Informatic innovations in glycobiology: relevance to drug discovery

Hiroshi Mamitsuka

Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji 611-0011, Japan

The recent development and applications of tree-based informatics on glycans have accelerated the biological analysis on glycans, particularly from structural viewpoints. We review three major aspects of recent informatics innovations on glycan structures: maturity of well-organized databases on glycan structures linking with other biological information, implementation of glycan structure matching algorithms and extensive development of methods for mining frequent patterns from glycan structures.

Introduction

Carbohydrate sugar chains, or glycans, are a major class of biological molecules. Located on the outer surface of cellular macromolecules, glycans are crucial for the functioning of multicellular organisms. In particular, the recognition of glycans on cellular surfaces can promote or inhibit various diseases, implying that glycans can be important drug targets. Glycans hold branch-shaped tree structures of monosaccharides, which are different from genes and proteins, that is, strings of only a small set of letters. More than 20 types of monosaccharides are already known, but the full extent of existing monosaccharides is not known as yet. Besides, a large number of different types of linkages exist in glycans, while only one type of linkage is used for genes (and also proteins). These complex features of glycans have hindered the advancement of sequence analysis of glycans for a long time. However, glycans have recently been modeled as labeled trees or labeled ordered trees from general informatics perspectives. This indicates that existing efficient and effective approaches used for trees in informatics can be applied to analyzing glycans. In addition, original informatics methods have been developed for the problems, where appropriate methods for trees had not been developed.

We emphasize that these recent informatics innovations in structural analysis in glycobiology are very important, because they will speed up the analysis of tree-shaped biological molecules, which have not been analyzed well by any data-intensive approach so far. For example, for genes and proteins, the amino/nucleic acid frequencies in sequences are well known and contained within any sequence database as supplementary statistics. On the contrary, even this type of information was totally unclear for glycans and monosaccharides. Currently, however, glycan structural information is becoming freely accessible on-line and sufficiently well maintained to allow this type of information to be easily computed by any user. Should a user care to analyze the data further, this would be also possible. For example, a user can have the frequency of a part of a tree. At the same time, all appearances can be retrieved from the database by tree matching between a query, that is, a tree part, and all entries in the database. This tree part can be extended to a set of tree parts by replacing exact local matching with global matching, meaning that any query works effectively to retrieve all corresponding sets from glycan databases. Furthermore, even without casting any query, the most frequent or likely patterns can be automatically retrieved from the database by using a data mining method for trees.

We further emphasize that these innovations contribute to informatics-based drug discovery. A simple and direct explanation for this is that glycans can be drug targets, because cellular recognition by glycans is deeply related with the alteration of diseases. For example, there are types of glycans which have the ability to confer drug resistance in some cancer cells [1], meaning that one or more structural patterns are hidden in the drug resistance set of *N*-glycans, which would be captured by a data-mining method for trees and may represent a way of identifying targets for drug discovery. More directly, glycans can be drugs, meaning that structural analysis on glycans, such as grouping glycans and

E-mail address: mami@kuicr.kyoto-u.ac.jp.

1359-6446/06/$ - see front matter doi:10.1016/j.drudis.2007.10.013

mining patterns in some set of glycans, could directly accelerate drug discovery processes. We emphasize again that these informatics-based drug discovery would not be possible without recent informatics innovations for trees or glycans.

In this review, we introduce three major aspects of recent informatics innovations for the analysis of the tree structures of glycans: rapid expansion of glycan structure databases; implementation of glycan structure matching algorithms and extensive development of algorithms for mining patterns from glycan structures. Below, we describe the notations that are used in the general informatics literature and that will be used throughout this review: a graph consists of nodes and edges; a *tree* is an acyclic connected graph. A *rooted tree* is a tree that has a special node, called the *root*. A node, which is on a unique path from the root to x, is an *ancestor* of x, and x is a *descendant* of this node. If an ancestor is only one edge away from node x, this is a parent, and x is a *child* of this node. A *subtree* is a tree consisting of all descendants of a node. Two nodes are *siblings* if they have the same parent. An *ordered tree* is a rooted tree in which siblings are ordered. A *labeled tree* is a tree in which a label is attached. For glycans, nodes and edges are monosaccharides and linkages, respectively. Labels are types of monosaccharides. Siblings can be ordered, according to carbon numbers on linkages to the same parent.

Glycan structure databases

Bioinformatics began approximately 30 or more years ago, with the development of gene/protein sequence databases. For glycans, the first database was the Complex Carbohydrate Structure Database (CCSD) or CarbBank [2,3], which was launched in the late 1980s and maintained, for approximately ten years, by a project of the University of Georgia's Complex Carbohydrate Research Center. This database of carbohydrate structures and annotations became the basis for a number of recent databases for glycan structures, such as SWEET-DB [4] and GlycoSuiteDB [5]. Currently, the three most significant, publicly available databases on glycan structures are KEGG GLYCAN [6], glycoSCIENCES.de [7] (containing SWEET-DB as part) and a database from the Consortium for Functional Glycomics (CFG) [8,9].

They are all well organized, each with their own unique, distinguishing features. For example, in KEGG GLYCAN, annotation information on each glycan is linked to KEGG's various other genomic data, particularly genes or enzymes which catalyze reactions for synthesizing the corresponding glycan [10]. This feature is likely to help those researching a glycan, or glycans, in order to obtain a comprehensive view of the biological phenomena in which it participates. On the contrary, a distinctive feature of the CFG database is the abundant information on microarray data, which is helpful for integrative analysis of glycan-related biological aspects as well. A special feature of glycoSCIENCES.de is that it contains the information on not only the two-dimensional tree structures of glycans, but also their three-dimensional structures. This information might be a significant advantage in the future, when two-dimensional branched tree structures prove insufficient for the analysis of glycans.

Comparing glycan structures

An important and necessary function of a database is to retrieve entries satisfying a query. For glycans, we need an algorithm for matching two given trees, which was developed in computer science in the 1970s [11] and applied, in bioinformatics, to the problem of matching two secondary structures of RNAs [12]. This problem of finding the maximum common subtree, that is, the common subtree with the largest number of nodes, of given two trees in the literature of computer science is solvable in polynomial time [13], based on dynamic programming [14]. In particular, for glycans, an appropriate scoring mechanism has been developed to find an optimal match score for two given glycans or labeled trees, meaning that we have an efficient algorithm to obtain a global optimum in glycan tree matching.

An implementation example of the above algorithm, KCaM [15,16] which is equipped in KEGG GLYCAN has a lot of options. One example is exact or approximate matching. Exact matching tries to find only a connected component (of nodes and edges) corresponding to a query, while approximate matching allows multiple components, which can be separated from one another. Another option is global or local matching. The most standard usage is global approximate matching, which is based on a dynamic programming procedure for strings. That is, all possible node pairs of two input trees are matched from leaves to the root in a dynamic programming manner. More concretely, when we match two nodes in given two trees, we have to consider the following two types: (1) two nodes are matched, and we then choose the combination in children which gives the maximum matching score; (2) two nodes are unmatched, and we then choose the pair of one node and a child of the other node which gives the maximum matching score. Figure 1 shows the schematic picture of these two types. A resulting maximum common subtree with the maximum score can be retrieved by backtracking to find the matching nodes that contributed to this score.

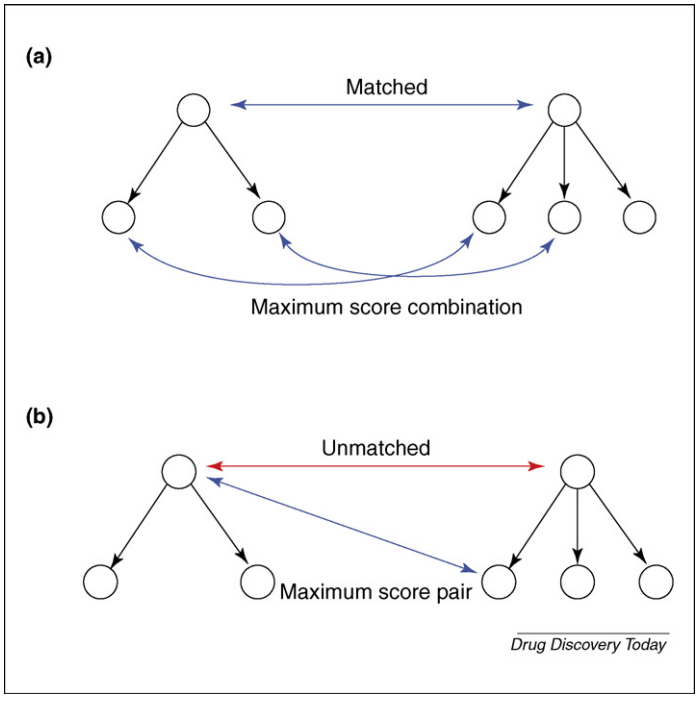

FIGURE 1

Two types of approximate matching: corresponding two nodes are (a) matched or (b) unmatched.

By applying a tree-matching algorithm to all possible pairs of glycans, we can have the statistics of matched disaccharides (parent–child) and their linkages (edges). That is, we can check how many times the combinations of disaccharides and their linkages appear in resultant pairwise tree alignments. A score matrix for these combinations was already computed based on the above statistics [17], just as PAM [18] or BLOSUM [19] for amino acids, which were computed from multiple sequence alignment of proteins.

Mining knowledge from glycan structures

Mining, or learning from data, can be classified into two categories: supervised and unsupervised learning. Supervised learning classifies examples into multiple classes; perhaps two classes, such as positives and negatives. A general procedure for supervised learning is that we first train rules from labeled examples to classify and then, for a new unknown example, a class can be assigned by using the trained rules. On the contrary, unsupervised learning uses unlabeled examples in training and attempts to find frequent patterns in those examples. Then, given a new example, it can be shown how similar this example is to training examples.

Currently, a widely used approach in supervised learning is a technique called support vector machine (SVM), which uses a so-called kernel function that can define a similarity between two arbitrary training examples [20,21]. A naïve approach of supervised learning is to identify a simple linear function that can classify examples in a given data space, but it is usually very hard to find. The basic idea of SVM is to expand the simple input data space to a complex data space that can be defined by a kernel function and then attempt to find a linear classification function in this new space. This implies that the classification accuracy of SVM heavily depends upon the kernel function. For biological strings, such as genes and proteins, a kernel function called 'string kernel' had already been proposed to apply [22]. For trees, in the literature of context trees in natural language processing, a kernel was proposed to compute a similarity between two given labeled trees [23]. This computation is based on a dynamic programming algorithm, which is very similar to the procedure in global approximate matching of given two trees, because the basic idea of 'tree kernel' is that the similarity of two labeled trees must be high if they share a common subtree. Such general tree kernels, which consider any labeled trees, have been presented, but so have tree kernels specialized for glycans [24], focusing on the features peculiar to glycans, such that subtrees including leaves or the root are more conserved than other subtrees. It has been already shown that in classification accuracy, SVM with glycan-specific kernels can outperform other kernels [24]. However, a disadvantage of kernel-based approaches is that it is hard to show clear rules to assign a class to a given example.

Unsupervised learning approaches include a variety of techniques. In this review, we focus on probabilistic model-based approaches which are very robust against the noise obtained from

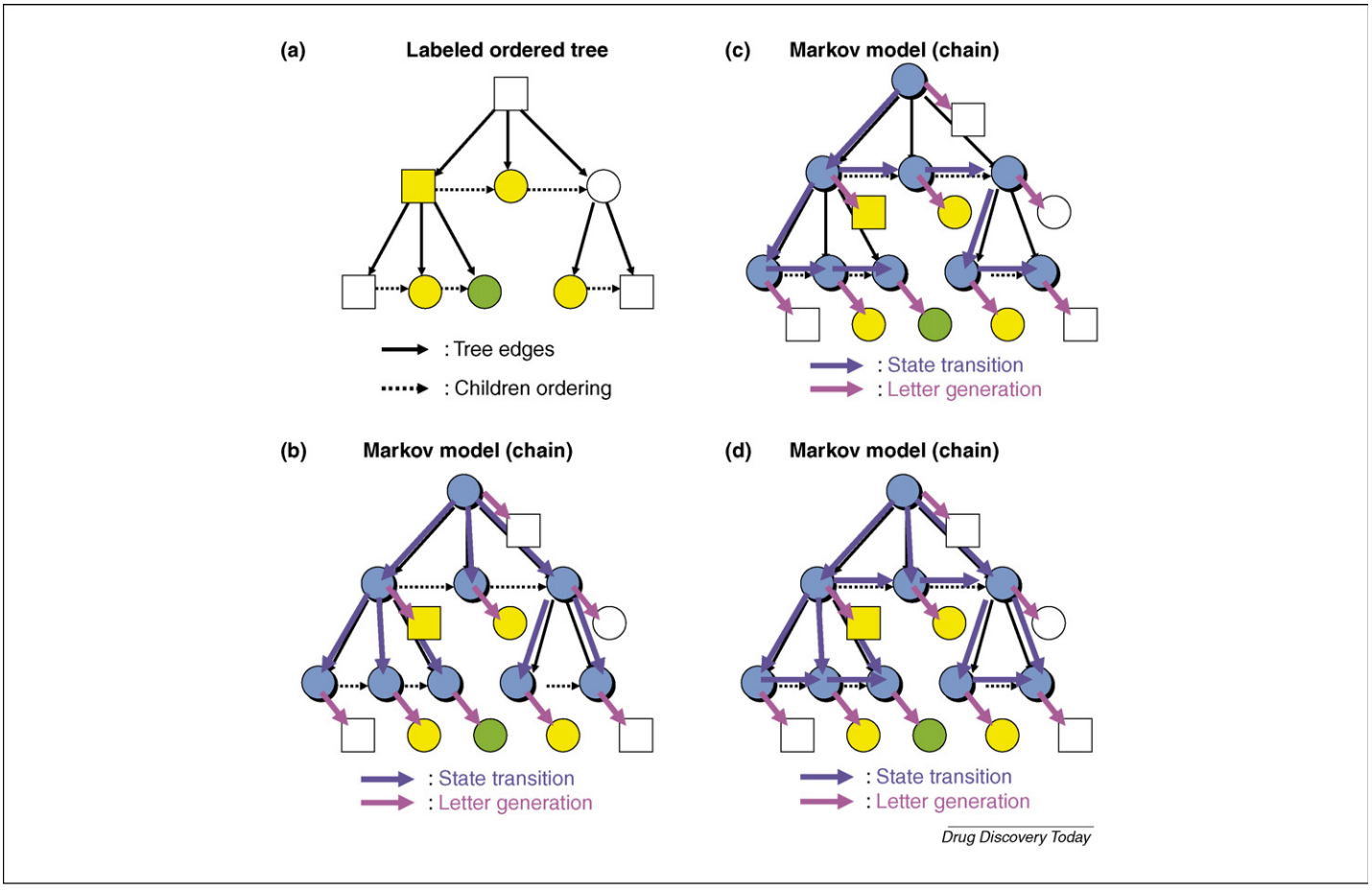

FIGURE 2

(a) An example of labeled ordered trees and probabilistic parameters of (b) HTMM, (c) OTMM and (d) PSTMM.

experimental data. A standard probabilistic model for temporal data or strings is a probabilistic Markov model (or simply a Markov model), which is defined by states and a state diagram. Normally, we assume the first-order Markov (chain) property that you can only come to a state from one state away. A hidden Markov model (HMM) [25] is a widely used Markov model for processing strings, such as speech recognition [25], natural language processing [26], and biological sequence analysis [27]. HMM has been extended or developed in a variety of ways, depending on the dataset applied. An example is ProfileHMM [27], which specifies the model structure for aligning multiple sequences and finding conserved domains in protein families. Other examples include context-free grammars (CFGs) [28] and probabilistic-tree grammars [29], both of which are also for analyzing strings in natural language processing. On the contrary, for labeled trees, HMM was already extended, in document image classification, to a model in which the Markov property is applied to the relation between nodes and parents [30]. This model, known as HTMM, or hidden tree Markov model, can be applied to glycans directly, but the recent development of probabilistic modeling for trees has revealed that more complex relations exist in glycans. We introduce two recent probabilistic models for trees, known as OTMM (ordered tree Markov model) [31] and PSTMM (probabilistic sibling-dependent tree Markov model) [32–34], and a real result obtained by applying OTMM to a current glycan database.

In these models, states correspond to nodes of a tree, and a label is generated at each state. So they have two types of probabilistic parameters: state transition probabilities attached to edges and label output probabilities attached to states in a state diagram. The Markov models for trees can be distinguished by state transition probabilities. The parameters of three tree Markov models are illustrated in Fig. 2. Figure 2(a) shows an example of labeled ordered trees. Figure 2(b) shows HTMM, in which a state transition probability is attached to the edge connecting two states corresponding to a node and a parent. Figure 2(c) shows OTMM in which a state transition probability is attached to the edge connecting two states corresponding to two siblings or a parent–child relation where the child is the eldest sibling. Figure 2(d) shows PSTMM in which a state transition probability is attached to an edge pair of two siblings and a parent–child where the child and the younger sibling are shared.

In unsupervised learning of these probabilistic models, a standard, reasonable approach for training probabilistic parameters from given data is to maximize the likelihood of given examples, and a time-efficient and widely used approach is the EM (expectation–maximization) algorithm, which guarantees to have a local maximum [35]. The EM algorithm of each of the above three models was already developed by modifying the forward–backward algorithm of HMM [25] (or the inside–outside algorithm of CFGs [28]). We note that the time and space complexities of OTMM and HTMM in training probability parameters are kept the same at the square of the number of states in an input state diagram, despite the incorporation of sibling dependencies in OTMM, because a state transition probability is always specified by two states only. On the contrary, the complexity of PSTMM reaches the cube of the number of states. For strings, the complexity of CFGs is also the cube, while kept as the square for HMM, and no more complex grammars than CFGs are practically used, meaning that the cube is a practical bound in computational complexity.

Once parameters of a probabilistic model are trained, the next issue is to assign a state to each node of a given labeled tree. This procedure is often called 'parsing', which can be solved by finding the most likely state transition path of a given tree. At the same time, parsing gives us, for an example (or a labeled tree), the likelihood which can be a score by which we can rank given examples and check the classification accuracy of a probabilistic model. Parsing algorithms of the above three models were already

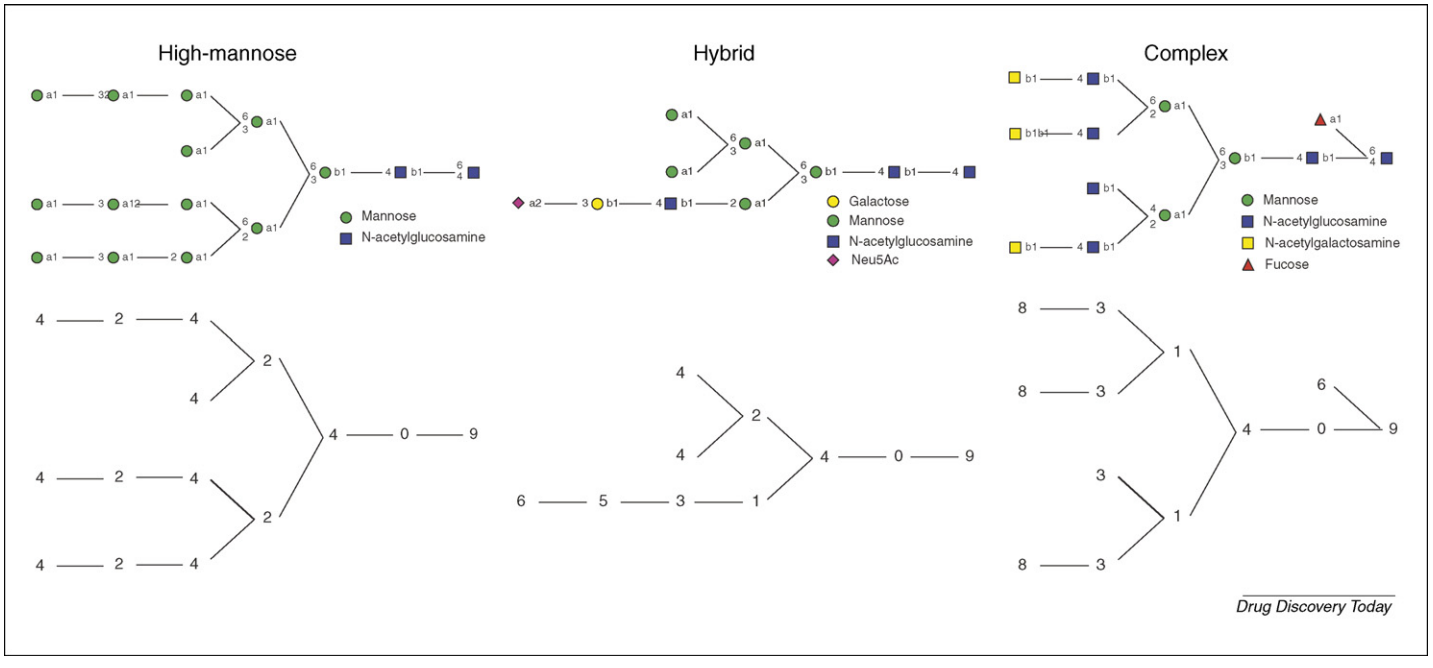

FIGURE 3

(Top) The actual glycans, and (bottom) the most likely state paths (state numbers show types, and order has no meaning.).

TABLE 1

Summary of informatics for glycan structures

Technique	DNA/protein equivalent	Refs
Glycan databases		
KEGG glycan	GenBank, EMBL, DDBJ, KEGG	[6]
Glycoscience.de		[7]
CFG		[8,9]
Tree-matching algorithms		
KCaM	Smith-Waterman, BLAST	[15,16]
Score matrices		
Glycan score matrices	BLOSUM/PAM	[17]
Supervised learning: kernel-based method		
Glycan kernel	String kernel	[24]
Unsupervised learning: probabilistic models		
OTMM	HMM	[31]
PSTMM	HMM/CFG	[32,33]
Profile PSTMM	Pfam (profile HMM) or Rfam	[34]

developed, and a thorough classification experiment on the three probabilistic models showed that OTMM and PSTMM outperformed HTMM, and the performance of OTMM and PSTMM was comparable with each other [31]. This result indicates that OTMM and PSTMM could capture patterns in training examples (glycans) more clearly than HTMM [31], implying that glycans have some particular pattern, not only in parent–child, but also in siblings. Readers interested in performance comparison should pay particular attention to reference [31].

Figure 3 shows three examples of glycans and their most likely state paths obtained by using a trained OTMM. The three examples are selected from three subclasses of *N*-glycans: high-mannose, hybrid and complex. The high-mannose type is basically made up of mannoses with the exception of the root and its children, while the complex type is made up of a variety of monosaccharides. The hybrid type is between those of the above two subclasses. For each of the three glycans, we run a parsing algorithm to establish the most likely state transition, which is shown in Fig. 3. From this figure, we can see that OTMM captured different state patterns on the same disaccharide, that is, Mannose and Mannose, depending on the subclasses. Furthermore, these three subclasses can be distinguished, based on the patterns learned. Concretely, the high-mannose type is featured by state 2 appearing at both child mannoses. These mannoses are both state 1 in the complex type. Interestingly, the hybrid type contains both types of states at child mannoses. Thus, this result indicates that the states learned can determine the subclass of glycans.

Finally, Table 1 summarizes the informatics approaches on glycan structures in a comparative table between glycans and genes/proteins.

Conclusion

A key point to make glycan structure databases more useful is to enrich the mutual links with other molecular-biological information, such as genes, proteins, biological functions, diseases and drugs. You can then obtain access to information related to particular glycans more easily. For example, you can start with a glycan, from which a link reaches to some genes, and one or more of them might be embedded in a regulatory pathway related with a disease. You can find a drug related with this disease, resulting in a new relation between drug and the starting point glycan. However, if links are too complex, the relations you can find may be too numerous to focus. In this case, the informatics approaches described may be useful. For example, probabilistic model learning can find major patterns in a glycan of interest, and glycans with the same pattern have a high likelihood of sharing links to the same biological information, meaning that the complex link network can be made simpler by reducing the number of links. Similarly, glycan-matching algorithms may assist in clustering glycans into a reasonably sized group. These are simple examples, but generalized informatics approaches to elucidating glycan structures will be vitally important to accelerate their participation in biological phenomena. In a similar vein, future work may possibly focus upon the development of new informatics approaches that can handle not only glycan structures, but also, at the same time, other biological information, such as gene sequences, microarray expression and chemical compounds.

References

1 Kudo, T. *et al.* (2007) *N*-Glycan alterations are associated with drug resistance in human hepatocellular carcinoma. *Mol. Cancer* 6, 32

2 Aoki-Kinoshita, K.F. and Kanehisa, M. (2006) Bioinformatics approaches in glycomics and drug discovery. *Curr. Opin. Mol. Ther.* 8, 514–520

3 Doubet, S. *et al.* (1989) The complex carbohydrate structure database. *Trends Biochem. Sci.* 14, 475–477

4 Loss, A. *et al.* (2002) SWEET-DB: an attempt to create annotated data collections for carbohydrates. *Nucleic Acids Res.* 30, 405–408

5 Cooper, C.A. *et al.* (2003) GlycoSuiteDB: a curated relational database of glycoprotein glycan structures and their biological sources. 2003 update. *Nucleic Acids Res.* 31, 511–513

6 Hashimoto, K. *et al.* (2006) KEGG as a glycome informatics resource. *Glycobiology* 16, 63R–70R

7 Lutteke, T. *et al.* (2006) GlycoSCIENCES.de: an internet portal to support glycomics and glycobiology research. *Glycobiology* 16, 71R–81R

8 Raman, R. *et al.* (2005) Glycomics: an integrated systems approach to structure-function relationships of glycans. *Nat. Methods* 2, 817–824

9 Raman, R. *et al.* (2006) Advancing glycomics: implementation strategies at the consortium for functional glycomics. *Glycobiology* 16, 82R–90R

10 Kanehisa, M. *et al.* (2006) From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.* 34, D354–D357

11 Tai, K. (1979) The tree-to-tree correction problem. *J. ACM* 26, 422–433

12 Lin, G. *et al.* (2001) Edit distance between two RNA structures. *RECOMB* 200–211

13 Edmonds, S. and Matula, D. (1968) An algorithm for subtree identification. *SIAM Rev.* 10, 273–274

14 Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197

15 Aoki, K.F. *et al.* (2003) Efficient tree-matching methods for accurate carbohydrate database queries. *Genome Informat.* 14, 134–143

16 Aoki, K.F. *et al.* (2004) KCaM (KEGG Carbohydrate Matcher): a software tool for analyzing the structures of carbohydrate sugar chains. *Nucleic Acids Res.* 32, W267–W272

17 Aoki, K.F. *et al.* (2005) A score matrix to reveal the hidden links in glycans. *Bioinformatics* 21, 1457–1463

18 Dayhoff, M.O. *et al.* (1983) Establishing homologies in protein sequences. *Methods Enzymol.* 91, 524
19 Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. U. S. A.* 89, 10915–10919
20 Cristianini, N. and Shawe-Taylor, J., eds (2000) *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press
21 Scholkopf, B. and Smola, A.J., eds (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press
22 Leslie, C. *et al.* (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics* 20, 467–476
23 Collins, M. and Duffy, N. (2002) New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (ACL 2002) 263–270
24 Yamanishi, Y. *et al.* (2007) Glycan classification with tree kernels. *Bioinformatics* 23, 1211–1216
25 Rabiner, L.R. and Juang, B.H. (1986) An introduction to hidden Markov model. *IEEE ASSP Mag.* 3, 4–16
26 Mannig, C. and Schutz, H., eds (1999) *Foundations of Statistical Natural Language Processing*, MIT Press
27 Durbin, R. *et al.* eds (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press
28 Lari, K. and Young, S.J. (1990) The estimation of stochastic context-free grammars using the inside-outside algorithm. *Comput. Speech Language* 4, 35–56
29 Abe, N. and Mamitsuka, H. (1997) Predicting protein secondary structure using stochastic tree grammars. *Mach. Learn.* 29, 275–301
30 Diligenti, M. *et al.* (2003) Hidden tree Markov models for document image classification. *IEEE Trans. Pat. Anal. Mach. Intel.* 25, 519–523
31 Hashimoto, K. *et al.* (2006) A new efficient probabilistic model for mining labeled ordered trees. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)* 177–186
32 Aoki, K.F. *et al.* (2004) Application of a new probabilistic model for recognizing complex patterns in glycans. *Bioinformatics* 20, i6–i14
33 Ueda, N. *et al.* (2005) A probabilistic model for mining labeled ordered trees: capturing patterns in carbohydrate sugar chains. *IEEE Trans. Know. Data Eng.* 17, 1051–1064
34 Aoki, K.F. *et al.* (2006) ProfilePSTMM: capturing tree-structure motifs in carbohydrate sugar chains. *Bioinformatics* 22, e25–e34
35 Dempster, A. (1977) Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B* 39, 1–38