UTILIZANDO AGRUPAMENTO COM RESTRIÇÕES E AGRUPAMENTO ESPECTRAL PARA INTEGRAÇÃO DE DADOS DE ENZIMAS

ELISA BOARI DE LIMA

UTILIZANDO AGRUPAMENTO COM RESTRIÇÕES E AGRUPAMENTO ESPECTRAL PARA INTEGRAÇÃO DE DADOS DE ENZIMAS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Wagner Meira Júnior

Coorientadora: Raquel Cardoso de Melo Minardi

Belo Horizonte

Fevereiro de 2011

ELISA BOARI DE LIMA

CONSTRAINED CLUSTERING AND SPECTRAL CLUSTERING FOR ENZYME DATA INTEGRATION

Thesis presented to the Graduate Program in Computer Science of Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Wagner Meira Júnior

Co-Advisor: Raquel Cardoso de Melo Minardi

Belo Horizonte

February of 2011

Lima, Elisa Boari de.

L732u Utilizando agrupamento com restrições e agrupamento espectral para integração de dados de enzimas / Elisa Boari de Lima. — Belo Horizonte, 2011.
xxiv, 84 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais. Departamento de Ciência da Computação.

Orientador: Meira Júnior, Wagner.
Coorientadora: Minardi, Raquel Cardoso de Melo.

1. Computação – Teses. 2. Mineração de Dados (Computação) – Teses. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*72 (043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Utilizando agrupamento com restrições e agrupamento espectral para integração de dados de enzimas

ELISA BOARI DE LIMA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. WAGNER MEIRA JÚNIOR - Orientador
Departamento de Ciência da Computação - UFMG

PROFA. RAQUEL CARDOSO DE MELO MINARDI - Co-orientadora
Departamento de Ciência da Computação - UFMG

PROF. ALTIGRAN SOARES DA SILVA
Departamento de Ciência da Computação - UFAM

PROF. CARLOS HENRIQUE DA SILVEIRA
Universidade Federal de Itajubá - MG

Belo Horizonte, 28 de fevereiro de 2011.

To my parents, who have always believed in me and shown unconditional support.

Acknowledgments

It is a pleasure to thank those who made this thesis possible. First and foremost, I would like to thank God for another incredible opportunity. I am deeply grateful to my parents, José and Annete, for all the love and encouragement, and for teaching me that even the toughest tasks can be achieved one step at a time. To my beloved siblings, Drielle and Alexandre, and family, for believing in me even without quite understanding my work.

My sincere gratitude to my advisor Wagner Meira Jr. for his insight, support and guidance throughout the writing of this thesis. To my co-advisor Raquel Minardi, for all the conversations, motivation, friendship and inspiration. To professor Mohammed J. Zaki, for generously welcoming me to Rensselaer Polytechnic Institute with invaluable comments and suggestions.

My special thanks to Henrique Rodrigues, for the immense patience and strength. To Douglas Pires, for helping me with Cutoff Scanning. To my friends and colleagues, for helping me persevere and without whom it would have been impossible to maintain my peace of mind. And to all those who somehow contributed to this thesis, my deepest appreciation.

Resumo

Quando múltiplas fontes de dados estão disponíveis para serem mineradas, geralmente é necessário um processo a priori de integração de dados. Tal processo pode ser custoso e não levar a bons resultados, visto que informação importante possivelmente será descartada. Nesta dissertação se propõe o uso de agrupamento com restrições e agrupamento espectral como estratégias para integrar fontes de dados sem perda de qualquer informação. O processo consiste basicamente em adicionar as fontes complementares na forma de restrições que os algoritmos de agrupamento devem satisfazer, ou utilizá-las para aumentar a similaridade entre pares de objetos para os algoritmos de agrupamento espectral.

Como uma aplicação concreta desta abordagem, esta dissertação foca no problema de previsão de funções enzimáticas, que é uma tarefa complexa, geralmente realizada por meio de trabalho experimental intensivo. Agrupamentos com restrições e espectral são empregados como meios de integração de informação proveniente de diversas fontes, e a forma como tal informação impacta a qualidade dos resultados em um cenário de agrupamento de enzimas é analisada. Os resultados mostram que o uso de conhecimento de domínio melhora, em geral, a qualidade dos agrupamentos em comparação com os resultados obtidos utilizando apenas a base de dados principal.

Palavras-chave: agrupamento com restrições, agrupamento de enzimas, agrupamento espectral, integração de dados.

Abstract

When multiple data sources are available for data mining, an a priori data integration process is usually required. This process may be costly and not lead to good results, since important information is likely to be discarded. In this master's thesis, we propose constrained clustering and spectral clustering as strategies for integrating data sources without losing any information. The process basically consists of adding the complementary data sources as constraints that the clustering algorithms must satisfy, or using them to increase the similarity between pairs of objects for the spectral clustering algorithms.

As a concrete application of our approach, we focus on the problem of enzyme function prediction, which is a hard task usually performed by intensive experimental work. We use constrained and spectral clustering as means of integrating information from diverse sources, and analyze how this additional information impacts clustering quality in an enzyme clustering application scenario. Our results show that the use of such additional information generally improves the clustering quality when compared to the results using only the main database.

Keywords: constrained clustering, data integration, enzyme clustering, spectral clustering.

List of Figures

List of Tables

List of Algorithms

Contents

Chapter 1

Introduction

In recent years, there has been a general increase in the amount of data publicly available worldwide. This is true for various areas of knowledge, particularly in the field of Bioinformatics, where massive amounts of data have been collected in the form of DNA sequences, protein sequences and structures, information on biological pathways, etc. This has lead to diverse and scattered sources of biological data.

Protein function prediction, and especially enzyme function prediction (which involves predicting the reaction it catalyzes, its mechanisms, substrates and products), is a very active Bioinformatics research topic. This is due to the exponential increase in the number of proteins being discovered because of sequenced genomes, to the difficulties in experimentally characterizing enzyme function and mechanisms, and to the potential biotechnological use of newly discovered enzyme functions. Predicting a protein's function is a hard task usually performed by labor-intensive experimental work or in a semi-automatic manner using sequence homology. This problem may vastly benefit from clustering techniques, since they allow the creation of groups of similar proteins that can be jointly studied. Seeing that similar proteins are likely to have similar functions, this would facilitate function prediction.

The manner in which biological information is scattered in so many different datasets poses a challenge for clustering algorithms. Valuable information is spread among mostly unstandardized, redundant and incomplete repositories across the World Wide Web. The Protein Data Bank (PDB), for instance, which is a repository of three-dimensional structural data, may have dozens or even hundreds of entries for the same molecule. The various data sources call for data integration, which usually would be performed before the actual clustering algorithm is applied.

1.1 Data Integration

Data mining often requires data integration, which is the combination of data from multiple sources into a coherent dataset. Various issues must be considered during data integration, such as what is referred to as the entity identification problem: how can equivalent real-world entities from multiple data sources be matched up? Also, some attributes representing a given concept may have different names in different databases, causing inconsistencies and redundancies. Metadata may be used to help avoid errors in schema integration [32].

An issue that must be faced is redundancy, which occurs when a given attribute can be derived from other attributes, or when there exist inconsistencies in attribute names. Having a large amount of redundant data may slow down or confuse the data mining process. According to Han and Kamber [32], some redundancies can be detected by correlation analysis, which measures how strongly one implies the other based on the available data. A strong correlation may imply that one of the attributes can be removed as a redundancy. One must note that correlation among attributes does not necessarily imply that one causes the other. Apart from attribute redundancy, tuple duplication also must be detected and dealt with. Failing to use normalized database tables, for example, is a source of redundancy. Duplicates can also cause inconsistencies due to inaccurate inputing or incomplete updating.

The data integration process must also deal with detecting and resolving conflicts between data values. Different sources may have different attribute values for the same real-world entity for example, possibly due to different representations, scaling or encoding [32]. The structure of the data must be carefully considered when matching attributes from one database to another in order to ensure that any attribute functional dependencies and constraints in the source system match those in the target system.

According to Han and Kamber [32], some challenges in data integration are the semantic heterogeneity and structure of data. A careful integration process can help reduce redundancies and inconsistencies in the resulting dataset, which in turn can help improve accuracy and speed of the subsequent data mining process. However, a careful data integration process inevitably presents a high cost.

Such a priori data integration is hard and may not lead to good results, since important information could be discarded in the process. A solution to the problem of integrating various data sources without losing any important information is constrained clustering, which is simply the process of starting from a basic clustering and

adding the supplementary information as constraints to be satisfied by the clustering algorithm. This allows the clustering problem to be incrementally solved, using the truly useful information without the cost of an a priori data integration process. Using additional datasets in the form of constraints or to increase the similarity between a pair of objects in a similarity matrix instead of performing a complete data integration process, as done in this thesis, saves time and computational resources, as well as avoids that valuable data be discarded.

In this master's thesis, we use constrained clustering and spectral clustering techniques as means of integrating information from diverse sources so as to verify the manner in which additional information other than the dataset itself impacts the clustering results. The chosen application scenario is that of clustering enzymes, and three different approaches are used: clustering enzyme families; clustering subfamilies when multiple families are combined; and clustering subfamilies inside a single enzyme family, which is the problem of determining different substrate specificities in a family of enzymes able to recognize the same overall substrates.

1.2 Objectives and Justification

The main goal of this master's thesis is to analyze how the integration of various data sources via constrained and spectral clustering affects the quality of the results in an enzyme clustering application scenario. The specific objectives of the thesis are:

- The study of constrained clustering and spectral clustering techniques reported in the literature;
- The extension of classic unconstrained clustering methods by introducing constraints;
- The implementation of constrained clustering and spectral clustering techniques;
- The application of constrained clustering and spectral clustering techniques to the problem of clustering enzymes;
- The comparison of the results with those obtained by unconstrained versions of the same clustering techniques.

The application of the proposed techniques to enzyme clustering might lead to important information about enzyme function and structure, as well as functional

diversification acquired throughout family evolution. This type of methodology, which joins information from diverse and possibly incomplete sources, is of great interest because such sources complement each other.

The integration of information from multiple knowledge domains allows the algorithms to work with as much information as possible. This is extremely relevant to the problem of clustering enzymes by specificity, since substrate specificity involves much more than just sequence similarity. The genomic context in which the genes that code such enzymes are located should be evaluated since, in general, proteins whose encoding genes are close to each other in the genome are more likely to have similar functions [36]. Proteins from the same family but whose encoding genes are in very different contexts probably present substrate differences. This way, proteins with high synteny in terms of genomic context should be assigned to similar clusters.

This kind of methodology is also of great interest for its ability to raise hypotheses about the active site and the residues that determine specificity in a family of enzymes of unknown function, and may subsequently be useful in laboratory experiments aimed at finding new enzymes of biotechnological potential and reengineering of such enzymes.

The main contributions of this work are the knowledge of whether or not adding information from external sources to the database is able to improve the clustering quality for this application; the different strategies for gathering and integrating such additional information to the main database for this particular biological problem; and, most importantly, the possibility of using domain knowledge to cluster enzymes.

Chapter 2

Clustering

Clustering is an important Data Mining technique which groups similar objects without the need for any supervised information. It can be defined as the process of dividing a set of objects into groups, each of which represent a significant subpopulation, so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters. Such objects may be database records, graph nodes, words, images or any collection of individuals described by a set of attributes or relationships [7]. According to Han and Kamber [32], a cluster of data objects may be treated collectively as one group. Therefore, clustering may be considered as a form of data compression.

Cluster analysis has been widely used in numerous applications, such as market research, pattern recognition, data analysis, and image processing. It is adaptable to changes and helps single out useful features that distinguish different groups. Various categories of clustering techniques exist, such as methods based on partitions, hierarchies, densities, grids and models, as well as methods for high-dimensional data and constraint-based clustering [32].

According to Han and Kamber [32], as a Data Mining function, clustering may be used as a stand-alone tool to gain insight into the distribution of data, to observe the characteristics of each cluster, or to focus on a particular set of clusters for further analysis. It may also serve as a preprocessing step for other algorithms such as characterization, attribute subset selection, and classification, which would operate on the detected clusters and the selected attributes or features.

Typical requirements of clustering are scalability; the ability to deal with different types of attributes and to discover arbitrarily shaped clusters; minimal domain

knowledge requirements in order to determine input parameters such as the desired number of clusters; the ability to deal with noisy data and with high dimensionality; incremental clustering and insensitivity to the order of input records; constraint-based clustering; and interpretability and usability [32].

2.1 Partitioning Methods

In this master's thesis we study partitioning methods, whose basic idea is, given a database of N objects, to construct K partitions of the data, with each partition representing a cluster and $K \leq N$. Partitioning methods divide the data into K groups (i.e., clusters), which must satisfy the following requirements:

- each group must contain at least one object;
- each object must belong to exactly one group.

Given the number K of partitions to construct, a partitioning method creates an initial partitioning, after which it uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. In general, a partitioning is considered good if objects in the same cluster are related to each other, whereas objects from different clusters are very dissimilar [32].

Finding the global optimal partitioning would require exhaustive enumeration of all the possible partitions. Instead, according to Han and Kamber [32], most applications adopt one of two popular heuristic methods:

- the K-Means algorithm, where each cluster is represented by the mean value of the objects in it; or
- the K-Medoids algorithm, where each cluster is represented by one of the objects located near the center of the cluster.

Such heuristics work well for finding spherical-shaped clusters in small to medium-sized databases. However, partitioning-based clustering methods need to be extended in order to find clusters with complex shapes and for clustering very large databases. Constrained clustering is a way of doing this. In this thesis, K-Means and K-Medoids are used to work with numerical and categorical attributes, respectively. Such algorithms have been chosen for this thesis in order to analyze the effect of integrating domain knowledge via constraints in constrained versions of these classic and widely applied clustering algorithms.

2.1.1 K-Means

The idea behind the K-Means clustering method is described in Algorithm 2.1. First, K of the N objects are randomly selected to initially represent a cluster mean or center. Each of the remaining objects is assigned to the cluster to which it is the most similar, based on its distance from the cluster mean. The algorithm then computes the new mean for each cluster. This process iterates until the clusters stop changing or a criterion function converges. K-Means is sensitive to outliers because an object with an extremely large value may substantially distort the data distribution.

Algorithm 2.1 K-Means Clustering Algorithm [32]

Input: number of clusters K and dataset D containing N objects

1: arbitrarily choose K objects from D as the initial cluster centers
2: **repeat**
3: (re)assign each object to the cluster to which it is the most similar, based on the mean value of the objects in the cluster
4: update the cluster means, i.e., calculate the mean value of the objects in each cluster
5: **until** no change

Output: a set of K clusters

2.1.2 K-Medoids

Instead of taking the mean value of the objects in a cluster as a reference point, actual objects can be picked to represent the clusters, with one representative object per cluster. Each remaining object is clustered with the representative to which it is the most similar. The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point. In general, the algorithm iterates until, eventually, each representative object is actually the medoid (i.e., the most centrally located object) of its cluster [32]. This is the basis of the K-Medoids clustering method, described in Algorithm 2.2.

As with K-Means, the initial representative objects are arbitrarily chosen. The iterative process of replacing representatives with nonrepresentative objects continues as long as the quality of the resulting clustering is improved. Such quality is estimated using a cost function that measures the average dissimilarity between an object and the representative of its cluster. In order to determine whether or not a nonrepresentative object o_{random} is a good replacement for a current representative o_j, four cases must be examined for each of the other nonrepresentative objects p:

Algorithm 2.2 K-Medoids Clustering Algorithm [32]

Input: number of clusters K and dataset D containing N objects

1: arbitrarily choose K objects from D as the initial representative objects
2: **repeat**
3: assign each remaining object to the cluster with the nearest representative object
4: randomly select a nonrepresentative object o_{random}
5: compute the total cost S of swapping representative object o_j with o_{random}
6: **if** $S < 0$ **then**
7: swap o_j with o_{random} to form the new set of K representative objects
8: **end if**
9: **until** no change

Output: a set of K clusters

1. p currently belongs to representative object o_j. If o_j is replaced by o_{random} and p is closest to one of the other representative objects o_i, $i \neq j$, then p is reassigned to o_i;

2. p currently belongs to representative object o_j. If o_j is replaced by o_{random} and p is closest to o_{random}, then p is reassigned to o_{random};

3. p currently belongs to representative object o_i, $i \neq j$. If o_j is replaced by o_{random} and p is still closest to o_i, then the assignment does not change;

4. p currently belongs to representative object o_i, $i \neq j$. If o_j is replaced by o_{random} and p is closest to o_{random}, then p is reassigned to o_{random}.

Each reassignment causes a change in absolute error E, so the cost function calculates the difference in absolute-error value if a current representative object is replaced by a nonrepresentative object. The total cost of swapping is the sum of costs incurred by all nonrepresentative objects. If the total cost is negative, then o_j is replaced with o_{random}, since the actual absolute error E would be reduced. If the cost is positive, the current representative object o_j is considered acceptable and nothing is changed in the iteration [32]. The process continues until no changes are made.

2.2 Constrained Clustering

Although clustering does not utilize supervised information, in various applications there is access to additional information or domain knowledge about the types of groups that are sought in the data. Such supplementary information may occur at object

level in the form of complementary information about actual similarities between pairs of objects, of class labels for a subset of the objects, or of user preferences about the manner in which the items should be clustered; or it may occur at cluster level, encoding knowledge about the groups themselves such as position, identity, distribution and size. Constrained clustering emerged from the need for ways to accommodate such information when available [7].

According to Basu et al. [7], while a clustering problem may be regarded as a scenario in which there is a need to partition a dataset into K groups, for a constrained clustering problem there is a priori knowledge about the desired partitioning. Constrained clustering may be seen as the process of dividing a set of N objects in some D-dimensional space into K significant groups while satisfying the imposed constraints.

A constrained clustering algorithm may consider either strict or flexible constraints. In the first case, all constraints must be satisfied, so that the purpose is to minimize the objective function while satisfying the constraints. In the latter case, the idea is to satisfy as many constraints as possible, but not necessarily all of them, so that the algorithm's objective function provides a bias towards good clusterings, while the constraints yield a bias towards a subset of good clusterings with additional desirable properties [7]. In this master's thesis we are considering strict instance-level constraints for the constrained clustering algorithms. However, the manner in which such constraints are applied to spectral clustering may be seen as a form of applying flexible constraints.

2.2.1 Types of Constraints

A set of instance-level constraints C consists of declarations about pairs of objects, where a positive or must-link constraint $c_{=}(i, j)$ indicates that instances i and j must be assigned to the same cluster, while a negative or cannot-link constraint $c_{\neq}(i, j)$ implies they must be placed in different clusters.

When constraints are available, the clustering algorithm must adapt its solution to accommodate C instead of simply outputting the partitioning that best satisfies its objective function [7]. Figure 2.1 presents an example of clustering that satisfies all three pairwise constraints, in which the unconstrained algorithm produces a clustering that prioritizes weight, while the constrained algorithm yields a completely different clustering, prioritizing height.

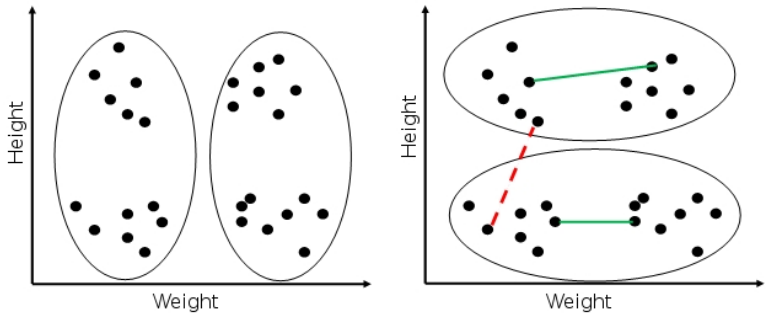

Figure 2.1. *Example of clustering that satisfies all constraints.* Straight green lines represent *must-link* constraints, while the dotted red line is a *cannot-link* constraint.

Despite seeming simple, these pairwise constraints have interesting properties. Must-link constraints are symmetric, reflexive and transitive, which allows for additional constraints to be inferred [9, 83]. Although the same is not true for the cannot-link constraint set, additional cannot-link constraints can be inferred from the must-link constraint set as described below.

Considering a graph whose nodes are instances of the dataset, whose edges (i, j) represent must-link constraints between instances i and j, and considering CC_1 and CC_2 to be connected components of this graph, it follows that:

- if there is a must-link constraint $c_{=}(x, y)$ where $x \in CC_1$ and $y \in CC_2$, then $c_{=}(a, b)$ constraints can be inferred for all $a \in CC_1$ and $b \in CC_2$;
- if there is a cannot-link constraint $c_{\neq}(x, y)$ where $x \in CC_1$ and $y \in CC_2$, then $c_{\neq}(a, b)$ constraints can be inferred for all $a \in CC_1$ and $b \in CC_2$ [17].

Constraints have typically been used in clustering algorithms to modify the stage when instances are assigned to clusters, so as to impose that the constraints be satisfied as much as possible; or to train the algorithm's distance function either before or during clustering. Also, constraints can be used in the initialization phase: the initial clusters are formed so that the instances with must-link constraints are assigned to the same cluster, while the instances with cannot-link constraints are placed in different clusters, after which an unconstrained clustering algorithm is applied [17].

2.2.2 Benefits and Problems of Using Constraints

Two main advantages of using constraints reported in the literature are the improvement of the precision in predicting labels for all instances when constraints are gen-

erated from few labeled data; and the generation of clusters with desirable geometric properties [17]. Many researchers have shown that, as the number of constraints increases, the precision of the clustering increases as well [5, 18, 41, 51, 83, 84]. When considering the average of several constraint sets, the performance on label prediction is typically higher than when constraints are not applied [17].

According to Davidson and Basu [17], two main limitations of constraint usage are normally disregarded in the literature: feasibility and the fact that not all constraint sets are useful. When constraints are employed, the clustering problem becomes that of "finding the best clustering that satisfies all of the constraints". However, if poorly specified, the constraints could directly or indirectly contradict each other so that a clustering that satisfies all of them does not exist. Davidson and Ravi [18] define the Feasibility Problem as "given a dataset X, a collection of constraints C, a lower bound K_l and an upper bound K_u on the number of clusters, does there exist a partition of X into K blocks such that $K_l \leq K \leq K_u$ and all constraints in C are satisfied?". The use of cannot-link constraints might make the feasibility problem intractable and, therefore, make constrained clustering intractable.

Davidson and Ravi [19] show that the addition of cannot-link constraints may rapidly over-restrain the constrained clustering problem so that satisfying all constraints becomes difficult. The authors also show that despite having no noise and being generated from facts, it is still possible that some constraint sets decrease the clustering precision. According to Wagstaff et al. [85], even when the number of constraints remains constant, the precision of the resulting partition greatly varies. The authors identified two properties that help explain such variations: inconsistency and incoherence. Inconsistency is the amount of conflict between the constraints and the algorithm's objective function and search bias. This measure quantifies to which degree the algorithm is incapable of discovering the constraints on its own. Incoherence, on the other hand, is the amount of internal conflict among the constraints given a distance metric, and is algorithm independent.

Wagstaff et al. [85] examined the consequences of supplying inconsistent or incoherent constraints to different constrained clustering methods and observed that constraint sets more consistent with the algorithm's bias or more internally coherent tend to produce the larger gains in precision. Therefore, for scenarios in which the user can generate multiple constraint sets, it is advisable to select the one with the least amount of inconsistency and incoherence. When multiple algorithms are available, choosing the one that presents the least inconsistency should help in yielding the best quality clustering with the smallest computational effort.

2.3 Spectral Clustering

Spectral clustering has become one of the most popular modern clustering algorithms. It is simple to implement and can be efficiently solved by standard linear algebra software, often outperforming traditional clustering algorithms such as K-Means [81]. However, why it works and what it really does are not obvious.

Two mathematical objects are used by spectral clustering: similarity graphs and graph Laplacians. Similarity graphs are a nice form of representing the data in case there is no additional information other than the similarities between objects. Each vertex v_i in the graph represents a data object x_i, and two vertices are connected if the similarity s_{ij} between the corresponding data objects is positive or larger than a certain threshold, with s_{ij} being the weight of the edge between them. The idea is to find partitions of the graph such that the edges between different groups have low weights, while the edges within a cluster have high weights [81].

2.3.1 Graph Construction Methods

Several popular methods exist for transforming a set of data points with pairwise distances or similarities into a graph. All share the same goal: to model the local neighborhood relationships between the data points. According to von Luxburg [81], theoretical results on how the choice of similarity graph affects the spectral clustering do not exist. The graph construction methods considered in this thesis are described below.

Fully Connected. Consists of simply connecting all pairs of vertices whose similarities are positive and weighting each edge by the respective s_{ij}. Since the graph is supposed to represent local neighborhoods, this fully connected graph is only useful if the similarity function itself models local neighborhoods [81].

ϵ-Neighborhood. All vertex pairs whose similarity is larger than ϵ are connected. The resulting graph is commonly considered unweighted, because weighting the edges would not introduce more information to the graph since the similarities between connected pairs are roughly at the same scale.

K-Nearest Neighbors (KNN). Vertex v_j is connected to vertex v_i if it is among the K nearest neighbors of v_i, i.e., if the edge between them is among the K edges with the largest weights involving v_i. This results in a directed graph, and two strategies may be used to make it undirected:

- simply ignore the directions of the edges, which yields the K-Nearest Neighbors graph; or
- only connect v_i and v_j if they are among the K nearest neighbors of each other, which is called Mutual KNN and yields the Mutual K-Nearest Neighbors graph.

In both cases, the edges are weighted with the similarity of their endpoints. The Mutual KNN graph is particularly well-suited if one wants to detect clusters of different densities, since it tends not to connect areas with contrasting densities.

Once the graph construction method is chosen, the values of parameters K or ϵ must be determined. According to von Luxburg [81], barely any theoretical results are known to guide this task. In case the similarity graph contains more connected components than the number of clusters the algorithm is asked to detect, spectral clustering will trivially return connected components as clusters. Therefore, unless one is certain that the connected components are the correct clusters, one should make sure that the similarity graph is either connected or only consists of few connected components and very few or no isolated vertices.

2.3.2 Graph Laplacians

Graph Laplacians are main tools for spectral clustering, but according to von Luxburg [81], there is no unique convention in the literature about which matrix exactly is called the "graph Laplacian". Consider G an undirected, weighted graph with a symmetrical non-negative weight matrix W. The degree d_i of vertex v_i is the sum of the weights of the edges that involve v_i. The graph's degree matrix D is a diagonal matrix, with $d_i, \ldots, d_N$ forming the diagonal. The two types of graph Laplacians considered in this thesis are described below.

Unnormalized graph Laplacian. Defined as $L = D - W$.

Normalized graph Laplacians. Two matrices are called normalized graph Laplacians in the literature. The first is a symmetric matrix (Equation 2.1) and the second is closely related to a random walk (Equation 2.2).

$$L_{sym} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} W D^{-1/2} \tag{2.1}$$

$$L_{rw} = D^{-1} L = I - D^{-1} L \tag{2.2}$$

Details on the properties of each Laplacian may be found in von Luxburg [81]. In this thesis, the unnormalized Laplacian matrix L and the normalized symmetric Laplacian matrix L_{sym} are used for performing spectral clustering.

Different spectral clustering algorithms exist, each using one of the graph Laplacians, as described by von Luxburg [81]. Algorithm 2.3 is applied when using the unnormalized Laplacian matrix L, while Algorithm 2.4 is applied when using the symmetric normalized Laplacian matrix L_{sym}.

Algorithm 2.3 Unnormalized Spectral Clustering [81]

Input: number of clusters K and similarity matrix $S \in \Re^{N \times N}$

1: construct a similarity graph using one of the aforementioned construction methods and let W be its weighted adjacency matrix
2: compute the unnormalized Laplacian L
3: compute the first K eigenvectors $u_i, \ldots, u_K$ of L
4: let $U \in \Re^{N \times K}$ be the matrix containing the vectors $u_i, \ldots, u_K$ as columns
5: **for** $i = 1 \ldots N$ **do**
6: let $y_i \in \Re^K$ be the vector corresponding to the i-th row of U
7: **end for**
8: cluster the points $(y_i)_{i=1\ldots N}$ in $\Re^K$ with K-Means into clusters $C_1, \ldots, C_K$

Output: clusters $A_1, \ldots, A_K$ with $A_i = \{j | y_j \in C_j\}$

Algorithm 2.4 Normalized Symmetric Spectral Clustering [81]

Input: number of clusters K and similarity matrix $S \in \Re^{N \times N}$

1: construct a similarity graph using one of the aforementioned construction methods and let W be its weighted adjacency matrix
2: compute the normalized symmetric Laplacian L_{sym}
3: compute the first K eigenvectors $u_i, \ldots, u_K$ of L_{sym}
4: let $U \in \Re^{N \times K}$ be the matrix containing the vectors $u_i, \ldots, u_K$ as columns
5: **form the matrix** $T \in \Re^{N \times K}$ **from** U **by normalizing the rows to norm 1, that is set** $t_{ij} = u_{ij} / (\sum_k u_{ik}^2)^{1/2}$
6: **for** $i = 1 \ldots N$ **do**
7: let $y_i \in \Re^K$ be the vector corresponding to the i-th row of T
8: **end for**
9: cluster the points $(y_i)_{i=1\ldots N}$ in $\Re^K$ with K-Means into clusters $C_1, \ldots, C_K$

Output: clusters $A_1, \ldots, A_K$ with $A_i = \{j | y_j \in C_j\}$

The idea behind spectral clustering is that the change in representation from a similarity matrix to a graph and, ultimately, to eigenvectors enhances the cluster-properties in the data so that the clusters can be trivially detected in the new representation. Therefore, simple clustering algorithms such as K-Means can then be applied, as in Algorithms 2.3 and 2.4.

Chapter 3

Application Scenario

Biotechnology is the use of knowledge about biological processes to solve problems and create useful products. This field has emerged with the use of living organisms in food fermentation (e.g., bread, wine, yogurt, beer), but is currently applied in a broader sense: the use of living organisms or parts thereof in the production of goods and services. Biotechnology has shown wide applicability in Agribusiness, in the Food Industry and in Medicine. It is highly interdisciplinary, involving disciplines such as Genetics, Biochemistry, Cellular Biology, Chemical Engineering, Information Technology, and Robotics [54].

Some of the main challenges that must be faced in order to make progress in Biotechnology are:

- recognition of protein interaction networks and metabolic pathways;
- prediction of sites in protein surfaces that interact with other biomolecules to perform the protein's function;
- *in silico* identification of the functions of proteins that arise with each newly sequenced genome.

The basic steps to understand the wide spectrum of biological processes that occur in a living organism are genome sequencing, gene identification and functional annotation of the genomic products, each of which pose their own challenges. However, experimental determination of protein functions is likely the most challenging [87]. According to Brown et al. [11], protein functional characterization has become the rate-limiting step in putting biological information to practical use. Large-scale functional annotation efforts have focused on automated strategies, since more traditional methods are only efficient when used on small subsets of the available data.

A gene classification is indispensable for organizing the huge complexity of biological relationships [34]. Genes and proteins are generally classified in terms of families, subfamilies and superfamilies in a taxonomy according to different unstandardized criteria, but usually based on sequential and structural similarity.

The two fundamental conditions to life are the organism's capacity to replicate itself and its ability to catalyze reactions efficiently and selectively [59]. Enzymes, which are the application scenario of this thesis, are a particular category of highly specialized proteins that catalyze the chemical reactions involved in the metabolism of all living organisms and represent a significant fraction of the proteome [27]. The activity of a living system is controlled by enzymes, which are needed so that almost all cellular processes can occur at significant rates, since enzymes commonly accelerate a reaction by a factor of 10^5 to 10^{17}. Reactions necessary for digesting food, sending nerve signals or contracting muscles, for example, simply would not occur at useful rates without catalysis. This extraordinary catalytic power is much larger than that of synthetic or inorganic catalysts [59].

A cell's individuality is largely due to the unique set of enzymes that it is genetically programmed to produce. The study of enzymes is of great practical importance. Some diseases, for example, may be caused by excessive enzyme activity. Others, especially inheritable genetic disorders, may be related to the deficiency or absence of one or more enzymes. Also, the measurement of enzyme activity in blood plasma, erythrocytes or tissue samples is important in the diagnosis of certain illnesses [59]. Enzymes are important practical tools, not only in medicine but also in the chemical industry, food processing, and agriculture.

In this master's thesis we analyze how the use of diverse sources of domain knowledge affects the success rate of the clustering algorithms. In our application scenario of enzyme clustering, we consider three different problems, all of which aim at determining patterns responsible for functional differentiation:

- clustering enzyme families;
- clustering enzyme subfamilies inside multiple families; and
- clustering enzyme subfamilies inside a single family.

All are challenging problems. In the last case, we aim at the enzymes' substrate specificity (i.e., their ability to discriminate between a substrate and competing molecules). We consider that an enzyme family is a group of enzymes that catalyze the same overall reaction, and different subfamilies recognize different substrates as inputs for the reaction.

Predicting an enzyme's function involves predicting the reaction it catalyzes, its mechanisms, substrates and products. The understanding of molecular function may be greatly facilitated by information such as structural similarity. However, unfortunately there is still a small fraction of structures that have been experimentally resolved compared to the large number of available amino acid sequences. The functions of 20% of the protein families in the Pfam database [26], for instance, remain unknown. Nevertheless, there are computational methods that allow modeling proteins that have significant degrees of identity with proteins of known structure.

Relevant Biological Concepts

Some concepts relevant to the understanding of this master's thesis and of the domain knowledge data used are described in this section.

Homology. Two proteins are homologous if they have an evolutionary relationship, i.e., if they descend from a common ancestor protein.

Amino Acid Residues. When an amino acid is joined to another, they both lose elements of water so a covalent bond can form between them. Since part of the original chemical molecule that defines the amino acid is lost, the resulting molecule is called "amino acid residue" or simply "residue".

Protein Evolution. As discussed by Nelson and Cox [59], the residues which are essential for the activity of a given protein are conserved over evolutionary time. The less functionally important residues may vary over time, i.e., one amino acid may replace another, and the varying residues may provide information used to trace evolution. Sometimes such amino acid substitutions are not random. At some positions in the protein's amino acid sequence, the need to maintain function may imply that only given substitutions can be tolerated.

Genomic Context. Proteins are chains of amino acids coded by genes, with each residue being coded by a triplet of nucleotides called a codon. In turn, genes are segments of a chromosome that correspond to the information required to produce proteins [59]. The genomic context is the set of neighboring genes in a DNA strand that may imply functional proximity, since close genes are commonly co-expressed and involved in the same biological process. Therefore, proteins coded by genes in similar genomic contexts have higher probability of being involved in

similar functions, while proteins from the same family but whose encoding genes are in very different contexts probably present different substrate specificities [36]. Possible measures of similarity between the genomic contexts of a pair of genes are called synteny. In this master's thesis, genomic context forms an additional data source to be incorporated to the main sequence-based dataset.

Sequence Alignments. Alignments are frequently used to compare biological sequences. A global pairwise alignment may be thought of as the process of sliding one sequence past the other until a good match is found [59]. In case the residues in the two sequences are identical in a given position, a positive score is assigned to the match. The sum of the scores provides a measure of alignment quality. Gaps may be introduced to maximize score in the case when two segments match well between the sequences, but are separated by a different number of residues in each sequence. Penalties are applied when gaps are introduced so as to reduce the total score of the alignment.

Instead of analyzing whether or not the residues are identical, their chemical properties may be considered so that more conserved amino acid substitutions receive higher scores. Amino acid substitution matrices are used to determine what scores to assign to the many possible substitutions [59]. Figure 3.1 shows an example of a multiple sequence alignment of five amino acid sequences. In this master's thesis, multiple sequence alignments are used as attributes for the constrained clustering algorithms and to form the basic similarity matrices for the spectral clustering algorithms. Amino acid substitution matrices BLOSUM62 and PAM30 are utilized as similarity measures.

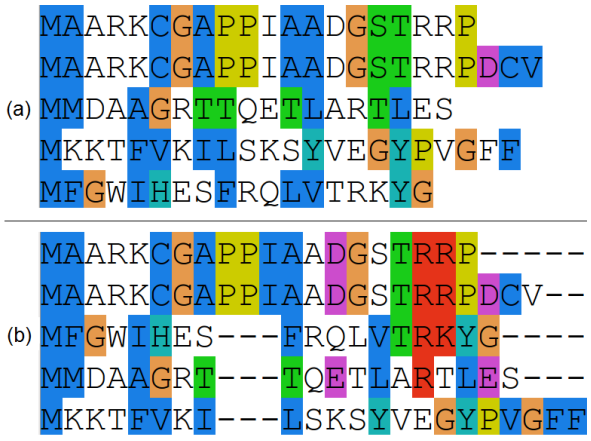

Figure 3.1. *Example of a multiple sequence alignment.* Part (*a*) shows five amino acid sequences of different lengths, while part (*b*) shows their multiple sequence alignment.

Protein Structure. The residues in a protein sequence interact with each other creating complex folding patterns that determine the protein's tertiary structure, which is directly related to its function. Different protein sequences may fold into similar three-dimensional structures, still maintaining function. Therefore, one can conclude that protein structures are more conserved than the corresponding sequences. Protein families with distinct folding patterns tend to have different functions. However, it is possible to find proteins with very dissimilar structures that have the same function (and vice versa) due to convergent evolution.

Simply put, Cutoff Scanning [74] is a method that represents three-dimensional structure as a histogram of the number of neighbors an atom presents within varying distances. Different families have distinct folding patterns and, consequently, characteristic histograms. Such histograms are used as attributes for the constrained clustering algorithms in this master's thesis.

Structural Alignment. Protein structural alignments are frequently used to detect functional similarity. Analogous to sequence alignments, the goal of a structural alignment is to find maximal protein substructures that can be superposed so as to maximize an objective score. A commonly used similarity measure is the coordinate distance-based Root Mean Square Deviation (RMSD), which measures the spatial Euclidean distance between superposed residues [67]. In this thesis, structural alignments are used as an additional data source to be incorporated to the main sequence-based dataset.

Enzyme Commission (EC) Numbers. EC numbers are a numerical hierarchical classification scheme for enzymes based on the chemical reactions they catalyze. This system for naming and classifying enzymes was adopted by international agreement because of ambiguities caused by previous naming systems and of the ever-increasing number of newly discovered enzymes [59]. As a system of enzyme nomenclature, every EC number is associated with a recommended name for the respective enzyme [55] and specifies enzyme-catalyzed reactions instead of enzymes themselves. Since the catalyzed reaction is the property that distinguishes enzymes from one another, it is logical to use it as the basis for enzyme classification and nomenclature.

The four levels that compose the EC number represent a progressively finer enzyme classification. The first level indicates the general class of the catalyzed reaction, the second and third levels depend on different criteria related to the chemical properties of the substrates and products of the reaction, while the

fourth level represents the substrate specificity [3]. The first level of EC number 3.4.11.4, for example, indicates the enzyme is a hydrolase, i.e., an enzyme that uses water to break up some other molecule. The second level (3.4) indicates that it is a hydrolase that acts on peptide bonds. The third level (3.4.11) indicates that it is a hydrolase that cleaves off the amino-terminal amino acid from a polypeptide, and the complete EC number indicates it is a hydrolase that cleaves off the amino-terminal end of a tripeptide.

A given enzyme's EC number can therefore be predicted at four levels, with the first being the easiest, since it can be done by detecting a remote homology. However, predicting the fourth level is extremely difficult. Schnoes et al. [68] estimate that 85% of the annotation errors are located in the lower level of the EC number. Rost [65] reports that less than 30% of enzyme-enzyme pairs above 50% sequence identity have entirely identical EC numbers, while Tian and Skolnick [77] report that pairwise sequence identity of at least 60% is required in order to transfer all four digits of an EC number with 90% accuracy. Our two approaches involving subfamily clustering focus on the problem of predicting the fourth level of the EC number.

Active Sites. An enzyme's active site is the set of residues that form a cavity in the enzyme's 3D structure where the substrate binds and the chemical reaction takes place. Chakrabarti and Panchenko [15] studied the co-evolution of residues in protein families and concluded that functionally important sites tend to be conserved, while specificity determining residues are correlated with mutations in certain positions, leading to functional diversification inside the family, thus creating subfamilies. In this work, active sites are used both as attributes (i.e., as a version of the main dataset) and as an additional data source to be incorporated to the sequence-based main dataset.

Since protein sequences are the target of different evolutionary pressures, certain positions are highly conserved while others seem to tolerate more alterations, insertions and deletions. Intuitively, multiple sequence alignments have been widely employed for detecting such conservations. A protein's function is much more related to its (more conserved) structure than to its sequence. However, few methods use structural data and none, to the best of our knowledge, use various forms of domain knowledge to cluster enzymes as in this master's thesis.

Chapter 4

Literary Review

This chapter presents the theoretical background for this thesis. Section 4.1 gathers some research on data integration applied to Bioinformatics, Section 4.2 presents research related to constrained clustering, Section 4.3 introduces research on spectral clustering, and Section 4.4 presents some research related to the application scenario.

4.1 Data Integration

Bioinformatics and genomics cover a wide range of different data formats and representations (e.g., sequences, structures, annotations, pathways) that are derived from experimental and in silico biological analysis and stored, used and manipulated by scientists and machines [71]. This huge volume of data is usually distributed in different locations, creating the need for tools that integrate the various knowledge domains, usually using complex information fusion processes.

Freier et al. [25] presented BioDataServer, a concept for user specific integration of life science data based on a mediator architecture in conjunction with freely adjustable data schemes. Rother et al. [66] developed COLUMBA, an integrated database of protein annotations which was centered around proteins whose structures had been resolved, and added as much annotations as possible to the proteins, describing their properties such as function, sequence, classification, textual description, and participation in pathways. The authors extracted annotations from seven external data sources without attempting to remove redundancies and overlaps among them. Instead, they viewed each data source as a proper dimension describing a protein.

Do and Rahm [22] presented the GenMapper system, which physically integrated heterogeneous annotation data in a flexible way and supported large-scale analysis of the integrated data. The system used a generic data model to uniformly represent different kinds of annotations originating from different data sources. Existing associations between objects were explicitly used to drive data integration and combine annotation knowledge from different sources.

Yamanishi et al. [91] presented a method to infer protein networks from multiple types of genomic data based on spectral clustering ideas and on a variant of kernel canonical correlation analysis. According to the authors, the originality was in the formalization of the protein network inference problem as a supervised learning problem, and in the integration of heterogeneous genomic data within this framework. Four types of widely available data were used: gene expressions, protein interactions measured by yeast two-hybrid systems, protein localizations in the cell, and protein phylogenetic profiles. The proposed method outperformed their unsupervised protein network inference algorithms. Later on, the authors [92] presented methodologies for inferring enzyme networks from the integration of multiple genomic data and chemical information in a supervised graph inference framework. The authors concluded that the prediction accuracy of the network reconstruction consistently improved because of the introduction of chemical constraints, the use of a supervised approach, and the weighted integration of multiple datasets.

The resource for distribution and query of three-dimensional structure data of the Research Collaboratory for Structural Bioinformatics (RCSB) was redesigned to expand the functionality of the previous website by integrating and providing searchability of data from over twenty other sources covering genomic, proteomic and disease relationships [21]. Wittig et al. [88] developed SABIO-RK, a curated database which contained and merged information about reactions such as reactants and modifiers, organism, tissue and cellular location, as well as the kinetic properties of the reactions. The data were manually extracted from literature and verified by curators, with concern for standards, formats and controlled vocabularies, with this process being supported by tools in a semi-automatic manner.

In summary, in recent years there has been an increasing concern from the scientific community about the need to integrate the various sources of biological knowledge in order to be able to use them for improving research results and to generally put such information to practical use.

4.2 Constrained Clustering

Initial research in the field of constrained clustering proposed algorithms capable of incorporating pairwise constraints on whether or not instances belong to clusters, and of learning distance metrics specific to the problem that lead to desirable clusterings. The field has expanded to include algorithms that use many other types of domain knowledge to aid the clustering process [7]. The first research initiative in the field proposed a modified version of COBWEB [24] that imposed strict pairwise constraints [82], followed by COP-KMeans [83], a constrained version of the well known K-Means clustering algorithm. Following the trend of adapting existing methods, Shental et al. [73] explored a constrained version of the Expectation Maximization (EM) algorithm.

To accommodate constraint noise or uncertainty, other methods consider flexible constraints, i.e., they aim at satisfying as many constraints as possible, but not necessarily the entire constraint set [6, 18, 84]. One approach treats constraints as statements about the true distance or similarity between instances. In this case, a must-link constraint, for example, implies that the instances involved should be close, whereas a cannot-link constraint implies they should be far enough apart so as to never be assigned to the same cluster. This distance may or may not be consistent with the distance implied by the feature space in which the instances reside. Such inconsistency may occur when some attributes are irrelevant or misleading with respect to the algorithm's objective function. Once the distance metric is learned, a regular unsupervised clustering algorithm may be applied to the data using the new metric.

Various methods of distance metric learning have been developed, some limited to learning only based on must-link constraints [5], while others may also accommodate cannot-link constraints [41, 89]. HMRF-Kmeans incorporates both constrained clustering approaches, as straightforward constraint satisfaction and distance metric learning are combined into a single probabilistic framework [6]. The results obtained by Xing et al. [89] suggested that combining distance metric learning with must-link constraint satisfaction leaded to better performance than simply learning the distance metric.

Davidson and Ravi [18, 20] also used constraints to specify interesting spatial properties. According to the authors, the constraint referred to as δ is used when a distance equal to or larger than δ must exist between instances of different clusters, which is equivalent to a conjunction of must-link constraints between all pairs of instances that are closer together than δ. Similarly, the constraint referred to as α, which limits the cluster diameters to at most α, is equivalent to a conjunction of cannot-link constraints between all pairs of instances more than a distance α apart. The constraint

the authors referred to as ϵ may be used when it is intended that every instance in a cluster have at least one neighbor within a distance ϵ, which could be replaced by a disjunction of must-link constraints [17].

Pensa et al. [62] presented interval constraints, which specify that a given cluster must include instances with values in a given range. This kind of constraint applies to attributes with real or otherwise sortable values. For hierarchical clustering, the constraint referred to as γ may be used when it is desirable that two clusters whose geometrical centroids are separated by a distance larger than γ cannot be joined [17].

Methods such as MPCK-Means [9] allow the user to specify individual weights for each constraint, thus treating the problem of varying confidence per constraint. MPCK-Means imposes a penalty for violating constraints proportional to the weight of the constraint that was violated.

A technique similar to constrained clustering in the sense that it also deals with different data sources is consensus or ensemble clustering. The idea of consensus clustering is to combine different clusterings into a single representative result which would bring out the common organization in the different datasets and reveal significant differences among them [28]. The main distinction between constrained and consensus clustering is that the first uses various possibly incomplete data sources as constraints to produce a single clustering, whereas the latter combines different clusterings into a single result.

Among the research that apply constrained clustering to biological scenarios, Zeng et al. [94] investigated the problem of clustering genes using gene expression data with additional information in the form of constraints generated from potentially diverse sources of biological information. The authors adapted MPCK-Means and explored methods of automatically generating constraints from multiple sources of data, investigating the effectiveness of different constraint sets and demonstrating that, when appropriate constraint sets are employed, constrained clustering yields more biologically significant clusters than those produced only using gene expression data.

Schönhuth et al. [69] used a flexible version of constrained clustering [45] to estimate a mixture model whose components were multivariate Gaussians with diagonal covariance matrices representing courses of time of gene expression. The secondary data consisted of occurrences of transcription binding sites in yeast genes. Lastly, Sese et al. [70] presented a constrained itemset clustering technique that computed the optimal cluster which maximized interclass gene expression variance between clusters based on the restriction that only divisions expressed using common features were allowed.

4.3 Spectral Clustering

Liu et al. [49] studied the ability of HIV-1 protease to develop mutations that conferred multi-drug resistance, which had been a major obstacle in designing therapies against HIV. The authors used spectral clustering on the covariance matrices resulting from sequence covariance analysis of HIV-1 protease sequences from patients subjected to different specific treatments and from untreated patients. The results of the study disclosed two distinctive clusters or correlated residues, demonstrating the possibility of distinguishing between the correlated substitutions with appropriate clustering analysis of sequence covariance data, and a connection between global dynamics and functional substitution of amino acids.

Perkins and Langston [63] applied spectral graph theory methods to develop a systematic method for gene expression threshold selection. The authors used a basic spectral clustering method to examine the set of gene-gene relationships and select a threshold dependent on the community structure of the data.

Li et al. [47] argued that a natural and promising approach for gene annotation was to integrate gene microarray expressions and sequences, especially in terms of their costs to be optimized in clustering. The authors developed an efficient gene annotation method with three steps containing spectral clustering over the integrated cost, based on the idea of network modularity. According to the authors, the results indicated an advantage in performance of their method over possible clustering or classification-based gene function annotation approaches using expressions and/or sequences.

4.4 Protein Clustering and Function Prediction

Large-scale protein functional annotation efforts have focused on automated strategies, since more traditional methods can only be used efficiently on small subsets of the available data. According to Brown et al. [11], automated analysis strategies have inherent and serious limitations, such as the fact that it has been shown that simple pairwise sequence comparisons are inadequate for functional classification of proteins with less than 30% to 40% identity. Thus, methods for automatic annotation must be able to also use information from domains other than sequence data.

Active site and binding site detection based on sequence and/or structure data are very active research fields [4, 8, 13, 29, 30, 35, 46, 58, 60, 64, 75]. In cases when a protein of unknown function may be linked to a superfamily of multiple foldings

or functions, determining possible subfamilies might lead to important information about the protein's function and structure, as well as functional diversification acquired throughout family evolution.

Livingstone and Barton [50] used multiple sequence alignments and physicochemical properties of residues to analyze conservations per group. Casari et al. [13] proposed a sequence representation in the form of vectors and used dimensionality reduction techniques to project such vectors in fewer dimensions and detect subgroups and functionally important residues. Lichtarge and collaborators [43, 44, 48, 52, 86] proposed Evolutionary Trace for predicting active sites and functional interfaces in proteins of known structure. The authors manually determined the clusterings based on sequence and structure similarity, and searched for conserved residues in each clustering as well as in the whole family. Afterwards, they mapped these residues to the three-dimensional structure and used them to search databases for proteins of similar function.

Hannenhalli and Russell [33] explored different hidden Markov models given multiple sequence alignments of protein subfamilies to detect positions which created subdivisions. A great disadvantage of such methods is the need to know the subfamilies a priori, which is a complicated task to be done manually because of the lack of experimental information on subfamilies. Furthermore, this is an extremely limiting factor when working with families of unknown function.

Sol et al. [75] presented a set of methods based on phylogenetic trees aiming at discovering which residues were important for tree ramifications. Later, the same group of researchers published another method, this time phylogeny-independent [60]: Xdet, a supervised method for detecting functional sites in conserved positions in multiple sequence alignments. Pei et al. [61] presented a method also based on phylogenetic trees and computed the evolutionary log-likelihood using a random model to compare and estimate the statistical significance of the predictions. SPEER [14] was another method that proposed dividing a family into subfamilies based on alignments and on the amino acid conservation patterns and physicochemical properties.

Capra and Singh [12] described a methodology for identifying the specificity determining positions that consisted in constructing multiple sequence alignments for protein families, selecting the residues which were structurally close to ligands, and filtering by the conserved residues to detect different subfamilies.

Various authors [43, 44, 48, 52, 86] mapped sequence evolutionary patterns into structures to search for proteins of possibly similar functions. Yu et al. [93] proposed a method that, given a multiple sequence alignment, a subfamily classification based

on Swiss-Prot [10] and a representative structure, determined surface residues that discerned groups.

Given that the structure tends to be more conserved than the sequence and that the number of different foldings is limited [16], in many cases it is possible to identify a homologous protein with known structure to serve as a model for a target sequence. Such methods have proven to be increasingly reliable [56, 78, 79].

Lichtarge et al. [48] demonstrated that mutations occur more frequently in surface residues. Residues closer to the protein's core tend to have a structural role, while functionally important residues commonly tend to occur in the protein's surface cavities [23]. Among the research related to function prediction, Kitson et al. [40] proposed a pipeline for functional annotation that used PSI-BLAST [1] and extracted information about the quality of the models built by homology. Tseng et al. [80] used a combination between cavity residues and evolutionary data in a small set of homologous proteins to compose a geometrical pattern that may be used in the search for proteins of similar functions.

Freilich et al. [26] predicted Enzyme Commission numbers using only sequence information. Other authors use structural or substrate-product information, such as Kotera et al. [42], who reported a method that, given pairs of substrates and products, attributes EC numbers up to the third level. Minardi et al. [53] proposed and implemented a methodology for determining enzyme subfamilies based on the residue composition of the active sites of the family's homologous enzymes. Hierarchical clustering [24] was used in a dataset that combined information about structural cavities and the evolution of the family's set of sequences, aiming at determining residue composition patterns responsible for functional differentiations in a family.

Tian et al. [76] presented EFICAz, an application for enzyme function inference that combined different methods based on family-dependent sequential similarity cutoffs, on the presence of patterns of functionally relevant domains and on the identification of discriminant residues. Afterwards, the authors added improvements to the method [2, 3]. Yamanishi et al. [90] proposed a method to predict potential EC numbers for substrate-product pairs or uncharacterized reactions, focusing on predicting the three first levels of the EC number.

The results of this research indicate that methods for predicting active sites or protein functions based on structure and other types of domain knowledge are more appropriate than those based only on sequence data, and that there exist constraints to be applied to the problem of clustering enzymes by specificity.

Chapter 5

Methodology

This chapter describes the methodology applied in this master's thesis. The main steps are outlined below.

1. *Database preparation.* Construction of a main database containing the amino acid sequences, as well as the gathering of additional data sources comprising information regarding three-dimensional structures, active sites and genomic contexts of enzyme families that include subfamilies of different substrate specificities;

2. *Data preprocessing.* Execution of sequence and structural alignments, along with the study of methods for calculating genomic context synteny;

3. *Algorithm design.* Definition of the constraints and of the constrained and spectral clustering methods more appropriate for the problem at hand;

4. *Implementation and application of constrained and spectral clustering algorithms.* Development of constrained versions of classic clustering methods and application of such algorithms to the problem of enzyme clustering;

5. *Evaluation.* Analysis of the results of applying constrained clustering to the application scenario and comparison with unconstrained versions of the algorithms. Furthermore, analysis of the results of employing spectral clustering with the additional data sources and comparison with the results of using simply the main sequence-based database.

5.1 Data Sources

Three enzyme families are studied in this master's thesis, namely nucleotidyl cyclases, protein kinases and serine proteases. Each family has a different number of subfamilies:

Nucleotidyl cyclases: adenylate cyclases and guanylate cyclases;

Protein kinases: serine/threonine kinases and tyrosine kinases, which will be referred to as *serthrkinases* and *tyrkinases*, respectively, in the remainder of the text;

Serine proteases: chymotrypsins, elastases, kallikreins and trypsins.

5.1.1 Main Dataset

The main database consists of the enzymes with known Enzyme Commission (EC) numbers. The information provided by Moss [55] was used to define the EC numbers for each of the three enzyme families, as shown in Table 5.1. In order to achieve a more reliable dataset to use as ground truth for analyzing the clustering results, we are considering only the enzymes that had a reviewed status in the Universal Protein Resources (UniProt) repository [37] at the time of the data collection, i.e., the enzymes that had been manually annotated and reviewed. This is due to the fact that automatic annotation methods might introduce annotation errors, which would jeopardize the analysis of the results.

Table 5.2 shows the number of enzymes in each subfamily after applying the aforementioned filtering process, which yields 56 nucleotidyl cyclases, 83 protein kinases and 102 serine proteases, totalizing 241 enzymes.

Table 5.1. Enzyme Commission numbers according to Moss [55].

Family	Subfamily	EC Number(s)
Nucleotidyl Cyclases	Adenylates	4.6.1.1
	Guanylates	4.6.1.2
Protein Kinases	*Serthrkinases*	2.7.11.1
	Tyrkinases	2.7.10.{1, 2}
Serine Proteases	Chymotrypsins	3.4.21.1
	Elastases	3.4.21.{36, 37, 71}
	Kallikreins	3.4.21.{34, 35, 118}
	Trypsins	3.4.21.4

Table 5.2. Number of enzymes in each subfamily.

Family	Subfamily	Enzymes
Nucleotidyl Cyclases	Adenylates	4
	Guanylates	52
Protein Kinases	*Serthrkinases*	73
	Tyrkinases	10
Serine Proteases	Chymotrypsins	4
	Elastases	13
	Kallikreins	21
	Trypsins	64

The resulting databases comprise each enzyme's UniProt identification, EC number and amino acid sequence. If different enzymes catalyze the same reaction, they receive the same EC number. UniProt identifiers, on the other hand, uniquely specify a protein by its amino acid sequence.

Given the subfamily labels derived from the EC numbers, the goal is to analyze how information from different data sources is able to aid an unsupervised clustering process to discover the actual subfamilies as determined by the labels.

5.1.2 Additional Data Sources

Besides the amino acid sequences, additional information on each enzyme in the database was gathered. The first two external sources of data are the tertiary structure model and the active site of each enzyme, provided by Minardi et al. [53]. The aligned active sites were obtained using Fpocket [31], a software that calculates structural cavities, and MultiProt [72], a software that superpositions structures. Active site residues belong to the enzyme family's most conserved structural cavities.

The third external data source comprises the genomic contexts for some of the enzymes. To obtain this information, the complete genomes of several organisms were downloaded from NCBI Entrez Genome[1]. Then, a mapping from GeneID to UniProt ID was performed, which was necessary since our databases use UniProt identifiers, while the genomes only present GeneIDs. To build the genomic context for the enzymes present in the genomes, a five-gene window was used. Thus, we consider that the genomic context of a given enzyme is simply an array containing the five proteins that come before it and the five that follow it in the genome, resulting in a total of

[1]Available at `http://www.ncbi.nlm.nih.gov/sites/genome`

ten proteins besides the enzyme itself. Unfortunately, we were unable to obtain the genomic context for all enzymes in our dataset.

The last additional data source is the set of vectors produced by applying Cutoff Scanning [74] to the structural model of each enzyme. This consists of calculating the Euclidean distance in angstroms between all pairs of amino acid residues in the enzyme's 3D structure. Then, the number of pairs whose distance to each other is less than a given cutoff is calculated. When this cutoff is varied, a vector is created so that each position in an enzyme's vector denotes the number of residue pairs within a given distance of each other. Therefore, each enzyme has a vector that represents its folding, i.e., the manner in which the residues are positioned in its 3D structure. Such vectors comprise important information that is complementary to the amino acid sequences.

5.2 Generating Constraints

This section describes in detail the methods used to create constraints based on each of the additional data sources. Apart from clustering each of the enzyme families separately in the search for the subfamilies, we also cluster all of them combined, either seeking the eight subfamilies or the three families.

5.2.1 Structural Alignment Based-Constraints

MultiProt [72] was used to perform pairwise structural alignments between the tertiary structure models of the enzymes in all three families. Because it is a heuristic method and searches not only for global alignments but for local alignments as well, MultiProt outputs more than one alignment. The Root Mean Square Deviation (RMSD) of the first result reported by MultiProt, which corresponds to the largest alignment found, is used to analyze the structural similarity between the pair of enzymes. Since the RMSD for aligning enzyme A to enzyme B may differ from that of aligning enzyme B to enzyme A, the average of the two results is used as the structural similarity score for the enzyme pair. The smaller the RMSD, the better the alignment and the more similar the enzymes are.

The RMSDs for each pair of enzymes in each family, as well as in all three families combined, were analyzed in the search for cutoffs that could be used to generate pairwise constraints. Since strict constraints (i.e., constraints that *must* be satisfied) are being considered in this thesis, this search was for the cutoffs that did not yield any

false positives, i.e., the RMSD thresholds that would not lead to must-link constraints between enzymes from different subfamilies or cannot-link constraints between enzymes from the same subfamily.

In order to generate must-link constraints, the following must hold: if the RMSD of the alignments between an enzyme pair is *at most* X, then the enzymes belong to the same subfamily. Therefore, a must-link constraint exists between them, placing both enzymes in the same cluster. Similarly, for cannot-link constraints it must hold that if the RMSD of the alignments between a pair of enzymes is *at least* Y, then they belong to different subfamilies and a cannot-link constraint exists between them, causing the enzymes to be placed in different clusters.

Table 5.3 presents each family's zero false positive RMSD thresholds for creating must-link (ML) and cannot-link (CL) constraints. The must-link cutoff is the largest RMSD value between a pair of enzymes from the same subfamily, while the cannot-link cutoff is the smallest value between pairs of enzymes belonging to different subfamilies.

Table 5.3. RMSD cutoffs in angstroms (Å) that yield zero false positive constraints for each family.

Family		**ML**	**CL**
Nucleotidyl Cyclases		0.42	1.08
Protein Kinases		0.76	1.50
Serine Proteases	Chymotrypsins	0.58	0.32
	Elastases	0.61	0.30
	Kallikreins	0.25	0.74
	Trypsins	0.25	0.86
All Three Families		0.25	1.50

In order to create constraints in a more general fashion, we employ cutoffs that are valid for all three families as well as for the original dataset, which also includes unreviewed enzymes. Therefore, we use thresholds $RMSD \leq 0.25$Å for creating must-link constraints and $RMSD \geq 1.62$Å for cannot-link constraints. Table 5.4 presents the number of pairwise constraints generated for each enzyme family before and after expanding the constraint set using the transitivity property described in Section 2.2.1.

Unfortunately, these cutoffs do not yield any structural alignment-based must-link constraints for the protein kinases family, nor cannot-link constraints for any of the three families. This can be explained by the fact that families and subfamilies are determined by function, and there exist proteins with very different structures and, consequently, whose alignments have high RMSD values, that still present the same

Table 5.4. Number of structural alignment-based constraints created for each family before and after transitively expanding the constraint set.

Family	Before		After	
	ML	CL	ML	CL
Nucleotidyl Cyclases	6	0	6	0
Protein Kinases	0	0	0	0
Serine Proteases	347	0	390	0
All Three Families	353	18,822	396	18,824

function. The opposite may also occur. Therefore, the cannot-link cutoff was set at a high RMSD level so that it would not lead to cannot-link constraints between enzymes annotated as being in the same subfamily.

The cannot-link cutoff could have been lowered so as to create constraints inside each family, but that would also go against the idea of generality because if a lower cutoff had been applied to the original set of enzymes, which includes automatically annotated enzymes, we would have ended up creating cannot-link constraints between enzymes annotated as belonging to the same subfamily. This would have inserted contradiction into the constraint set, making the strict constraint approach intractable, as it would be impossible to satisfy all constraints. When considering all three families combined, however, several cannot-link constraints are created between pairs of enzymes belonging to different families and, consequently, to different subfamilies.

5.2.2 Genomic Context-Based Constraints

After obtaining genomic contexts for some enzymes of each family as previously described, we observed that if two enzymes have at least one protein in common between their genomic contexts, it always happens that they belong to the same subfamily. Therefore, must-link constraints are created between each pair of enzymes whose genomic contexts have at least one protein in common. Table 5.5 shows the number of pairwise constraints created for each family before and after expanding the constraint sets with the transitivity property.

Very few genomic context-based must-link constraints are created due to the fact that most of the enzymes in our dataset do not appear in the genomes we had access to. Nevertheless, additional constraints might be created using the transitivity property when combined with other constraint sets.

Table 5.5. Number of genomic context-based must-link constraints created for each family before and after transitively expanding the constraint set.

Family	Before	After
Nucleotidyl Cyclases	3	3
Protein Kinases	3	3
Serine Proteases	50	57
All Three Families	56	63

5.2.3 Active Site-Based Constraints

Since the active sites of all enzymes in a given family are aligned, they all have the same number of residues. That said, two strategies are employed to create active site-based constraints: must-link constraints are created between all pairs of enzymes with 100% identical active sites, and between all pairs with active sites at least 95% identical.

We considered the strategy of creating cannot-link constraints between all pairs of enzymes with less than a given percentage of active site identity. A 30% cutoff, for example, would create some constraints for the protein kinases family. However, if we were to apply the same strategy to the original database, which also contains automatically annotated enzymes, cannot-link constraints would have been created between enzymes annotated as being in the same subfamily, thus introducing contradiction to the constraint set and making the problem intractable. Therefore, only must-link constraints are created using active site information.

Because the number of residues in the active sites slightly varies between families, Clustal X [39] was used to perform multiple sequence alignment of the active sites so that the same strategies could be applied when clustering the three families altogether. Table 5.6 presents the number of pairwise constraints created in each case for each family, before and after expanding the constraint sets with the transitivity property.

Table 5.6. Number of active site-based must-link constraints created for each family before and after transitively expanding the constraint sets.

Family	Before		After	
	100%	95%	100%	95%
Nucleotidyl Cyclases	88	249	88	434
Protein Kinases	10	12	10	12
Serine Proteases	50	158	50	296
All Three Families	148	419	148	742

5.3 Constrained Clustering Algorithms

In order to analyze the effect of integrating multiple data sources into the clustering process as constraints, we study constrained and unconstrained versions of well known clustering algorithms K-Medoids and K-Means. The unconstrained versions are implemented as described by Han and Kamber [32], while the constrained versions are adapted from COP-KMeans [83].

5.3.1 Constrained K-Medoids

Constrained and unconstrained versions of K-Medoids have been implemented because the main data source, which consists of the amino acid sequences, is categorical. Clustering algorithms require that all objects have the same attributes in order for the difference between each attribute of an object pair to have a meaning, so the distance or similarity between them can be calculated. Since the lengths of the sequences vary, multiple sequence alignments were performed between all sequences in each family using Clustal X [39], and the results are used as the enzymes' attributes for K-Medoids. A multiple sequence alignment was also performed among the enzymes from all three families in order to apply K-Medoids to the approaches that involve clustering families and clustering subfamilies inside multiple families.

Using multiple sequence alignments as attributes is a straightforward approach, since sequence information is much more readily available than structural data. Regardless, we also test using the active sites as attributes instead. This way we are able to assess how the use of structural information as the main dataset to which constraints are added compares to using the more common sequence information.

Three different similarity measures are used for these categorical attributes: BLOSUM62 and PAM30, which are amino acid substitution matrices commonly used in protein sequence alignments, and the complement of the Hamming distance, which is simply the number of identical residues in the sequences, excluding gaps.

The classic (unconstrained) K-Medoids clustering algorithm was previously described in Chapter 2 and detailed in Algorithm 2.2. Algorithm 5.1 illustrates the constrained K-Medoids algorithm implemented in this thesis. In line 7, a nonrepresentative object o_{nr} can replace a representative o_r if there are no must-link constraints between o_{nr} and any object belonging to a different cluster, as well as no cannot-link constraints between o_{nr} and any object in the cluster represented by o_r.

Algorithm 5.1 Constrained K-Medoids Algorithm

Input: number of clusters K, dataset D containing N objects and constraint set C

1: expand constraint set C using the transitivity property
2: build initial representative set R arbitrarily choosing K objects from D such that must-link constraints do not exist between representatives
3: build initial nonrepresentative set NR with the remaining objects
4: run *constrainedClustering(R, NR, C)* to obtain total similarity S of the clustering
5: **repeat**
6: **for** each representative $o_r \in R$ and each nonrepresentative $o_{nr} \in NR$ **do**
7: **if** o_{nr} can replace o_r **then**
8: switch o_r and o_{nr} and run *constrainedClustering(R, NR, C)* to obtain the total similarity S' of the new clustering
9: **if** $S' > S$ **then**
10: the new clustering is better, so it replaces the current one, with $S \Leftarrow S'$
11: **end if**
12: **end if**
13: **end for**
14: **until** the total similarity S converges

Output: a set of K clusters

constrainedClustering(R, NR, C)

1: **for** each representative $o_r \in R$ **do**
2: **if** a nonrepresentative $o_{nr} \in NR$ has a must-link constraint with o_r **then**
3: assign o_{nr} to the cluster represented by o_r
4: **end if**
5: **end for**
6:
7: **for** each remaining nonrepresentative $o_{nr} \in NR$ **do**
8: **if** o_{nr} has a must-link constraint with any already clustered object o_i **then**
9: assign o_{nr} to the same cluster as o_i
10: **else**
11: assign o_{nr} to the cluster with the nearest representative o_r for which a cannot-link constraint does not exist between o_{nr} and the objects already in the cluster
12: **end if**
13: **end for**
14: **return** the total similarity S of the clustering

5.3.2 Constrained K-Means

K-Means has been implemented so as to take advantage of the information provided by the previously described Cutoff Scanning [74] vectors, which are created using a step of 0.2Å varying from 0Å to 30Å. Therefore, each vector has 151 positions, with the first element being the number of residue pairs in the enzyme's three-dimensional structure whose alpha-carbons are virtually in the same position (0Å distance), the second element being the number of pairs whose alpha-carbons are at most 0.2Å apart, and so on. The vectors are either normalized or not.

Squared Euclidean distance is used as the distance measure for these numerical attributes. The classic K-Means clustering method was detailed by Algorithm 2.1 in Chapter 2. The constrained version of the algorithm implemented in this thesis is illustrated in Algorithm 5.2.

Algorithm 5.2 Constrained K-Means Algorithm

Input: number of clusters K, dataset D containing N objects and constraint set C

1: expand constraint set C using the transitivity property
2: build initial cluster center set A arbitrarily choosing K objects from D such that must-link constraints do not exist between them and they are preferably not involved in cannot-link constraints
3: **repeat**
4: **for** each object $o \in D$ **do**
5: find the objects with which o has a must-link constraint, forming subset ML
6: assign all objects $o_{ML} \in ML$ to the cluster whose current objects do not have cannot-link constraints with ML and whose center is closest to ML (using the sum of the distances from the center to o_{ML})
7: **end for**
8: update the cluster centers by calculating the mean value of the objects in each cluster
9: **until** the clustering converges

Output: a set of K clusters

5.4 Spectral Clustering

This section elucidates all the decisions that were made for each step necessary to perform spectral clustering, as outlined in Section 2.3.

5.4.1 Similarity Matrices

The same data used by the constrained clustering algorithms are employed to build the similarity matrices for spectral clustering. Several similarity matrices are adopted, involving all data sources previously used for creating constraints and their combinations. The initial or basic similarity matrices, to which the additional data sources are integrated, consist of the similarity scores between pairs of aligned amino acid sequences (the same used as attributes by K-Medoids) calculated using the same three similarity measures: BLOSUM62 and PAM30 amino acid substitution matrices, and the complement of the Hamming distance. Each of these three initial matrices (one for each similarity measure) is linearly normalized for each enzyme family. The matrix diagonal, which includes the similarities of each enzyme with itself, are not included in this normalization in order to prevent that such high similarities push the other normalized similarities to zero.

Three strategies are used for integrating the additional datasets containing information on structural alignments, genomic contexts, and active sites:

1. **Using the same cutoffs applied for creating constraints as described in Section 5.2.** In this case we add 1 to the similarity between i and j if there exists a must-link constraint between them, and subtract 1 from their similarity in case a cannot-link constraint exists. Example: consider similarity $s_{ij} = 0.6$ according to the initial BLOSUM62 similarity matrix. If the RMSD of the structural alignment between i and j is less than or equal to the established RMSD cutoff for creating must-link constraints ($RMSD \leq 0.25$), then $s_{ij} = 1.6$ in the modified similarity matrix. Similarly, if a cannot-link constraint existed between the pair, the new similarity would be $s_{ij} = -0.4$. However, negative values are truncated to zero since the similarity matrix must be non-negative.

2. **Using the actual values from the additional datasets.** In this case, the percentage of identical amino acids between the active sites and/or the number of different proteins in common between the genomic contexts are added to the initial similarity. The normalized RMSD of the structural alignment between the pair is subtracted from the similarity, since the larger the RMSD, the less similar the enzymes are. Again, the new similarity is truncated to zero in case the subtraction produces a negative value.

3. **Enforcing edges after the graph is constructed.** In this case, if there is a cannot-link constraint between a pair of enzymes according to the previously

described cutoffs, the similarity between them is set to zero, guaranteeing that an edge does not exist between them in the graph. Similarly, if there is a must-link constraint between them but their similarity has been set to zero after applying the graph construction method, the similarity is reset to the value in the initial similarity matrices (i.e., those with purely the normalized BLOSUM62, PAM30, and Hamming scores). This makes no difference when using the fully connected graph, but might create additional edges for the other graph construction methods, guaranteeing that an edge exists between a pair of enzymes for which a must-link constraint exists.

For each enzyme family, all three initial similarity matrices are studied, as well as all combinations of additional data sources for all three integration strategies. This yields 63 similarity matrices for each family and 87 matrices when all three families are combined, in which case structural alignment-based cannot-link constraints exist.

5.4.2 Graph Construction Methods

As described in Section 2.3, four graph construction methods are used in this thesis: the fully connected graph, K-nearest neighbors (KNN), mutual KNN and ϵ-neighborhood. The fully connected graph is simply the similarity matrix with negative values truncated to zero. The other methods, however, require parameters K and ϵ to be defined.

Three parameter values are used for the ϵ-neighborhood graph construction method. To determine the ϵ values for each enzyme family, the average of each column of the initial similarity matrices (i.e., purely the normalized BLOSUM62, PAM30, and Hamming scores) is calculated. The diagonal of the matrix (i.e., the similarity of an enzyme with itself) is excluded when calculating this average, so that it does not influence the choice of ϵ values . Afterwards, the averages are sorted and the quartiles are used as the parameter values. Therefore, the three ϵ values are the median of the averages, which splits the sorted average set into two subsets, and the median of each subset. Thus, the ϵ values are determined such that 25% (first quartile), 50% (second quartile), and 75% (third quartile) of the average similarities are below ϵ. There are three ϵ values for each family and each of the initial similarity matrices, as shown in Table 5.7.

A similar approach is used for defining K for KNN and mutual KNN: the values of K are determined as the ceilings of 25%, 50%, and 75% of the total number of neighbors, i.e., the total number of enzymes in the family minus one. Therefore, there are three values of K for each family. It may happen that there are multiple candidates

Table 5.7. ϵ values for the ϵ-neighborhood graph construction method.

Enzyme Family	Matrix	25%	50%	75%
Nucleotidyl cyclases	BLOSUM62	0.28819	0.32216	0.34808
	Hamming	0.33636	0.36493	0.39924
	PAM	0.38032	0.40627	0.41788
Protein kinases	BLOSUM62	0.55463	0.61767	0.62875
	Hamming	0.69899	0.78401	0.79662
	PAM	0.66180	0.72643	0.73690
Serine proteases	BLOSUM62	0.41872	0.43227	0.44709
	Hamming	0.64547	0.66020	0.69676
	PAM	0.57139	0.58705	0.59929
All 3 Families	BLOSUM	0.57385	0.62411	0.62781
	Hamming	0.71971	0.79222	0.81868
	PAM	0.67612	0.73055	0.73603

for the Kth neighbor of vertex i, since more than one vertex may be at the same distance from v_i. In this case, all the candidates at the Kth smallest distance from vertex i are added as its neighbors, resulting in more neighbors than the specified value of K. Table 5.8 shows the values of K used for each family for both KNN and mutual KNN.

Table 5.8. K values for KNN and mutual KNN graph construction methods.

Enzyme Family	Total Neighbors	25%	50%	75%
Nucleotidyl cyclases	55	14	28	42
Protein kinases	82	21	41	62
Serine proteases	101	26	51	76
All 3 Families	240	60	120	180

5.4.3 Graph Laplacian Matrices

As stated in Section 2.3, two spectral clustering algorithms are applied, one considering the unnormalized graph Laplacian matrix and the other using the normalized symmetric graph Laplacian matrix. The Laplacian matrices are calculated as described in that section, and Matlab R2010a is used to calculate the eigenvalues and eigenvectors of the Laplacians. The matrix built from using the K first eigenvectors as columns, where K is the desired number of clusters, is the input for unconstrained K-Means.

5.5 Parameter Settings

As previously described, the attributes (i.e., feature vectors) used by K-Medoids are either the enzymes' amino acid sequences or their active sites, with multiple sequence alignments in both cases. The complement of the Hamming distance and amino acid substitution matrices BLOSUM62 and PAM30 are used by K-Medoids as similarity measures, and by spectral clustering to build the initial similarity matrices to which the additional data sources are integrated. For K-Means, the attributes are the 151-dimension distance vectors produced by Cutoff Scanning, either normalized or not, and squared Euclidean distance is used as the algorithm's distance metric.

For all algorithms, when clustering each family separately, the number of clusters K is the actual number of subfamilies in each family, namely $K = 2$ for nucleotidyl cyclases and protein kinases, and $K = 4$ for serine proteases. This corresponds to the approach of clustering subfamilies inside a single enzyme family. When clustering all three families combined, we analyze results for $K = 8$, which is the total number of subfamilies and corresponds to clustering subfamilies inside multiple families, as well as the results for $K = 3$, so as to verify the performance of the algorithms at separating the three enzyme families, which is the third and last approach of this thesis.

When applying constrained and spectral clustering to the application scenario, all additional data sources and combinations thereof are studied. Furthermore, many parameter configurations exist for each algorithm, as depicted below.

- **Constrained K-Medoids:** the factors to be considered are the constraint set being applied, the similarity measure (BLOSUM62, PAM30 or the Hamming distance), and what information is employed as attributes for the enzymes (either amino acid sequences or active sites, both aligned);
- **Constrained K-Means:** the parameters are the constraint set being applied and whether or not the attributes (i.e., the distance vectors) are normalized;
- **Spectral Clustering:** the four parameters to be considered are the similarity measure (same options as K-Medoids), the graph construction method (fully connected, ϵ-neighborhood, KNN or mutual KNN), the type of graph Laplacian (unnormalized or normalized symmetric); and the similarity matrix given as input to the algorithms.

All possible parameter configurations are studied for each algorithm and for each enzyme family. The results are discussed in Chapter 6.

5.6 Evaluation Criteria

Different groups of clustering validation indexes exist, namely external, internal and relative validation. Simply put, external validation employs criteria that are not inherent to the dataset such as a priori knowledge about the data; internal validation employs criteria derived from the dataset such as intra and intercluster distances, while relative validation compares clusterings generated by different algorithms or by different parameter configurations for the same algorithm. In this thesis, the clustering results are being compared to the ground truth provided by the Enzyme Commission (EC) numbers. Therefore, external validation is used to analyze the clusters.

External criteria measure whether or not the objects being clustered are randomly structured. Among the measures commonly used when applying external validation, in this work three measures are employed in order to compare the results of the various settings, all of which are based on the purity of the clusters in comparison to the actual partitions (i.e., the true (sub)families as determined by the EC numbers): purity, entropy and the ratio between them, which we call EP-Ratio. The probability ρ_{ij} that cluster c_i contains objects from partition p_j is given by Equation 5.1, where $|c_i|$ is the number of objects (i.e., enzymes) in cluster c_i.

$$\rho_{ij} = \frac{|c_i \cap p_j|}{|c_i|} \tag{5.1}$$

Purity P_i of cluster c_i is the maximum value of ρ_{ij}, whereas the purity of the whole clustering is the weighted sum of the purity of each cluster as shown in Equation 5.2, where $|c|$ is the total number of objects. Purity quantifies to what extent a cluster contains entities from a given partition. Therefore, the larger the purity, the better the clustering. However, since maximum purity would be achieved if each object was put in its own cluster, extra measures are required to analyze the results.

$$P_C = \sum \frac{|c_i|}{|c|} P_i \tag{5.2}$$

Entropy is a popular supervised learning metric that measures the amount of uncertainty associated with a cluster assignment. It is based on the probability that cluster c_i contains objects from partition p_j. The smaller the entropy, the better the clustering, with a perfect clustering having zero entropy. Each cluster's entropy E_i is given by Equation 5.3, while the entropy for the entire clustering is the weighted sum of the individual cluster entropies, as shown in Equation 5.4.

$$E_i = -\sum \rho_{ij} \log_2 \rho_{ij} \tag{5.3}$$

$$E_C = \sum \frac{|c_i|}{|c|} E_i \tag{5.4}$$

The EP-Ratio, which is the main measure we use to analyze the quality of the clustering results, is simply the ratio between entropy and purity. Therefore, the closest the EP-Ratio is to zero, the better the clustering.

5.7 Assessment Methodologies

This section describes the two methodologies employed in this thesis to analyze the obtained clustering results.

5.7.1 Comparison of Paired Observations

In order to analyze the effect of each constraint set and combinations thereof, the EP-Ratios for thirty replications of the constrained clustering algorithms are compared against the EP-Ratios of the clusterings obtained by the unconstrained algorithms. In order to do so, paired observations are employed in a straightforward analysis: the two sets of EP-Ratios (one for each version of the algorithm) are treated as a sample of thirty pairs, each corresponding to the same random seed in each algorithm.

The difference between the values in each EP-Ratio pair and the confidence interval of such differences are calculated. In case the confidence interval includes zero, the quality of the result obtained by the constrained algorithm is not significantly different from the clustering found by the unconstrained version of the algorithm at the given level of confidence [38]. In this thesis we consider 99% and 95% confidence intervals.

5.7.2 General Full Factorial Designs

The goal of an experimental design is to obtain the maximum information with the minimum number of experiments. A proper experiment analysis helps single out the effects of various factors that might affect the performance, as well as allows to determine if a given factor has a significant effect or if the difference in results is simply due to random variations caused by measurement errors or uncontrolled parameters [38].

Details on general full factorial designs may be found in the work of Jain [38]. Simply put, the model for a K-factor full factorial design contains $2^K - 1$ effects: it estimates the effect that each factor has on the response variable (i.e., the clustering's EP-Ratio) as well as the effect of each interaction between factors.

The model for three factors A, B and C with a, b and c levels (i.e., values), respectively, and r replications is given by Equation 5.5, where y_{ijkl} is the response obtained by lth replication of the experiment with factors A, B and C at levels i, j and k, respectively; μ is the mean response; α_i is the effect of factor A at level i; β_j is the effect of factor B at level j; ξ_k is the effect of factor C at level k; γ_{ABij} is the effect of the interaction between A and B at levels i and j; γ_{ABCijk} is the effect of the interaction between A, B and C at levels i, j and k, and so on. Lastly, e_{ijkl} is the error of the lth replication when factors A, B and C are at levels i, j and k, and is simply the difference between the observed value and the value calculated by the model.

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \xi_k + \gamma_{ABij} + \gamma_{ACik} + \gamma_{BCjk} + \gamma_{ABCijk} + e_{ijkl} \tag{5.5}$$

The parameters can be estimated from the means taken along the dimensions, such that the grand mean is $\mu = \bar{y}_{....}$, the effect of factor A at level i is $\alpha_i = \bar{y}_{i...} - \bar{y}_{....}$, the effect of the interaction between A and B at levels i and j is $\gamma_{ABij} = \bar{y}_{ij..} - \bar{y}_{....}$ and so on. The error in the lth replication of the experiment is given by $e_{ijkl} = y_{ijkl} - \bar{y}_{ijk}$.

In order to estimate the variation of the response variable that each factor is responsible for, the sums of squares must be calculated using Equation 5.6, where a, b and c are the numbers of levels that parameters A, B and C have, respectively, and r is the number of replicates.

$$\begin{aligned}\sum_{ijkl} y_{ijkl}^2 = \\ abcr\mu^2 + bcr\sum_i \alpha_i^2 + acr\sum_j \beta_j^2 + abr\sum_k \xi_k^2 + \\ cr\sum_{ij} \gamma_{ij}^2 + br\sum_{ik} \gamma_{ik}^2 + ar\sum_{jk} \gamma_{jk}^2 + \\ r\sum_{ijk} \gamma_{ijk}^2 + \sum_{ijkl} e_{ijkl}^2\end{aligned} \tag{5.6}$$

This is equivalent to Equation 5.7, where the various sums of squares have been appropriately placed with their corresponding terms in Equation 5.6.

$$\begin{aligned} SSY = & \\ & SS0 + SSA + SSB + SSC + \\ & SSAB + SSAC + SSBC + \\ & SSABC + SSE \end{aligned} \tag{5.7}$$

The total variation SST is given by Equation 5.8, and can be divided into parts explained by each factor and factor interactions and an unexplained part attributed to experimental errors. The percentage of variation explained by a factor or an interaction can be used to measure the importance of the corresponding effect. The percentage explained by factor A, for example, can be calculated by $\frac{SSA}{SST}$ [38].

$$\begin{aligned} & SST \\ & \quad = SSY - SS0 \\ & \quad = SSA + SSB + SSC + SSAB + SSAC + SSBC + SSABC + SSE \end{aligned} \tag{5.8}$$

The statistical significance of a factor is tested by dividing its sum of squares (SS) by its degrees of freedom to obtain its mean square (MS). The degrees of freedom are given in Equation 5.9, in the same order as the sums of squares appear in Equation 5.7. Therefore, $MSE = \frac{SSE}{abc(r-1)}$, $MSA = \frac{SSA}{a-1}$, $MSAB = \frac{SSAB}{(a-1)(b-1)}$, and so on.

$$\begin{aligned} abcr = & \\ & 1 + (a-1) + (b-1) + (c-1) + \\ & (a-1)(b-1) + (a-1)(c-1) + (b-1)(c-1) + \\ & (a-1)(b-1)(c-1) + abc(r-1) \end{aligned} \tag{5.9}$$

If the ratio $\frac{MSA}{MSE}$ is larger than the F-distribution table value of $F_{[a-1,abc(r-1)]}$, where $a-1$ and $abc(r-1)$ are the degrees of freedom of factor A and of the error, respectively, as shown in Equation 5.9, then the effect of factor A is statistically significant at the given level of confidence. The same process, which is called the F-test, is applied to evaluate the effect of other factors and interactions, using the appropriate degrees of freedom. The F-test makes various assumptions (e.g., the errors are normally distributed) which are detailed in Jain [38]. In this thesis 99% confidence is considered for performing the F-test.

Chapter 6

Results and Discussion

Both constrained and spectral clustering are performed using all additional data sources and combinations thereof. We first integrate each type of domain knowledge separately in order to assess the effect it has on clustering quality. Additionally, all combinations of data sources are applied in order to analyze the improvement multiple additional knowledge domains provide, as well as the effect of increasing the number of constraints.

6.1 Constrained Clustering Results

This section presents the results for the constrained clustering algorithms. Table 6.1 shows the codes used for each constraint set described in Section 5.2. Thirty repetitions are performed for each algorithm and each parameter configuration, and all following tables present the average EP-Ratios of the constraint sets that yield results significantly different from those produced by the unconstrained algorithms according to the paired observations comparison, with at least a 95% level of confidence. Asterisks indicate the difference is significant at the 95% level of confidence but not at the 99% level, while hyphens indicate the differences are not significant. The best results are in bold.

Table 6.1. Codes used for each constraint set.

Code	Constraint Set
CL	Structural Alignment-Based Cannot-Links
ML	Structural Alignment-Based Must-Links
GC	Genomic Context-Based Must-Links
$100AS$	100% Identical Active Site-Based Must-Links
$95AS$	95% Identical Active Site-Based Must-Links

6.1.1 K-Medoids with Multiple Sequence Alignments

In this subsection we present and discuss the results of applying K-Medoids using the multiple sequence alignments as attributes, as described in Section 5.3.1. Each individual constraint set and each combination of constraint sets are employed. For the general full factorial design, two factors are considered: the similarity measure (BLOSUM62, PAM30, or Hamming distance) and the constraint set being employed.

Clustering Subfamilies Inside a Single Family

Based on the paired observations comparison, although constrained versions of K-Medoids achieve improvements for the protein kinases and serine proteases families when compared to the unconstrained algorithm, none of the constraint sets yield results significantly different from those produced by unconstrained K-Medoids for any of the three measures at the 95% confidence level. This can be explained by the small number of constraints created, and suggests that the unconstrained algorithm's objective function is able to satisfy the constraints naturally, indicating that the constraints are consistent with it. However, significant differences exist when applying the constraint sets for nucleotidyl cyclases and for all three families combined, as follows.

The average EP-Ratios for the constraint sets with significant results compared to the unconstrained K-Medoids for the nucleotidyl cyclases family are presented in Table 6.2. When the complement of the Hamming distance is used as the algorithm's similarity measure, 100% identical active site-based constraints and their combination with genomic context-based constraints actually yield worse results (i.e., higher EP-Ratios) than the unconstrained version, which means that the subfamilies are more mixed in the resulting clusters. Using 95% identical active site-based constraints, as well as their combination with genomic context-based constraints, produces better results (i.e., lower EP-Ratios) than the unconstrained version for any of the similarity measures. When the similarity measure is either BLOSUM62 or PAM30, all four constraint sets yield better results than the unconstrained algorithm.

Regarding the general full factorial design, for the nucleotidyl cyclases family, 79.68% of the variation in EP-Ratio is explained by the constraint set, 15.24%, by the interaction between constraint set and similarity measure, and 5%, by the similarity measure. Since the smaller the EP-Ratio, the better the clustering is, the best constraint set is that with the largest negative effect on EP-Ratio. Using no constraint sets has a positive effect, which means the use of constraints improves the results. For

Table 6.2. K-Medoids with MSA - Nucleotidyl Cyclases.

Constraint Set	BLOSUM62	Hamming	PAM30
Unconstrained	0.3186	0.3186	0.3186
100AS	0.3142	0.3268	0.3142
GC & 100AS	0.3142	0.3268	0.3142
95AS	**0.3050**	**0.3050**	**0.3050**
GC & 95AS	**0.3050**	**0.3050**	**0.3050**

this family, the best constraint sets are the 95% identical active site-based constraints and their combination with genomic context-based constraints, which is in accordance with the results in Table 6.2.

For the protein kinases family, the results of the general full factorial design show that all of the variation in EP-Ratio is explained by the similarity measure. This is likely due to the very small number of constraints created for this family, which the unconstrained algorithm's objective function satisfies on its own. For the serine proteases family, the only parameter which is significant according to the F-test is the similarity measure. Both are in accordance with the paired observations comparison, which concluded that none of the constraint sets yields results significantly different from the unconstrained algorithm.

Clustering Subfamilies Inside Multiple Families

Table 6.3 presents the average EP-Ratios for the constraint sets with significant results based on the paired observations comparison when all three families are combined and $K = 8$, which is the total number of subfamilies. Many structural alignment-based cannot-link constraints exists when the enzyme families are combined.

When using BLOSUM62 as the algorithm's similarity measure, the resulting clusters are either worse or not significantly different from those obtained by the unconstrained K-Medoids, unless the cannot-link constraint set is added, in which case the results are always better. When using PAM30 or the complement of the Hamming distance as similarity measures, all constraint sets lead to improved clusters, except when using genomic context-based constraints, structural alignment-based must-links and their combination, in which case the results are not significantly different from unconstrained clustering. For the Hamming distance, the results of using the combination of structural alignment-based must-link constraints and 100% identical active site-based constraints is significant at 95% confidence, but not at the 99% level.

Table 6.3. K-Medoids with MSA - All 8 Subfamilies.

Constraint Set	BLOSUM62	Hamming	PAM30
Unconstrained	0.9143	1.0974	0.9792
100AS	-	1.0838	0.9604
GC & 100AS	-	1.0131	0.9573
95AS	0.9272	1.0086	0.9511
ML & 100AS	-	1.0869*	0.9604
GC & 95AS	0.9342	1.0145	0.9511
ML, GC & 100AS	-	1.0002	0.9584
ML & 95AS	0.9272	1.0086	0.9511
ML, GC & 95AS	0.9342	1.0145	0.9511
CL & GC	0.8926	1.0079	0.9071
CL & 100AS	0.8798	0.9941	0.8973
CL, GC & 100AS	0.8777	0.9236	0.8973
CL & ML	0.8926	1.0056	0.9071
CL, ML & GC	0.8926	1.0103	0.9071
CL & 95AS	**0.8752**	**0.9092**	**0.8913**
CL, ML & 100AS	0.8806	0.9909	0.8965
CL, GC & 95AS	0.8762	0.9210	**0.8913**
CL, ML, GC & 100AS	0.8800	0.9217	0.8966
CL, ML & 95AS	**0.8752**	**0.9092**	**0.8913**
CL, ML, GC & 95AS	0.8762	0.9210	**0.8913**

The general full factorial design shows that 48.79% of the variation in EP-Ratio is explained by the similarity measure, 36.12%, by the constraint set and 12.74%, by the interaction between them. The best constraint sets, i.e., those with the largest negative effects, are the set that combines structural alignment-based cannot-link constraints with 95% identical active site-based constraints and its combination with structural alignment-based must-link constraints, which agrees with the results in Table 6.3. Using no constraint sets has a positive effect on EP-Ratio, implying that the use of constraints improves the results.

Clustering Families

Based on the paired observations comparison, when clustering in the search for the three families instead of the subfamilies (i.e., $K = 3$), the result of adding the cannot-link constraint set stands out. The constraint sets with significant results are shown in Table 6.4. The average EP-Ratio for unconstrained K-Medoids is fairly high, and most sets of must-link constraints improve it. However, perfect clusters are found when the

cannot-link constraint set is applied, no matter which must-link constraint set is used. This implies that K-Medoids is able to perfectly separate the three enzyme families when using structural information in the form of cannot-link constraints. The cases involving cannot-link constraints have been omitted from the table because they all yield perfect clusters (i.e., the EP-Ratios are zero) for any of the similarity measures.

Table 6.4. K-Medoids with MSA - All 3 Families.

Constraint Set	BLOSUM62	Hamming	PAM30
Unconstrained	0.4542	0.8393	0.4929
GC	-	0.8383	-
100AS	0.4430	-	0.4814
GC & 100AS	0.4430	-	0.4814
95AS	0.4202	0.7745	0.4583
ML & 100AS	0.4430	-	0.4814
GC & 95AS	0.4202	0.7762	0.4583
ML, GC & 100AS	0.4430	-	0.4814
ML & 95AS	0.4202	0.7879	0.4583
ML, GC & 95AS	0.4202	0.7651	0.4583

The results of the general full factorial design show that 84.37% of the variation in EP-Ratio is explained by the constraint set, 7.98%, by the similarity measure and 7.36%, by the interaction between them. The constraint sets with the largest negative effects are all those which involve cannot-link constraints, since perfect clusters are obtained when they are applied. Using no constraint sets has a positive effect on EP-Ratio, again suggesting that using constraints improves the clustering results.

Summary

When clustering subfamilies in the nucleotidyl cyclases family, the significant improvements occur when using active site-based and genomic context-based constraint sets. This shows that using active sites obtained from enzyme structures and the genomic context of the corresponding genes (when available) can aid the problem of predicting the fourth and most challenging level of the EC number, which represents substrate specificity. The effect of using constraint sets may not have been significant for the other two families because of the small number of constraints we were able to generate. Both structural and genomic context data are still rare in comparison with sequence data. However, structural genome initiatives such as the Protein Structure Initiative (PSI)[1]

[1] `http://www.structuralgenomics.org/`

and the several genome projects that exist worldwide will contribute to expand the amount of available data. This information can then be used to further improve these results, since even small numbers of additional constraints could imply larger constraint sets when combined with those that already exist because of the transitivity property.

For the problems of clustering families or subfamilies in multiple families, the effect of the cannot-link constraints is very noticeable, even leading to perfect clusters in the first case. Structural alignment-based cannot-link constraints are very useful since they add structural information that the sequence-based attributes do not carry, allowing the separation into (sub)families based on structural dissimilarity. It is likely that cannot-link constraints would have had the same positive effect when clustering subfamilies inside a single family, however, unfortunately, we were unable to create them in this case, as previously discussed. Another factor that contributes to the larger effect of the cannot-link constraint set is simply the large number of constraints.

The results in general attest to the importance and significance of using information from knowledge domains other than sequence data alone, which constitutes the main database, since sequence similarity does not imply functional similarity. The cases in which the constrained clustering produces results worse than the unconstrained clustering seem to agree with Davidson and Ravi [19], who state that, despite not presenting contradictions and being generated from facts, it is still possible that some constraint sets decrease the clustering precision.

6.1.2 K-Medoids with Active Sites

Since amino acid sequences are much more readily available than protein structures, using the results of a multiple sequence alignment as attributes is a straightforward approach. However, in this work we also study the use of active sites as attributes. Since inside each enzyme family the active sites are already aligned, we simply repeat the process employed when multiple sequence alignments are used as attributes.

When combining all three families, however, there are small differences in the number of residues in the active sites of enzymes from different families. Since the clustering algorithms require that all objects be described by the same attributes, which in our case are the positions in the active sites, we perform a multiple sequence alignment in order to be able to compare residues at the same position in the active sites of all enzyme pairs. In this case, active site-based constraints are not employed, since the information they would add is already embedded in the attributes. For the general full factorial designs the factors are the similarity measure and the constraint set applied.

Clustering Subfamilies Inside a Single Family

For the protein kinases family, only genomic context-based constraints were created, since none of the enzyme pairs have an RMSD smaller or equal to the cutoff employed. Based on the comparison of paired observations, the results yielded by these constraints are not significantly different from those of the unconstrained algorithm for any of the similarity measures. This can be explained by the very small number of constraints generated for this family, which the unconstrained algorithm's objective function satisfies naturally.

For the nucleotidyl cyclases family, using structural alignment-based constraints does not produce results significantly different from the unconstrained K-Medoids for any of the similarity measures. For PAM30, the results of using genomic context-based constraints are non-significant, and using the combination of genomic context-based and structural alignment-based constraints yields worse results than when no constraints are applied. For BLOSUM62 and the complement of the Hamming distance, using genomic context-based constraints and their combination with structural alignment-based constraints produces significantly better results than the unconstrained algorithm. The average EP-Ratios for the constraint sets with significant results are shown in Table 6.5. All constraint sets produce significantly better results than the unconstrained K-Medoids for the serine proteases family, as shown in Table 6.6.

Table 6.5. K-Medoids with Active Sites - Nucleotidyl Cyclases.

Constraint Set	BLOSUM62	Hamming	PAM30
Unconstrained	0.3770	0.3784	**0.3872**
GC	**0.3721**	0.3734	-
ML & GC	0.3733	**0.3687**	0.3936

Table 6.6. K-Medoids with Active Sites - Serine Proteases.

Constraint Set	BLOSUM62	Hamming	PAM30
Unconstrained	0.4216	0.3624	0.4485
GC	0.3007	0.2889	**0.3003**
ML	0.3121	0.3030	0.3541
ML & GC	**0.2926**	**0.2697**	0.3049

The results of the general full factorial designs show that, for the nucleotidyl cyclases family, 57.6% of the variation in EP-Ratio is explained by the similarity measure, 9.65%, by the interaction between similarity measure and constraint set, and 2.33%, by the constraint set alone. The constraint set with the largest negative effect is the genomic context-based constraint set, followed by its combination with structural alignment-based constraints, which agrees with the results in Table 6.5. Using no constraints at all has a positive effect on EP-Ratio, indicating that the use of constraints improves the clustering quality.

For the protein kinases family, 24.16% of the variation in EP-Ratio is explained by the similarity measure, 1.32%, by the interaction between similarity measure and constraint set, and 0.76% by the constraint set. Again, the small importance of the constraint set is likely because of the very small number of constraints created for this family. The constraint set with the largest negative effect is the genomic context-based constraint set, while using no constraints at all has a positive effect. For the serine proteases family, 52.02% of the variation in EP-Ratio is explained by the constraint set, 7.85%, by the similarity measure, and 3.43%, by the interaction between them. The constraint sets with the largest negative effects are the genomic context-based constraint set and its combination with the structural alignment-based constraints, which agrees with the results in Table 6.6. Again, using no constraints has the largest positive effect, implying that constraints improve the clustering quality.

Clustering Subfamilies Inside Multiple Families

When all three families are combined, structural alignment-based cannot-link constraints exist. Based on the comparison of paired observations, significantly better results are produced by all constraint sets when clustering all eight subfamilies, as shown in Table 6.7.

The result of the general full factorial design shows that 22.27% of the variation in EP-Ratio is explained by the similarity measure, 17.74%, by the constraint set and 1.31%, by the interaction between them. The constraint set with the largest negative effect is the one that combines structural alignment-based cannot-link and must-link constraints with genomic context-based constraints. Both the genomic context-based constraint set and its combination with structural alignment-based must-link constraints also have large negative effects, which agrees with the results in Table 6.7. Not using constraints at all has the largest positive effect, showing that using constraints improves the clustering quality.

Table 6.7. K-Medoids with Active Sites - All 8 Subfamilies.

Constraint Set	BLOSUM62	Hamming	PAM30
Unconstrained	0.2812	0.2805	0.3312
GC	0.2329	0.2441	**0.2662**
ML	0.2355	0.2479	0.2897
ML & GC	**0.2273**	0.2418	0.2705
CL & GC	0.2338	0.2476	0.2684
CL & ML	0.2355	0.2537	0.2897
CL, ML & GC	0.2281	**0.2391**	0.2721

Clustering Families

When using active sites as attributes for clustering in the search for the three families, perfect clusters are obtained for all similarity measures and both constrained and unconstrained versions of K-Medoids. This can be explained by the active site being directly responsible for the enzyme's function, which in turn is what determines enzyme families. Therefore, using active sites as attributes may be considered as a way of separating enzyme families. Unfortunately, this information is often unavailable.

The results for the general full factorial design show that 84.37% of the variation in EP-Ratio is explained by the constraint set, 7.98%, by the similarity measure and 7.36%, by the interaction between them. The largest negative effect is obtained by all constraint sets involving the cannot-link constraints, which yield perfect clusters. Using no constraints at all has a positive effect, which implies the use of constraints improves the clustering quality.

Summary

When using the active sites as attributes, most of the significant improvements are obtained when using either the genomic context-based constraints or their combination with structural alignment-based constraints. This attests to the quality of genomic context information, since it allows to differentiate enzymes belonging to the same family but with different substrate specificities. Also, the improvements involving combinations of constraint sets again suggest that larger numbers of constraints lead to better results.

Even when using active site information as the basis for the clustering, adding information from other knowledge domains still improves the results. This shows the potential that using data from many different domains has on clustering enzymes and

predicting their function. Again, the cases in which constrained clustering yields results worse than unconstrained clustering are in agreement with the statement of Davidson and Ravi [19] that some constraint sets might decrease the clustering precision.

6.1.3 K-Means with Distance Arrays

In this section we present and discuss the results of applying K-Means using the 151-dimension vectors as attributes and applying all constraint sets and their combinations. The attributes are either normalized or not. For the general full factorial design, the factors are the constraint set and the normalization mode (i.e., whether or not the attributes are normalized).

Clustering Subfamilies Inside a Single Family

Based on the comparison of paired observations, none of the constraint sets yield results significantly different from those produced by the unconstrained K-Means for the protein kinases family when the attributes are unnormalized. However, perfect clusters are achieved for both constrained and unconstrained versions of K-Means when we normalize the attributes.

Table 6.8 shows the average EP-Ratios of the constraint sets that yield significant results for the nucleotidyl cyclases family. Using genomic context-based constraints or their combination with structural alignment-based constraints produces worse results than the unconstrained K-Means for both normalized and unnormalized attributes. When normalizing the attributes, using 95% identical active site-based constraints and their combination with genomic context-based constraints yield better results than when no constraints are employed. The improvement of using 100% identical active site-based constraints is significant at the 95% confidence level, but not at the 99% level.

Table 6.8. K-Means - Nucleotidyl Cyclases.

Constraint Set	Unnormalized	Normalized
Unconstrained	**0.3702**	0.3965
GC	0.3726	0.3979
ML & GC	0.3726	0.3979
100AS	-	0.3848*
95AS	-	**0.3583**
GC & 95AS	-	0.3604

For the serine proteases family, none of the constraint sets produce results significantly different at the 99% confidence level. However, at 95% confidence level, using 95% identical active site-based constraints yields better results for both normalized and unnormalized attributes, as shown in Table 6.9. Using structural alignment-based constraints and their combination with 95% identical active site-based constraints yield worse results than the unconstrained K-Means.

Table 6.9. K-Means - Serine Proteases.

Constraint Set	Unnormalized	Normalized
Unconstrained	1.0307	1.0095
95AS	**0.9720***	**0.9550***
ML	-	1.0914*
ML & 95AS	-	1.0867*

Regarding the general full factorial design results, for the nucleotidyl cyclases family, 12.09% of the variation in EP-Ratio is explained by the constraint set, 4.69%, by the normalization mode, and 3.63%, by the interaction between them. The constraint set with the largest negative effect is the 95% identical active site-based constraint set. Using no constraints at all has a positive effect, implying that using constraints improves the clustering quality.

For the protein kinases family, the factorial design results are that 97.12% of the variation in EP-Ratio is explained by the normalization mode. The constraint sets have no effect at all. This, again, can be attributed to the very small number of constraints generated for this family, which the algorithm's objective function is capable of satisfying on its own. For the serine proteases family, only the effect of the constraint set is significant according to the F-test. However, it explains only 5.66% of the variation in EP-Ratio. The constraint set with the largest negative effect is the 95% identical active-site based constraint set, which agrees with the results in Table 6.9. Using no constraints at all has a positive effect, which means constrained clustering leads to better results.

Clustering Subfamilies Inside Multiple Families

Based on paired observations comparison, when clustering all eight subfamilies, none of the constraint sets yield significant results when the attributes are normalized. For unnormalized attributes, however, almost all constraint sets yield improved results when compared to unconstrained K-Means, as shown in Table 6.10. Only the results

for four constraint sets are non-significant at a 95% confidence level; four are significant at the 95% confidence level but not at 99% confidence; and all others are significant with 99% confidence, especially when the cannot-link constraint set is applied.

Table 6.10. K-Means - All 8 Subfamilies.

Constraint Set	Unnormalized	Normalized
Unconstrained	0.6754	**0.6181**
GC	0.6557*	-
100AS	0.6375*	-
GC & 100AS	0.6287*	-
95AS	0.5998	-
GC & 95AS	0.5902	-
ML & 95AS	0.5903	-
ML, GC & 95AS	0.6036*	-
CL & GC	0.5767	-
CL & 100AS	0.5888	-
CL, GC & 100AS	0.5760	-
CL & ML	0.5726	-
CL, ML & GC	0.5628	-
CL & 95AS	0.5706	-
CL, ML & 100AS	0.5698	-
CL, GC & 95AS	**0.5406**	-
CL, ML, GC & 100AS	0.5602	-
CL, ML & 95AS	0.5592	-
CL, ML, GC & 95AS	0.5478	-

The results of the general full factorial design show that 3.23% of the variation in EP-Ratio is explained by the interaction between the constraint set and the normalization mode, and 2.56%, by the constraint set. The effect of the normalization mode alone is not significant according to the F-test. Two constraint sets have the largest negative effects: the one that combines structural alignment-based cannot-link and must-link constraints, genomic context-based constraints and 100% identical active site-based constraints; and the one that combines structural alignment-based cannot-link constraints with genomic context-based and 95% identical active site-based constraints. Using no constraint sets at all has a positive effect, again suggesting the use of constraints improves the clustering quality.

Clustering Families

When clustering in the search for the three families, applying the cannot-link constraint set always yields better results than when no constraints are used with or without normalizing the attributes according to the comparison of paired observations. When the attributes are normalized, all must-link constraint sets yield better results, as shown in Table 6.11. Whenever the cannot-link constraint set is employed, perfect clusters are achieved for 29 of the 30 repetitions, so that the score is 0.1214 for all combinations of constraint sets and both normalized and unnormalized attributes. Therefore, the results when using cannot-link constraints were omitted from the table.

Table 6.11. K-Means - All 3 Families.

Constraint Set	Unnormalized	Normalized
Unconstrained	0.5353	0.6431
GC	-	0.5481
GC & 100AS	-	0.5481
ML & GC	-	0.5481
GC & 95AS	-	0.5137
ML, GC & 100AS	-	0.5481
ML, GC & 95AS	-	0.5137

The results of the general full factorial design show that the only parameter whose effect is significant according to the F-test is the constraint set, which explains 8% of the variation in EP-Ratio. The constraint sets with the largest negative effects are all those involving the cannot-link constraints, which is in agreement with Table 6.11. Again, using no constraint has a positive effect on the EP-Ratio, suggesting constrained clustering improves the clustering results.

Summary

The difference observed between normalized and unnormalized attributes is due to the fact that, in the latter case, the effect is that each attribute (i.e., position in the vector) has a different weight in the clustering algorithm's distance function. Because of the manner in which the vectors are created (the last position corresponds to the number of amino acid pairs within the largest distance from each other), discrepancies in the last positions of the vectors have higher weights. The constraint sets are less effective when unnormalized attributes are used, except when clustering subfamilies inside multiple

families, in which case none of the constraint sets produce results significantly different from unconstrained K-Means with normalized attributes. When normalizing the attributes, the best results are obtained when using the active site-based constraint sets.

In the case of clustering the three families, active site-based constraints combined with genomic context-based, as well as their combination with structural alignment-based constraints yield the best results, especially when the cannot-link constraint set is applied. Again, this suggests that the more constraints, the better the clustering quality, i.e., the more additional information we gather from external data sources, the better the results of enzyme clustering and function prediction. However, as stated by Davidson and Ravi [19], it is possible that some constraint sets decrease the clustering precision, even if they are based on facts and there is no contradiction among them.

6.2 Spectral Clustering Results

This section presents the results obtained applying spectral clustering to the problem of clustering enzymes by specificity. The same codes described in Table 6.1 are used to identify the data sources being integrated to the initial similarity matrices, which contain purely the normalized BLOSUM62, PAM30 or Hamming scores. The exception is the active site data, simply represented by AS instead of 95AS and 100AS. Additionally, Table 6.12 shows the codes that identify the possible manners in which the data sources are integrated to the initial similarity matrices, as described in Section 5.4.

Table 6.12. Codes used for each manner of adding the data sources to the initial similarity matrices.

Code	Integration Method
cutoff	Using the same cutoffs applied for creating constraints
value	Using the actual values from the additional datasets
edges	Enforcing edges after the graph is constructed

Regarding the general full factorial design, there are four parameters that vary for spectral clustering, as detailed in Section 5.4:

- the similarity measures, with three different values (BLOSUM62, PAM30, or the complement of the Hamming distance);
- the graph construction method, with four options (fully connected, KNN, mutual KNN and ϵ-neighborhood) and three values for K and ϵ, totalizing ten methods;

- the type of graph Laplacian, with two options (unnormalized or normalized symmetric Laplacian);
- the similarity matrix, with 21 different similarity matrices for each family and 29 when all three families are combined. The larger number of matrices when the families are combined is due to the existence of structural alignment-based cannot-link constraints.

This yields the total of 1,260 parameter configurations for each enzyme family and 1,740 configurations when all three families are combined. Furthermore, there are ten repetitions for each configuration. In order to analyze the results of all these configurations, a general full factorial design with four factors [38] was built for each family.

6.2.1 Full Factorial Designs with Four Factors

This subsection describes the results of the full factorial designs for each of the enzyme families and for when they are combined. Table 6.13 lists the four factor codes and which parameter they represent.

Table 6.13. Codes used for each parameter.

Code	Parameter
A	Similarity Measure
B	Similarity Matrix
C	Graph Laplacian
D	Graph Construction Method

Nucleotidyl Cyclases

The analysis of variance for enzyme family nucleotidyl cyclases at the 99% confidence level is shown in Table 6.14, which contains the factor or interaction, its sum of squares (SS), the percentage of the total variation in response variable (i.e., EP-Ratio) that it explains, its mean square (SS) and the F-test values. The F-test results show that the effects of all factors and all interactions are significant at this level of confidence, since all calculated values are larger than the corresponding F-distribution table values.

The most important individual parameter is D, which represents the graph construction method and explains 18.33% of the variation in EP-Ratio. The interaction between graph construction method and similarity matrix, and the triple interaction

Table 6.14. ANOVA Table for Nucleotidyl Cyclases.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	0.5285	1.84	0.2643	1257.7689	4.65
B	0.4717	1.64	0.0236	112.2521	1.92
C	1.2277	4.27	1.2277	5843.3858	6.69
D	5.2729	**18.33**	0.5859	2788.6436	2.44
AB	1.4301	4.97	0.0358	170.1669	1.63
AC	0.0061	0.02	0.0031	14.6206	4.65
AD	0.9508	3.30	0.0528	251.4301	1.97
BC	0.3230	1.12	0.0018	8.5418	1.92
BD	5.6629	**19.68**	0.0315	149.7453	2.30
CD	1.4595	**5.07**	0.1622	771.8794	2.44
ABC	0.1565	0.54	0.0039	18.6214	1.63
ABD	6.4132	**22.29**	0.0178	84.7922	2.30
ACD	0.1649	0.57	0.0092	43.6159	1.97
BCD	1.4926	**5.19**	0.0083	39.4688	2.30
ABCD	0.8269	2.87	0.0023	10.9328	2.30
Error	2.3825	**8.28**	0.0002	-	-

between them and the similarity measure are even more significant, since they explain 19.68% and 22.29% of the variation in EP-Ratio, respectively. Since the smaller the EP-Ratio, the better the clustering is, the best value for a given parameter is the one with the largest negative effect. That said, the best graph construction method is mutual KNN with K set as 75% of the neighbors (effect of -0.0341), closely followed by the mutual KNN with $K = 50\%$ of the neighbors, which has the effect of -0.0323.

Protein Kinases

Table 6.15 presents the analysis of variance for enzyme family protein kinases at the 99% confidence level. The F-test results show that the effects of all factors and combinations thereof are statistically significant.

Again, the most important individual factor is the graph construction method, which explains 13.67% of the variation in EP-Ratio. The interaction between graph construction method and similarity matrix is even more significant, explaining 17.12% of the variation. The method with the largest negative effect for the protein kinases family is mutual KNN with $K = 50\%$ of the neighbors (with effect -0.0626), followed by mutual KNN with $K = 25\%$ of the neighbors, which has an effect of -0.0551.

Table 6.15. ANOVA Table for Protein Kinases.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	0.8471	0.29	0.4235	154.2458	4.65
B	19.0601	**6.68**	0.9530	347.0747	1.92
C	9.1827	3.22	9.1827	3344.2473	6.69
D	38.9937	**13.67**	4.3326	1577.8992	2.44
AB	5.0603	1.77	0.1265	46.0728	1.63
AC	1.3706	0.48	0.6853	249.5802	4.65
AD	9.1763	3.22	0.5098	185.6622	1.97
BC	3.6198	1.27	0.0201	7.3239	1.92
BD	48.8362	**17.12**	0.2713	98.8089	2.30
CD	9.7144	3.41	1.0794	393.0975	2.44
ABC	4.3325	1.50	0.1083	39.4466	1.63
ABD	33.6975	**11.81**	0.0936	34.0896	2.30
ACD	7.5621	2.65	0.4201	153.0024	1.97
BCD	35.0350	**12.28**	0.1946	70.8854	2.30
ABCD	27.6709	**9.69**	0.0769	27.9929	2.30
Error	31.1377	**10.91**	0.0027	-	-

Serine Proteases

The analysis of variance for enzyme family serine proteases at the 99% confidence level is presented in Table 6.16. The F-test results show all effects are statistically significant. For this family, the most important individual factor is also the graph construction method, which explains 30.75% of the variation in EP-Ratio. The method with the largest negative effect is mutual KNN with $K = 25\%$ of the neighbors, which has an effect of -0.6853, followed by mutual KNN with $K = 50\%$ of the neighbors, with an effect of -0.2719.

Clustering Subfamilies Inside Multiple Families

The analysis of variance for clustering all eight subfamilies when the three families are combined is presented in Table 6.17. All effects are statistically significant according to the F-test results.

The most important individual factor is the graph construction method, which is responsible for 32.29% of the variation in EP-Ratio. The interaction between the graph construction method and the similarity matrix is also very important, explaining

Table 6.16. ANOVA Table for Serine Proteases.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	2.3093	0.06	1.1546	29.8903	4.65
B	232.6511	**6.05**	11.6326	301.1311	1.92
C	488.8935	**12.71**	488.8935	12655.9512	6.69
D	1182.4165	**30.75**	131.3796	3401.0144	2.44
AB	34.4082	0.89	0.8602	22.2681	1.63
AC	10.4604	0.27	5.2302	135.3933	4.65
AD	24.7356	0.64	1.3742	35.5738	1.97
BC	100.8773	2.62	0.5604	14.5078	1.92
BD	318.8938	**8.29**	1.7716	45.8621	2.30
CD	535.6238	**13.93**	59.5138	1540.6283	2.44
ABC	16.6310	0.43	0.4158	10.7632	1.63
ABD	95.4977	2.48	0.2653	6.8671	2.30
ACD	85.9104	2.23	4.7728	123.5531	1.97
BCD	196.2639	**5.10**	1.0904	28.2259	2.30
ABCD	81.8236	2.13	0.2273	5.8838	2.30
Error	438.0589	**11.39**	0.0386	-	-

37.21% of the variation. The graph construction methods with the largest negative effects are mutual KNN with $K = 25\%$ of the neighbors (effect of -1.8687), the fully connected graph (effect of -1.7576) and mutual KNN with $K = 50\%$ of the neighbors (effect of -1.6255).

Clustering Families

The analysis of variance for clustering all three families combined, with $K = 3$, is presented in Table 6.18. The F-test results show that all effects are statistically significant. When clustering families, the most important individual factor is, again, the graph construction method, which is responsible for 35% of the variation in EP-Ratio. The interaction between graph construction method and similarity matrix is also important, since it explains 27.69% of the variation. The graph construction methods with the largest negative effects are mutual KNN with $K = 25\%$ of the neighbors (effect of -2.1833), the fully connected graph (effect of -2.0343) and mutual KNN with $K = 50\%$ of the neighbors (effect of -1.5032).

Table 6.17. ANOVA Table for All Eight Families.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	1577.6151	1.54	788.8076	1786.5279	4.65
B	8510.6875	**8.33**	293.4719	664.6689	1.74
C	223.9324	0.22	223.9324	507.1724	6.69
D	32990.7408	**32.29**	3665.6379	8302.1065	2.44
AB	996.0645	0.98	17.1735	38.8954	2.30
AC	24.7444	0.02	12.3722	28.0211	4.65
AD	2879.6872	2.82	159.9826	362.3361	1.97
BC	225.5799	0.22	0.8643	1.9575	1.74
BD	38017.3647	**37.21**	145.6604	329.8984	2.30
CD	489.5291	0.48	54.3921	123.1898	2.44
ABC	130.2645	0.13	2.2459	5.0867	2.30
ABD	5338.6705	**5.23**	10.2273	23.1633	2.30
ACD	466.6421	0.46	25.9246	58.7151	1.97
BCD	1694.8759	1.66	6.4938	14.7074	2.30
ABCD	1438.7559	1.41	2.7562	6.2425	2.30
Error	7152.8031	**7.00**	0.4415	-	-

General Results

The analysis of paired observations was employed to compare the results obtained when using the initial similarity matrices (i.e., those containing purely the normalized BLOSUM62, PAM30 or Hamming scores) against the similarity matrices that combine all information possible. There is one of such matrices for each combination of similarity measure and strategy for integrating additional data sources (i.e., using cutoffs, values or edges as described in Section 5.4). In most cases, the more "informed" similarity matrices yield results significantly better than the initial matrices, which shows the potential and importance of using information from diverse knowledge domains for clustering enzymes and predicting their function, since sequence similarity alone does not imply functional similarity or substrate specificity. There are, however, few cases in which the results are worse than those produced when using the initial matrices.

6.2.2 Fixing the Graph Construction Method

The previous tables show that for all cases, whether clustering each enzyme family separately or all three families combined, the most important individual factor is the graph construction method. Therefore, we fixed the graph construction method, repeating

Table 6.18. ANOVA Table for All Three Families.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	24.2774	0.02	12.1387	28.8688	4.65
B	11192.2849	**10.01**	385.9409	917.8625	1.74
C	970.0224	0.87	970.0224	2306.9525	6.69
D	39126.6717	**35.00**	4347.4079	10339.2082	2.44
AB	538.0619	0.48	9.2769	22.0628	2.30
AC	120.3282	0.11	60.1641	143.0851	4.65
AD	2565.2407	2.29	142.5134	338.9319	1.97
BC	740.1353	0.66	2.8358	6.7442	1.74
BD	30947.7766	**27.69**	118.5739	281.9979	2.30
CD	2056.1202	1.84	228.4578	543.3289	2.44
ABC	622.5089	0.56	10.7329	25.5255	2.30
ABD	5490.5831	4.91	10.5184	25.0153	2.30
ACD	547.7616	0.49	30.4312	72.3729	1.97
BCD	6451.8033	**5.77**	24.7196	58.7892	2.30
ABCD	3571.2902	3.19	6.8416	16.2709	2.30
Error	6811.7411	**6.09**	0.4205	-	-

the analysis varying only the other three factors. The mutual KNN with $K = 50\%$ of the neighbors was chosen, since it is consistently among the best for all cases. Additionally, the analysis of a given family was repeated fixing the graph construction method as the best for that particular family. This always involves a mutual KNN, only varying the K among the families: $K = 75\%$ of the neighbors for the nucleotidyl cyclases family; $K = 50\%$ for the protein kinases; and $K = 25\%$ for serine proteases and when all three families are combined.

Nucleotidyl Cyclases

The analysis of variance for the nucleotidyl cyclases family when fixing the graph construction method as mutual KNN with $K = 50\%$ of the neighbors is shown in Table 6.19. The effect of factor C (i.e., the graph Laplacian) has been omitted because it is non-significant according to the F-test. The most important individual parameter is the similarity matrix, which explains 34.63% of the variation in EP-Ratio. The largest percentage of variation, however, is explained by the interaction between similarity measure and similarity matrix: 56.39%. The similarity matrices with the largest negative effects are $AS\_\,value$ and $AS\_\,GC\_\,value$, which, using the codes in Tables 6.1 and 6.12, correspond to the matrices that contain active site information, and both active

site and genomic context information, respectively. In both cases such information has been integrated to the initial similarity matrices (i.e., those containing only the normalized BLOSUM62, PAM30, or Hamming scores) using the actual values as described in Section 5.4. Using the initial similarity matrices has a positive effect on the EP-Ratio, which means that adding information to the initial sequence similarity-based matrices improves the clustering quality.

Table 6.19. ANOVA Table for Nucleotidyl Cyclases with mutual KNN $K = 50\%$.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	0.6359	**7.39**	0.3179	7931.3994	4.65
B	2.9817	**34.63**	0.1491	3719.0175	1.92
AB	4.8559	**56.39**	0.1214	3028.3544	1.63
AC	0.0012	0.01	0.0006	14.7655	4.65
BC	0.0045	0.05	0.0002	5.5695	1.92
ABC	0.0398	0.46	0.0009	24.7912	1.63
Error	0.0909	1.06	4E-5	-	-

When using mutual KNN with $K = 75\%$ of the neighbors, which is the best for this family, the effect of the graph Laplacian is significant, but only explains 0.03% of the variation in EP-Ratio. Also, the percentages explained by the factors and interactions are somewhat different, but maintain the same order of importance as with mutual KNN with $K = 50\%$. The best similarity matrix is, again, *AS_value*, closely followed by *AS_GC_value*. The effect of using the initial similarity matrices is, again, positive, implying that using additional information yields better clusterings.

Protein Kinases

Table 6.20 presents the results of the analysis of variance for protein kinases using mutual KNN with $K = 50\%$ of the neighbors, which is also the best graph construction method for this family. Again, factor C (i.e., the graph Laplacian) has been omitted because it is non-significant according to the F-test. The most important individual factor is the similarity matrix, which explains 26.58% of the variation in EP-Ratio. The interaction between similarity matrix and similarity measure is also important, explaining 24.53% of the variation. The best similarity matrix for this family is *ML_AS_value*, which contains information on structural similarity and percentage of active site identity. Using the initial similarity matrices has a positive effect, which implies the extra information improves the clustering quality.

Table 6.20. ANOVA Table for Protein Kinases with mutual KNN $K = 50\%$.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	1.5695	2.18	0.7847	141.6027	4.65
B	19.1608	**26.58**	0.9580	172.8734	1.92
AB	17.6817	**24.53**	0.4420	79.7646	1.63
AC	0.6747	0.94	0.3373	60.8712	4.65
BC	8.9482	**12.41**	0.4474	80.7328	1.92
ABC	11.4604	**15.89**	0.2865	51.6992	1.63
Error	12.5689	**17.44**	0.0055	-	-

Serine Proteases

The analysis of variance for the serine proteases family using mutual KNN with $K = 50\%$ of the neighbors is presented in Table 6.21. Factor A (i.e., the similarity measure) and its combination with factor B (i.e., the similarity matrix) have been omitted because their effects are non-significant according to the F-test. The most important individual parameter is the similarity matrix, which explains 40.83% of the variation in EP-Ratio. The effect of the graph Laplacian is also important, explaining 33.91% of the variation. The best similarity matrix is AS_GC_value, closely followed by AS_value, which contain active site and genomic context information, and active site information, respectively. Using the initial similarity matrices has a positive effect on the EP-Ratio, implying that the additional information helps achieve better clusterings.

Table 6.21. ANOVA Table for Serine Proteases with mutual KNN $K = 50\%$.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
B	203.1473	**40.83**	10.1574	313.8662	1.92
C	168.7086	**33.91**	168.7086	5213.1555	6.69
AC	1.3759	0.2766	0.6879	21.2580	4.65
BC	46.2327	**9.29**	2.3116	71.4303	1.92
ABC	2.4291	0.49	0.0607	1.8765	1.63
Error	73.3972	**14.75**	0.0324	-	-

When using mutual KNN with $K = 25\%$ of the neighbors, which is the best for the family, the results are different. Both factor A and its interaction with factor B have significant effects, explaining 4.53% and 15.98% of the variation in EP-Ratio, respectively. The interaction between factors A and C is non-significant. The order of importance according to the percentage of variation explained by the factors or

interactions is very different than when using $K = 50\%$ of the neighbors: 51.78% of the variation in EP-Ratio is unexplained and attributed to experimental errors; 15.98% is explained by the interaction between similarity measure and similarity matrix; 15.45%, by the similarity matrix; 4.92%, by the interaction between the three factors; 4.53%, by the similarity measure; 3.99%, by the interaction between similarity matrix and graph Laplacian; and 3.33%, by the graph Laplacian. With either graph construction method, however, the most important individual parameter is the similarity matrix. The best similarity matrix in this case is *AS_ value*, closely followed by *ML_ AS_ GC_ value* (i.e., the matrix containing the most information possible) and *AS_ GC_ value*. Using the initial similarity matrices has a negative effect, implying that some matrices containing additional data yield worse clusterings and some yield better clusterings than the initial similarity matrices.

Clustering Subfamilies Inside Multiple Families

Table 6.22 shows the analysis of variance for clustering all eight subfamilies when the three families are combined, using mutual KNN with $K = 50\%$ of the neighbors. The interaction between factors A (i.e., the similarity measure) and C (i.e., the graph Laplacian) has been omitted because its effect is non-significant according to the F-test. Again, the most important individual factor is the similarity matrix, responsible for 16.52% of the variation in EP-Ratio. The percentage of variation unexplained by the parameters and attributed to experimental error is high. The best similarity matrix is *ML_ AS_ value*, followed by *CL_ ML_ cutoff*, *CL_ ML_ GC_ cutoff*, *CL_ ML_ AS_ cutoff*, *CL_ ML_ AS_ GC_ cutoff* and *AS_ GC_ value*. When clustering the three families combined, the effect of the cannot-link structural alignment-based constraints stands out, just as it does for constrained clustering. Using the initial similarity matrices has a positive effect, implying, yet again, that the additional information improves the clustering results.

When using mutual KNN with $K = 25\%$ of the neighbors, which is the best graph construction method for when all three families are combined, the interaction between A and C is significant, but explains only 1.05% of the variation in EP-Ratio. The percentages explained by the factors and interactions are different than when $K = 50\%$ of the neighbors: 57.52% of the variation in EP-Ratio is unexplained and attributed to errors; 13.83% is explained by the similarity matrix; 11.23%, by the triple interaction between the factors; 6.24%, by the interaction between similarity measure and similarity matrix; 5.35%, by the interaction between similarity matrix and graph

Table 6.22. ANOVA Table for All Eight Subfamilies with mutual KNN $K = 50\%$.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	469.2121	**9.19**	234.6061	301.6215	4.65
B	843.7189	**16.52**	29.0938	37.4044	1.74
C	31.0993	0.61	31.0993	39.9829	6.69
AB	667.6988	**13.08**	11.5120	14.8005	2.30
BC	338.5828	**6.63**	11.6753	15.0103	1.74
ABC	232.3492	4.55	4.0060	5.1503	2.30
Error	2520.1241	**49.35**	0.7778	-	-

Laplacian; 3.16%, by the similarity measure; and 1.62%, by the graph Laplacian. With both graph construction methods, the similarity matrix is the most important individual factor. The best similarity matrix is *ML_ AS_ value*, followed by *AS_ GC_ value* and *AS_ value*. Using the initial matrices has, again, a positive effect on EP-Ratio.

Clustering Families

Table 6.23 shows the analysis of variance for clustering all three families using mutual KNN with $K = 50\%$ of the neighbors. Factor C (i.e., the graph Laplacian) has been omitted because its effect is non-significant according to the F-test. The most important individual factor is the similarity matrix, responsible for and impressive mark of 65.03% of the variation in EP-Ratio. The best similarity matrices are *CL_ AS_ cutoff*, *CL_ AS_ GC_ cutoff*, *CL_ ML_ AS_ cutoff* and *CL_ ML_ AS_ GC_ cutoff*, closely followed by *AS_ GC_ value* and *AS_ value*. Using the initial matrices has a positive effect on EP-Ratio, again suggesting adding information improves the clustering results.

Table 6.23. ANOVA Table for All Three Families with mutual KNN $K = 50\%$.

Factor	SS	% Explained	MS	F-Test	
				Calculated	Table
A	2189.7551	**9.53**	1094.8776	1784.3894	4.65
B	14940.4138	**65.03**	515.1867	839.6315	1.74
AB	2526.6426	**10.99**	43.5628	70.9969	2.30
AC	19.8925	0.09	9.9463	16.2100	4.65
BC	436.5224	1.89	15.0525	24.5319	1.74
ABC	874.2677	3.81	15.0736	24.5663	2.30
Error	1988.0208	**8.65**	0.6136	-	-

When using mutual KNN with $K = 25\%$ of the neighbors, which is the best graph construction method when clustering all three families, the effect of the graph Laplacian is significant and explains 4.98% of the variation in EP-Ratio. The percentages explained by the factors and interactions are different than when $K = 50\%$ of the neighbors: 22.33% of the variation is explained by the interaction between similarity measure and similarity matrix; 22.27%, by the similarity matrix; 19.31%, by the triple interaction between the factors; 10.09%, by the interaction between similarity matrix and graph Laplacian; 9.47%, by the similarity measure; 6.76% is unexplained and attributed to experimental errors; and 4.79% is explained by the interaction between similarity measure and graph Laplacian. The similarity matrix is the most important individual parameter for both graph construction methods. The best similarity matrix in this case is *ML_GC_value*, followed by *ML_AS_value* and *AS_GC_value*. The initial matrices have a negative value, which shows that some of the "informed" similarity matrices yield worse results.

Discussion

All the above results show that the factor that explains most of the variation in EP-Ratio is the similarity matrix. In each case there is a specific matrix that presents the largest negative effect, i.e., is the best option. However, the best matrix is always one that involves additional information other than the basic sequence similarity represented by the normalized BLOSUM62, PAM30, or Hamming scores. In most cases, the effect of using the initial similarity matrices (i.e., those containing purely normalized BLOSUM62, PAM30, or Hamming similarities) increases the EP-Ratio, which implies that they lead to poor quality results in comparison to the matrices containing domain knowledge. This attests to the importance of introducing information from different sources and domains to the problem of clustering enzymes by specificity, since sequence similarity alone does not imply functional similarity, let alone substrate specificity.

The results indicate that using the actual values is the best of the three strategies for integrating the additional data sources we considered, since most of the similarity matrices that yield the best results were constructed by adding the data in this way. Also, the active site information stands out, since it is always involved in the best similarity matrices. Since the enzyme's active site is the area where the reaction it catalyzes takes place, this information is very important and valuable to the problem of clustering enzymes by substrate specificity. This serves to underline the importance of adding information to the process other than the basic sequence similarity.

Chapter 7

Conclusions

Bioinformatics is an active research field with virtually endless data sources, since new information on biological processes is constantly being discovered. Unfortunately, this invaluable data are scattered throughout the World Wide Web, so that combining the various information sources is a challenge in itself.

Despite the various structural genomics initiatives, which aim at obtaining protein structures in large scale, the structures of the majority of newly discovered enzymes will remain unknown for a long time. In this master's thesis, we use constrained and spectral clustering techniques to integrate multiple data sources without the need for a full complex and expensive a priori data integration process. This kind of method, which combines information from diverse and possibly incomplete sources and knowledge domains, is of great importance for annotating (i.e., predicting the function of) newly discovered enzymes, especially when the more common sequence data are used as basis. This is due to the fact that an enzyme's function is determined by many factors besides its amino acid sequence. Therefore, the use of information from different and complementary knowledge domains can likely improve the quality of the annotations.

In this work we verified that the use of such additional information, which in our case involves three-dimensional structure, genomic context and active site data, is in fact able to improve the quality of the results of clustering enzymes based on sequence information. Our experimental results for both constrained and spectral clustering show that the additional data source which yields the largest improvements compared to using only the main sequence-based dataset is the one containing active site information. This may be explained by the fact that an enzyme's active site is directly related to its function, as well as its substrate specificity. Active site information even leads to

perfect clusters when clustering all three families using active sites as attributes for both constrained and unconstrained K-Medoids, which shows the quality of active site information for clustering enzyme families. When clustering the enzymes into subfamilies, though, even when the valuable active site information is used as attributes, integrating information from other knowledge domains still improves the clustering quality, showing that the more information is integrated, the better the results. Unfortunately, this highly useful active site information is often unavailable.

The structural alignment-based cannot-link constraints also stand out, since they are often involved in the best constraint sets and similarity matrices when all three families are combined, even leading to perfect clusters when clustering families using constrained K-Medoids. This attests to the quality of introducing structural information to a sequence-based dataset, since structure is much more related to function than sequence, and function, in turn, is what determines families and subfamilies. These results also suggest that the more constraints and, consequently, the more domain knowledge that is integrated to the problem of clustering enzymes, the better the quality of the results. Also, the spectral clustering results suggest that integrating the domain knowledge via the values in the additional dataset is better than using cutoffs to boost pairwise similarity or to alter edges in the similarity graphs.

This framework is valuable and appropriate for this application scenario because it allows using the most readily available information (i.e., the amino acid sequences) as foundation, while the additional information is integrated in order to improve the clustering quality, even if such information is limited to a subset of the original dataset.

Future Work

As future work, we intend to collect all reviewed enzymes with known Enzyme Commission numbers and structures in order to perform large-scale experiments, as well as validate the constraint generation strategies, especially those involving cutoffs. Also, we intend to apply the proposed strategies to the EzCatDB database [57], which classifies enzyme catalytic mechanisms based on data derived from PDB entries.

Additionally, we intend to expand this research to flexible constraints and to other domains of biological information such as substrate-product pairs, enzyme folding patterns and ligand binding sites. Also, we intend to apply other types of clustering algorithms such as COBWEB [24], which is a hierarchical clustering technique, and adapt them to use constraints.

Bibliography

[1] Altschul, S., Madden, T., Schaffer, A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25(17):3389–3402.

[2] Arakaki, A. K., Huang, Y., and Skolnick, J. (2009). EFICAz2: enzyme function inference by a combined approach enhanced by machine learning. *BMC Bioinformatics*, 10:107.

[3] Arakaki, A. K., Tian, W., and Skolnick, J. (2006). High precision multi-genome scale reannotation of enzyme function by EFICAz. *BMC Genomics*, 7:315.

[4] Armon, A., Graur, D., and Ben-Tal, N. (2001). ConSurf an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. *J. Mol. Biol.*, 307(1):447–463.

[5] Bar-Hillel, A., Hertz, T., Shental, N., and Weinshall, D. (2005). Learning a mahalanobis metric from equivalence constraints. *J. Mach. Learn. Res.*, 6:937–965.

[6] Basu, S., Bilenko, M., and Mooney, R. J. (2004). A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 59–68, Seattle, WA.

[7] Basu, S., Davidson, I., and Wagstaff, K. L. (2008). *Constrained Clustering: Advances in algorithms, theory, and applications.* Chapman & Hall/CRC Press, Boca Raton.

[8] Bernardes, J. S., Fernandez, J. H., and Vasconcelos, A. T. R. (2008). Structural descriptor database: a new tool for sequence-based functional site prediction. *BMC Bioinformatics*, 9:492.

[9] Bilenko, M., Basu, S., and Mooney, R. J. (2004). Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-First International Conference on Machine Learning*. ACM.

[10] Boeckmann, B., Bairoch, A., Blatter, R. A. M. C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., and M., M. S. (2003). The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, 31:365–370.

[11] Brown, S. D., Gerlt, J. A., Seffernick, J. L., and Babbitt, P. C. (2006). A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biol.*, 7:R8.

[12] Capra, J. A. and Singh, M. (2008). Characterization and prediction of residues determining protein functional specificity. *Bioinformatics*, 24(13):1473–1480.

[13] Casari, G., Sander, C., and Valencia., A. (1995). A method to predict functional residues in proteins. *Nat. Struct. Biol.*, 2(2):171–178.

[14] Chakrabarti, S., Bryant, S. H., and Panchenko, A. R. (2007). Functional specificity lies within the properties and evolutionary changes of amino acids. *J. Mol. Biol.*, 373(3):801–810.

[15] Chakrabarti, S. and Panchenko, A. R. (2009). Coevolution in defining the functional specificity. *Proteins*, 75(1):231–240.

[16] Chothia, C. (1992). One thousand families for the molecular biologist. *Nature*, 357:543–544.

[17] Davidson, I. and Basu, S. (2007). A survey of clustering with instance level constraints. *ACM Trans. Knowl. Discov. Data*, w:1–41.

[18] Davidson, I. and Ravi, S. S. (2005). Clustering with constraints: Feasibility issues and the k-means algorithm. In *Proceedings of the 2005 SIAM International Conference on Data Mining*, pages 138–149, Newport Beach, CA.

[19] Davidson, I. and Ravi, S. S. (2006). Identifying and generating easy sets of constraints for clustering. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*.

[20] Davidson, I. and Ravi, S. S. (2007). The complexity of non-hierarchical clustering with instance and cluster level constraints. *Data Min. Knowl. Disc.*, 14:25–61.

[21] Deshpande, N., Addess, K. J., Bluhm, W. F., Merino-Ott, J. C., Townsend-Merino, W., Zhang, Q., Knezevich, C., Xie, L., Chen, L., Feng, Z., Green, R. K., Flippen-Anderson, J. L., Westbrook, J., Berman, H. M., and Bourne, P. E. (2005). The RCSB protein data bank: a redesigned query system and relational database based on the mmCIF schema. *Nucleic Acids Res.*, 33:D233–D237.

[22] Do, H. H. and Rahm, E. (2004). Flexible integration of molecular-biological annotation data: The GenMapper approach. In Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., and Ferrari, E., editors, *Advances in Database Technology - EDBT 2004*, volume 2992 of *Lecture Notes in Computer Science*, pages 557–558. Springer Berlin / Heidelberg.

[23] Dundas, J., Ouyang, Z., Tseng, J., Binkowski, A., Turpaz, Y., and Liang, J. (2006). CASTp: computer atlas of surface topography of proteins with structural and topographical mapping of functionally annotated residues. *Nucleic Acids Res.*, 34(Web Server Issue):W116–W118.

[24] Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. *Mach. Learn.*, 2:139–172.

[25] Freier, A., Hofestädt, R., Lange, M., Scholz, U., and Stephanik, A. (2002). BioDataServer: A SQL-based service for the online integration of life science data. *In Silico Biol.*, 2(2):37–57.

[26] Freilich, S., Spriggs, R. V., George, R. A., Al-Lazikani, B., Swindells, M., and Thornton, J. M. (2005). The complement of enzymatic sets in different species. *J. Mol. Biol.*, 349(4):745–763.

[27] Friedberg, I. (2006). Automated protein function prediction: the genomic challenge. *Brief. Bioinform.*, 7(3):225–242.

[28] Goder, A. and Filkov, V. (2008). Consensus clustering algorithms: Comparison and refinement. In *Proceedings of the Ninth SIAM Workshop on Algorithm Engineering and Experiments (ALENEX)*.

[29] Goldenberg, O., Erez, E., Nimrod, G., and Ben-Tal, N. (2009). The ConSurf-DB: pre-calculated evolutionary conservation profiles of protein structures. *Nucleic Acids Res.*, 37(Database Issue):D323–D327.

[30] Gong, S. and Blundell, T. L. (2008). Discarding functional residues from the substitution table improves prediction of active sites within three-dimensional structures. *PLoS Comput. Biol.*, 4(10):e1000179.

[31] Guilloux, V. L., Schmidtke, P., and Tuffery, P. (2009). Fpocket: An open source platform for ligand pocket detection. *BMC Bioinformatics*, 10:168.

[32] Han, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd edition.

[33] Hannenhalli, S. S. and Russell, R. B. (2000). Analysis and prediction of functional sub-types from protein sequence alignments. *J. Mol. Biol.*, 303:61–76.

[34] Henikoff, S., Greene, E. A., Pietrokovski, S., Bork, P., Attwood, T. K., and Hood, L. (1997). Gene families: the taxonomy of protein paralogs and chimeras. *Science*, 278:609–614.

[35] Henschel, A., Winter, C., Kim, W. K., and Schroeder, M. (2007). Using structural motif descriptors for sequence-based binding site prediction. *BMC Bioinformatics*, 8(Suppl. 4):S5.

[36] Huynen, M., Snel, B., III, W. L., and Bork, P. (2000). Predicting protein function by genomic context: Quantitative evaluation and qualitative inferences. *Genome Res.*, 10:1204–1210.

[37] Jain, E., Bairoch, A., Duvaud, S., Phan, I., Redaschi, N., Suzek, B. E., Martin, M. J., McGarvey, P., and Gasteiger, E. (2009). Infrastructure for the life sciences: design and implementation of the UniProt website. *BMC Bioinformatics*, 10:136.

[38] Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling*. Wiley - Interscience.

[39] Jeanmougin, F., Thompson, J. D., Gouy, M., Higgins, D. G., and Gibson, T. J. (1998). Multiple sequence alignment with Clustal X. *Trends Biochem. Sci.*, 23:403--405.

[40] Kitson, D. H., Badretdinov, A., Zhu, Z. Y., Velikanov, M., Edwards, D. J., Olszewsk, K., Szalma, S., and Yan, L. (2002). Functional annotation of proteomic sequences based on consensus of sequence and structural analysis. *Brief. Bioinform.*, 3(1):32–44.

[41] Klein, D., Kamvar, S. D., and Manning, C. D. (2002). From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 307–313.

[42] Kotera, M., Okuno, Y., Hattori, M., Goto, S., and Kanehisa, M. (2004). Computational assignment of the EC numbers for genomic-scale analysis of enzymatic reactions. *J. Am. Chem. Soc.*, 126:16487–16498.

[43] Kristensen, D., Chen, B. Y., Fofanov, V. Y., Ward, R. M., Lisewske, A. M., K., M., Kavraki, L. E., and Lichtarge, O. (2006). Recurrent use of evolutionary importance for functional annotation of proteins based on local structural similarity. *Protein Sci.*, 15(6):1530–1536.

[44] Kristensen, D. M., Ward, R. M., Lisewski, A. M., Erdin, S., Chen, B. Y., Fofanov, V. Y., Kimmel, M., Kavraki, L. E., and Lichtarge, O. (2008). Prediction of enzyme function based on 3D templates of evolutionarily important amino acids. *BMC Bioinformatics*, 9:17.

[45] Lange, T., Law, M. H. C., Jain, A. K., and Buhmann, J. M. (2005). Learning with constrained and unlabelled data. In *Proceedings of the 2005 IEEE Conference on Computer Vision and Patter Recognition*.

[46] Langraf, R., Xenarios, I., and Eisenberg, A. (2001). Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins. *J. Mol. Biol.*, 307(5):1487–1502.

[47] Li, L., Shiga, M., Ching, W. K., and Mamitsuka, H. (2010). Annotating gene functions with integrative spectral clustering on microarray expressions and sequences. *Genome Inform. Ser.*, 22:95–120.

[48] Lichtarge, O., Bourne, H. R., and Cohen, F. E. (1996). An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.*, 257(2):342–358.

[49] Liu, Y., Eyal, E., and Bahar, I. (2008). Analysis of correlated mutations in HIV-1 protease using spectral clustering. *Bioinformatics*, 24:1243–1250.

[50] Livingstone, C. D. and Barton, G. J. (1993). Protein sequence alignments: a strategy for the hierarchical analysis of residue conservation. *Comput. Appl. Biosci.*, 9(6):745–756.

[51] Lu, Z. and Leen, T. K. (2005). Semi-supervised learning with penalized probabilistic clustering. *Adv. Neur. In,*, 17:849–856.

[52] Madabushi, S., Yao, H., Marsh, M., Kristensen, D. M., Philippi, A., Sowa, M. E., and Lichtarge, O. (2002). Structural clusters of evolutionary trace residues are statistically significant and common in proteins. *J. Mol. Biol.*, 316(1):139–154.

[53] Minardi, R. C. M., Bastard, K., and Artiguenave, F. (2010). Structure-based protein family clustering and prediction of specificity-determining position patterns. *Bioinformatics*, 26(24):3075–3082.

[54] Moses, V., Springham, D. G., and Cape, R. (1999). *Biotechnology: the science and the business*. Taylor & Francis.

[55] Moss, G. P. (2010). Enzyme nomenclature. `http://www.chem.qmul.ac.uk/iubmb/enzyme/`.

[56] Moult, J. (2005). A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr. Opin. Struc. Biol.*, 15:285–289.

[57] Nagano, N. (2005). EzCatDB: the enzyme catalytic-mechanism database. *Nucleic Acids Res.*, 33 (s1):D407–D412.

[58] Najmanovich, R., Kurbatova, N., and Thornton, J. (2008). Detection of 3D atomic similarities and their use in the discrimination of small molecule protein-binding sites. *Bioinformatics*, 26(16):i105–i111.

[59] Nelson, D. L. and Cox, M. M. (2004). *Lehninger Principles of Biochemistry*. W. H. Freeman.

[60] Pazos, F., Rausell, A., and Valencia, A. (2006). Phylogeny-independent detection of functional residues. *Bioinformatics*, 22(12):1440–1448.

[61] Pei, J., Cai, W., Kinch, L. N., and Grishin, N. V. (2006). Prediction of functional specificity determinants from protein sequences using log-likelihood ratios. *Bioinformatics*, 22(2):164–171.

[62] Pensa, R. G., Robardet, C., and Boulicaut, J. F. (2008). Constraint-driven co-clustering of 0/1 data. In *Constrained Clustering: Advances in algorithms, theory, and applications*, pages 123–148. CRC Press.

[63] Perkins, A. D. and Langston, M. A. (2009). Threshold selection in gene co-expression networks using spectral graph theory techniques. *BMC Bioinformatics*, 10:S4.

[64] Pupko, T., Bell, R. E., Mayrose, Y., Glaser, F., and Ben-Tal, N. (2002). Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants with their homologues. *Bioinformatics*, 18(Suppl. 1):S71–77.

[65] Rost, B. (2002). Enzyme function less conserved than anticipated. *J. Mol. Biol.*, 318:595--608.

[66] Rother, K., Müller, H., Trissl, S., Koch, I., Steinke, T., Preissner, R., Frömmel, C., and Leser, U. (2004). COLUMBA: Multidimensional data integration of protein annotations. In Rahm, E., editor, *Data Integration in the Life Sciences*, volume 2994 of *Lecture Notes in Computer Science*, pages 156–171. Springer Berlin / Heidelberg.

[67] Salem, S., Zaki, M. J., and Bystroff, C. (2009). Interative non-sequential protein structural alignment. *J. Bioinformatics and Computational Biology*, 7:571--596.

[68] Schnoes, A. M., Brown, S. D., Dodevski, I., and Babbitt, P. C. (2009). Annotation error in public databases: Misannotation of molecular function in enzyme superfamilies. *PLoS Comput. Biol.*, 5(12):e1000605.

[69] Schönhuth, A., Costa, I. G., and Schliep, A. (2009). Semi-supervised clustering of yeast gene expression data. In *Cooperation in Classification and Data Analysis*. Springer.

[70] Sese, J., Kurokawa, Y., Monden, M., Kata, K., and Morishita, S. (2004). Constrained clusters of gene expression profiles with pathological features. *Bioinformatics*, 20(17):3137–3145.

[71] Shaban-Nejad, A., Baker, C., Haarslev, V., and Butler, G. (2005). The FungalWeb ontology: Semantic web challenges in bioinformatics and genomics. In Gil, Y., Motta, E., Benjamins, V., and Musen, M., editors, *The Semantic Web*, volume 3729, pages 1063–1066. Springer Berlin / Heidelberg.

[72] Shatsky, M., Nussinov, R., and Wolfson, H. (2002). Multiprot - a multiple protein structural alignment algorithm. In Guidó, R. and Gusfield, D., editors, *Algorithms in Bioinformatics*, volume 2452 of *Lecture Notes in Computer Science*, chapter 18, pages 235--250. Springer Berlin Heidelberg.

[73] Shental, N., Bar-Hillel, A., Hertz, T., and Weinshall, D. (2004). Computing gaussian mixture models with EM using equivalence constraints. *Adv. Neur. In,*, 16:465–472.

[74] Silveira, C. H., Pires, D. E. V., Minardi, R. C. M., Veloso, C. J. M., Lopes, J. C. D., Meira, W., Neshich, G., Ramos, C. H. I., Habesch, R., and Santoro, M. M. (2009). Protein cutoff scanning: A comparative analysis of cutoff dependent and cutoff free methods for prospecting contacts in proteins. *Proteins*, 74:727–743.

[75] Sol, A. D., Pazos, F., and Valencia, A. (2003). Automatic methods for predicting functionally important residues. *J. Mol. Biol.*, 326(4):1289–1302.

[76] Tian, W., Arakaki, A. K., and Skolnick, K. (2004). EFICAz: a comprehensive approach for accurate genome-scale enzyme function inference. *Nucleic Acids Res.*, 32(21):6226–6239.

[77] Tian, W. and Skolnick, J. (2003). How well is enzyme function conserved as a function of pairwise sequence identity? *J. Mol. Biol.*, 333:863–882.

[78] Tramontano, A., Leplae, R., and Morea, V. (2001). Analysis and assessment of comparative modeling predictions in CASP4. *Proteins*, Suppl. 5:22–38.

[79] Tramontano, A. and Morea, V. (2003). Assessment of homology-based predictions in CASP5. *Proteins*, 53 (Suppl. 6):652–368.

[80] Tseng, Y. Y., Dundas, J., and Liang, J. (2009). Predicting protein function and binding profile via matching of local evolutionary and geometric surface patterns. *J. Mol. Biol.*, 387(2):451–464.

[81] von Luxburg, U. (2007). A tutorial on spectral clustering. *Stat. Comput.*, 17(4):395–416.

[82] Wagstaff, K. and Cardie, C. (2000). Clustering with instance-level constraints. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 1103–1110.

[83] Wagstaff, K., Cardie, C., Rogers, S., and Schroedl, S. (2001). Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584.

[84] Wagstaff, K. L. (2002). *Intelligent Clustering with Instance-Level Constraints.* PhD thesis, Cornell University.

[85] Wagstaff, K. L., Basu, S., and Davidson, I. (2006). When is constrained clustering beneficial, and why? In *Proceedings of the Twenty-First National Conference on Artificial Intelligence.*

[86] Ward, R. M., Venner, E., Daines, B., Murray, S., Erdin, S., Kristensen, D. M., and Lichtarge, O. (2009). Evolutionary trace annotation server: automated enzyme function prediction in protein structures with 3D templates. *Bioinformatics*, 25(11):1426–1427.

[87] White, R. H. (2006). The difficult road from sequence to function. *J. Bacteriol.*, 188(10):3431–3432.

[88] Wittig, U., Golebiewski, M., Kania, R., Krebs, O., Mir, S., Weidemann, A., Anstein, S., Saric, J., and Rojas, I. (2006). SABIO-RK: Integration and curation of reaction kinetics data. In Leser, U., Naumann, F., and Eckman, B., editors, *Data Integration in the Life Sciences*, volume 4075, pages 94–103. Springer Berlin / Heidelberg.

[89] Xing, E. P., Ng, A. Y., Jordan, M. I., and Russel, S. (2003). Distance metric learning, with application to clustering with side-information. *Adv. Neur. In,*, 15:505–512.

[90] Yamanishi, Y., Hattori, M., Kotera, M., M., S. G., and Kanehisa (2009). E-zyme: predicting potential EC numbers from the chemical transformation pattern of substrate-product pairs. *Bioinformatics*, 25:i179–i186.

[91] Yamanishi, Y., Vert, J. P., and Kanehisa, M. (2004). Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20:i363–i370.

[92] Yamanishi, Y., Vert, J. P., and Kanehisa, M. (2005). Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics*, 21:i468–i477.

[93] Yu, G. X., Park, B. H., Chandramohan, P., Munavalli, R., Geist, A., and Samatova, N. F. (2005). In silico discovery of enzyme-substrate specificity-determining residue clusters. *J. Mol. Biol.*, 352(5):1105–1117.

[94] Zeng, E., Yang, C., Li, T., and Narasimhan, G. (2007). On the effectiveness of constraints sets in clustering genes. In *Proceedings of the Seventh IEEE International Conference on Bioinformatics and Bioengineering*, pages 79–86.