Application of a Novel and Fast Information-Theoretic Method to the Discovery of Higher-Order Correlations in Protein Databases

Evan W. Steeg
Molecular Mining Corporation, PARTEQ Innovations, Queen's University, Kingston,
Ontario K7L 3N6, Canada
steeg@qucis.queensu.ca

Hai Pham
Dept. of Computing and Information Science, Queen's University, Kingston,
Ontario K7L 3N6, Canada
pham@qucis.queensu.ca

We present a fast, discrete data-mining approach to the problem of finding k-tuples of correlated amino acid residues in protein sequence data. When sets of sequence-distant sites display high mutual information, they may bespeak important structural or functional features. Our novel methodology overcomes the limitations of previous methods which examined only single-residue features or pairwise interactions.

1 Introduction

Many important scientific, medical, and industrial problems concern the discovery and analysis of higher-order features (HOFs) in biological data, both continuous and discrete, and in increasingly large databases. One such problem is the detection of inter-column correlations in aligned macromolecular sequence data, from which one might extract valuable structural, functional, and/or evolutionary knowledge. Because of the daunting computational complexity and sampling difficulties involved in the collection of very high-order probability and entropy terms, many previous approaches to this problem have ignored higher-order features entirely or made restrictive locality assumptions, often to the detriment of structure prediction and sequence classification efficacy. In this paper, we introduce a fast information-theoretic method that takes advantage of the discrete amino acid symbol representations and the very sparse nature of the high-dimensional joint probability density spaces encountered in this application.

The structure of this article is as follows. We first motivate the problem with a discussion of HOFs in protein sequence and structure modelling. Next is a precise mathematical formulation of the problem, along with a characterization of its complexity and a brief outline of previous approaches to the problem.

Finally, we present our own novel data mining methodology and report on its application to the discovery of significant k-ary inter-residue interactions in an HIV protein sequence database.

2 Motivation: Feature Detection in Protein Sequence Modelling

Given a set of aligned sequences, such as shown in Example 2.1 below, representing a protein family or superfamily, one can begin to characterize the family by finding and selecting its representative *features*.

2.1 First-Order Features

Consider this toy dataset of aligned sequences of symbols[a]:

Example 2.1

$col1$	$col2$	$col3$	$col4$	$col5$	$col6$
A	B	C	D	E	F
W	U	C	V	E	G
Z	L	C	M	W	M
V	U	C	V	A	A
G	B	C	D	Z	Z
W	L	C	M	E	Z
	$\Uparrow$		$\Uparrow$		

Each of the $M = 6$ rows represents a different protein sequence, and each of the $N = 6$ columns represents a particular position in the aligned sequences, that is, a particular amino acid residue. The first (and often only) step in a typical feature-selection process is the analysis of features corresponding to individual residue numbers (a.k.a. "positions" or "sites"); these are primitive or *first-order* features (FOFs). Features relating two or more positions are *higher-order* (HOFs). Typically, first-order and higher-order features in protein sequence analysis derive from simple calculations of the amino acid frequencies for particular positions. Estimates of underlying probability distributions from some hypothesized *population* of possible sequences (the entire protein family) are inferred from empirical frequencies over the *sample* (the known members of the family). These estimates may be based entirely and directly on frequency counts, or may combine frequency counts with other factors, such as prior probabilities and regularizing terms, as in a proper Bayesian approach [1]; extensive "bootstrap" and similar re-sampling computations may also assist.

[a] These are not real amino acid symbols and sequences. The symbols were chosen, with the top row consisting of the first six letters of the English alphabet, to aid in understanding the algorithm presented in later sections.

Following this basic approach, one can further compute the variability of a position using the information-theoretic definition of *entropy*. For a discrete variable c_j we define $H(c_j) = -\sum_{i=1}^{|\mathcal{A}|} p(a_i@c_j) \log p(a_i@c_j)$, where $p(a_i@c_j)$ is the probability of the amino acid a_i appearing at position c_j, and $\mathcal{A}$ is the alphabet of naturally occurring amino acids, so $|\mathcal{A}| = 20$ in this application. The entropy $H(c_j)$ ranges from 0 to $\log |\mathcal{A}|$. Thus a completely conserved position c_j like Column 3 above, having no variability, has entropy $H(c_j) = 0$. The use of entropy-like measures to detect perfectly- and relatively-conserved positions is crucial to the multiple alignment task itself and these first-order features often elucidate important evolutionary relationships and structural and functional constraints, as when one speaks of the "invariant tryptophan" or the "conserved cysteines for the disulphide bridges in the Immunoglobulin constant domain". Such features and their frequency estimation also form the basis for several of the most widely-used template generation and template matching methods of sequence classification, notably including weight-matrix methods and the Profile methods of Gribskov *et. al.* [2].

2.2 Higher-Order Features

Although the single-position features are of great use in understanding protein sequence families and motifs, it is clear that higher-order features — representing associations among $k \geq 2$ residue positions — can capture important constraints and relationships missed by first-order analysis.

Referring again to Example 2.1, one notes that position 3 is very conserved; and, reading down from the top, one notes further that positions 2 and 4 (indicated by arrows) seem highly variable. However, whereas positions 2 and 4 display several mutations, they "mutate in lockstep" — their mutations covary perfectly. Given a statistically significant (manifested in *many* more sequences) example like this, one could extract a second-order feature, perhaps in the form of a "rule" relating positions 2 and 4, such as, "at (2,4) in protein family $\mathcal{F}$, B goes with D, U goes with V, and L goes with M". If statistically significant, such a relationship would strongly suggest a structural or functional constraint imposed on the two positions, whereby in the course of molecular evolution a mutation at one site must always have been compensated by an appropriate mutation in the other site. Of course, such covariances are not often so perfectly simple — typically, an amino acid may be paired with any of several other amino acids, with each pairing in the rule having an associated probability.

The use of such detectable correlations in structure prediction and classification, and the empirical evidence supporting such use, is discussed below.

Three main questions need to be addressed: (1) What kinds of evolutionarily conserved multi-residue structural or functional constraints might one expect to find by detecting correlations between columns in a multiple sequence alignment? (2) Have correlation-detection efforts in fact found important structural or functional constraints? (3) How much information do such discoveries provide towards predicting or determining a molecule's native tertiary structure?

2.3 What Do We Expect to Observe?

A protein family is the set of amino acid sequences that are believed to share a common global tertiary structure. The theory and observation of protein folding and evolution supports the general idea of evolution and conservation within a protein family: *Functional constraints* are conserved in *surface* residues; *structural constraints* are conserved in *core* residues; *mutational drift* dominates in *loop* residues. Functional constraints often involve other molecules — such as other proteins, nucleic acids, lipids, metals, O_2 or other small molecules. The kind of structural constraints expected to be conserved throughout evolution of a protein family are mainly those involving a few key residues that stabilize a conformation. Where electrostatic interactions are deemed important, one might expect to find a conservation of net charge across two or more sequence positions. When one of two electrostatically interacting residues carries a positive charge, its "partner" residue (presumably close in $3D$ structure even if distant in sequence) should be negatively charged, and *vice versa*. The situation is similar for packing constraints. One might expect to find pairs or small k-tuples of residues that display mutually compensatory mutations with respect to side-chain volume — when a "Large" mutates to a "Small", another "Small" must mutate into a "Large", to put it simplistically.

2.4 What Has been Observed?

Several studies performed in the last ten years offer evidence that functional and structural constraints can be detected from covariation analysis of aligned sequence positions [3,4], especially if the results of such mathematical analysis are supported by subsequent double-mutant cycle analysis [5,6]. As always, one's discoveries are constrained by one's methodology. For example, one study [4] of the myoglobin family of protein sequences found the degree of compensatory mutation to be low for the property of side-chain volume but high for electrical charge — close to the correlation level expected for perfect conservation of local charge. The authors speculate that because their analyses focused only on contact-neighbour pairs of residues, they were able to detect a very locally-acting constraint like charge conservation but not a more distributed constraint

like conservation of volume. It is reasonable to suggest that a search for still higher-order ($k > 2$) interactions might find some.

2.5 How Can These Observations Be Used?

Even if we cannot always figure out exactly which constraints are conserved among a set of correlated residues, it is worth asking whether the mere detection of such correlations can be used to predict 3D structural proximity. If so, then even a few such predictions might provide crucial information in global structure prediction, through the use of distance-geometry constraints [7] or empirical contact potentials [8]. There is much ongoing work in this area, but results thus far indicate that the prediction of pairwise inter-residue distances from correlation information can provide an improvement of 1.4 to 5.1 times over random contact predictions [9]. By itself, this kind of information is not nearly enough for *ab initio* prediction of global conformation, although it may be enough in many cases to distinguish between two or more alternative conformational models, or to provide additional constraints for energy minimization and molecular dynamics simulation[b].

Clearly, to miss or ignore higher-order interactions is to impair structure prediction capabilities. It can also hamper the simpler task of mere sequence classification. This fact is discussed at greater length in other sources [10,11], where it is also shown that after the significant inter-residue correlations have been discovered, they can be built into representations that make for fast and sensitive classifiers based upon, for example, the methodologies of graphical models [11].

3 Mathematical Formulation and Complexity of the Problem

Having motivated the problem, we now state it formally. Assume that we are given a database of M *objects* $\bar{s}_i$ ("s" for **s**equence), each of which is characterized by particular *values* $a_{ij} \in \mathcal{A}$ for each of N discrete-valued *variables* c_j ("c" for **c**olumn). A particular value for a particular variable is an *attribute* and denoted $a_l@c_j$. We further assume that there is some "true" underlying probability distribution $p()$ which, for all orders $k = 1, 2, \ldots, N$ specifies the probabilities for each possible k-tuple of attributes. For example, for $k = 1$, we have $p(c_j) : \mathcal{A} \longrightarrow [0, 1]$.

[b] We remind the reader that the detection of correlated mutations does not always imply spatial proximity. Putting aside the merely spurious correlations, those deriving purely from phylogenetic branching, and the artifacts of poor estimation methodology, there will likely remain many instances of coordinated mutation that reflect non-local *functional* synergism.

The problem, then, is: Given a real number $\theta \in [0,1]$, return a list of all k-ary joint attribute patterns $\alpha = (a_1^\alpha @ c_1^\alpha, a_2^\alpha @ c_2^\alpha, \ldots, a_k^\alpha @ c_k^\alpha)$ such that

$$P(Observed(a_1^\alpha @ c_1^\alpha, a_2^\alpha @ c_2^\alpha, \ldots, a_k^\alpha @ c_k^\alpha) | Independent(c_1^\alpha, c_2^\alpha, \ldots, c_k^\alpha), \mathcal{M}) < \theta,$$

for some Observed number of occurrences of the pattern

$$\alpha = (a_1^\alpha @ c_1^\alpha, a_2^\alpha @ c_2^\alpha, \ldots, a_k^\alpha @ c_k^\alpha)$$

and some model $\mathcal{M}$ which underlies one's sampling and hypothesis testing method. That is, we want to find all combinations of attributes which, under our own sampling and counting mechanism, are observed to occur significantly more often than one would expect given only the marginal probabilities of the individual attributes. That is, we discover the "suspicious coincidences". Inherent in this correlation-detection problem is the problem of estimating or approximating the distribution $p()$, or at least parts of it. Other, closely-related formulations of the problem are possible. For example, several previous analyses of protein sequence and structure focused on a pairwise mutual information value [6,12]. Our formulation, in which the search for k-tuples of residues showing high mutual-information is less direct, reflects the logical structure and efficiency needs of our algorithm, described in Section 5.

We note that a more general database-theoretic formulation of this problem (e.g., with different alphabets for each column) is applicable to many problems across many disciplines, and indeed "association mining" is one of the central problems in the emerging field of knowledge discovery and data mining.

3.1 Complexity of the Problem

To test all possible k-tuples of attributes for "suspiciousness" requires at least $O(\left(\begin{array}{c} N \\ k \end{array} \right) \cdot M)$ computational steps. To do this for *all* $2 \leq k \leq N$ is an $O(2^N)$ computation, because one has to enumerate the powerset of a set of N columns. This powerset expansion of all joint probability terms, known variously as the full Gibbs model [13] or the Bahadur-Lazarsfeld expansion, is at the heart of information complexity [14], computational complexity, sample complexity, and the bias-variance dilemma in statistical modelling and machine learning [15]. This combinatorial barrier makes any direct, exhaustive approach to association mining infeasible.

4 Previous Methods

Calculation and Extension of Pairwise Correlations: In the last few years, several groups have recognized and addressed the importance of at least *2nd-order features*, in proteins [6] and RNA [16]. One might make the heuristic guess that a set of $k > 2$ columns characterized by high pairwise correlations also display significant higher-order ($k > 2$) correlations. This corresponds, more or less, to considering the transitive closure of the "Correlated With" binary relation, and there are many possible ways to do this. Like any heuristic, it can lead to trouble; both false positives and false negatives are possible.

Hidden Markov Models and Grammar Induction: Several groups have reported significant success in modelling protein sequence families with Hidden Markov Models (HMMs) [17]. For some of the same reasons why HMMs are very good at aligning the sequences in the first place, using local sequential correlations, these methods are less useful for finding the important sequence-distant correlations in data that has already been partially or completely aligned. The phenomenon responsible for this dilemma, termed "diffusion", is examined in some detail in recent work by Bengio and Frasconi [18]. Essentially, a first-order HMM, by definition, assumes independence among sequence columns, given a hidden state sequence. Multiple alternative state sequences can in principle be used to capture longer-range interactions, but the number of these grows exponentially with the number of k-tuples of correlated columns.

Artificial Neural Networks: Many neural network architectures and learning algorithms are able to capture higher-order relationships among their inputs [19]. MacKay's "density networks" use Bayesian learning to build componential latent variable models [20], and have been applied to protein sequence modelling. However, the combinatorial explosion of priors and hyper-priors that need to be set may severely limit this method's application to real-world dataset sizes.

5 Coincidence Detection: A Novel Data-Mining Method

Like HMMs and Gibbs models [13], the MacKay approach would benefit from a fast preprocessing stage that could find candidate subsets of correlated observable variables and allow one to pre-set some of the priors (or HMM state transitions, or Gibbs potentials) accordingly.

Clearly, several well-studied and somewhat effective methodologies exist for modelling protein sequence families. In each case, the mathematical machinery is in place to handle and detect very local and low-order statistical structure in the data; but the difficulties with computational complexity and

statistical estimation arise in the attempt to account comprehensively for all possible non-local and higher-order interactions between residues, i.e., columns, in the aligned sequence data. Our Coincidence Detection method is designed to get around the central obstacle to higher-order feature discovery: we do not want to specify or limit, *a priori*, the number of possible k-tuples of correlated columns, the width k of any of them, or the degrees of correlation involved; and yet we do not want to explicitly represent and process latent variables or parameters for the exponentially-many possible k-tuples. Therefore, the method must be able to recognize the occurrence of patterns that provide evidence for k-ary correlations *whenever* they arise, and to analyze such patterns *only* when they arise — rather than set up data structures for higher-order patterns that may not ever appear.

5.1 Outline of Procedure

More detail on the several variants of our method may be found elsewhere[10,21]. We summarize here the four basic components:

Representation: The occurrences of an attribute in a set of data items are summarized in a binary *incidence vector*. An incidence vector of length r has a 1 in the ith position iff the corresponding attribute, e.g., $B@2$, occurs in the ith data item in the set.

Sampling: Take r sequence records at a time, from a uniform distribution.

Binning, and Coincidence Detection: For each sampling iteration, throw the attributes into bins, according to their incidence vectors. These vectors act like r-bit addresses into a very sparse subset of 2^r address space. All the attributes in one bin constitute a *coincidence set*, or *cset*. Record the cset and the number $h : 0 \leq h \leq r$ of occurrences. (Note that h is the number of 1s in the incidence vector "address".)

Hypothesis Tests: After all the sampling and binning, compare the observed number of occurrences of each cset with the number expected under the null hypothesis of statistically independent columns. The basis for the "expected" part of the hypothesis test is the probability of a match, or coincidence, of size h in a given r-sample for a cset $\alpha = (a_1^\alpha @ c_1^\alpha, \ldots, a_k^\alpha @ c_k^\alpha)$:

$$f_{match}(\alpha, h, r) = \frac{r!}{h!(r-h)!} p(a_{i_1} @ c_{j_1}, \ldots, a_{i_k} @ c_{j_k})^h p(\tilde{a}_{i_1} @ c_{j_1}, \ldots, \tilde{a}_{i_k} @ c_{j_k})^{r-h}),$$

where the joint probability terms reflect the independence assumption:

$$p(a_{i_1} @ c_{j_1}, \ldots, a_{i_k} @ c_{j_k}) \quad = \quad \prod_{l=1}^{k} p(a_{i_l} @ c_{j_l})$$

$$p(\tilde{a}_{i_1}@c_{j_1}, \ldots, \tilde{a}_{i_k}@c_{j_k}) \quad = \quad \prod_{l=1}^{k}(1 - p(a_{i_l}@c_{j_l})),$$

and $\tilde{a}$ means the appearance of any symbol other than a. Finally, a Chernoff-Hoeffding bound [22] is used to implement the hypothesis testing, and so our procedure produces an estimate of the probability p^* of seeing n_{obs} occurrences of a cset α when the marginal probabilities of the components, and the independence assumption, predict only n_{exp} occurrences. Finally, the list of observed csets is sorted by their p^* values, and the procedure returns a small list of only the most "interesting" or "surprising" higher-order features, e.g., those which have $p^* < 0.001$.

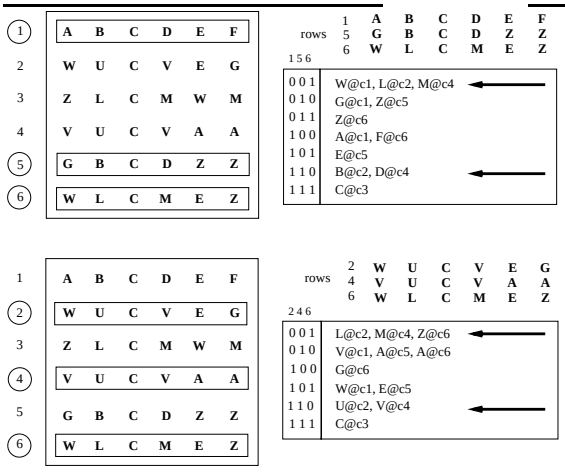

Figure 1: Operation of the Coincidence Detection Method: Two iterations of the r-sampling (for $r = 3$) on the toy dataset are depicted, top to bottom. For each iteration, the left-hand box represents the dataset, with outlined entries representing the sampled rows. The right-hand-box represents the set of bins into which the attributes collide. For example, in the first iteration, $B@2$ and $D@4$ both occur in the first and second of the three sampled rows, so they each have incidence vector 110 and collide in the bin labelled by that binary address. Bins containing only a single attribute are ignored; and "empty" bins are never created at all. All bins are cleared and removed after each iteration, but collisions (coincidences) are recorded in the *Csets* global data structure.

This sampling/binning trick allows any number k of sequence-distant attributes to collide together, and it builds small-scale ($r \ll M$) coincidences into evidence for large-scale correlation. With use of appropriate data structures for storage and updating of the csets, the time, space, and sample complex-

ity of our method are kept quite manageable. While proving tight, meaningful complexity bounds for this probabilistic algorithm is impossible without severe distributional assumptions, we observed linear or sub-linear growth in runtime and memory use as a function of M, N, and desired confidence levels, in tests on several specially-constructed datasets, as well as on real protein families [10].

Note that in our problem formulation, no limiting assumptions are made about how many such cliques of correlated variables there are, how *wide* they may be (the maximal size of k), nor on the absolute or relative degrees of correlation. Nor do we make other assumptions about the statistical structure of the data distribution.

5.2 Discovering Sites of Functional and Immunological Importance in HIV V3 Loop

While, to date, most studies of compensatory mutation focus on highly-conserved "core"-type regions of protein structures, Korber *et al.* analyzed the highly-variable V3 loop of the HIV-1 envelope protein. The researchers performed robust bootstrapped estimates of the pairwise mutual information for all column-pairs from a set of 31 columns representing V3 residues. They found a small set of pairs that showed considerable and statistically-significant mutual information, and their analysis of the particular attributes (amino acids) suggested a pattern of highly likely compensatory mutations. Subsequent mutational analysis experiments in the laboratory indicated functional linkage between some of the pairs of sites with high mutual information. Because the V3 region is known to be both functionally and immunologically important, it is suggested that such analyses might be important in the search for HIV/AIDS vaccine design.

We applied our method to a newer, larger version of this same Los Alamos HIV Sequence Database, in order to rediscover the significant site-pairs found by the Los Alamos group and to try to discover additional highly correlated k-tuples. As reported in detail in our first author's Ph.D. thesis [10], we indeed discovered, using the same overall amount of computation, the originally-reported pairs of correlated residues, with two out of seven exceptions[c]. The Coincidence Detection method also discovered additional significant pairs, as well as several 3-tuples, a 4-tuple and a 5-tuple. It is easy to show that a direct search for all significant 4-tuples alone requires more computation than we performed with our method.

[c] This discrepancy might result from algorithmic differences; however, because one of the Los Alamos group's site-pairs did not show significance on our own controlled application of their algorithm, we believe that some differences stem from our use of the newer, larger, and perhaps otherwise different database.

6 Conclusions and Future Work

A unique strength of the Coincidence Detection method is that it can discover, e.g., 41-ary correlations in the same time it takes to find pairwise correlations of equal statistical significance. It also requires no assumptions about the number, size or sequential separation of the hidden higher-order features in the data. However, the relative advantage of the method is greatest on datasets in which there exist very strong and significant inter-attribute correlations, of whatever widths $2 \leq k \leq N$. The method is at a disadvantage when applied to data with no correlations, or very weak ones; in such cases, the number of sampling iterations needed before correlations are detected, or before they can be ruled out, may be prohibitive. Another association mining method [23] has gained recent popularity, and a note of comparison is in order: While their method, with its emphasis on "support" tends to find the *most frequent* associations (e.g., those (A, B) with high values of $p(A)$, $p(B)$, and $p(A, B)$), our method finds the *most surprising* associations (e.g., those with $p(A, B) \gg p(A)p(B)$). Different applications may require different criteria.

We are currently applying our methods to additional protein families, to RNA sequence data, and to non-biological applications. We are also developing hardware implementations, for the binary attribute representations and bin addressing scheme lend themselves to very fast and cheap circuit designs, and many aspects of the algorithm are parallelizable.

Acknowledgments

The authors wish to thank Derek Robinson for his inspiration and effort in developing earlier versions of our association mining methods, and for help with figures. We thank Ed Willis for proofreading and Geoffrey Hinton for helpful discussions. We also thank the reviewers for their comments and suggestions. Any mistakes that remain are the responsibility of the first author.

References

1. D.H. Wolpert and D. R. Wolf. Technical Report LA-UR-93-833, Los Alamos National Laboratory, 1993.
2. M. Gribskov, M. Homyak, J. Edenfield, and D. Eisenberg. *Comput. Appl. Biosci.*, (4):61–66, 1988.
3. D. Altschuh, T. Vernet, P. Berti, D. Moras, and K. Nagai. *Prot. Eng.*, 2:193–199, 1988.
4. E. Neher. *PNAS*, 91:98–102, 1994.

5. A. Horovitz, E.S. Bochkareva, O. Yifrach, and A.S. Girshovich. *J. Mol. Biol.*, 238:133–138, 1994.
6. B.T.M. Korber, R.M. Farber, D.H. Wolpert, and A.S. Lapedes. *Proc. Nat. Acad. Sci.*, 90, 1993.
7. P.R. Sibbald. *J. Theor. Biol.*, 173:361–375, 1995.
8. M.J. Sippl. *J. Mol. Biol.*, 213:859–883, 1990.
9. I.N. Shindyalov, N.A. Kolchanov, and C. Sander. *Prot. Eng.*, 7:349–358, 1994.
10. E. W. Steeg. Ph. D. thesis, Department of Computer Science, University of Toronto, 1997.
11. T. M. Klingler and D. L. Brutlag. *Prot. Sci.*, 3:1847–1857, 1994.
12. A. S. Lapedes, E. W. Steeg, and R. M. Farber. *Machine Learning*, 21:103–124, 1995.
13. J. W. Miller. Ph. D. thesis, California Institute of Technology, 1993.
14. H. H. Ku and S. Kullback. *IEEE Transactions on Information Theory*, 4:444–447.
15. S. Geman, E. Bienenstock, and R. Doursat. *Neural Computation*, 4:1–58, 1992.
16. R.R. Gutell, A. Power, G.Z. Hertz, E.J. Putz, and G.D. Stormo. *Nucl. Acids Res.*, 20:5785–5795, 1992.
17. A. Krogh, M. Brown, I.S. Mian, K. Sjolander, and D. Haussler. *J. Mol. Biol.*, 235:1501–1531.
18. Y. Bengio and P. Frasconi. Technical Report, 96. Dept. I.R.O., Universite de Montreal, Canada / Dipartimento di Sistemi e Informatica, Universita di Firenze, Italy.
19. S. Becker and M. Plumbley. *International Journal of Applied Intelligence*, 6(3), 1996.
20. David J.C. MacKay. In *Proceedings of Workshop on Neutron Scattering Data Analysis*, 1994.
21. E. W. Steeg and D. A. Robinson. Coincidence Detection: A Fast Method for Discovering Associations in High-Dimensional Data. *In preparation.*
22. W. Hoeffding. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
23. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo. In *Advances in Knowledge Discovery and Data Mining*, pages 307–328. American Association for Artificial Intelligence, 1996.