Data Integration via Constrained Clustering: An Application to Enzyme Clustering

Elisa Boari de Lima* Raquel Cardoso de Melo Minardi* Wagner Meira Jr.*
Mohammed Javeed Zaki†

Abstract
When multiple data sources are available for clustering, an a priori data integration process is usually required. This process may be costly and may not lead to good clusterings, since important information is likely to be discarded. In this paper we propose constrained clustering as a strategy for integrating data sources without losing any information. It basically consists of adding the complementary data sources as constraints that the algorithm must satisfy.

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 clustering as a means of integrating information from diverse sources as constraints, and analyze how this additional information impacts clustering quality in an enzyme clustering application scenario. Our results show that constraints generally improve the clustering quality when compared to an unconstrained clustering algorithm.

Keywords: constrained clustering, data integration, enzyme clustering.

1 Introduction

In recent years, there has been a general increase in the amount of data publicly available worldwide. This is true for various research areas, 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 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 can vastly benefit from clustering techniques, since they allow the creation of groups of similar proteins which 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 gathered in so many different data sets 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 3D structural data, can have dozens or even hundreds of entries for the same molecule. The various data sources call for data integration, which is a complex process due to the various issues that must be addressed. Inconsistencies and redundancies, for example, may arise because attributes representing a given concept may have different names in different databases. Conflicts between data values may also exist, since different sources may have different attribute values for the same real-world object, due to different representations, scaling or encoding [10].

This a priori data integration is hard and may not lead to good clustering results, since important information is likely to 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.

In this work we use constrained clustering techniques as a means of integrating information from diverse sources in the form of constraints, 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 mul-

[*] Department of Computer Science, Universidade Federal de Minas Gerais, Brazil – {eblima, raquelcm, meira}@dcc.ufmg.br

[†] Computer Science Department, Rensselaer Polytechnic Institute, USA – zaki@cs.rpi.edu

tiple 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. Our purpose is to analyze how the integration of various data sources in the form of constraints affects the success of enzyme clustering, which might lead to important information about the functions and structures of the enzymes, as well as functional diversification acquired throughout family evolution.

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 manner in which we gathered such information and transformed it into constraint sets for this particular biological problem; and, most importantly, the possibility of using constraints to cluster enzymes.

2 Related Work

Clustering is a data mining technique that groups similar objects without any supervised information. However, in various applications there is access to additional information or domain knowledge about the types of groups that are sought in the data, such as complementary information about actual similarities between object pairs. Constrained clustering emerged from the need for ways to accommodate such information when available [3]. It can be seen as the process of dividing a set of N objects in some D-dimensional space into K groups, each of which represent a significant subpopulation, 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, whereas in the latter case the idea is to satisfy as many as possible, but not necessarily all of them. In this work we are considering strict instance-level constraints. A set of instance-level constraints C consists of declarations about pairs of objects. A must-link constraint $c_=(i, j)$ indicates that instances i and j must be assigned to the same cluster, while a cannot-link constraint $c_{\neq}(i, j)$ implies they must be placed in different clusters.

Must-link constraints are symmetric, reflexive and transitive, which allows for additional constraints to be inferred [4, 25]. Cannot-link constraints are not transitive, however additional cannot-link constraints can be inferred from the set of must-link constraints. Considering a graph whose nodes are instances of the dataset, whose edges (i, j) represent a must-link constraint between instances i and j, and considering CC_1 and CC_2 to be connected components of this graph, it follows that:

1. 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 $b \in CC_2$;
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 $b \in CC_2$ [8].

Initial research in the field of constrained clustering proposed algorithms that are able to incorporate pairwise constraints on whether or not instances belong to clusters, and to learn distance metrics specific to the problem that lead to a desirable clustering. The field has expanded to include algorithms that use many other types of domain knowledge to aid the clustering [3].

The first research in the field proposed a modified version of COBWEB that imposes strict pairwise constraints [24], followed by COP-KMeans [25], a constrained version of the well known K-Means clustering algorithm. Shental et al. [21] explored a constrained version of the Expectation Maximization (EM) clustering algorithm. To accommodate constraint noise or uncertainty, other methods aim at satisfying as many constraints as possible, but not necessarily the entire constraint set [2, 7, 26].

In recent years, much research has been done incorporating instance-level constraints to clustering methods [1, 4, 13, 25, 27]. This semi-supervised approach has led to improved performance in various real-world applications such as resolving co-references in nominal phrases, refinement of GPS-based maps [25], and person identification from surveillance videos [1].

Two main advantages of using constraints reported in the literature are the improvement of the precision to predict labels for all instances when constraints are generated from few labeled data, and generation of clusters with desirable geometric properties [8]. Many researchers have shown that as the number of constraints increases, the precision of the clustering increases as well [1, 7, 13, 15, 25, 26]. When considering the average of several constraint sets, the performance on label prediction is typically higher than when constraints are not used [8].

Among the research that apply constrained clustering to biological problems, Zeng et al. [28] investigate 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 adapt MPCK-Means [4] and explore methods for automatically generating constraints from multiple sources of biological 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. Casari et al. [5] propose a sequence representation in the form of vectors and use techniques of dimensionality reduction to project such vectors in fewer dimensions and detect subgroups and functionally important residues.

A technique that is 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 clustering, which would bring out the common organization in the different data sets and reveal significant differences among them [9]. The main distinction between constrained and consensus clustering is that the first uses various data sources as constraints to produce a single clustering, whereas the latter combines different clusterings into a single result.

3 Application Scenario

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. Enzymes are a particular category of proteins that catalyze chemical reactions, converting a set of substrates in a set of products. There are different classifications and nomenclatures for enzymes. 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.

We want to analyze how the use of different data sources as constraints 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, in which case we aim at their ability to recognize different substrates. These are all challenging problems.

Enzyme Commission (EC) numbers are a numerical classification scheme for enzymes based on the chemical reactions they catalyze. As a system of enzyme nomenclature, every EC number is associated with a recommended name for the respective enzyme [17] and specifies enzyme-catalyzed reactions instead of the enzymes themselves. The four numbers that compose the EC number represent a progressively finer enzyme classification. Therefore, if we wanted to predict a given enzyme's EC number, for instance, we could do it at four levels. Predicting the first level is the easiest, since it could be done by detecting a remote homology. But predicting the fourth level is extremely difficult. Schnoes et al. [22] estimate that 85% of the annotation errors are caused by overprediction, which implies that the errors are located in the lower level of the EC number. Our two approaches involving subfamily clustering focus on this problem.

The understanding of molecular function may be greatly facilitated by structural information. 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. Nevertheless, there are computational methods that allow modeling proteins that have significant degree of identity with a protein of known structure.

Some concepts relevant to the understanding of this work and of the data used to generate constraints are described in the remainder of this section.

Genomic Context. Proteins are chains of amino acid residues coded by genes, with each amino acid 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. 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 processes. 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. In this work, genomic context forms an additional data source incorporated in the form of constraints.

Sequence Alignments. Alignments are frequently used to compare biological sequences. A global pairwise alignment can be thought of as the process of sliding one sequence past the other until a good match is found [18]. In case the amino acid residues in the two sequences are identical in a given position, a positive score is assigned. 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 can 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 [18]. In this work multiple sequence alignments are used as attributes, and BLOSUM62 and PAM30 substitution matrices are used as similarity measures for the clustering algorithms.

Protein Structure. The amino acid 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 3D structures, still maintaining function. Therefore, protein structures are more conserved than sequences. Protein families with different folding patterns tend to have different functions. Cutoff Scanning [23] is a method that represents 3D structure as a histogram of the number of neighbors an atom has. Different families have distinct folding patterns and, consequently, characteristic histograms. Such histograms are used as attributes for the clustering algorithms in this work.

Structural Alignments. Protein structural alignments are frequently used to detect functional similarity. Analogous to the 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 [19]. We use structural alignments to create constraints for the clustering algorithms.

Active Sites. An enzyme's active site is the set of amino acid residues where the substrate binds and the chemical reaction takes place. Chakrabarti & Panchenko [6] 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 for the clustering algorithms and for creating constraints.

4 Data Sources

In this work we study three enzyme families, namely nucleotidyl cyclases, protein kinases and serine proteases, which have varying numbers 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.

Our main database consists of the enzymes with known Enzyme Commission (EC) numbers. The information provided by Moss [17] was used to define the EC numbers for each of our enzyme families, as shown Table 1. In order to achieve a more reliable dataset to use as ground truth for analyzing the clustering results, we decided to work only with the enzymes that had a reviewed status in the UniProt repository, 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 1: EC Numbers according to Moss [17].

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 2 shows the number of enzymes in each subfamily after this filtering process. The resulting databases comprise of each enzyme's Universal Protein Resources (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.

Table 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

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

4.1 Additional Data Sources

Besides the amino acid sequences, additional information on each enzyme in the database was gathered. The first external piece of data is the enzyme's tertiary structure model, provided by Minardi et al. [16], who also supplied the active site of each enzyme. The aligned active sites were obtained using Fpocket [14], a software that calculates structural cavities, and MultiProt [20], 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, the 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 ten proteins besides the enzyme itself. Unfortunately, we were not able 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 [23] 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 atoms are positioned in the 3D structure. Such vectors comprise important information that is complementary to the amino acid sequences.

5 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, we also clustered all of them together.

5.1 Structural Alignment-Based Constraints

MultiProt [20] 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, 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 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 in this work we are considering strict constraints (i.e. constraints that *must* be satisfied), this search was for the cutoffs that did not yield any false positives.

To generate must-link constraints, the following statement must hold: "If the RMSD of an enzyme pair is *at most* X, then the enzymes belong to the same subfamily". Therefore, a must-link constraint exists between them, thus placing both enzymes in the same cluster. Similarly, for cannot-link constraints it must hold that "If the RMSD 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 3 presents the zero false positive cutoffs for each family for creating must-link (ML) and cannot-link (CL) constraints.

Table 3: RMSD cutoffs that yield zero false positives 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 apply to all three families as well as to the original dataset, which also includes unreviewed enzymes. Therefore, we use $RMSD \leq 0.25$ for creating must-link constraints and $RMSD \geq 1.62$ for cannot-link constraints. Table 4 presents the number of constraints generated for each enzyme family before and after expanding the constraint set using the aforementioned transitivity property.

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. However, when considering all three families combined, several cannot-link constraints are created between pairs of enzymes belonging to different families and, consequently, to different subfam-

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

Table 4: Number of structural alignment-based constraints created for each family.

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

ilies. We could have lowered the cannot-link cutoff so as to create constraints inside each family, but that would go against our idea of generality because if a lower cutoff had been applied to the original set of enzymes, which also includes automatically annotated enzymes, we would end up creating cannot-link constraints between enzymes annotated as belonging to the same subfamily. This would add contradiction into the constraint sets making the strict constraint approach intractable, as it would be impossible to satisfy all constraints.

5.2 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 shows the number of constraints created for each family before and after expanding the constraint sets with the transitivity property.

Table 5: Number of genomic context-based must-link constraints created for each family.

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

Very few genomic context-based must-link constraints are created due to the fact that most of the enzymes 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.

5.3 Active Site-Based Constraints Since the active sites of all the enzymes in a given family are aligned, they all have the same number of amino acid 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 identical active sites. A 30% cutoff would create some constraints exclusively 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 be created between enzymes annotated as being in the same subfamily, thus introducing contradiction into the constraint set and making the problem intractable. Since a lower cutoff would not create any cannot-link constraints at all, only must-link constraints are created using active site information.

Since the number of amino acid residues in the active sites slightly varies between families, Clustal X [12] was used to perform Multiple Sequence Alignment (MSA) of the active sites so we could be able to apply the above strategies when clustering the three families altogether. Table 6 presents the number of constraints created in each case for each family, before and after expanding the constraint sets with the transitivity property.

Table 6: Number of active site-based must-link constraints created for each family.

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

6 Constrained Clustering Algorithms

In order to analyze the effect of integrating multiple data sources as constraints into the clustering process, 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 & Kamber [10]. The constrained versions for both algorithms are adapted from COP-KMeans [25].

6.1 K-Medoids We implement the constrained and unconstrained versions of K-Medoids because the main data source, which consists of the amino acid sequences, is categorical. Clustering algorithms require that all objects have the same number of attributes in order for the difference between each attribute of an object pair to have a meaning. Since the lengths of the sequences vary, we performed multiple sequence alignments (MSAs) between all sequences in each family using Clustal X [12], and the results are used as the enzymes' attributes for the clustering algorithm. A MSA was also performed between all enzyme sequences in all three families in order to apply K-Medoids to the problems of clustering families and clustering subfamilies inside multiple families.

Using the multiple sequence alignments 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 of the MSAs. 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 the 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 amino acids excluding gaps.

6.2 K-Means We implement K-Means so as to take advantage of the information provided by the previously described Cutoff Scanning [23] vectors, which are created using a step of 0.2Å varying from 0Å to 30Å, totalizing 151 distances. Therefore, each vector has 151 positions, with the first element being the number of amino acid residue pairs in the enzyme's 3D 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. For these numerical attributes we use squared Euclidean distance as the distance measure for K-Means.

7 Results and Discussion

Both K-Medoids and K-Means are performed with all constraint sets and combinations thereof. We study each constraint set separately in order to assess the effect that the corresponding data source has on clustering quality. Additionally, the constraint set combinations are studied in order to analyze the improvement multiple additional data sources provide, as well as the effect of increasing the number of constraints.

Parameter Settings. As previously described in Section 6, the attributes (feature vectors) used by K-Medoids are either the results of multiple sequence alignments or the aligned active sites. The complement of the Hamming distance, and BLOSUM62 and PAM30 substitution matrices are used by K-Medoids as similarity measures. For K-Means, the attributes are the 151-dimension distance vectors, with the attributes being either normalized or not, and squared Euclidean distance is used as the algorithm's distance metric.

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. When clustering all three families combined, we analyze results for $K = 8$, which is the total number of subfamilies, as well as the results for $K = 3$ so as to verify the performance of the algorithms at separating the three enzyme families.

Evaluation Criteria. Since we are comparing the clustering results to the ground truth provided by the EC numbers, we use external validation to analyze the clusters. External criteria measure whether or not the objects are randomly structured. Three metrics are used 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 subfamilies): purity, entropy and EP-Ratio.

The probability that cluster c_i contains objects from partition p_j is given by $\rho_{ij} = \frac{|c_i \cap p_j|}{|c_i|}$. The 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: $P_C = \sum \frac{|c_i|}{|c|} P_i$, with $|c|$ being the total number of objects. 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 metrics are required to properly analyze the clusters.

Entropy is a popular supervised learning metric which measures the amount of uncertainty in the cluster assignment. The smaller the entropy, the better the clustering. The entropy E_i for each cluster is given by $E_i = -\sum \rho_{ij} \log_2 \rho_{ij}$, with the entropy for the entire clustering being the weighted sum of the individual cluster entropies: $E_C = \sum \frac{|c_i|}{|c|} E_i$.

The main metric we use for comparing results, which we call the clustering's EP-Ratio, is simply the ratio between entropy and purity. Therefore, the smaller the EP-Ratio, the better the clustering.

Assessment Methodology. In order to analyze the effect of each constraint set and combinations thereof, we compare the EP-Ratios for the thirty repetitions of the constrained algorithms against the EP-Ratios for the unconstrained algorithms. To do so, we use paired observations, performing a straightforward analysis: the two sets of thirty EP-Ratios are treated as one sample of thirty pairs, each corresponding to the same random seed in the constrained and unconstrained algorithms. We calculate the difference in EP-Ratios for each pair and then the confidence interval of these differences. If such confidence interval includes zero, the result of the constrained version is not significantly different from that obtained by the unconstrained version of the algorithm at the given level of confidence [11]. In this work we consider 99% and 95% confidence intervals.

Table 7 shows the codes used for each constraint set. All following tables present the average EP-Ratios of the constraint sets that yield results significantly different from those produced by unconstrained versions of the algorithms 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 7: 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

7.1 K-Medoids with MSA In this subsection we present and discuss the results of applying K-Medoids using the multiple sequence alignments as attributes. Each individual constraint set and each combination of constraint sets are employed.

Clustering subfamilies inside a single family. Although constrained versions of K-Medoids achieve improvements for the protein kinases and serine proteases enzyme 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 metrics at the 95% confidence level. 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 8. When the complement of the Hamming distance is used as the algorithm's similarity metric, 100% identical active site-based constraints and their combination with genomic context-based constraints actually yield worse results (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 (lower EP-Ratios) than the unconstrained version. When the similarity metric is either BLOSUM62 or PAM30, all four constraint sets yield better results than the unconstrained algorithm.

Table 8: 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**

Clustering subfamilies in multiple families. Table 9 presents the average EP-Ratios for the constraint sets with significant results when all three families are combined and $K = 8$, which is the total number of subfamilies. Unlike when clustering each family separately, when all are combined many structural alignment-based cannot-link constraints exist.

Table 9: 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**

When using BLOSUM62 as the algorithm's similarity metric, 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 metrics, all constraint sets lead to improved clusters, except when using genomic context-based constraints, structural similarity-based must-links and their combination, in which case the results are not significant. For Hamming, the results of using the combination of structural alignment-based and 100% identical active site-based constraints is significant at 95% confidence, but not at the 99% level.

Clustering families. When clustering in the search for the three families instead of the subfamilies

($K = 3$), the result of adding the cannot-link constraint set stands out. The constraint sets with significant results are shown in Table 10. The average EP-Ratio for unconstrained K-Medoids is fairly high, and most sets of must-link constraints improve it. But 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.

Table 10: 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

Summary. When clustering subfamilies in a single family, the significant improvements occur when using active site-based and genomic context-based constraint sets. This shows that using active sites obtained from available enzyme structures and the genomic context of the corresponding gene (when available) can aid the problem of predicting the fourth and most challenging level of the EC number. 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)[2] 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 using cannot-link constraints is very noticeable, even leading to perfect clusters in the first case. These structural alignment-based cannot-link constraints are very useful since they add structural information that the sequence-based attributes do not carry. This shows that structural information allows to separate (sub)families based on structural dissimilarity. It is likely that the 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, as previously discussed. Another factor that contributes to the larger effect of the cannot-link constraint set is simply the large number of constraints.

7.2 K-Medoids with Active Sites Since amino acid sequences are much more readily available than protein structures, using the result 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. So we perform a multiple sequence alignment in order to be able to compare amino acid residues at the same position of different active sites. In this case, active site-based constraints are not employed, since the information they would add is already embedded in the attributes.

Clustering subfamilies inside a single family. For the protein kinases family, only genomic context-based constraints were created, since none of the enzyme pairs has a RMSD smaller or equal to the cutoff employed. The results yielded by these constraints are not significantly different from those of the unconstrained algorithm for any of the similarity metrics. This can be explained by the very small number of constraints generated for this family.

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 metrics. 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. The average EP-Ratios for the constraint sets with significant results are shown in Table 11. All constraint sets produce significantly better results than the unconstrained K-Medoids for the serine proteases family, as shown in Table 12.

Clustering subfamilies in multiple families. Significantly better results are produced by all constraint sets when clustering all eight subfamilies, as shown in Table 13.

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

Table 11: 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 12: 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

Clustering families. When using active sites as attributes for clustering in the search for the three families, perfect clusters are obtained for all similarity metrics 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 is an effective way of separating enzyme families. Unfortunately, this information is rarely available.

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 proteins 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.

7.3 K-Means with Distance Arrays

In this subsection 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.

Clustering subfamilies inside a single family. 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 not normalized. However, when we normalize the attributes perfect clusters are achieved for both constrained and unconstrained versions of K-Means.

Table 13: 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

Table 14 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 yields 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 14: 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 a 95% confidence level using 95% identical active site-based constraints yields better results for both normalized and unnormalized attributes, as shown in Table 15. Using structural alignment-based constraints and their combination with 95% identical active site-based constraints yields worse results than the unconstrained K-Means.

Table 15: 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*

Clustering subfamilies in multiple families. 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 16. 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 at the 99% confidence level, especially when the cannot-link constraint set is applied.

Table 16: 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	-

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. When the attributes are normalized, all must-link constraint sets yield better results, as shown in Table 17. Whenever the cannot-link constraint set is employed, perfect clusters are achieved for 29 of the 30 repetitions.

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 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 and 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.

Table 17: 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
CL & GC	**0.1214**	**0.1214**
CL & 100AS	**0.1214**	**0.1214**
CL, GC & 100AS	**0.1214**	**0.1214**
CL & ML	**0.1214**	**0.1214**
CL, ML & GC	**0.1214**	**0.1214**
CL & 95AS	**0.1214**	**0.1214**
CL, ML & 100AS	**0.1214**	**0.1214**
CL, GC & 95AS	**0.1214**	**0.1214**
CL, ML, GC & 100AS	**0.1214**	**0.1214**
CL, ML & 95AS	**0.1214**	**0.1214**
CL, ML, GC & 95AS	**0.1214**	**0.1214**

8 Conclusions and Future Work

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

Despite the various initiatives in structural genomics 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 work we integrate multiple data sources in the form of constraints, so as to use their additional information without the need of a full a priori data integration process. This kind of method is of great importance for annotating newly discovered enzymes, especially when using sequence data as a basis, as we do by using multiple sequence alignments as attributes.

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

As future work, we intend to collect all reviewed enzymes with known EC numbers and structures in order to perform large-scale experiments, and to validate the constraint creation strategies, especially those that involve cutoff values, such as structural alignment-based constraints. We also intend to expand this research to flexible constraints and to use other types of clustering algorithms, such as COBWEB and spectral clustering.

Acknowledgments

This work was partially supported by Capes, CNPq, Fapemig, Finep, NSF grant EMT-0829835, and NIH grant 1R01EB0080161-01A1.

References

[1] A. Bar-Hillel, T. Hertz, N. Shental and D. Weinshall, *Learning a Mahalanobis Metric from Equivalence Constraints*, J. of Machine Learning Research, 6 (2005), pp. 937–965.

[2] S. Basu, M. Bilenko and R. J. Mooney, *A Probabilistic Framework for Semi-Supervised Clustering*, 10th ACM SIGKDD, (2004), pp. 59–68.

[3] S. Basu, I. Davidson and K. L. Wagstaff, *Constrained Clustering: Advances in algorithms, theory, and applications*, CRC Press, 2008.

[4] M. Bilenko, S. Basu and R. J. Mooney, *Integrating Constraints and Metric Learning in Semi-Supervised Clustering*, 21st Int. Conf. on Machine Learning, (2004), pp. 11–18.

[5] G. Casari, C. Sander and A. Valencia, *A Method to Predict Functional Residues in Proteins*, Nat. Str. Biol., 2(2), 1995, pp. 171–178.

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

[7] I. Davidson, I. and S. S. Ravi, *Clustering with Constraints: Feasibility issues and the k-means algorithm*, SIAM Int. Conf. on Data Mining, 2005, pp. 138–149.

[8] I. Davidson and S. Basu, *A Survey of Clustering with Instance Level Constraints*, ACM Transactions on Knowledge Discovery from Data, w(x), 2007, pp. 1–41.

[9] A. Goder and V. Filkov, *Consensus Clustering Algorithms: Comparison and Refinement*, 9th Workshop on Algorithm Engineering and Experiments (ALENEX), SIAM, 2008.

[10] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, 2nd ed, 2006.

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

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

[13] D. Klein, S. D. Kamvar and C. D. Manning, *On the Effectiveness of Constraints Sets in Clustering Genes From Instance-Level Constraints to Space-Level Constraints: Making the most of prior knowledge in data clustering*, 19th Int. Conf. on Machine Learning, (2002), pp. 307–313.

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

[15] Z. Lu and T. K. Leen, *Semi-Supervised Learning with Penalized Probabilistic Clustering*, Advances in Neural Info. Processing Systems, 17 (2005).

[16] R. M. Minardi, K. Bastard and F. Artiguenave *Structure-Based Protein Family Clustering and Prediction of Specificity-Determining Position Patterns*, Bioinformatics, 2010 (under review).

[17] G. P. Moss, *Enzyme Nomenclature*, Available at `http://www.chem.qmul.ac.uk/iubmb/enzyme/`, Accessed in August, 2010.

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

[19] S. Salem, M. J. Zaki and C. Bystroff, *Iterative Non-Sequential Protein Structural Alignment*, J. of Bioinformatics and Comp. Biol., 7(3), (2009), pp. 571–596.

[20] M. Shatsky, R. Nussinov, and H. Wolfson, "Multiprot a multiple protein structural alignment algorithm," in Algorithms in Bioinformatics, ser. Lecture Notes in Computer Science, R. Guigó and D. Gusfield, Eds. Berlin, Heidelberg: Springer, 2002, vol. 2452, ch. 18, pp. 235–250.

[21] N. Shental, A. Bar-Hillel, T. Hertz and D. Weinshall, *Computing Gaussian Mixture Models with EM using Equivalence Constraints*, Advances in Neural Info. Processing Systems, 16 (2004).

[22] A. M. Schnoes, S. D. Brown, I. Dodevski, P. C. Babbitt, *Annotation Error in Public Databases: Misannotation of Molecular Function in Enzyme Superfamilies*, PLoS Comput Biol 5(12), 2009.

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

[24] K. Wagstaff and C. Cardie, *Clustering with Instance-Level Constraints*, 17th Int. Conf. on Machine Learning, (2000), pp. 1103–1110.

[25] K. Wagstaff, C. Cardie, S. Rogers and S. Schroedl, *Constrained K-Means Clustering with Background Knowledge*, 18th Int. Conf. on Machine Learning, (2001), pp. 577–584.

[26] K. L. Wagstaff, *Intelligent Clustering with Instance-Level Constraints*, Ph.D. Dissertation. Cornell University, Ithaca, NY, USA, 2002.

[27] E. P. Xing, A. Y. Ng, M. I. Jordan and S. Russel, *Distance Metric Learning, with Application to Clustering with Side-Information*, Advances in Neural Info. Processing Systems, 15 (2003).

[28] E. Zeng, C. Yang, T. Li and G. Narasimhan, G, *On the Effectiveness of Constraints Sets in Clustering Genes*, 7th IEEE Int. Conf. on Bioinformatics and Bioengineering, (2007), pp. 79–86.