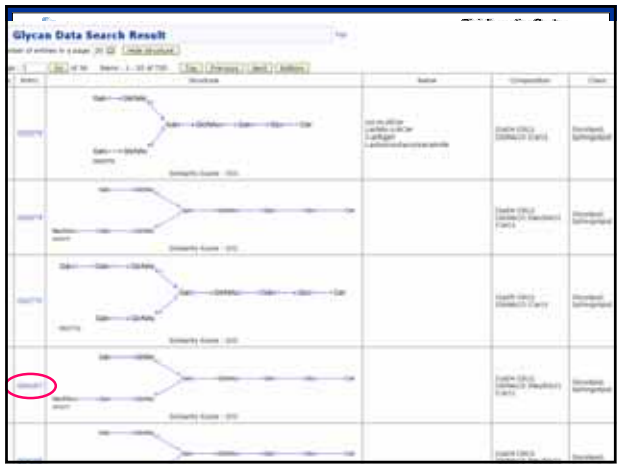
Glycan Biosynthesis and Metabolism Pathways

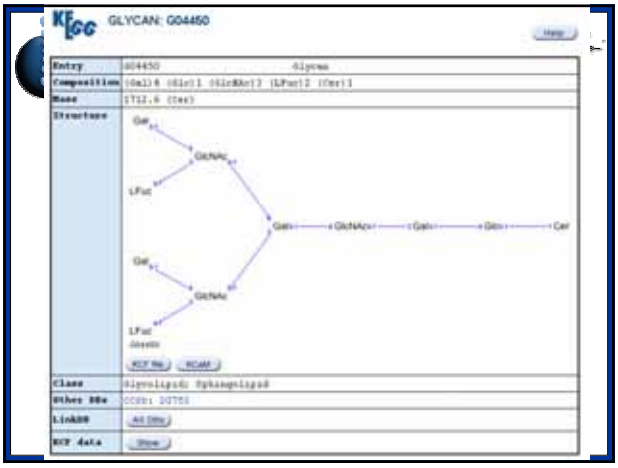
- N-Glycan biosynthesis
- High-mannose type N-glycan biosynthesis
- N-Glycan degradation
- O-Glycan biosynthesis
- Chondroitin / heparan sulfate biosynthesis
- Keratan sulfate biosynthesis
- Glycosaminoglycan degradation
- Lipopolysaccharide biosynthesis
- Peptidoglycan biosynthesis
- Glycosylphosphatidylinositol(GPI) -anchor biosynthesis
- Glycosphingolipid metabolism
- Blood group glycolipid biosynthesis - lactoseries
- Blood group glycolipid biosynthesis - neo-lactoseries
- Globoside metabolism
- Ganglioside biosynthesis
- Glycan structures - biosynthesis 1
- Glycan structures - biosynthesis 2
- Glycan structures - degradation

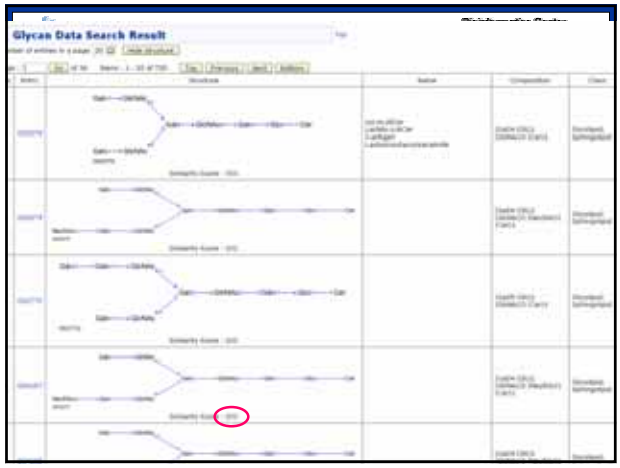
Overall Relationship of Glycan Related Pathway Maps

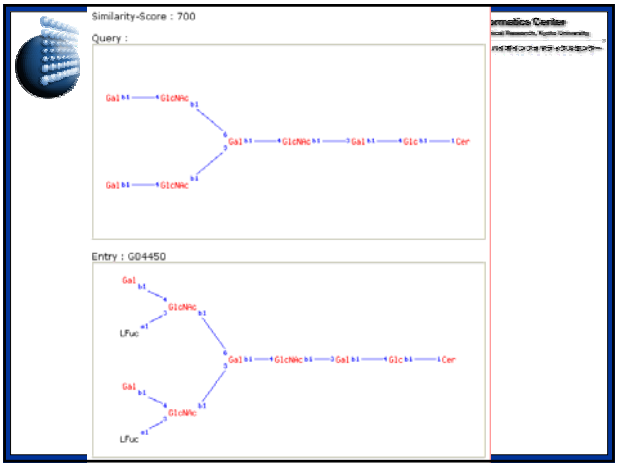


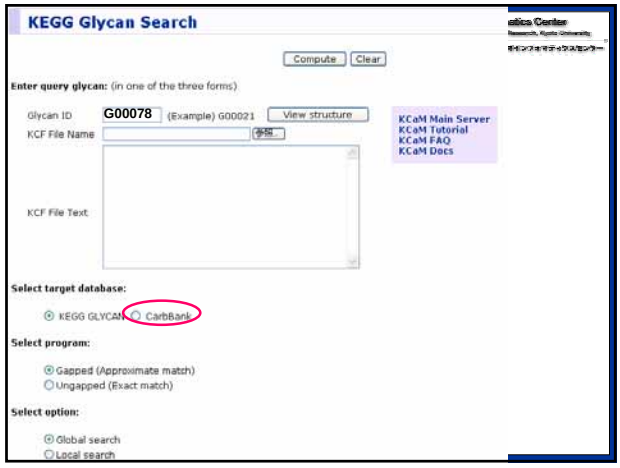

“View Structure”
KEGG Glycan Search
Compute
Return
Query
G00078
Structure
Galβ1—4 GlcNAc
Galβ1—4 GlcNAcβ1—3 Galβ1—4 Glcβ1—1 Cer
Galβ1—4 GlcNAc
G00078
Database
KEGG GLYCAN
CarbBank
Program
Gapped (Approximate match)
Ungapped (Exact match)
Option
Global search
Local search
-> Show advanced options

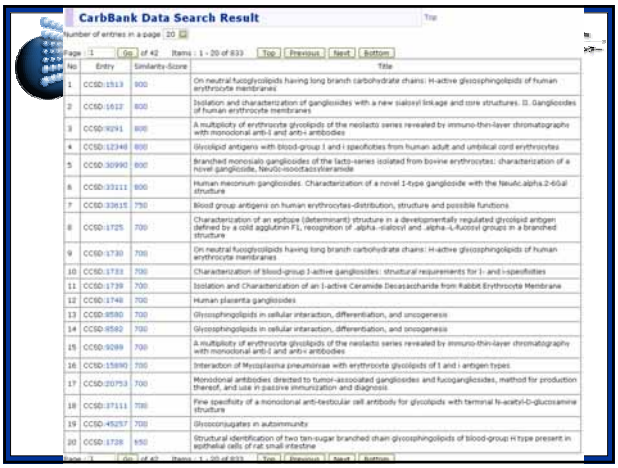

Glycan Data Search Result


GLYCAN: G04450


Glycan Data Search Result

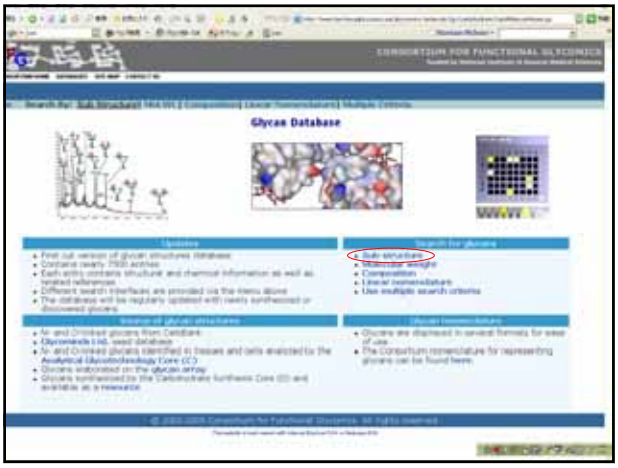

Similarity-Score : 700
Query :
Entry : G04450

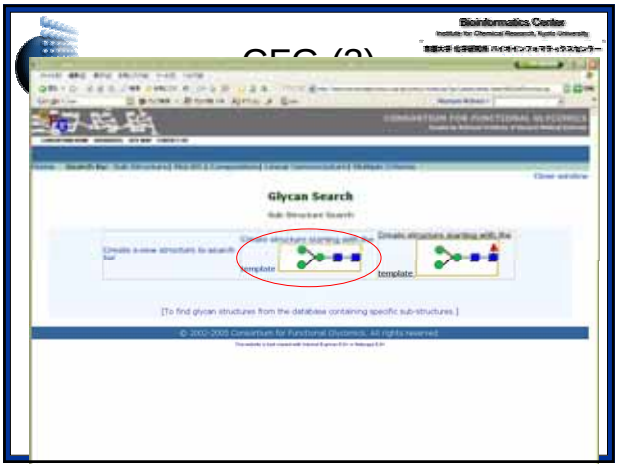

KEGG Glycan Search
Compute
Clear
Enter query glycan: (in one of the three forms)
Glycan ID
G00078
(Example) G00021
View structure
KCF File Name
KCF File Text
KCaM Main Server
KCaM Tutorial
KCaM FAQ
KCaM Docs
Select target database:
KEGG GLYCAN
CarbBank
Select program:
Gapped (Approximate match)
Ungapped (Exact match)
Select option:
Global search
Local search

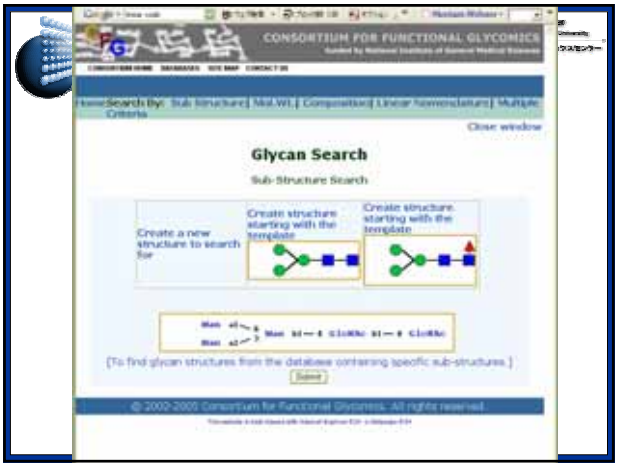


Consortium for Functional Glycomics

- Consortium home page: http://www.functionalglycomics.org/
- Consortium of major universities and research institutes worldwide
- Aim: to provide a central resource for glycomics research
- Also provides requested resources to promote participating investigators' research
  - Glycan arrays and data
  - Mass spectra analysis…
- CFG glycan database web page: http://www.functionalglycomics.org/glycomics/molecule/jsp/carbohydrate/carbMoleculeHome.jsp

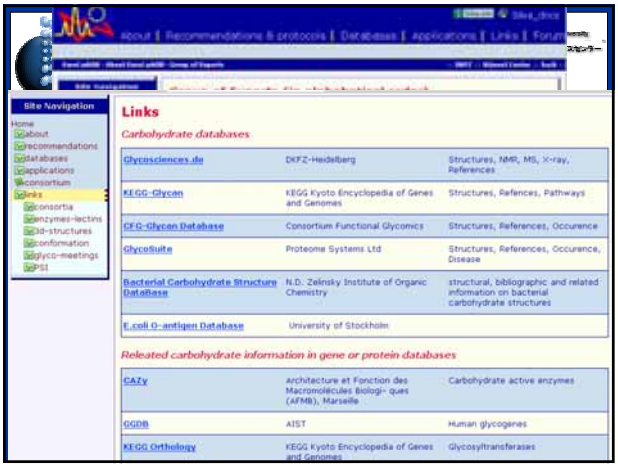






EuroCarbDB

- http://www.eurocarbdb.org/
- Based in Europe, but participants from universities and research groups worldwide
- Distributed infrastructure to integrate multiple resources with a single interface



Outline

- What is a Glycan?
- Glycan = Labeled Ordered Tree
- Databases on Glycans
- Probabilistic Models for Labeled Ordered Trees and their empirical experimental results
  - Probabilistic Sibling Dependent Markov Model (PSTMM)
  - Profile PSTMM
  - Ordered Tree Markov Model
- Concluding Remarks

Where to Begin?

- Glycan Structure:
  - Rooted Tree:
    - Tree with root
  - (Rooted) Labeled Tree:
    - Tree whose nodes have labels attached
      - Monosaccharide names
  - (Rooted) Labeled Ordered Tree
    - Labeled tree whose children are ordered

Motivation (from Biological Side)

- Glycans = Labeled Ordered Trees
- Many unknowns in glycobiology
  - High uncertainty and noisy
- The leaves of glycans are important in recognition by various pathogens
- Differences in these patterns affect biological functions
- Pattern mining method robust against noise required

Motivation (from Informatics Side)

- Labled Ordered Trees: Semi-structured (or Unstructured) data
  - Other examples found in web and text mining, e.g. XML
- Mining semi-structured data, like graphs and/or trees, becoming important in machine learning and data mining
- Frequent pattern mining and kernels already developed recently
- New mining approaches robust against noise required

Probabilistic Modeling

- Statistical machine learning
- Represent uncertainty
- Robust against noise in data
- Efficient learning schemes already known
- Modeling labeled ordered trees not developed yet

Mining Labeled Ordered Trees Based-on Probabilistic Modeling

- To "learn" patterns from the tree structures of glycans by estimating the probability parameters of our model

Three Problems

- Must be solved to be used in real-world applications
  1. Computing likelihood: computing how likely a given example can be generated from a model
  2. Learning: estimating probability parameters of a model from given data
  3. Parsing: finding the most likely state transition on an example given a model

Three Problems

1. Computing the likelihood of a set of trees:
   - To determine which data are considered to belong to same class as training (learned) data
2. Estimating the parameters:
   - To "learn" patterns found in given data
3. Finding the most likely state transition:
   - To retrieve the learned patterns
   - To apply to multiple alignments

Three Problems

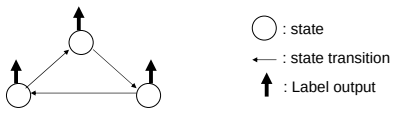
- Learning (estimating probability parameters) is the most important, since …
- Computing the likelihood is a part of learning
- Parsing can be done by modifying the likelihood computation

Three Problems

- Must be solved to be used in real-world applications
  1. Computing likelihood: computing how likely a given example can be generated from a model
  2. Learning: estimating probability parameters of a model from given data
  3. Parsing: finding the most likely state transition of a tree given a model

Hidden Markov Model (HMM)

- Model: set of states connected by directed edges
- Parameters:
  - State transition probability: a_{ij}
  - Label output probability (at state): $b_j(k)$
- Operation: series of Markov transition on states, generating sequence of labels (string) with likelihood

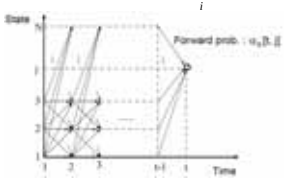


Three Problems

- Hidden Markov model case:
  1. Computing likelihood: computing auxiliary probabilities: Forward or Backward probabilities
  2. Learning: Maximizing the likelihood by Baum-Welch (Forward-Backward) algorithm, an EM (Expectation-Maximization) algorithm
  3. Parsing: Viterbi algorithm

Forward Probability, HMM Case

- The probability that the current state is *j* and string [*1..t*] is already generated: $\alpha_\sigma[t, j]$
- Dynamic programming over *t* available
- Updating formula: $\alpha_\sigma[t,i] = \sum_i a_{ij} b_j(\sigma_t) \alpha_\sigma[t-1,i]$



Computing Likelihood, HMM Case

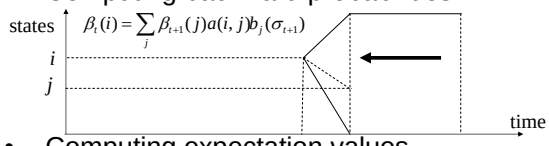
- Compute forward probabilities over whole a given string
- Use final forward probabilities
- Likelihood: $\sum_i \alpha_\sigma[T,i]$

Learning, HMM Case

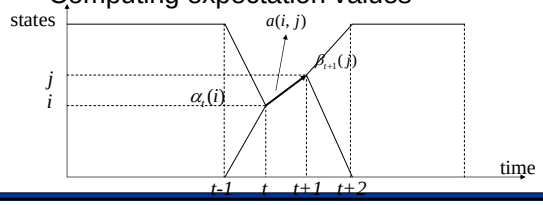
- Baum-Welch (Expectation-Maximization) algorithm
  1. Compute forward probabilities
  2. Compute backward probabilities: $\beta_\sigma[t, j]$
     - Probability that the current state is j and string [t..T] is already generated
  3. Compute expectation value of transition *ij*:
     $$E_{P_\sigma}[\#((i,j),\sigma] \propto \sum_t \alpha_\sigma[t,i] a_{ij} b_j(\sigma_{t+1}) \beta_\sigma[t+1, j]$$
  4. Update transition probability a_{ij} using expectation values:
     $$\hat{a}_{ij} = \frac{E_{P_\sigma}[\#((i,j),\sigma]}{\sum_j E_{P_\sigma}[\#((i,j),\sigma]}$$
- The above steps iterated until convergence

Supplement for HMM Learning

- Computing backward probabilities



- Computing expectation values



Learning, HMM Case Summary

- Maximizing the Likelihood
- Baum-Welch: EM algorithm
- Repeat the following two steps alternately until some stopping condition satisfied
  - E-step:
    1. Compute forward and backward
    2. Compute expectation values
  - M-step:
    1. Update transition probabilities

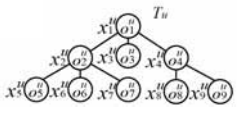
Back to Probabilistic Models for Trees

- Notations
  - Tree: *T*
  - Node: *x*
  - State type: *s*
  - State of node: *z*
  - Label: *o*

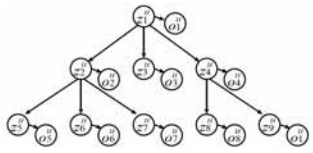
Related work: Hidden Tree Markov Model (HTMM)

[Deligenti et al., 2003]

- Probabilistic model for labeled trees, not for labeled ordered trees
- State depends on that of the parent only

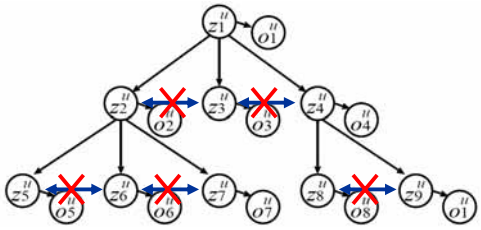


Tree



Dependencies

HTMM Cannot Capture Sibling Dependencies!



Outline

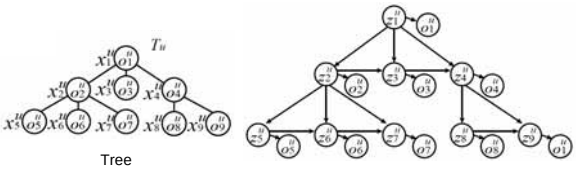
- What is a Glycan?
- Glycan = Labeled Ordered Tree
- Databases on Glycans
- Probabilistic models for Labeled Ordered Trees and their empirical experimental results
  - Probabilistic Sibling Dependent Markov Model (PSTMM)
  - Profile PSTMM
  - Ordered Tree Markov Model
- Concluding Remarks

Proposed model

- Probabilistic Sibling dependent Tree Markov Model (PSTMM)
  - Modeling sibling dependency as well as parent-child dependency
  - Extension of hidden Markov model (HMM) and hidden tree Markov model (HTMM)

PSTMM

- State depends on those of both the parent and the immediately elder sibling



Tree

Dependencies

Define PSTMM Parameters

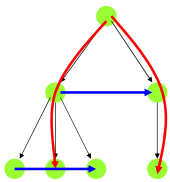
- Three probability parameters
  - Initial state probability: $\pi[s_l](=P(z_1^u = s_l))$
    - Probability that the state of the root is S_l
  - State transition probability:

$$a[\{s_q, s_l\}, s_m](= P(z_j^u = s_m \mid z_p^u = s_q, z_i^u = s_l)$$

    - Probability that the state of j is S_m given that the state of the immediately elder sibling is S_l and the sate of the parent is S_q
  - Label output probability: $b[s_l, \sigma_h]$
    - Probability that the state S_l outputs σ_h

Auxiliary Probabilities for PSTMM

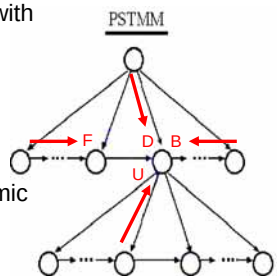
- Extension of HMM for Labeled Ordered Trees
- Labeled Ordered Tree
  - Two directions:
    - Parent-Child directions
    - Sibling directions
  - Sibling: Forward and Backward
  - Parent-Child: ?



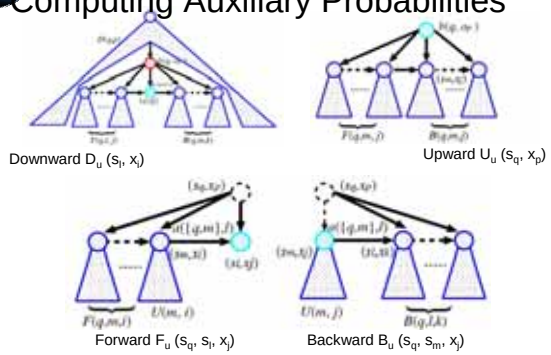
Downward and Upward!

Four Auxiliary Parameters

- Define the following four probabilities for a tree *u* with nodes $x_1, \ldots, x_n$
  - Forward $F_u(s_q, s_l, x_i)$
  - Backward $B_u(s_q, s_m, x_i)$
  - Upward $U_u(s_q, x_p)$
  - Downward $D_u(s_l, x_i)$
- Each computed by dynamic programming (DP)



Dynamic Programming for Computing Auxiliary Probabilities

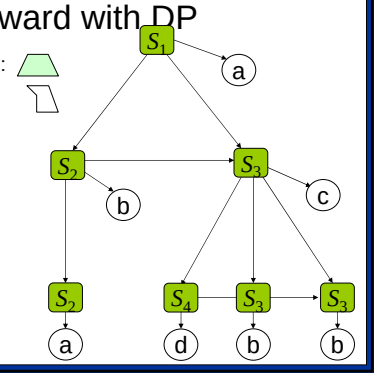


Computing Backward and Upward with DP

- Example

Backward:

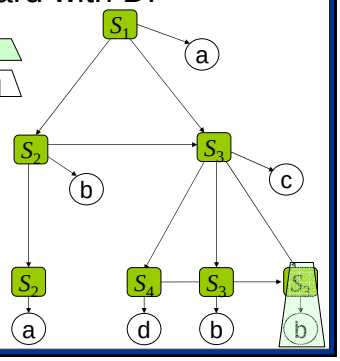
Upward:



Computing Backward and Upward with DP

Backward:

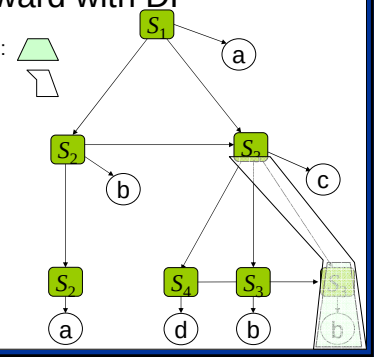
Upward:



Computing Backward and Upward with DP

Backward:

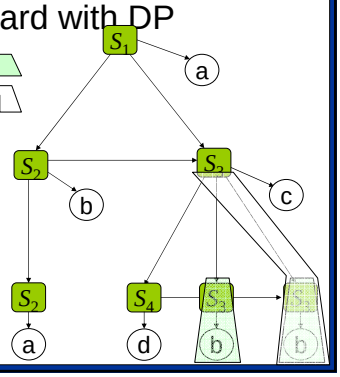
Upward:



Computing Backward and Upward with DP

Backward:

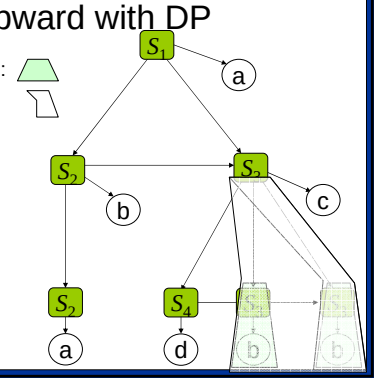
Upward:

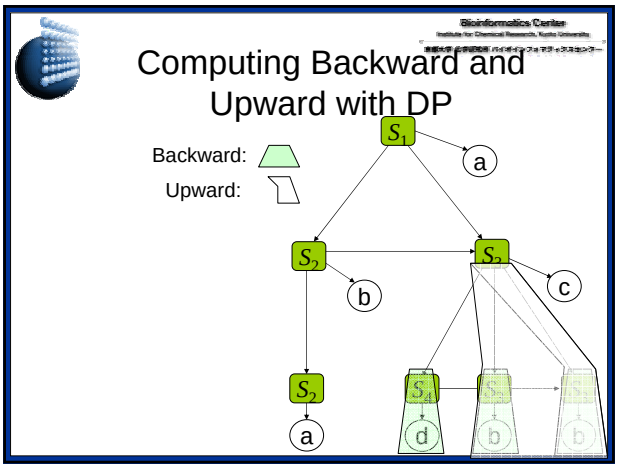


Computing Backward and Upward with DP

Backward:

Upward:

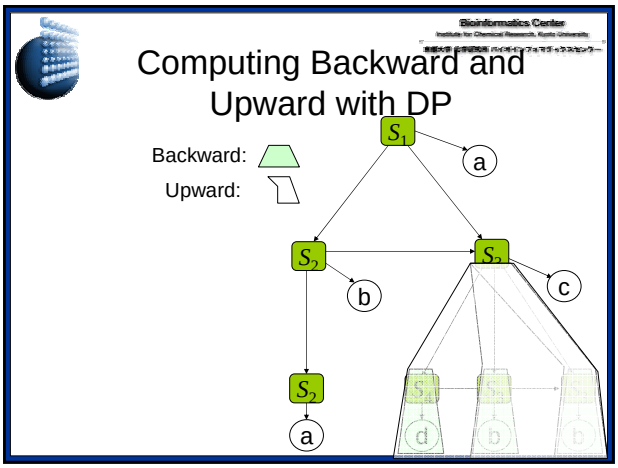




Computing Backward and Upward with DP

Backward: ⏢

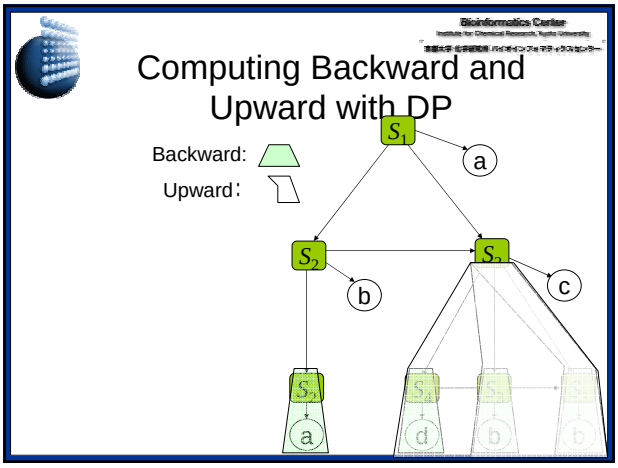
Upward: ▱



Computing Backward and Upward with DP

Backward: ⏢

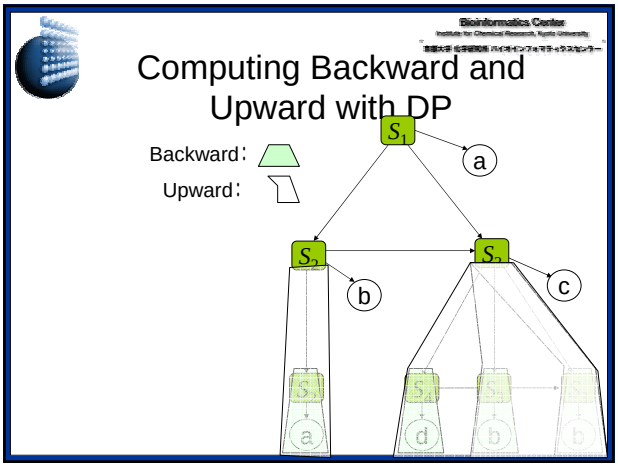
Upward: ▱



Computing Backward and Upward with DP

Backward: ⏢

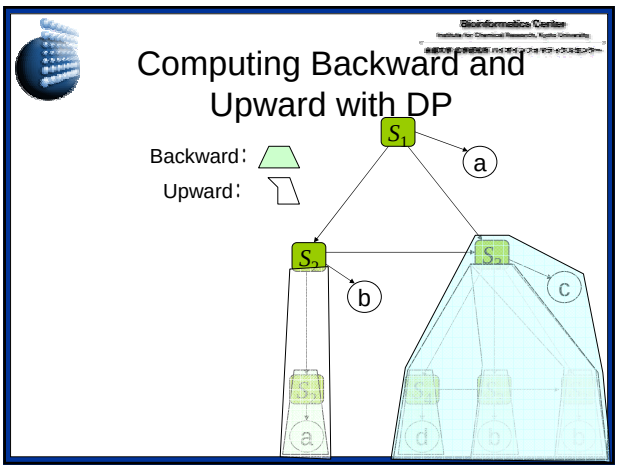
Upward: ▱



Computing Backward and Upward with DP

Backward: ⏢

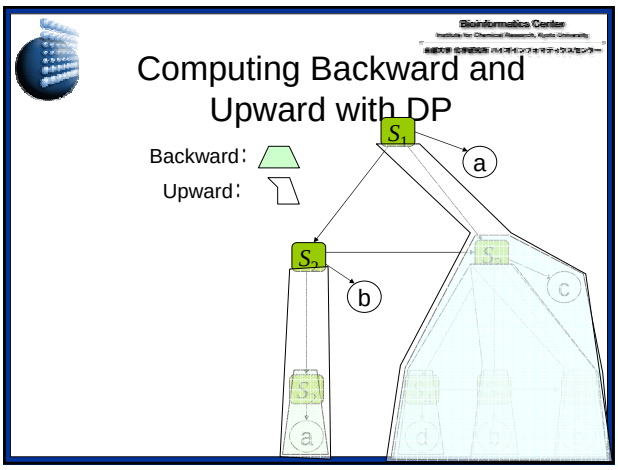
Upward: ▱



Computing Backward and Upward with DP

Backward: ⏢

Upward: ▱

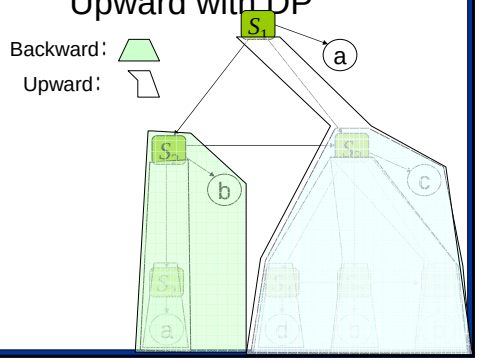


Computing Backward and Upward with DP

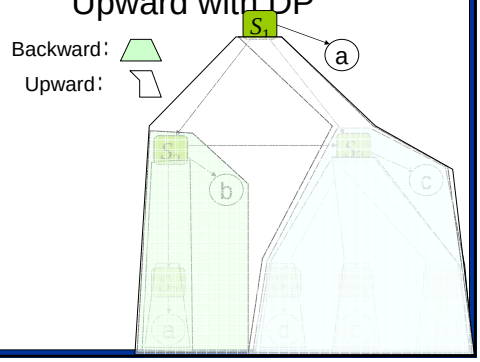
Backward: ⏢

Upward: ▱

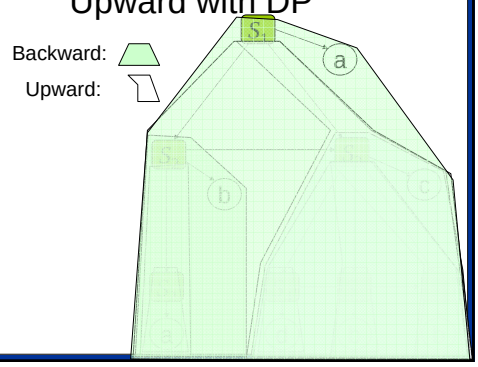
Computing Backward and Upward with DP



Computing Backward and Upward with DP



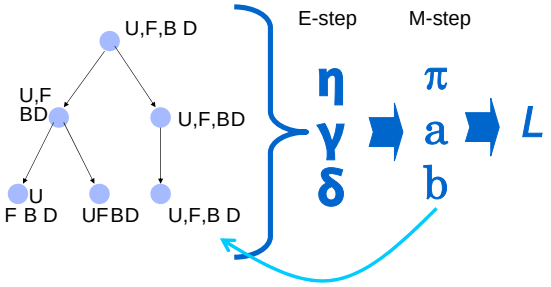
Computing Backward and Upward with DP



Estimating Parameters

- Maximum Likelihood and EM (Expectation-Maximization) as shown for HMMs
- E-step computes three expectation values in E-step:
  - $\eta(s_m)$ = initial state expectation value
  - $\gamma(\{s_q, s_m\}, s_l)$ = state transition expectation value
  - $\delta(s_l, \sigma_h)$ = label output expectation value
- M-step updates our probability parameters $\pi[s_l]$, a $[\{s_q, s_l\}, s_m]$, and b $[s_l, \sigma_h]$ using these expectation values
- Repeat E-M until likelihood is maximized

Estimating Parameters: Iteration Procedure



Computing the Likelihood

- The likelihood of tree T_u given a set of parameters $\boldsymbol{\theta}$ can thus be found at the root node x_1 as:
  - $L(T_u; \boldsymbol{\theta}) = \Sigma\, \pi[s_l]\, U_u(s_l, x_1)$
- The likelihood for a set of trees $\mathbf{T} = \{T_u, \ldots, T_n\}$ given a set of parameters $\boldsymbol{\theta}$ can thus be computed as a product of the likelihood of each tree:
  - $L(\mathbf{T}; \boldsymbol{\theta}) = \Pi_u\, \Sigma\, \pi[s_l]\, U_u(s_l, x_1)$
    $= \Pi_u\, L(T_u; \boldsymbol{\theta})$

Most Likely State Transition

- HMMs:
  - Viterbi algorithm
  - Used for multiple sequence alignment
- PSTMMs:
  - Viterbi algorithm
  - For Multiple tree alignment

The Most Likely State Transition

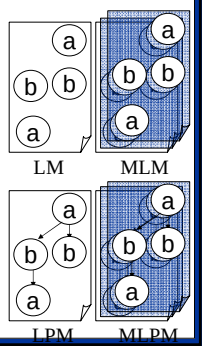
- Based on maximum likelihood parameters, we can calculate:
  - $\boldsymbol{\phi}_B(s_q, s_m, x_j)$ = maximum state transition probability
  - $\boldsymbol{\phi}_U(s_q, x_p)$ = maximum output label probability
  - $\boldsymbol{\tau}_B(s_q, s_m, x_j)$ and $\boldsymbol{\tau}_U(s_q, x_p)$, which retrieve the actual states corresponding to these maximum values (argmax of $\boldsymbol{\phi}$)

Empirical Experiments

- Experimental Setting
  - Synthetic
    - Performance comparison
      - Capture sibling-dependent patterns
      - Discriminate between those that do and do not contain these patterns
      - Use five-fold cross validation
  - Real data: glycans
    - Performance comparison
    - Analyzing patterns found

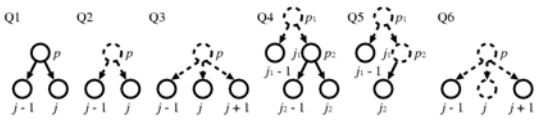
Synthetic Data Experiment

- Compared to five models that do not incorporate any dependencies among children.
  - HTMM
  - Label model (LM)
  - Mixture of label model (MLM)
  - Label pair model (LPM)
  - Mixture of label pair model (MLPM)



Synthetic Data Experiment

- Positive data set consisted of trees of equivalent size, embedded with various types of patterns:
- Negative data set (which is used only for test) consisted of trees of equivalent size, keeping the same distribution of parent-child pairs as that in positive data set



Results

- Trees with patterns vs. ones without patterns.
  - AUC (Area under the ROC curve)
    - Equivalent to Mann-Whitney-Wilcoxon text and Gini index
  - PSTMM outperformed being statistically significant (*t*-test).

Pattern	PSTMM	HTMM	MLPM	LPM	MLM	LM
Q1	**87.6**	56.9 (**13.0**)	71.8 (**12.5**)	50.7 (**40.0**)	49.9 (**64.0**)	57.6 (**40.9**)
Q2	**89.1**	51.4 (**24.4**)	48.5 (**32.8**)	58.7 (**27.6**)	52.6 (**39.0**)	53.9 (**33.9**)
Q3	**96.1**	53.0 (**55.0**)	51.2 (**75.2**)	58.4 (**36.8**)	55.6 (**37.8**)	56.3 (**31.2**)
Q4	**80.0**	48.9 (**10.3**)	58.9 (**6.2**)	60.3 (**6.5**)	49.9 (**9.7**)	54.2 (**8.9**)
Q5	**70.3**	50.5 (**13.1**)	51.8 (**16.1**)	50.0 (**18.3**)	49.9 (**21.7**)	47.1 (**16.2**)
Q6	**84.5**	50.3 (**48.4**)	49.1 (**21.5**)	51.5 (**39.1**)	49.9 (**29.6**)	49.8 (**27.4**)

Results

- Trees with patterns vs. ones without patterns.
  - Accuracy
    - (TP+TN)/(TP+FP+FN+TN)
    - Threshold is selected such that the accuracy is maximized.
  - PSTMM outperformed being statistically significant (*t*-test).

Pattern	PSTMM	HTMM	MLPM	LPM	MLM	LM
Q1	**80.9**	58.1 (**14.9**)	66.8 (**10.7**)	52.9 (**30.4**)	50.0 (**35.0**)	58.3 (**21.8**)
Q2	**83.8**	53.5 (**30.0**)	51.2 (**47.5**)	58.0 (**32.7**)	58.6 (**26.9**)	55.9 (**31.6**)
Q3	**90.7**	54.3 (**29.9**)	53.2 (**43.4**)	58.7 (**37.4**)	60.7 (**35.0**)	56.5 (**25.9**)
Q4	**73.9**	52.2 (**8.1**)	58.1 (**5.97**)	59.5 (**4.86**)	50.0 (**8.73**)	56.6 (**6.27**)
Q5	**66.1**	52.5 (**16.5**)	53.3 (**10.4**)	52.0 (**15.1**)	50.0 (**18.5**)	52.2 (**12.9**)
Q6	**78.2**	52.3 (**36.2**)	51.4 (**18.7**)	52.9 (**30.2**)	50.0 (**25.1**)	53.3 (**20.1**)

Results

- Trees with patterns vs. ones without patterns.
  - Precision at recall of 30%
    - TP/(TP+FP)
    - Threshold is selected where the recall TP/(TP+FN) = 0.3
  - PSTMM outperformed being statistically significant (*t*-test).

Pattern	PSTMM	HTMM	MLPM	LPM	MLM	LM
Q1	**99.2**	53.5 (**22.0**)	76.8 (**14.0**)	49.7 (**28.3**)	50.0 (**123.5**)	54.3 (**28.8**)
Q2	**95.3**	50.3 (**37.1**)	48.8 (**61.5**)	59.2 (**35.4**)	46.9 (**75.0**)	50.0 (**38.4**)
Q3	**99.0**	51.2 (**46.2**)	53.8 (**39.1**)	64.0 (**13.7**)	48.8 (**55.7**)	55.8 (**36.4**)
Q4	**87.3**	47.6 (**7.3**)	59.1 (**4.64**)	57.8 (**6.44**)	50.0 (**6.99**)	51.2 (**7.03**)
Q5	**75.7**	51.0 (**16.9**)	51.6 (**14.8**)	48.9 (**13.2**)	50.0 (**13.6**)	45.4 (**13.9**)
Q6	**88.2**	50.0 (**31.6**)	49.2 (**19.0**)	52.5 (**44.2**)	50.0 (**29.2**)	47.4 (**28.4**)

Applying PSTMM to Glycans

- Glycan classes

Table 2. Glycan classes from KEGG Glycan, sorted by total number of structures. (A) is the total number of glycans, (B) the total number of sibling pairs, and (B) the average number of nodes within each glycan.

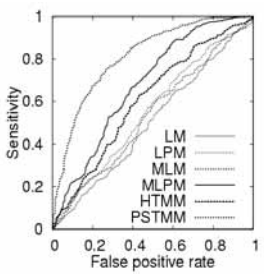
Tested classes

Class	(A)	(B)	(C)
N-Glycan	2068	1865	11.1
Glycoside	931	380	4.1
Sphingolipid	914	421	8.1
O-Glycan	746	515	6.6
Glycosaminoglycan	596	431	8.3
LPS	447	213	7.3
Polysaccharide	344	153	8.6
Neoglycoconjugate	111	57	6.4
GPI	75	47	7.3
Oligosaccharide	34	2	4.5
Glycerolipid	30	2	3.5

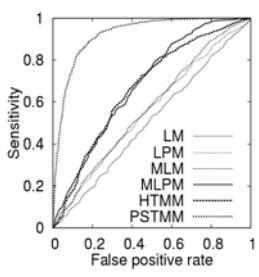
Glycan Data Experiment

- Data set consisted of
  - true glycans
  - "negative" data
- Negative data: randomly generated trees containing <u>same distribution</u> of *parent-child labels* as true glycans
- Five-fold cross validation

Glycan Results

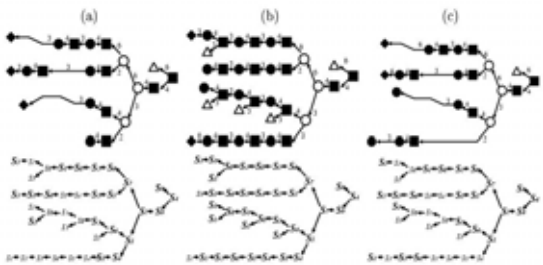


O-Glycan: AUC 80.1

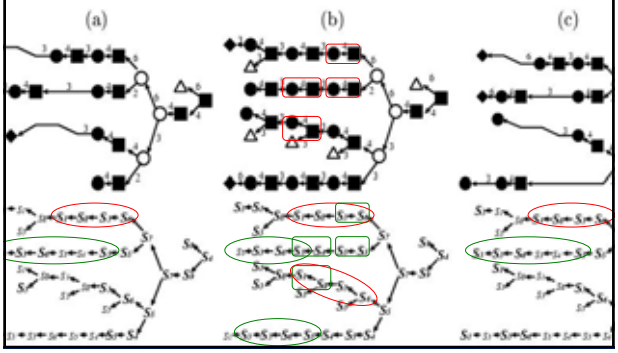


N-Glycan: AUC 92.0

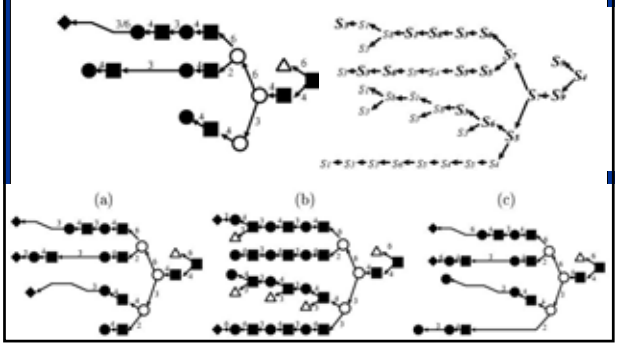
Glycan Pattern Mining



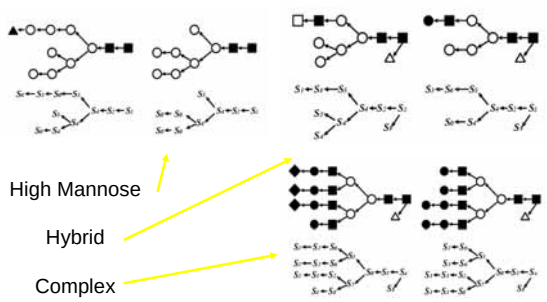
Glycan Pattern Mining



Glycan Tree Alignment



Learned Classification



Computational Complexity of PSTMM

- Equivalent to context free grammars for strings: maximal practical bound

	Time	Space
U, ϕ_U, τ_U	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^2 \cdot \lvert V\rvert)$	$O(\lvert S\rvert \cdot \lvert V\rvert)$
B, ϕ_B, τ_B	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^3 \cdot \lvert V\rvert)$	$O(\lvert S\rvert^2 \cdot \lvert V\rvert)$
F	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^3 \cdot \lvert V\rvert)$	$O(\lvert S\rvert^2 \cdot \lvert V\rvert)$
D	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^3 \cdot \lvert V\rvert)$	$O(\lvert S\rvert \cdot \lvert V\rvert)$
$\mu(a)$	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^3 \cdot \lvert V\rvert \cdot \lvert C\rvert)$	$O(\lvert S\rvert^3)$
$\mu(b)$	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert \cdot \lvert V\rvert)$	$O(\lvert S\rvert \cdot \lvert\Sigma\rvert)$
$\mu(\pi)$	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert \cdot \lvert V\rvert)$	$O(\lvert S\rvert)$
$\hat{a}$	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^3)$	$O(\lvert S\rvert^3)$
$\hat{b}$	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert \cdot \lvert V\rvert)$	$O(\lvert S\rvert \cdot \lvert\Sigma\rvert)$
$\hat{\pi}$	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert)$	$O(\lvert S\rvert)$

Drawbacks of PSTMM

- Computational complexity is maximal practical bound
- Overfitting problems
- Difficult to retrieve patterns from learned states

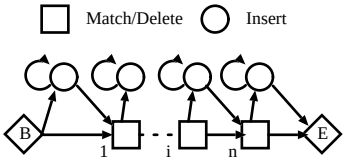
New Models of Labled Ordered Trees

- Profile PSTMM
  - Incorporate match, insert and delete states
  - Utilize new state transitions: Down and Right
- Ordered Tree Markov Model (OTMM)
  - Reduce dependencies on parents: State depends on that at the immediately elder sibling only, except that the eldest siblings which depend on their parents

Outline

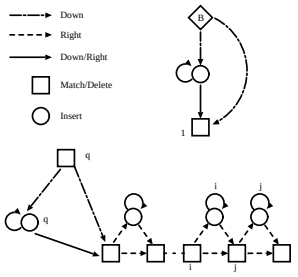
- What is a Glycan?
- Glycan = Labeled Ordered Tree
- Databases on Glycans
- Probabilistic Models for Labeled Ordered Trees and their empirical experimental results
  - Probabilistic Sibling Dependent Markov Model (PSTMM)
  - Profile PSTMM
  - Ordered Tree Markov Model
- Concluding Remarks

ProfileHMMs



- Match and Delete states can be found together at same positions.
- Insert states loop back to themselves or to the next position.
- Begin state at start and End state at end.

ProfilePSTMM State Model



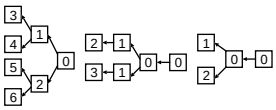
- Two types of transitions:
  - parent-child
  - between siblings
- Begin state transitions down to root node
  - Also represents End state
- Positions (1, …,i, … n) are fixed
- Each position has C(i) children positions

Probability Parameters

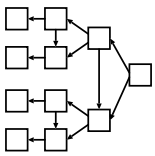
- Similar Forward, Backward, Upward and Downward parameters
- State positions are fixed:
  - Tree is traversed together with state model
  - No need to traverse every combination of states (as when they are free)
  - Much more efficient
- Profiles could be retrieved directly from match states' label output probabilities

Synthetic Experiment

- Three types of profiles
- Fifty trees containing a profile
- Fifty negative data trees with same parent-child label distribution
- Fixed state model

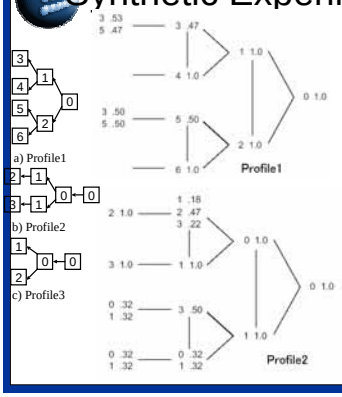


a) Profile1 b) Profile2 c) Profile3



State model

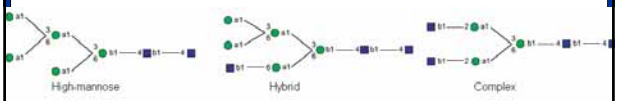
Synthetic Experiment Results



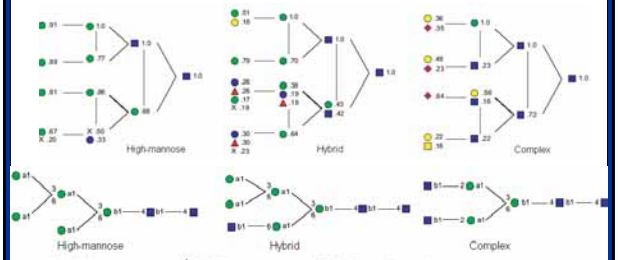
	P1	P2	P3
Accuracy	.914	.788	.892
Precision	.843	.974	.926
AUC	.910	.868	.903

Glycan Class Experiment

- Test the three sub-classes of N-Glycans:
  - High-mannose, Hybrid, Complex type
- 50 examples for each subtype (positive examples)
- 50 negative examples generated from distribution of positive examples



Glycan Class Experiment

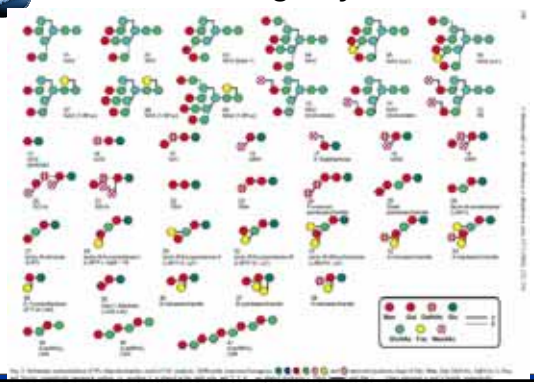


	High-mannose	Hybrid	Complex
Accuracy	.978	.982	.970
Precision	.882	.904	.882
AUC	.959	.966	.954

Galectin Experiment

- Galectins are glycan-binding proteins
- Recognize galactose at leaves
- Details still not completely understood
- Binding affinity for specific glycans tested for major galectins in Hirabayashi et al. 2002.
- Positive data: 30 weighted glycans
- Negative data: 30 glycans with same parent-child label distributions

Galectin Binding Glycan Data

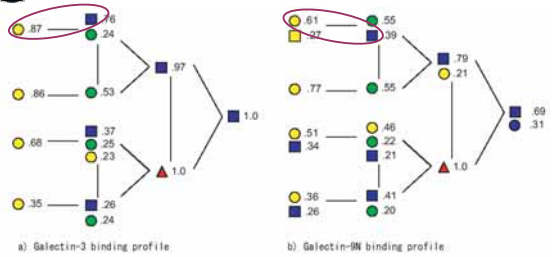


Galectin Binding Affinity Data

Table 3. Binding affinities and weights for Galectin-3 and Galectin-9N. Affinity values are normalized and inverted from the original data by Hirabayashi [16] such that higher values indicate higher affinity. Abbreviations: NA3: triantennary N-Glycan; fuc. NA3: core-fucosylated NA3; NA4: tetraantennary N-Glycan; fuc. NA4: core-fucosylated NA4; penta.: pentasaccharide; A-hexa: A-hexasaccharide; LN3: LAcNAc; LN5: $(LacNAc)_2$

	Gal-3 affinity (weight)	Gal-9N affinity (weight)
NA3	1.28205 (1)	2.6316 (2)
fuc. NA3	1.21951 (1)	2.2222 (2)
NA3 type1	1.08696 (1)	1.6949 (0)
NA4	1.44928 (1)	5.5556 (5)
fuc. NA4	1.40845 (1)	4.3478 (4)
Galili penta.	1.47059 (1)	0.2273 (0)
Forssman penta.	0.16129 (0)	11.111 (11)
A-hexa	1.5873 (1)	3.8462 (3)
LN3	2.85714 (2)	1.2346 (0)
LN5	5.26316 (5)	8.3333 (8)

Galectin Binding Glycan Profiles



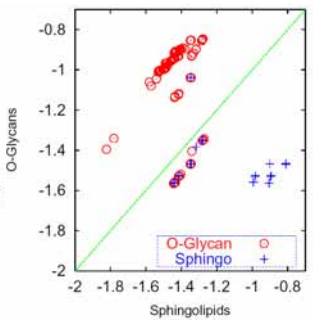
	Gal-3	Gal-9N
Acc	.847	.91
Prec	1.0	.918
AUC	.93	.931

Testing Profile Differentiation

- Train two profiles on different data sets
- Test both on both data sets to assess differentiation ability

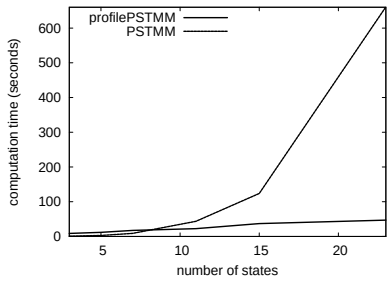
Table 5. 2x2 Contingency matrix for O-Glycan vs. Sphingolipid (Sphingo) class differentiation. The resulting discrimination rate is 923/968=95.4%.

	O-Glycan model	Sphingo model
O-Glycans	445	44
Sphingo	1	478



Improvement in Computation Time

- Efficiency increased by O(|S|)

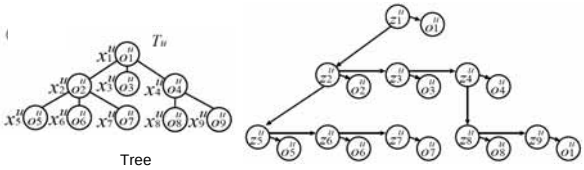


Outline

- What is a Glycan?
- Glycan = Labeled Ordered Tree
- Databases on Glycans
- Probabilistic Models for Labeled Ordered Trees and their empirical experimental results
  - Probabilistic Sibling Dependent Markov Model (PSTMM)
  - Profile PSTMM
  - Ordered Tree Markov Model
- Concluding Remarks

Ordered Tree Markov Model (OTMM)

- State depends on that of the immediately elder sibling, except the eldest siblings which depend on the parents



Tree

Dependencies

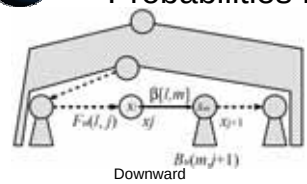
Define OTMM Parameters

- Three probability parameters
  - Initial state probability: $\pi[s_l](=P(z_1^u = s_l))$
    - Probability that the state of the root is s_l
  - State transition probability (two cases):
    $a[s_q, s_m](=P(z_j^u = s_m \mid z_p^u = s_q)$
    - Probability that the state of *j* is s_m given that the state of the parent is s_q (eldest siblings)
    $a[s_l, s_m](=P(z_j^u = s_m \mid z_i^u = s_l)$
    - Probability that the state of *j* is s_m given that the state of the immediately elder sibling is s_l (otherwise)
  - Label output probability: $b[s_l, \sigma_h]$
    - Probability that the state s_l outputs σ_h

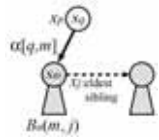
Ordered Tree Markov Model (OTMM)

- Same learning scheme applied, i.e. EM algorithm
- Need four auxiliary probabilities again: Forward, Backward, Upward and Downward
- However, significant modification required for dynamic programming updating, since a state in OTMM does not depend on that of a parent, except the eldest siblings

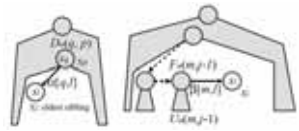
DP for Computing Four Auxiliary Probabilities for OTMM



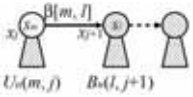
Downward



Upward



Forward

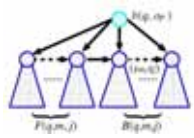


Backward

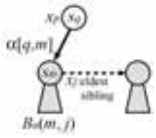
Comparison with PSTMM

- Relatively similar two auxiliary probabilities
  - Upward: child to parent

PSTMM:

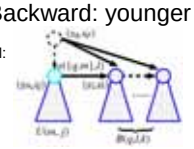


OTMM:

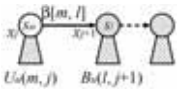


  - Backward: younger sibling to elder

PSTMM:



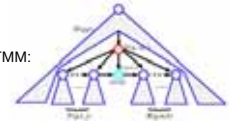
OTMM:



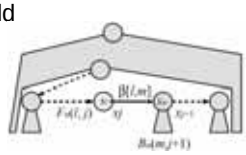
Comparison with PSTMM

- Significantly different auxiliary probabilities
  - Downward: parent to child

PSTMM:

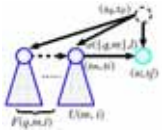


OTMM:

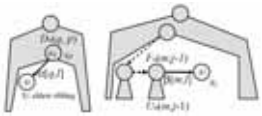


  - Forward: older sibling to younger

PSTMM:



OTMM:



EM update

- The same as that of PSTMM
- E-step computes three expectation values:
  - $\eta(s_m)$ = initial state expectation value
  - $\gamma(s_m, s_l)$ = state transition expectation value
  - $\delta(s_l, \sigma_h)$ = label output expectation value
- M-step updates our probability parameters $\pi[s_l]$, a $[s_l, s_m]$, and b $[s_l, \sigma_h]$ using these expectation values
- Repeat E-M until some stopping condition satisfied

Computational Complexity of OTMM

- Comparison with PSTMM and HTMM
  - Efficiency always increased by O(|S|)

	Time
OTMM	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^2 \cdot \lvert V\rvert)$
HTMM	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^2 \cdot \lvert V\rvert)$
PSTMM	$O(\lvert\mathbf{T}\rvert \cdot \lvert S\rvert^3 \cdot \lvert V\rvert \cdot \lvert C\rvert)$

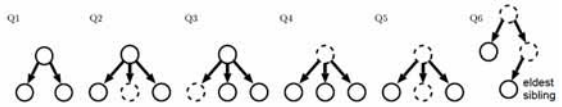
	Space
OTMM	$\max\{O(\lvert S\rvert \cdot \lvert V\rvert), O(\lvert S\rvert^2), O(\lvert S\rvert \cdot \lvert\sum\rvert)\}$
HTMM	$\max\{O(\lvert S\rvert \cdot \lvert V\rvert), O(\lvert S\rvert^2), O(\lvert S\rvert \cdot \lvert\sum\rvert)\}$
PSTMM	$\max\{O(\lvert S\rvert^2 \cdot \lvert V\rvert), O(\lvert S\rvert^3), O(\lvert S\rvert^2 \cdot \lvert\sum\rvert)\}$

Experimental Setting

- Synthetic
  - Performance comparison with PSTMM
    - Capture sibling-dependent patterns
    - Discriminate between those that do and do not contain these patterns
    - Evaluate predictive accuracy and computation time using five-fold cross validation
- Real data: glycans
  - Performance comparison
  - Analyzing patterns found

Synthetic Data Experiment

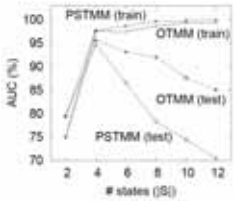
- Data set consisted of trees of equivalent size, embedded with various types of patterns:



- Parameters:
  - K: generated patterns
  - |T|: #training trees
  - |S|: #states

Synthetic Data Results

- Trees with patterns vs. ones without patterns.
  - Fixed Q1, |T|=100 and, K=1,
  - Changed |S|=2,…,12
  - AUC (Area under the ROC curve) for both training and test
  - OTMM avoided overfitting to the data found in PSTMM.



Synthetic Data Results

- Trees with patterns vs. ones without patterns.
  - Fixed Q1 and K=3,
  - Changed |S|=2,…,12 and |T|=100,…,600
  - Computation time in training
  - AUC (Area under the ROC curve) for test
  - OTMM avoided overfitting, keeping much less computation time

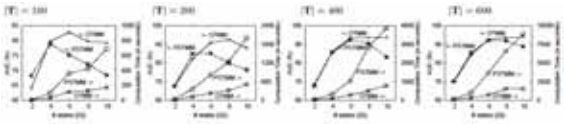


Figure 11: AUC and computation time for $K = 3$.

Synthetic Data Results

- Trees with patterns vs. ones without patterns.
  - Fixed Q1 and |T|=200
  - Changed |S|=2,…,12 and K=1,…,4
  - Computation time in training
  - AUC (Area under the ROC curve) for test
  - OTMM avoided overfitting, keeping much less computation time

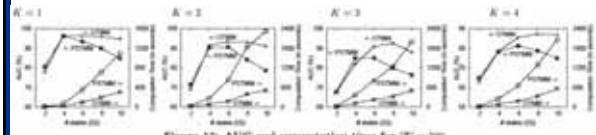


Figure 12: AUC and computation time for $|T| = 200$.

Synthetic Data Results

- Trees with patterns vs. ones without patterns.
  - Fixed K=2, |T|=400 and |S|=6 where overfitting avoided for both OTMM and PSTMM
  - Computation time in training
  - AUC (Area under the ROC curve)
  - OTMM reduced the computation time drastically, keeping the same predictive performance.

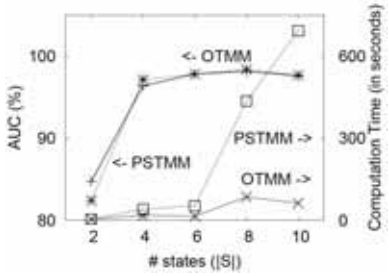
	Model	Q1	Q2	Q3	Q4	Q5	Q6
AUC (%)	OTMM	**93.4**	87.2	88.6	**96.6**	**81.8**	**82.0**
	PSTMM	92.3	**89.9**	**91.8**	95.0	79.9	75.2
Comp. Time (in seconds)	OTMM	**438.7** (0.204)	**583.8** (0.269)	**608.8** (0.309)	**379.5** (0.193)	**829.8** (0.239)	**581.4** (0.179)
	PSTMM	2145.6 (1.0)	2173.8 (1.0)	1970.8 (1.0)	1961.9 (1.0)	3475.4 (1.0)	3257.1 (1.0)

Glycan Data Experiment Performance Comparison

- Data set consisted of
  - Positives: N-glycans
  - Negatives: O-glycans
- Used cross-validation in the same manner as synthetic data
- Used parameter settings achieved the best performance in synthetic data

Glycan Data Experiment Performance Comparison Results

- OTMM more efficient computationally, keeping the same predictive performance



Glycan Data Experiment Glycan Pattern Mining 1

- N-Glycan

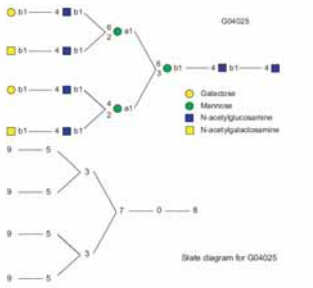


Figure 14: Example of the states learned using OTMM for a specific glycan structure.

Glycan Data Experiment Glycan Pattern Mining 2

- Three subclasses in N-glycan

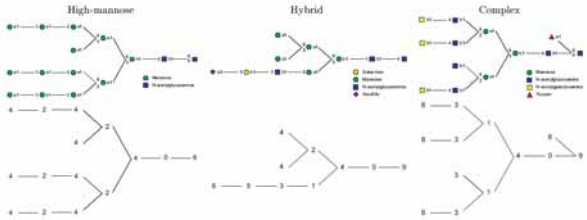


Figure 15: (top) The actual glycans, and (bottom) the most likely state paths.

Summary

- Proposed a family of probabilistic models for labeled ordered trees and their efficient learning scheme
  - OTMM reduced the complexity reasonably, avoiding overfitting and keeping the predictive performance
- Structure-based analysis: First step of glycome informarics
- There indeed seem to exist sibling-dependent relationships in glycans!
- Statistical analysis of glycans seem appropriate considering the noisiness of the data
- Important to link with other information
  - Functional annotations of genes and proteins that interact and bind with glycans

Acknowledgement

- Kiyoko F. Aoki-Kinoshita
- Nobuhisa Ueda
- Kosuke Hashimoto
- Minoru Kanehisa

(Bioinformatics Center, Kyoto University)

References

- N. Ueda, K. F. Aoki and H. Mamitsuka, SDM 2004
- K. F. Aoki et al, SIGMOD Record 2004
- K. F. Aoki et al, ISMB 2004
- N. Ueda et al, TKDE 2005