We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

4

Vector Space Information Retrieval Techniques for Bioinformatics Data Mining

Eric Sakk and Iyanuoluwa E. Odebode
Department of Computer Science, Morgan State University, Baltimore, MD USA

1. Introduction

Information retrieval (IR) can be defined as the set of processes involved in querying a collection of objects in order to extract relevant data and information Dominich (2010); Grossman & Frieder (2004). Within this paradigm, various models ranging from deterministic to probabilistic have been applied. The goal of this chapter is to invoke a mathematical structure on bioinformatics database objects that facilitates the use of vector space techniques typically encountered in text mining and information retrieval systems Berry & Browne (2005); Langville & Meyer (2006).
Several choices and approaches exist for encoding bioinformatics data such that database objects are transformed and embedded in a linear vector space Baldi & Brunak (1998). Hence, part of the key to developing such an approach lies in invoking an algebraic structure that accurately reflects relevant features within a given database. Some attention must therefore be devoted to the numerical encoding of bioinformatics objects such that relevant biological and chemical characteristics are preserved. Furthermore, the structure must also prove useful for operations typical of data mining such as clustering, knowledge discovery and pattern classification. Under these circumstances, the vector space approach affords us the latitude to explore techniques analogous to those applied in text information retrieval Elden (2004); Feldman & Sanger (2007); Grossman & Frieder (2004).
While the methods presented in this chapter are quite general and readily applicable to various categories of bioinformatics data such as text, sequence, or structural objects, we focus this work on amino acid sequence data. Specifically, we apply the BLOCKS protein sequence database Henikoff et al. (2000); Pietrokovski et al. (1996) as the template for testing the applied techniques. It is demonstrated that the vector space approach is consistent with pattern search and classification methodologies commonly applied within the bioinformatics literature Baldi & Brunak (1998); Durbin et al. (2004); Wang et al. (2005). In addition, various subspace decomposition approaches are presented and applied to the pattern search and pattern classification problems.
To summarize, the main contribution of this work is directed towards bioinformatics data mining. We demonstrate that information measures derived from the vector space approach are consistent with and, in many cases, reduce to those typically applied in the bioinformatics literature. In addition, we apply the BLOCKS database in order to demonstrate database search and information retrieval techniques such as

- Pattern Classification

- Compositional Inferences from the Vector Space Models
- Clustering
- Knowledge Discovery

The chapter is outlined in Figure 1 as follows. Section 2 provides basic background regarding information retrieval and bioinformatics techniques applied in this work. Given this foundation, Section 3 presents various approaches to encoding bioinformatics sequence data. Section 4 then introduces the subspace decomposition methodology for the vector space approach. Finally, Section 5 develops the approach in the context of various applications listed in Figure 1.

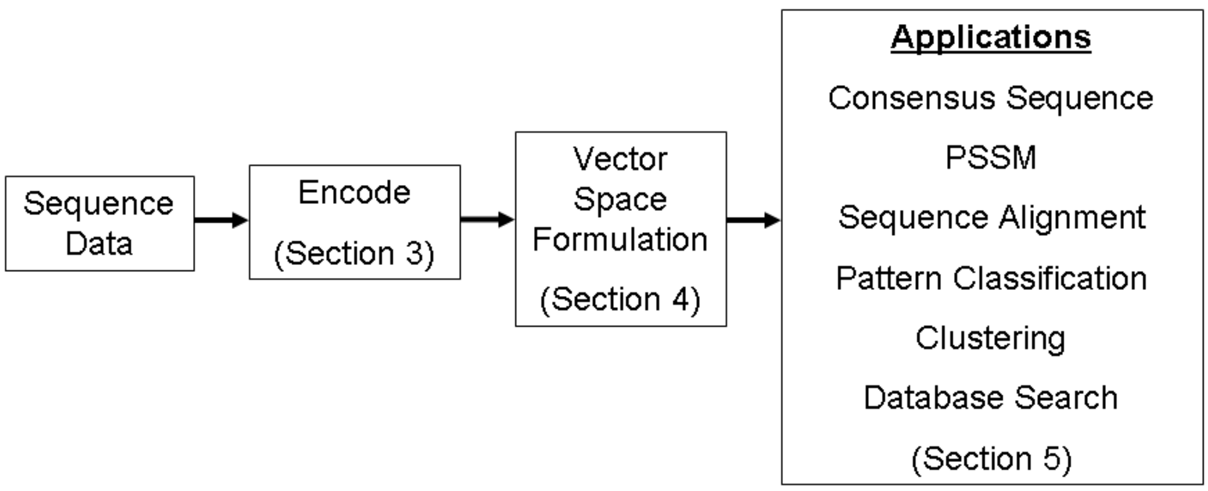


Fig. 1. Flowchart for the chapter

2. Overview and notation

Part of the goal of this chapter is to phrase the bioinformatics database mining problem in terms of vector space IR (information retrieval) techniques; hence, this section is devoted toward reviewing terms and concepts relevant to this work. In addition, definitions, mathematical notation and conventions for elements such as vectors and matrices are introduced.

2.1 Vector space approach to information retrieval

Information retrieval can be thought of as a collection of techniques designed to search through a set of objects (e.g. contained within a database, on the internet, etc) in order to extract information that is relevant to the query. Such techniques are applicable, for example, to the design of search engines, as well as performing data mining, text mining, and text categorization Berry & Browne (2005); Elden (2004); Feldman & Sanger (2007); Hand et al. (2001); Langville & Meyer (2006); Weiss et al. (2005). One specific category of this field that has proven useful for the design of search engines and constructing vector space models for text retrieval is known as Latent Semantic Indexing (LSI) Berry et al. (1999; 1995); Deerwester et al. (1990); Salton & Buckley (1990). Using the LSI approach, textual data is transformed (or '***encoded***') into numeric vectors. Matrix analysis techniques Golub & Van Loan (1989) are then applied in order to quantify semantic relationships within the textual data.

	Document 1	Document 2	Document 3	Document 4
Term 1	1	0	1	0
Term 2	0	1	1	0
Term 3	1	1	0	1
Term 4	1	1	0	1
Term 5	1	0	1	0

Table 1. Example of a 5×4 term-document matrix.

Consider categorizing a set of m documents based upon the presence or absence of a list of n selected terms. Under these circumstances, an $n \times m$ term-document matrix can be constructed where each entry in the matrix might reflect the weighted frequency of occurrence of each term of interest. Table 1 provides an example; in this case, a matrix column vector defines the frequency of occurrence of each term in a given document. Such a construction immediately facilitates the application of matrix analysis for the sake of quantifying the degree of similarity between a query vector and the document vectors contained within the term-document matrix.

Given an $n \times m$ term-document matrix A, consider an $n \times 1$ vector q constructed from a query document whose components reflect the presence or absence of entries in the same list of n terms used to construct the matrix A. The question then naturally arises how one might quantify the similarity between the query vector q and the term-document matrix A. Defining such a similarity measure would immediately lead to a scoring scheme that can be used to order results from most relevant to least relevant (ie induce a '***relevance score***').

Given the vector space approach, a natural measure of similarity arises from the inner product. Assuming an ℓ_2-norm, if both q and the columns of A have been normalized to unit magnitude, then the inner product between q and the j^{th} column vector of A becomes

$$q^T a_j = ||q|| \, ||a_j|| \cos\theta_j = \cos\theta_j \tag{1}$$

(where the 'T' superscript denotes the transpose). Since all components of q and A are non-negative, all inner products will evaluate to a value such that $0 \leq \cos\theta_j \leq 1$. Similar queries approach a value of one indicating a small angle between the query and column vector, dissimilar queries approach a value of zero indicating orthogonality. This specific measure is called the '***cosine similarity***' and is abbreviated as

$$\cos\theta = q^T A \tag{2}$$

where $\cos\theta$ represents a row vector whose components quantify the ***relevance*** between the query and each column vector of A.

Given the vector space approach, LSI (latent semantic indexing) goes a step further in order to infer semantic dependencies that are not immediately obvious from the raw data contained in the term-document matrix. In terms of linear algebra, the LSI methodology translates into characterizing the column space of A based upon some preferred matrix decomposition. A tool commonly applied in this arena is the Singular Value Decomposition (SVD) Golub & Van Loan (1989) where the term-document matrix is factored as follows:

$$A = U\Sigma V^T \tag{3}$$

where U is an $n \times n$ orthogonal matrix (i.e. $U^{-1} = U^T$), V is an $m \times m$ orthogonal matrix (i.e. $V^{-1} = V^T$). Furthermore, Σ is an $n \times m$ diagonal matrix of singular values such that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0 \tag{4}$$

where $r = rank(A)$ and $\sigma_i \equiv \Sigma_{ii}$. It turns out that the first r columns of U define an orthonormal basis for the column space of the matrix A. This basis defines the underlying character of the document vectors and can be used to infer linear dependencies between them. Furthermore, it is possible to expand the matrix A in terms of the SVD:

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^T \tag{5}$$

where u_j and v_j represent the j^{th} columns of U and V. This expansion weights each product $u_j v_j^T$ by the associated singular value σ_j. Hence, if there is a substantial decreasing trend in the singular values such that $\sigma_j/\sigma_1 << 1$ for all $j > L$, one is then led to truncate the above series in order to focus on the first L terms that are responsible for a non-negligible contribution to the expansion. This truncation is called the ***low rank approximation*** to A:

$$A \approx \sum_{j=1}^{L} \sigma_j u_j v_j^T \tag{6}$$

The low rank approximation describes, among other aspects, the degree to which each basis vector in U contributes to the matrix A. Furthermore, the subspace defined by the first L columns of U is useful for inferring linear dependencies in the original document space.

2.2 Bioinformatics

Given this abbreviated overview of vector space approaches to information retrieval, we now put it in the context of bioinformatics research. In particular, the SVD has been applied in many contexts as it can be thought of as a deterministic version of principal component analysis Wall et al. (2003). One specific area of honorable mention is pioneering work dealing with the analysis of microarray data Alter et al. (2000a;b); Kuruvilla et al. (2004).

With regard to information retrieval and LSI in bioinformatics Done (2009); Khatri et al. (2005); Klie et al. (2008), research in this area devoted to phylogenetics and multiple sequence alignment Couto et al. (2007); Stuart & Berry (2004); Stuart, Moffett & Baker (2002) has been reported. Much of this work can be traced back to initial foundations where the encoding of protein sequences has been performed using the frequency of occurrence of amino acid k-grams Stuart, Moffett & Leader (2002). Using the k-gram approach, column vectors in the data matrix (i.e. what was previously referred to as the 'term-document matrix') are encoded amino acid sequences and their components are the frequency of occurrence of each possible k-gram within each sequence. For example, if amino acids are taken $k = 3$ at a time, then there exist $n = 20^k = 8000$ possible 3-grams. Assuming there are m amino acid sequences, the associated data matrix will be $n \times m = 8000 \times m$. For each amino acid sequence, a sliding, overlapping window of length k is used to count the frequency of occurrence of each k-gram and entered into the data matrix A.

The goal of this chapter is to build upon the IR and bioinformatics foundation in order to introduce novel perspectives on operations and computations commonly encountered in bioinformatics such as the consensus sequence, position specific scoring matrices (PSSM), database searches, pattern classification, clustering and multiple alignments. In doing so, it is our intent that the reader's view of these tools will be expanded toward novel applications beyond those presented here.

3. Sequence encoding

Many choices exist for the encoding of and weighting of entries within the term-document matrix; in addition, there exist a wide range of possibilities for matrix decompositions as well as the construction of similarity and scoring measures Elden (2004); Feldman & Sanger (2007); Hand et al. (2001); Weiss et al. (2005). The goal of this chapter is not to expand on the set of choices for the sake of text retrieval and generic data mining; instead, we must focus on techniques and approaches that are relevant to bioinformatics. Specifically, our attention in this section is devoted toward developing and presenting novel encoding schemes that preserve relevant biological and chemical properties of genomic data.

An assortment of methods have been proposed and studied for converting a protein from its amino acid sequence space into a numerical vector Bacardit et al. (2009); Baldi & Brunak (1998); Bordo & Argos (1991); Stuart, Moffett & Leader (2002). Scalar techniques generally assign a real number that relates an amino acid to some physically measurable property (e.g. - volume, charge, hydrophobicity) Andorf et al. (2002); Eisenberg et al. (1984); Kyte & Doolittle (1982); Wimley & White (1996). On the other hand, orthogonal or 'standard' vector encoding techniques Baldi & Brunak (1998) embed each amino acid into a k dimensional vector space where k is the number of symbols. For example, if $k = 20$ (as it would be for the complete amino acid alphabet), the j^{th} amino acid where $1 \leq j \leq 20$ is represented by a 20 dimensional vector that is assigned a one at the j^{th} position and zero in every other position. In general, standard encoding transforms a sequence of length L into an $n = Lk$ dimensional vector. As an example consider the DNA alphabet $\mathcal{A} = \{A, G, C, T\}$. In this case $k = 4$ and standard encoding transforms the alphabet symbols as

$$A = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} . \; G = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} . \; C = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} . \; T = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} . \tag{7}$$

Therefore, for an example sequence $s = AT$ with $L = 2$, this encoding method yields the following vector of dimension $n = Lk = 8$:

$$x^T = \begin{pmatrix} 1 \; 0 \; 0 \; 0 \; 0 \; 0 \; 0 \; 1 \end{pmatrix} .$$

Observe that, for typical values of L, assuming a data set of m sequences, standard encoding leads to an $n \times m$ data matrix that is sparse.

In bioinformatics, given the limitations on biological measurement, the number of experimental observations tends to be limited and values of m are often small with respect to n. Under these conditions, it is often the case that vector encoding methodologies lead to sparse data matrices (as is the case for text retrieval applications) in high dimensional vector spaces. Observe, for example, that the k-gram method reviewed in Section 2.2 fits this description.

We can expand upon the standard encoding approach by categorizing the standard amino acid alphabet into families that take into account physical and chemical characteristics derived from the literature Andorf et al. (2002); Baldi & Brunak (1998). In addition, entries within the data matrix can be weighted based upon their hydrophobicity Eisenberg et al. (1984); Kyte & Doolittle (1982). Table 2 introduces alphabet symbols used to group amino acids according to hydrophobicity, charge and volume. Tables 3-5 show examples of various encoding schemes that we apply for this analysis.

Hydrophobicity	R=hydrophobic, H=hydrophilic
Charge	P=positive, N=negative, U=uncharged
Volume	S=smal, M=medium, ML=medium-large, L=medium

Table 2. Encoding symbols applied in Tables 3-5

R	1	A, I, L, M, F, P, W, V, D, E
H	3	R, H, K, N, C, Q, G, S, T, Y

Table 3. Hydrophobic/Hydrophilic Encoding

RU	1	A, I, L, M, F, P, W, V
HN	2	D, E
HP	3	R, H, K
HU	4	N, C, Q, G, S, T, Y

Table 4. Charged Hydrophobic/Hydrophilic Encoding

RUS	1	A
RUM	2	F
RUML	3	I, L, M, V
RUL	4	F, W
HPML	5	R, H, K
HNM	6	D
HNML	7	E
HUS	8	G, S
HUM	9	N, C, V
HUML	10	Q
HUL	11	Y

Table 5. Volume/Charged Hydrophobic/Hydrophilic Encoding

4. Subspace decompositions for pattern classification

LSI techniques necessarily require the application of matrix decompositions such as the SVD to infer column vector dependencies in the data matrix. Decompositions of this kind can lead to the construction of subspaces that can mathematically categorize subsets of sequences into families. Furthermore, since these families define specific classes of data, they can be used as training data in order to perform database searches and pattern classification. The application of linear subspaces for the sake of pattern classification Oja (1983) consists of applying orthogonal projection operators based upon the training classes (an orthogonal projection operator P obeys $P = P^T$ and $P^2 = P$).

4.1 Orthogonal projections

To begin, let us assume there are training sequences of known classification that can be categorized into M distinct classes and that the i^{th} class contains m_i encoded vectors of dimension n. For each class, an $n \times m_i$ matrix A_i can be constructed (assuming the training vectors are column vectors). To characterize the linear subspace generated by each class, we can apply the singular value decomposition (SVD) Golub & Van Loan (1989). In addition to providing us with an orthonormal basis for each class, we can also glean some information about the influence of the singular values and singular vectors from the rank approximants. Class data matrices are therefore decomposed as

$$A_i = U_i \Sigma_i V_i^T \tag{8}$$

where U_i is $n \times n$ orthogonal matrix, Σ_i is $n \times m_i$ whose diagonal contains the singular values and V_i is an $m_i \times m_i$ orthogonal matrix. Assume the rank of each data matrix A_i is r_i and let Q_i denote the $n \times r_i$ matrix formed from first r_i columns of U_i. Given the properties of the SVD, the columns of Q_i define an orthonormal basis for the column space of A_i. Hence, an orthogonal projection operator for the i^{th} class is established by computing

$$P_i = Q_i Q_i^T. \tag{9}$$

(given that the SVD induces $U_i^T = U_i^{-1}$, it is straightforward to check that $P_i^2 = P_i$ and $P_i^T = P_i$).
Consider an $n \times 1$ query vector x whose classification is unknown. The class membership of x can be ascertained by identifying the class yielding the maximum projection norm:

$$C(x) \equiv \underset{i=1,\cdots,M}{\arg\max} \, ||P_i x||. \tag{10}$$

One computational convenience of constructing the orthonormal bases Q_i is that it is not necessary to compute the projections when making this decision. Given any Q with orthonormal columns and orthogonal projection $P = QQ^T$ such that $P^2 = P$ and $P = P^T$, observe that

$$\begin{aligned} ||Px||^2 &= x^T P^T P x = x^T P^2 x \\ &= x^T P x = x^T Q Q^T x \\ &= ||Q^T x||^2 = ||x^T Q||^2. \end{aligned} \tag{11}$$

Under these circumstances, to decide class membership, Equation (10) reduces to

$$C(x) \equiv \underset{i=1,\cdots,M}{\arg\max} \, ||x^T Q_i||. \tag{12}$$

Furthermore, the values $||x^T Q_i||$ immediately yield relevance scores and confidence measures for each class.

4.2 Characterization of the orthogonal complement

It is important to note that the union of all the class subspaces *need not* be equal to the n dimensional vector space from which all data vectors are derived. To perform a complete orthogonal decomposition of the n dimensional vector space in terms of the data, we first define the matrix

$$A \equiv [A_1 \ \cdots \ A_M]. \tag{13}$$

The goal then is to characterize the null space $\mathcal{N}(A^T)$, the subspace which is orthogonal to the column space of A. Assuming the rank of A is r_A, computing the SVD

$$A = U_A \Sigma_A V_A^T \tag{14}$$

and forming the matrix Q_A from the the first r_A columns of U_A yields an orthogonal decomposition of the subspace generated by ***all*** class vectors. Hence, a projection operator for this subspace is constructed as

$$P_A = Q_A Q_A^T. \tag{15}$$

In addition, a projection for the orthogonal complement $\mathcal{N}(Q_A^T)$ of A is then easily formed via

$$P_{A^\perp} = I_n - P_A \tag{16}$$

where I_n is the $n \times n$ identity matrix. A complete orthogonal decomposition Lay (2005) of a vector $x \in \mathcal{R}^n$ can then be determined from

$$x = P_A x + P_{A^\perp} x. \tag{17}$$

4.3 Information retrieval

Before attempting to decide the class membership of a vector $x \in \mathcal{R}^n$ based upon Equation (12), it is sensible to characterize the portion of the vector that contributes to the class subspace defined by Q_A. Given Equation (17), this is most easily done by comparing $||P_A x||$ with $||P_{A^\perp} x||$ as

$$\tan(\phi) = \frac{||P_{A^\perp} x||}{||P_A x||} \tag{18}$$

where ϕ is the angle between x and the subspace defined by Q_A. Ideally, if the class subspaces have been completely characterized, $\tan(\phi)$ should be small. Conversely, larger values of $\tan(\phi)$ would indicate that x is a member of a class subspace that has not yet been defined. Under these circumstances, the orthogonal complement would have to be further characterized and partitioned in order to define more classes beyond the known M existing classes.

It is also possible to phrase the tangent measure as a scalar version of the more familiar cosine similarity defined above in Equation (2). If $||x|| = 1$, the cosine similarity measure takes on a convenient form

$$\cos(\phi) = ||x^T Q_A||. \tag{19}$$

To see why, consider the inner product

$$x^T (P_A x) = x \cdot P_A x = ||x|| \, ||P_A x|| \cos(\phi). \tag{20}$$

If $||x|| = 1$, then

$$\cos(\phi) = \frac{x^T P_A x}{||P_A x||}. \tag{21}$$

However, since P_A is an orthogonal projection

$$||P_A x|| = \sqrt{x^T P_A^T P_A x} = \sqrt{x^T P_A x} \tag{22}$$

and Equation (21) can therefore be rewritten as

$$\cos(\phi) = \sqrt{x^T P_A x}. \tag{23}$$

On the other hand, by applying Equation (11) to Equation (22), it follows that

$$||x^T Q_A|| = \sqrt{x^T P_A x} \tag{24}$$

as well; hence, the equality of Equations (23) and (24) establishes Equation (19).
Equation (19) should also be clear from the geometric fact that

$$\cos(\phi) = \frac{||P_A x||}{||x||}. \tag{25}$$

Assuming $||x|| = 1$, Equation (19) then easily follows by applying Equation (11) to Equation (25). Equations (23) and (24) are presented in order to offer additional insight by relating the inner product to the projection operator.
Of central focus in the next section will be to apply the above projection framework to information retrieval in bioinformatics. Since the classification problem will be of significance, we note that, given the identity in Equation (19), Equation (12) can be rephrased in term of the cosine similarity measure

$$C(x) \equiv \arg\max_{i=1,\cdots,M} \cos(\phi_i) \tag{26}$$

where

$$\cos(\phi_i) \equiv ||x^T Q_i||. \tag{27}$$

In addition, this measure of class membership becomes more reliable if the contribution of x to the orthogonal complement of the data set is small. For instance, when ϕ is small, $cos(\phi)$ in Equation (19) approaches unity. Therefore, $\cos(\phi)$ can be applied as a measure of data set reliability while $\cos(\phi_i)$ can be used to produce relevance scores for $i = 1, \cdots, M$. These conclusions are summarized in Table 6.

Similarity Measure	Purpose	Reference
$\cos(\phi)$	Data Set Reliability	Equation (19)
$\cos(\phi_i)$	Relevance Score	Equations (26) - (27)

Table 6. Reliability and relevance measures to be applied in Section 5
.

5. Applications

In bioinformatics, families with similar biological function are often formed from sets of protein or nucleic acid sequences. For example, databases such as Pfam Finn et al. (2010), PROSITE Sigrist et al. (2010) and BLOCKS Pietrokovski et al. (1996) categorize sequence domains of similar function into distinct classes. Given the encodings discussed in Section 3, we seek to demonstrate how Equations (19) and (26) can applied in order to perform sequence modeling, pattern classification and database search computations typically encountered in bioinformatics Baxevanis & Ouellette (2005); Durbin et al. (2004); Mount (2004).

5.1 Consensus sequence

A set of m sequences of length L having some related function (e.g. DNA promoter sites for a common sigma factor) is often represented in the form of an $m \times L$ matrix where each column refers to a common position in each sequence. A consensus sequence s_C of length L is constructed by extracting the symbol having the highest frequency in each column. This approach to sequence model construction, while quite rudimentary, is often useful for visualizing obvious qualitative relationships amongst sequence elements.

Using the vector space approach, it is possible to recover the consensus sequence. Assuming each sequence symbol is encoded into a k dimensional vector, each sequence will be encoded into a vector of length $n = Lk$ (see Section 3). Hence the original $m \times L$ matrix of sequences will be transformed into an $n \times m$ data matrix of the form described in Section 2.1. In this case, each column vector in the data matrix represents an encoded amino acid sequence.

To recover the consensus, it is useful to introduce notation for describing an empirically derived average vector μ_A from an $n \times m$ data matrix A as follows:

$$\mu_A \equiv (\frac{1}{m})A\mathbf{e} \tag{28}$$

where $\mathbf{e}$ and $m \times 1$ vector of ones. Then, μ_A is an $n \times 1$ column vector made up of L contiguous 'subvectors' of dimension k where the value of k depends upon the encoding method applied. Let v_i for $i = 1, \cdots, L$ represent each subvector in μ_A; then, the i^{th} symbol in the consensus sequence $s_C(i)$ can be inferred by associating the component of v_i yielding the highest average with the originally encoded symbol. To be precise, let the alphabet of k sequence symbols (e.g. DNA, amino acids, structural, text, etc) be defined as

$$\mathcal{A} \equiv \{a_1, a_2, \cdots, a_k\} \tag{29}$$

and let the j^{th} component of v_i be written as v_{ij} for $j = 1, \cdots, k$. The subscript index of the component with the maximum average in v_i can therefore be extracted as

$$J = \arg\max_{j=1,\cdots,k} v_{ij} \tag{30}$$

and the associated alphabet symbol is entered into the i^{th} position of the consensus sequence as

$$s_C(i) = a_J \tag{31}$$

where $a_J \in \mathcal{A}$. The algorithm for recovering the consensus sequence can be summarized as follows:

1. Given the $n \times m$ encoded data matrix A, compute μ_A.
2. For each v_i where $i = 1, \cdots, L$, apply Equation (30).
3. Given the alphabet $\mathcal{A}$, apply Equation (31) in order to construct the consensus sequence s_C.

5.2 Position specific scoring matrix

The consensus sequence, while qualitatively useful, is an incomplete sequence model in that it does not consider cases where two or more symbols in a given position are close to equiprobable. Under these circumstances, one is forced to arbitrarily choose one symbol for the consensus at the expense of loosing information about the other symbols. In contrast, the position specific scoring matrix (PSSM) is a sequence model that considers the frequency of occurrence of all symbols in each position. Furthermore, the PSSM can be used to score and rank sequences of unknown function in order to quantify their similarity to the sequence model.

Given an $m \times L$ matrix of of m related sequences of length L and an alphabet of k symbols, a $k \times L$ 'profile' matrix of empirical probabilities is first constructed by computing the symbol frequency for each position. The profile matrix can be thought of as the preimage of the PSSM. While it can provide important statistical details regarding the sequence model, it does not have the capability to score sequences in an additive fashion position by position. To do this requires converting the profile into a $k \times L$ PSSM of additive information scores. Given a sequence s of length L, the PSSM can then be used to compute a score for s in order to determine its relationship to the sequence model.

Recovering the PSSM from the vector space approach is straightforward. Given an $n \times m$ data matrix of encoded sequences, the i^{th} subvector v_i in the average vector μ_A computed from Equation (28) is equivalent to the i^{th} column in the $k \times L$ profile matrix. Simply reshaping the $kL \times 1$ vector μ_A into a $k \times L$ matrix recovers the profile. However, since the goal is to score sequences of unknown function, we are more interested in showing how μ_A can be applied to recover a PSSM score. Assume that the components of μ_A have been transformed by applying the same information measure $\mathcal{I}_{PSSM}$ used to convert the profile to the PSSM. Assuming an encoding alphabet with k symbols, a query sequence s of length L can be encoded to form a $kL \times 1$ vector x. The PSSM score S_{PSSM} of x can then be recovered via the inner product:

$$S_{PSSM} = x^T \mathcal{I}_{PSSM}(\mu_A) \tag{32}$$

where $\mathcal{I}_{PSSM}(\mu_A)$ represents the conversion of a probability vector into an vector of additive information scores.

The similarity of Equation (32) with Equation (2) is worth noting. Assume several families of sequences of equal length L are encoded into separate data matrices A_i where $i = 1, \cdots, M$ and M is the number of families. It should be clear that the relevance score for the query vector x can be produced using the cosine similarity according to

$$S_{PSSM} = x^T \mathcal{I}_{PSSM} \tag{33}$$

where

$$\mathcal{I}_{PSSM} = [\mathcal{I}_{PSSM}(\mu_{A_1})\ \mathcal{I}_{PSSM}(\mu_{A_2})\ \cdots\ \mathcal{I}_{PSSM}(\mu_{A_M})] \tag{34}$$

is the $n \times M$ information matrix that describes the sequence families.

It is of important theoretical interest that the vector space approach recovers both the PSSM and its information capacity to score sequences. However, it is more useful to observe that invoking an algebraic structure on a set of sequences induces a spectrum of novel possibilities. For instance, the SVD can be applied to the data matrix and a scoring scheme can be derived from the computed orthogonal basis. In addition, as mentioned at the end of Section 3, it is possible to weight both the data matrix and the encoded sequence according to more biologically significant measures such as hydrophobicity. Finally, and probably most importantly, the vector space formulation allows for powerful optimization techniques Golub & Van Loan (1989); Luenberger (1969) to be applied in order to maximize the scoring capacity of the sequence model.

5.3 Clustering

Our goal in this section is to investigate how clustering encoded sets of vectors will partition an existing set of data. While there are several approaches to performing data clustering Theodoridis & Koutroumbas (2003), we choose to invoke techniques that characterize the mean behavior of a data cluster. Specifically, we analyze one supervised method (Section 5.3.2) and one unsupervised method (Section 5.3.3). As we shall see, these approaches will enable us to construct 'fuzzy' regular expressions capable of algebraically describing the behavior of a given data set. It will become clear that this approach will offer additional insight to sequence clustering techniques typically encountered in the literature Henikoff & Henikoff (1991); Smith et al. (1990). As the BLOCKS database Henikoff et al. (2000); Pietrokovski et al. (1996) has been constructed from sequence clusters using ungapped multiple alignment, we choose to apply this database as the template in order to compare it against the vector space model.

5.3.1 The BLOCKS database

The BLOCKS database consists of approximately 3000 protein families (or 'blocks'). Each family has a varying number of sequences that have been derived from ungapped alignments. Therefore, while sequence lengths between two different families may differ, sequences contained within each family, by the definition of a 'block', must all have the same length. Furthermore, the number of sequences in each family can vary and there is can be a considerable degree of redundancy within some families; hence, it is sensible to analyze how the data is distributed with respect to each BLOCKS family.

The histogram in Figure 2 illustrates the number of BLOCKS families as function of sequence length. For example, there are 90 families containing sequences of length $L = 40$. From this figure, we can conclude that it is generally possible to find at least 40 families containing nominal sequence lengths. It is also important to characterize how the number of sequences contained within each family is distributed throughout the database. The histogram in Figure 3 illustrates the number of BLOCKS families as function of the number of sequences contained within each family. From this figure, we observe that many families contain somewhere between 9 and 20 representative sequences. Finally, for the sake of clarity, we restrict our attention to sequences having the same lengths. The extension of these results to variable length sequences is the subject of current research based upon existing methodologies cited in the literature Couto et al. (2007); T. Rodrigues (2004). The histogram in Figure 4 illustrates the number of BLOCKS families as function of the number of sequences contained within in each family; however, observe that this representative sample has been restricted to those families containing sequences of equal length (in this case $L = 30$). The behavior in this graph is typical in that most families contain on the order of 10-12 sequences of equal length. For the purposes of illustration and without loss of generality, we choose to demonstrate the techniques in the upcoming sections using families containing sequences of equal length.

5.3.2 Centroid approach

In this section, we cluster sequences whose BLOCKS classification is known a priori in order to algebraically characterize each family. To do this, each family in the analysis is encoded separately and Equation (28) is applied to each family data matrix in order to derive a family centroid. Since the families are already partitioned, this approach is a supervised clustering technique that will enable us to derive symbol contributions from the centroid vectors.

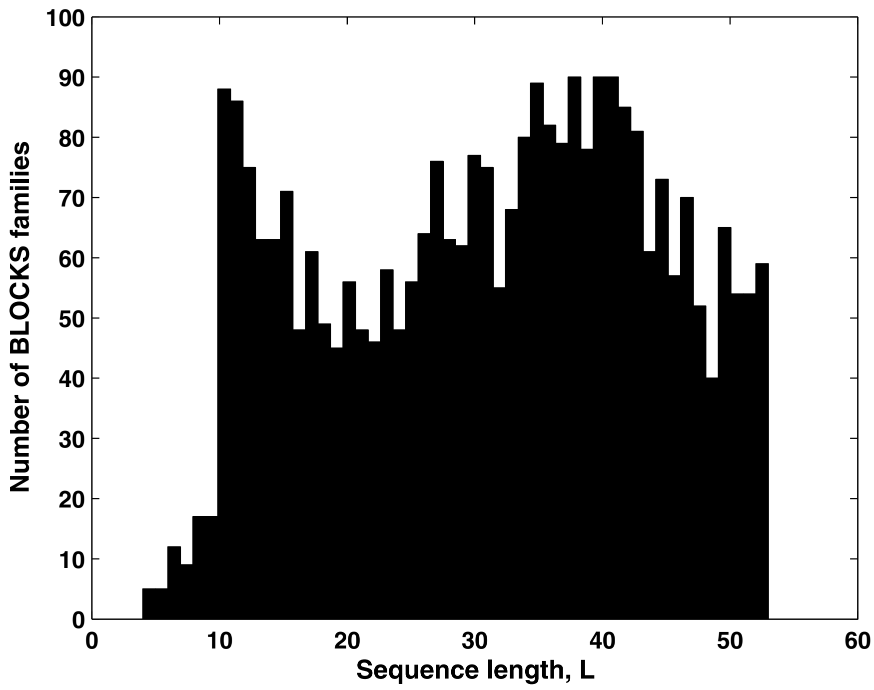


Fig. 2. Histogram of the number of BLOCKS families as function of sequence length.

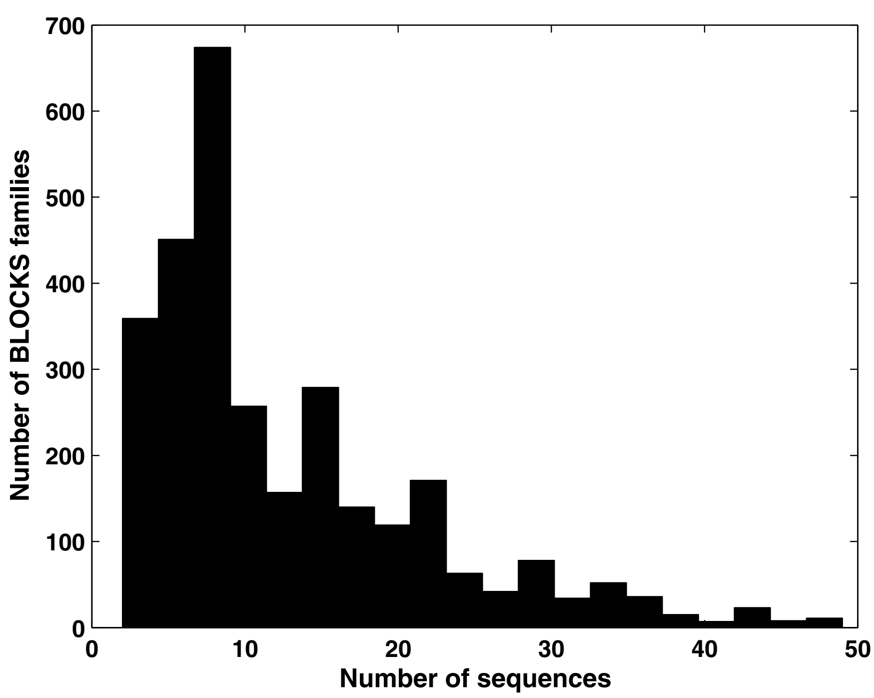


Fig. 3. Histogram of the number of BLOCKS families as function of the number of sequences contained in each family.

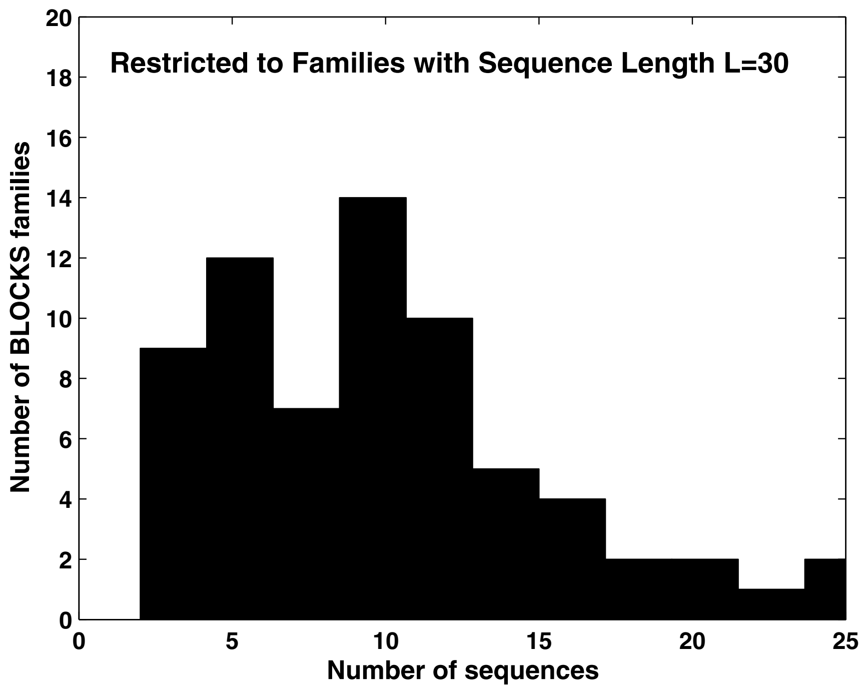


Fig. 4. Histogram of the number of BLOCKS families as function of the number of sequences contained in each family (restricted to families with sequences of length L=30)

For this numerical experiment, we apply Table 5 as the encoding scheme and choose the BLOCKS family sequence length to be $L = 30$. Under these conditions, sequences will be encoded into column vectors of dimension $n = (30)(11) = 330$. In addition, all encoded data vectors are normalized to have unit magnitude.

There are 73 families in the BLOCKS database that have block length $L = 30$. Furthermore, there are a total of 910 sequences distributed amongst the 73 families. As mentioned above, there is a small degree of sequence redundancy within some BLOCKS families. After removing redundant sequences, a total of $J = 755$ sequences of length $L = 30$ are distributed amongst $I = 73$ families. Given the encoding method, the dimensions of the non-redundant data matrix A will be 330×755.

Figure 5 shows the results of computing the distance between all centroids. From this histogram, we observe that database families are fairly well-separated since the minimum distance between any two centroids is greater than 0.6.

In order to analyze the performance of the encoding method, we apply the inner product. Specifically, each data vector v_j is classified by choosing the family associated with the centroid yielding the largest inner product:

$$C(v_j) \equiv \arg\max_{i=1,\cdots,I} v_j^T \mathcal{M}. \tag{35}$$

where $j = 1, \cdots, J$ and

$$\mathcal{M} = \left[\mu_{A_1}\ \mu_{A_2}\ \cdots\ \mu_{A_I}\right] \tag{36}$$

For standard encoding (i.e. $k = 20$, $n = 600$), all 755 data vectors were classified correctly using Equation (35). On the other hand, when applying the encoding method in Table 5, there

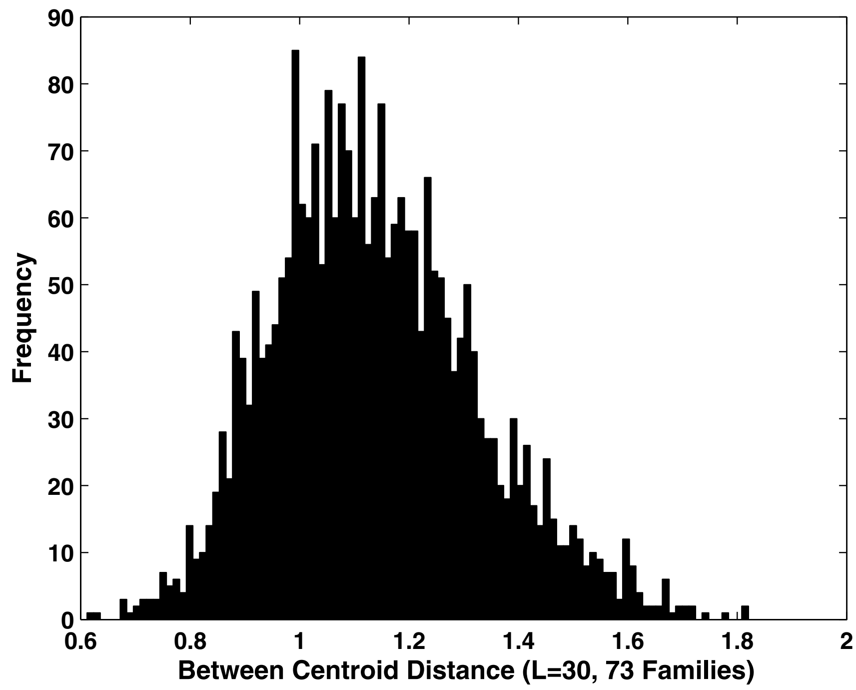


Fig. 5. Histogram of between centroid distance.

was one misclassification. Figure 6 illustrates that data vector number 431 (which as member of family 30, 'HlyD family secretion proteins') was misclassified into family 54 (Osteopontin proteins). So, while the vector dimension is reduced from 600 to 330 (because k is reduced from 20 to 11), a minor cost in classification accuracy is incurred. At the same time, we observe a substantial reduction in dimensionality.

We note one final application of the centroid approach for deriving 'fuzzy' regular expressions extracted from the vector components of the centroid vectors. Consider the sum normalized i^{th} family centroid

$$\mathcal{N}_{A_i} \equiv \left(\frac{1}{\sum_{j=1}^{n} (\mu_{A_i})_j} \right) \; \mu_{A_i}. \tag{37}$$

For each subvector associated with each sequence position in $\mathcal{N}_{A_i}$, it is then possible to write an expression describing the percentage contribution of each symbol to analytically characterize the i^{th} sequence family.

5.3.3 K-means approach

In contrast to the supervised approach, we now wish to take all sequences of length L in the database and investigate how they are clustered when the unsupervised K-means algorithm is applied. When this algorithm is applied to small numbers of families (e.g. < 10), our results indicate that this algorithm will accurately determine the sequence families for the encoding method presented. However, as the number of data vectors grow, the high-dimensionality of the encoding method tends to obscure distances and, hence, can obscure the clusters. We briefly address this issue in the conclusions section of this chapter.

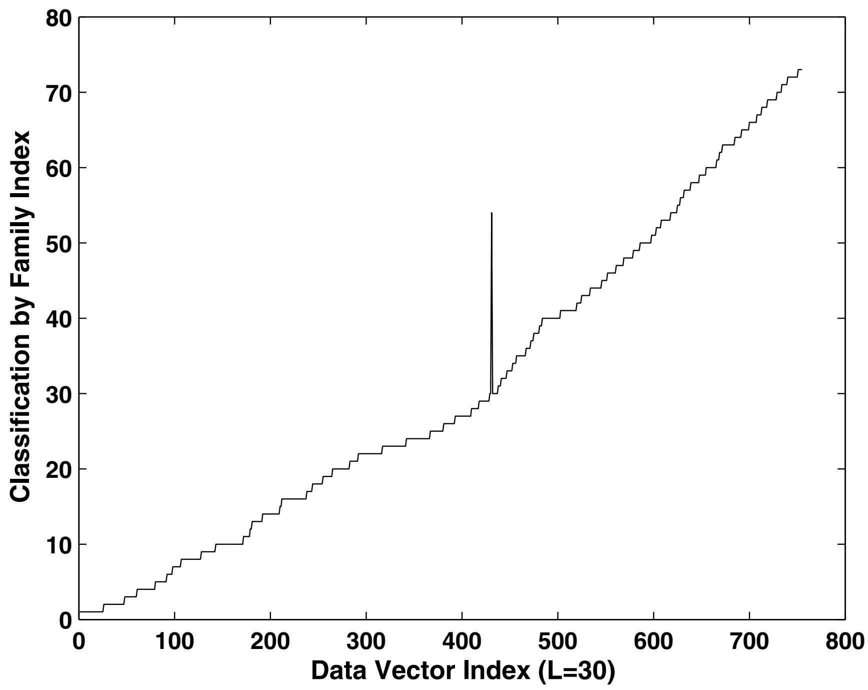


Fig. 6. Family classification of each data vector.

5.4 Database search and pattern classification

We now come to what is arguably one of the most important applications in this chapter. In this section, we will apply the reliability and relevance measures summarized in Table 6 to perform BLOCKS database searches and pattern classification Bishop (2006); Hand et al. (2001).

5.4.1 Characterization of BLOCKS orthogonal complement

When constructing a database, it is critical to understand and analytically characterize the spectrum of objects *not* contained within the database. This task is easily achieved by considering the orthogonal complement. As first step, we consider families with sequence lengths $L = 15$ (70 families) and $L = 30$ (73 families). Furthermore, we compare encodings from Table 3 and Table 5 with standard encoding. Specifically, for each encoding method, an $n \times m$ non-redundant data matrix A consisting of all data vectors of from all families with sequence length L is constructed. The SVD is then applied to construct an orthogonal basis Q_A for the column space of A. The rank r of A (r=$\mathcal{D}[Q_A]$) and the dimension of the null space of A are then compared ($\mathcal{D}[\mathcal{N}(Q_A^T)]$). Using this approach, it is then possible to assess the quantity $n - \mathcal{D}[Q_A]$ to determine the size of the subspace left uncharacterized by the database. Table 7 summarizes the results. From this table, it is clear that, after redundant encoded vectors are removed, the BLOCKS database thoroughly spans the pattern space. Furthermore, the histogram in Figure 5 further indicates that, while the sequence subspace is well represented, there is also a good degree of separation between the family classes.

5.4.2 Pattern classification

Another important database characterization is to examine how the projection method classifies data vectors after the class subspace bases have been constructed using the SVD.

renewcommand11.2

Sequence Length	Encoding Method	n	m	$\mathcal{D}[Q_A]$	$\mathcal{D}[\mathcal{N}(Q_A^T)]$
$L = 15$	Standard	300	949	286	14
$L = 15$	Table 5	165	949	165	0
$L = 15$	Table 3	30	936	30	0
$L = 30$	Standard	600	785	576	13
$L = 30$	Table 5	330	785	330	0
$L = 30$	Table 3	60	774	60	0

Table 7. Characterization of BLOCKS orthogonal complement for various sequence lengths and encodings

In a manner similar to Figure 6, we classify all encoded data vectors in order to determine their family membership by applying Equation (26). Figures 7 - 8 show results where the $L = 15$ and $L = 30$ cases have been tested. For the $L = 15$ case, as the vector space dimension decreases more classification errors arise since a reduced encoding will result in more non-unique vectors. The $L = 30$ case leads to longer vectors, hence, it is more robust to reduced encodings.

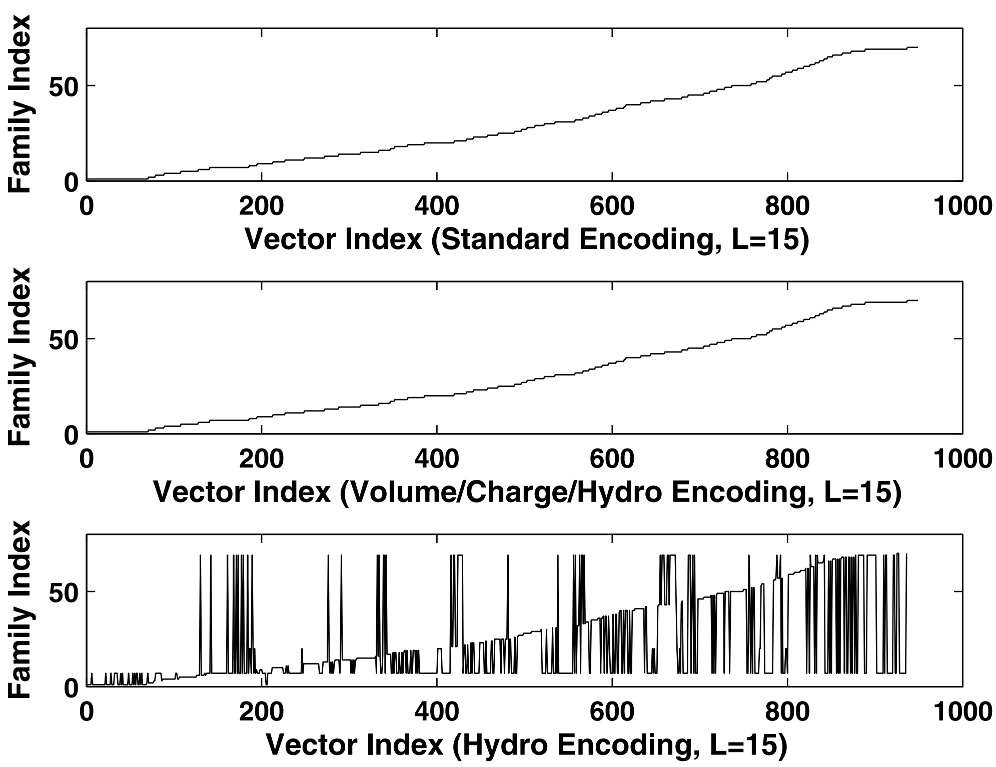


Fig. 7. Family classification of each data vector.

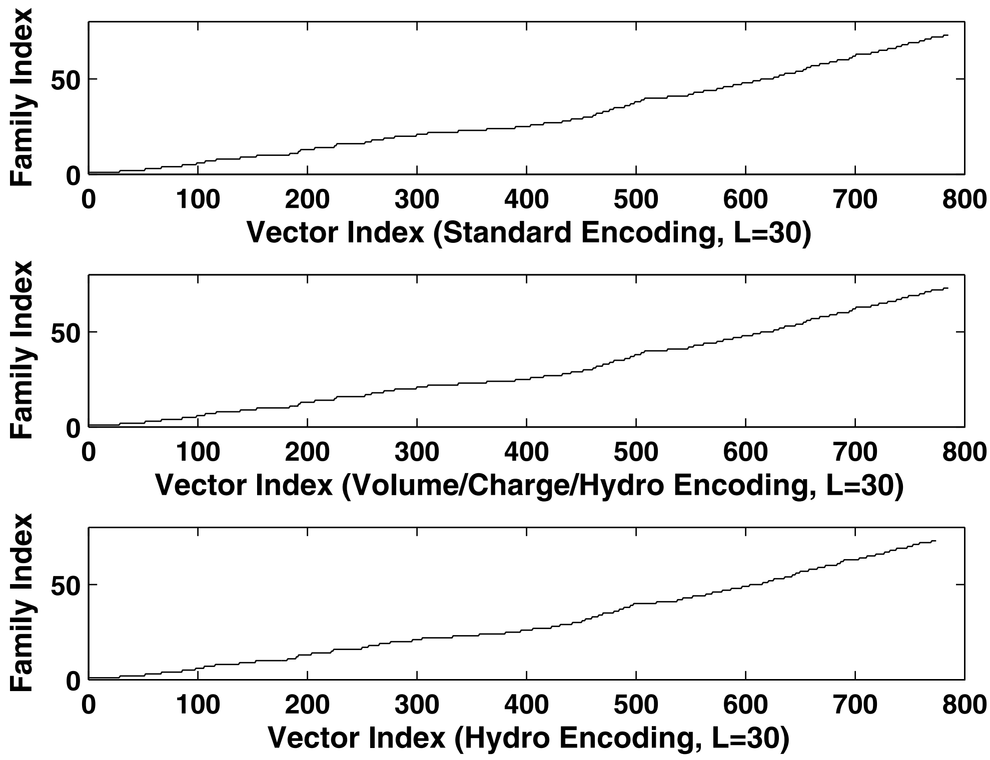


Fig. 8. Family classification of each data vector.

5.4.3 BLOCKS database search

In this section, we demonstrate how to perform database searches using the relvance and reliability equations summarized in Table 6. Database search examples have been reported using the BLOCKS database Henikoff & Henikoff (1994). In this work, we analyze the effect of randomly mutating sequences within the BLOCKS database to analyze family recognition as a function sequence mutation. For the purposes of illustration, we consider a test sequence from the Enolase protein family (BL00164D) in order to examine relevancy and database reliability. For this test sequence with $L = 15$, amino acids are randomly changed where the number of positions mutated is gradually increased from 0 to 12. Furthermore, encodings from Table 3 are compared with standard encoding.

For this series of tests, the reliability always gives a value of $\cos(\phi) = 1$, implying that the randomization test did not result in a vector outside the subspace defined by the database. This corroborates conclusions drawn in Section 5.4.1. Figure 9 shows that the classification remains stable for both encodings until about 5-6 positions out of 15 have been mutated (the family index for the original test sequence is 10). In addition, the relevance can be summarized by computing the *difference* between the maximum value of $\cos(\phi_i)$ and the second largest value. For the sake of illustration, if the BLOCKS family with index 10 does not yield the maximum projection, then the relevance difference is assigned a negative value. Figure 10 show the results of this computation. In this test, we observe a consistent decrease in the relevance difference indicating that secondary occurrences are gaining influence against the family class of the test sequence.

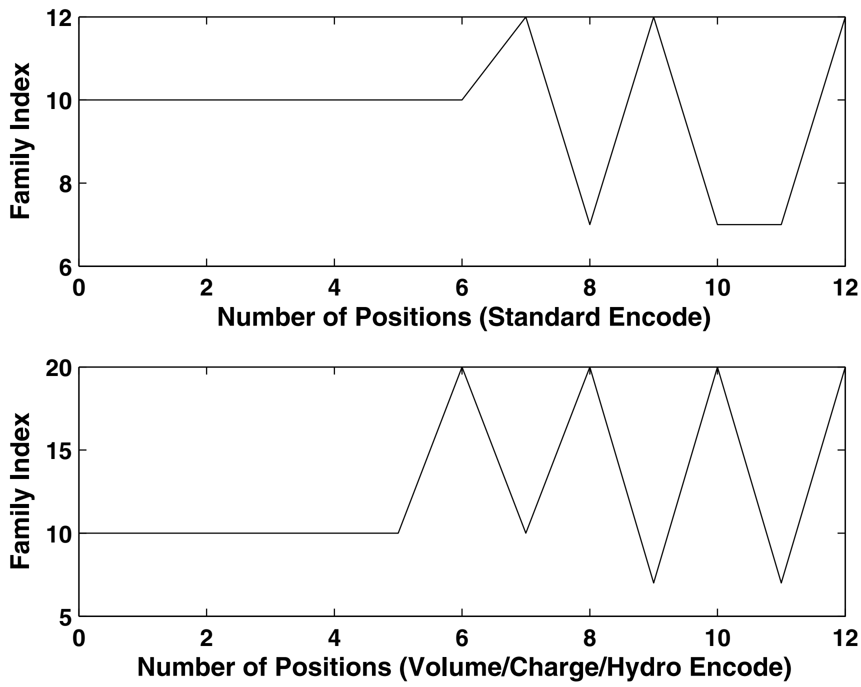


Fig. 9. Family classification as a function of the number of positions randomized.

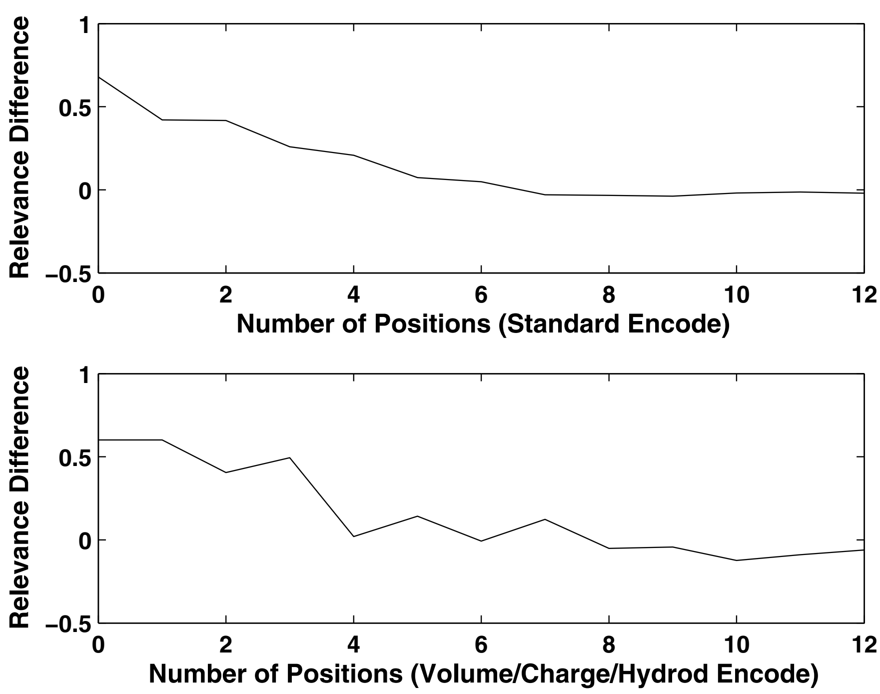


Fig. 10. Relevance differential as a function of the number of positions randomized.

6. Conclusions

This chapter has elaborated upon the application of information retrieval techniques to various computational approaches in bioinformatics such as sequence modeling, clustering, pattern classification and database searching. While extensions to multiple sequence alignment have been alluded to in the literature Couto et al. (2007); Stuart, Moffett & Baker (2002), there is a need to include and model gaps in the approaches proposed in this body of work. Extensions to the vector space methods outlined in this chapter might involve including a new symbol to represent a gap. Regardless of the symbol set employed, it is clear that the approach described can lead to sparse elements embedded in high dimensional vector spaces. While data sets of this kind can be potentially problematic Beyer et al. (1999); Hinneburg et al. (2000); Houle et al. (2010); Steinbach et al. (2003), subspace dimension reduction techniques are derivable from LSI approaches such as the SVD.
The IR techniques introduced above are readily applicable in any setting where bioinformatics data (sequence, structural, symbolic, etc) can be encoded. This work has focused primarily on amino acid sequence data; however, given existing structural encoding techniques Bowie et al. (1991); Zhang et al. (2010), future work might be directed toward vector space approaches to structural data. The methods outlined in this chapter allow for novel biologically meaningful weighting schemes, algebraic regular expressions, matrix factorizations for subspace reduction as well as numerical optimization techniques applicable to high dimensional vector spaces.

7. Acknowledgements

This work was made possible by funding from grant DHS 2008-ST-062-000011, "Increasing the Pipeline of STEM Majors among Minority Serving Institutions". The authors would like to thank David J. Schneider of the USDA-ARS for many helpful discussions.

8. References

Alter, O., Brown, P. O. & Botstein, D. (2000a). Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms, *PNAS* 100: 3351–3356.

Alter, O., Brown, P. O. & Botstein, D. (2000b). Singular value decomposition for genome-wide expression data processing and modeling, *PNAS* 97: 10101–10106.

Andorf, C. M., Dobbs, D. L. & Honavar, V. G. (2002). Discovering protein function classification rules from reduced alphabet representations of protein sequences, *Proceedings of the Fourth Conference on Computational Biology and Genome Informatics*, Durham, NC, pp. 1200–1206.

Bacardit, J., Stout, M., Hirst, J. D., Valencia, A., Smith, R. E. & Krasnogor, N. (2009). Automated alphabet reduction for protein datasets, *BMC Bioinformatics* 10(6).

Baldi, P. & Brunak, S. (1998). *Bioinformatics: The Machine Learning Approach*, MIT Press, Cambridge, MA.

Baxevanis, A. D. & Ouellette, B. F. (2005). *Bioinformatics: A practical guide to the analysis of genes and proteins*, Wiley.

Berry, M. W. & Browne, M. (2005). *Understanding Search Engines: Mathematical Seacrh Engines and Text Retrieval*, SIAM.

Berry, M. W., Drmac, Z. & Jessup, E. R. (1999). Matrices, vector spaces, and information retrieval, *SIAM Rev.* 41: 335–362.

Berry, M. W., Dumais, S. T. & OŠBrien, G. W. (1995). Using linear algebra for intelligent information retrieval, *SIAM Rev.* 37: 573–595.

Beyer, K., Goldstein, J., Ramakrishnan, R. & Shaft, U. (1999). When is "nearest neighbor" meaningful?, *In Int. Conf. on Database Theory*, pp. 217–235.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*, Springer.

Bordo, D. & Argos, P. (1991). Suggestions for safe residue substitutions in site-directed mutagenesis, *Journal of Molecular Biology* 217: 721–729.

Bowie, J. U., Luthy, R. & Eisenberg, D. (1991). A method to identify protein sequences that fold into a known three-dimensional structure, *Science* 253: 164–170.

Couto, B. R. G. M., Ladeira, A. P. & Santos, M. A. (2007). Application of latent semantic indexing to evaluate the similarity of sets of sequences without multiple alignments character-by-character, *Genetics and Molecular Research* 6: 983–999.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. (1990). Indexing by latent semantic analysis, *Journalof the American Society for Information Science* 41: 391–407.

Dominich, S. (2010). *The Modern Algebra of Information Retrieval*, Springer.

Done, B. (2009). Gene function discovery using latent semantic indexing, *Wayne State University (Ph.D.. Thesis)* .

Durbin, R., Eddy, S., Krogh, A. & Mitchison, G. (2004). *Biological Sequence Analysis*, Cambridge University.

Eisenberg, D., Schwarz, E., Komaromy, M. & Wall, R. (1984). Analysis of membrane and surface protein sequences with the hydrophobic moment plot, *Journal of Molecular Biology* 179: 125–142.

Elden, L. (2004). *Matrix Methods in Data Mining and Pattern Recognition*, SIAM.

Feldman, R. & Sanger, J. (2007). *The Text Mining Handbook*, Cambridge.

Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., Gavin, O., Gunesekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E., Eddy, S. & Bateman, A. (2010). The Pfam protein families database, *Nucl. Acids Res.* 38: D211–222.

Golub, G. H. & Van Loan, C. F. (1989). *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD.

Grossman, D. A. & Frieder, O. (2004). *Information Retrieval: Algorithms and Heuristics*, Springer.

Hand, D., Mannila, H. & Smyth, P. (2001). *Principles of Data Mining*, MIT Press.

Henikoff, J. G., Greene, E. A., Pietrokovski, S. & Henikoff, S. (2000). Increased coverage of protein families with the blocks database servers, *Nucl. Acids Res.* 28: 228–230.

Henikoff, S. & Henikoff, J. G. (1991). Automated assembly of protein blocks for database searching, *Nucleic Acids Research* 19: 6565–6572.

Henikoff, S. & Henikoff, J. G. (1994). Protein family classification based on searching a database of blocks, *Genomics* 19: 97–107.

Hinneburg, E., Aggarwal, C., Keim, D. A. & Hinneburg, A. (2000). What is the nearest neighbor in high dimensional spaces?, *In Proceedings of the 26th VLDB Conference*, pp. 506–515.

Houle, M., Kriegel, H., Kröger, P., Schubert, E. & Zimek, A. (2010). Can shared-neighbor distances defeat the curse of dimensionality?, *in* M. Gertz & B. Ludascher (eds), *Scientific and Statistical Database Management*, Vol. 6187 of *Lecture Notes in Computer Science*, Springer, pp. 482–500.

Khatri, P., Done, B., Rao, A., Done, A. & Draghici, S. (2005). A semantic analysis of the annotations of the human genome, *Bioinformatics* 21: 3416–3421.

Klie, S., Martens, L., Vizcaino, J. A., Cote, R., Jones, P., Apweiler, R., Hinneburg, A. & Hermjakob, H. (2008). Analyzing large-scale proteomics projects with latent semantic indexing, *Journal of Proteome Research* 7: 182–191.

Kuruvilla, F. G., Park, P. J. & Schreiber, S. L. (2004). Vector algebra in the analysis of genome-wide expression data, *Genome Biology* 3(3).

Kyte, J. & Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein, *Journal of Molecular Biology* 157: 105–132.

Langville, A. N. & Meyer, C. D. (2006). *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press.

Lay, D. C. (2005). *Linear Algebra and Its Applications*, Wiley.

Luenberger, D. G. (1969). *Optimization by Vector Space Methods*, Wiley.

Mount, D. W. (2004). *Bioinformatics: Sequence and Genomic Analysis*, Cold Spring Harbor Laboratory Press.

Oja, E. (1983). *Subspace Methods of Pattern Recognition*, Wiley, New York, NY.

Pietrokovski, S., Henikoff, J. G. & Henikoff, S. (1996). The blocks database - a system for protein classification, *Nucl. Acids Res.* 24: 197–200.

Salton, G. & Buckley, C. (1990). Improving retrieval performance by relevance feedbackl, *J. Amer. Soc. Info. Sci.* 41: 288–297.

Sigrist, C. J. A., Cerutti, L., de Castro, E., Langendijk-Genevaux, P. S., Bulliard, V., Bairoch, A. & Hulo, N. (2010). PROSITE: a protein domain database for functional characterization and annotation, *Nucl. Acids Res.* 38: D161–166.

Smith, H., Annau, T. & Chandrasegaran, S. (1990). Finding sequence motifs in groups of functionally related proteins, *PNAS* 87: 826–830.

Steinbach, M., Ertöz, L. & Kumar, V. (2003). The challenges of clustering high-dimensional data, *In New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*, Springer-Verlag.

Stuart, G. W. & Berry, M. W. (2004). An SVD-based comparison of nine whole eukaryotic genomes supports a coelomate rather than ecdysozoan lineage, *BMC Bioinformatics* 5(204).

Stuart, G. W., Moffett, K. & Baker, S. (2002). Integrated gene and species phylogenies from unaligned whole genome protein sequences, *Bioinformatics* 18: 100–108.

Stuart, G. W., Moffett, K. & Leader, J. J. (2002). A comprehensive vertebrate phylogeny using vector representations of protein sequences from whole genomes, *Mol. Biol. Evol.* 19: 554–562.

T. Rodrigues, L. Pacífico, S. T. (2004). Clustering and artificial neural networks: Classification of variable lengths of Helminth antigens in set of domains, *Genetics and Molecular Biology* 27: 673–678.

Theodoridis, S. & Koutroumbas, K. (2003). *Pattern Recognition*, Elsevier.

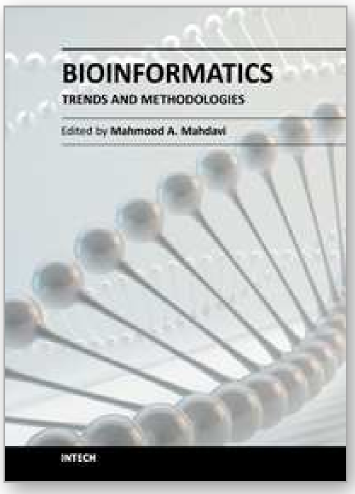
Wall, M. E., Rechtsteiner, A. & Rocha, L. M. (2003). *Singular value decomposition and principal component analysis*, KLuwer, pp. 91–109.

Wang, J. T. L., Zaki, M. J., Toivonen, H. T. T. & Sasha, D. (eds) (2005). *Data Mining in Bioinformatics*, Spinger.

Weiss, S. M., Indurkhya, N., Zhang, T. & Damerau, F. J. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer.

Wimley, W. C. & White, S. H. (1996). Experimentally determined hydrophobicity scale for proteins at membrane interfaces, *Nature Structural Biology* 3: 842–848.

Zhang, Z. H., Lee, H. K. & Mihalek, I. (2010). Reduced representation of protein structure: implications on efficiency and scope of detection of structural similarity, *BMC Bioinformatics* 11(155).



Bioinformatics - Trends and Methodologies
Edited by Dr. Mahmood A. Mahdavi

ISBN 978-953-307-282-1
Hard cover, 722 pages
Publisher InTech
Published online 02, November, 2011
Published in print edition November, 2011