### A general framework for genome interpretation using evolutionary signatures

by

Guilherme Issao Camarinha Fujiwara

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

© Massachusetts Institute of Technology 2008. All rights reserved.

Author ..... . . . . . . . . . . . Department of Electrical Engineering and Computer Science May 21st, 2008 Certified by ..... Manolis Kellis Associate Professor Thesis Supervisor

Accepted by ...... Arthur C. Smith Chairman, Department Committee on Graduate Theses

MASSACHUSETTS INSTITUTE OF TECHNOLOGY NOV 1 3 2008 LIBRARIES

#### ARCHIVES

## A general framework for genome interpretation using evolutionary signatures

by

#### Guilherme Issao Camarinha Fujiwara

Submitted to the Department of Electrical Engineering and Computer Science on May 21st, 2008, in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science

#### Abstract

In the post-genomic era, characterized by the availability of the genome sequence data for many species, one of the biggest challenges to be solved is to identify the functional elements in our genome: the small subsequences containing units of biological function.

Work has been done to computationally identify specific functional elements such as protein coding genes [11], RNA genes [17], microRNA genes [16], regulatory motifs and individual binding sites for transcription factors and microRNAs [10]. This work has benefited from the use of evolutionary signatures obtained by observing the genomics changes across the sequence data of related species.

We propose in this work a general framework to perform functional element identification using evolutionary signatures. We first design several metrics of evolutionary signatures that are meant to capture different patterns of evolution expected from elements that have different biological function as well as novel patterns capturing diverse properties of evolutionary changes. We then compute these metrics for each of the elements in the human genome that are conserved across mammals and other vertebrate species in order to identify classes of functional elements.

Based on these metrics, we first perform classification of specific known types of functional elements, such as protein coding sequences, RNA coding sequences and CpG-rich promoters. With success in this step, we go one step further and establish an unsupervised clustering framework for conserved elements based on these metrics. With this approach, we obtain clusters of known and unknown classes of functional elements. We find that some of these clusters correspond to known functional elements, while others are depleted for known functions, while showing strong evidence of transcription and epigenetic modifications, suggesting these may correspond to novel classes of functional clusters. This illustrates the power of this method in identifying elements of known classes of functionality and to discover elements of novel classes of functionality. Thesis Supervisor: Manolis Kellis Title: Associate Professor

### Acknowledgments

I would like to acknowledge my advisor, Manolis Kellis, that not only provided great guidance in this project, but also was very encouraging and always willing to sit down to share great insights.

I also thank all of my other colleagues in the group, Mike Lin, Matt Rasmussen, Giorgos Papachristoudis, Alexander Stark, Pouya Kheradpour and Rogerio Candeias for inspiration, company, help and many fruitful conversations.

Last, I would like to thank my family and friends that provide me immense support. Without them, getting up to do research every day would be a lot harder. In particular, Katya for her continued support and for bringing baked goodies to the lab and my brother, my mother and my father for being this amazing family that I owe everything I accomplish to.

# Contents

0	Introduction			11	
	0.1	Backg	round	11	
	0.2	Thesis	s statement	13	
1	A general language for evolutionary events				
	1.1	A lang	guage for evolution	15	
		1.1.1	Primary evolutionary events - The letters	15	
		1.1.2	Secondary evolutionary events - The words	16	
		1.1.3	Tertiary evolutionary events - The sentences	17	
	1.2	Imple	menting this language of evolution	18	
		1.2.1	Implementation of the specific metrics	18	
2	Mu	lti-spe	cies setting	21	
	2.1	Aggre	gating	22	
	2.2	Discri	minating	22	
	2.3	Imple	menting these metrics	23	
		2.3.1	Parsimony	23	
3	Ac	lassific	ation framework for feature selection	25	
	3.1	Single	-element classification	25	
		3.1.1	Exons	26	
		3.1.2	CpG promoters	27	
		3.1.3	RNA Genes	28	

	3.2	Multiple element type classification	29			
4	Uns	supervised clustering based on evolutionary patterns				
	4.1	Clustering				
	4.2	Feature selection	33			
	4.3	Computational issues	34			
5	Mea	Measuring success				
	5.1	Authoritative sequence annotation data	37			
	5.2	Experimental data	39			
		5.2.1 Histone modification data	39			
		5.2.2 DNAse Hypersensitivity	39			
		5.2.3 EST data	40			
	5.3	Measuring enrichment method	40			
6	Em	Empirical results				
	6.1	Species				
	6.2	Conserved elements				
	6.3	Performance of clustering				
	6.4	Interpretation of results				
	6.5	Functional annotation	44			
	6.6	Experimental data	46			
7	Conclusion					
	7.1	Next steps	52			

# List of Figures

3-1	Scatter plot of exons	27
3-2	Classification results	28
3-3	Scatter plot of CpG islands	29
3-4	Multiple element classification	30
4-1	Clustering exons	34
4-2	Clustering CpG promoters	35
4-3	Clustering RNA genes	36
6-1	Phylogeny tree	42
6-2	Heat map of clusters	44
6-3	Clusters sensitivity for known elements	45
6-4	Clusters specificity for known elements	46
6-5	Enrichment for functional elements	47
6-6	Experimental regions enrichment	48
6-7	Experimental regions enrichment	49

## Chapter 0

## Introduction

#### 0.1 Background

A major goal in the field of genomics is to, given the genomic sequence, identify the individual elements of the genomic sequence that correspond to units of biological function. An example of some of these functional elements are protein-coding exons, introns, diverse families of RNA genes and many non transcribed elements that have regulatory functions such as binding sites, promoters and enhancers.

The digital nature of genomic sequence data motivates a computational approach to this problem of functional element identification. As an example, computational approaches have been used for identifying protein-coding genes early in computational functional genomics when genomes for single species was all there was available based on the distinct sequence patterns of protein-coding genes and precise structural properties of tRNA genes, for example.

With the recent sequencing of many sets of closely related species of yeasts, flies and mammals, a new method for discriminating function in sequence data has recently become possible: by comparing the aligned sequence of closely related species, one can observe the changes occurred to the genome of these species throughout evolution and use these changes to discriminate genomic regions of distinct functions.

The key principle in comparative genomics is that changes that disrupt biologically functional elements are negatively selected. In contrast, changes that are silent are tolerated and accumulate across evolutionary time. Thus genomic regions that show increased sequence conservation with closely related species correspond to functional elements.

Having this paradigm in mind, methods were developed to identify genomic regions that are significantly conserved. One such popular method is PhastCons [15]. It uses a phylo-HMM generative model for how sequence is generated jointly across aligned species. Given generative models for both conserved and non-conserved regions, one can compute the probability that any given region is generated by each of the models, thus being able to identify significantly conserved regions.

With such methods, we can identify conserved elements that are candidate functional elements. However, increased conservation alone does not provide us any information about the type of biological function associated with these elements. Thus, in addition to using the overall level of conservation as a discriminating feature, one can look at the specific profile of evolutionary changes that happen to each of these elements to be able to understand the specific function associated with each of them.

As an example, Kellis et al [9] defined a Reading Frame Conservation test based on the pressure to preserve the the 3-periodicity of codons in protein-coding regions, leading to the exclusion of gaps that are not multiples of 3. With this test they were able to accurately identify protein-coding sequences in the yeast genome and make corrections to the yeast gene catalogue. This method did not use the amount of sequence conservation for discrimination; it uses instead the specific types of changes observed for a given biological function. More recently, more work has been done with Drosophila species using this strategy [17]: Lin et al [11] use evolutionary signatures to find novel protein-coding genes and exons as well as discard false ones; Stark et al [16] use evolutionary signatures identify novel micro RNAs; and Kheradpour et al [10] apply evolutionary signatures to the discovery of regulatory motifs and their instances. More recently, with the sequencing of dozens of mammalian genomes, similar methods are being used to identify functional elements in the human genome.

#### 0.2 Thesis statement

The goal of this thesis is to create a general framework to perform genome analysis using evolutionary signatures. We want this framework to be able to perform genome interpretation based on a flexible set of evolutionary signatures, thus allowing us to identify different kinds of functional elements in an unbiased way. Ultimately, we are interested in identifying novel classes of functional elements that we may not suspect yet in the human genome..

We first develop this framework in the context of classification, allowing us to use a training set. We then move the framework towards unsupervised identification of functional elements using a clustering approach. A great advantage of this approach is that it is not limited to finding a particular type of functional element, but it can be used to discover novel classes of functional elements. It is thus essential to carefully design these evolutionary signatures towards general and perhaps novel classes of functional elements.

## Chapter 1

# A general language for evolutionary events

Designing evolutionary signatures that discriminate elements according to the different types of function is a core challenge of this project. It is necessary that these metrics be specific enough to provide sufficient discrimination power and yet general so that different types of biological function can be identified. In this sense, we would like to think of these metrics as a general language that describes the evolutionary events. We first discuss the idea of this language and then implement concrete metrics based on this idea.

#### 1.1 A language for evolution

We describe our concept of a language of evolution in terms of different events, elaborating on three different levels of abstraction

#### **1.1.1** Primary evolutionary events - The letters

The lowest level evolutionary events are single nucleotide mutations, insertions and deletions. By itself, each of these events provides very little information about the underlying biological function of the genomic region where it occurs. However, these are the building blocks of any change in the genomie. By aggregating many of these events we can produce a meaningful picture of the evolutionary selection pressures present in an element.

#### 1.1.2 Secondary evolutionary events - The words

By looking at sets of primary evolutionary events, one can capture more discriminative higher-level evolutionary events, the words of the language of evolution.

**Conservation.** Perhaps the simplest and most important of these higher-level events is the overall level of nucleotide conservation, which captures in a simplistic way as the total amount of information across mutations, insertions and deletions. This higher-level event describes the overall amount of evolutionary constraint in the sequence.

Gap Lengths. The length of gaps in a sequence, i.e., the number of contiguous insertion or deletion events given an alignment, is a higher-level event that captures important sequence constraints such as relative distance between certain nucleotides, in particular, when the underlying code has some periodicity that constrains these distances in order to preserve the offset of the code. This has been shown to be a very discriminative event for protein coding genes, which have an underlying 3 base code, but it may also be relevant for other types of biological function.

**Periodicity of mutations.** Mutation events can be distributed according to some periodic pattern for a given period length. Such a higher-level event is very descriptive of an underlying period in the biological function of the sequence. The classical example of this event is protein-coding sequences where mutations tend to occur more frequently in the third base of the 3-letter codons due to degeneracy in the genetic code (codons differing in only their third position typically encode the same amino acid).

Substitution rates. An important type of higher-level event of a more statistical nature is the frequency with which we observe specific mutation events. Namely, given a word length k, we can look at the  $4^k \times 4^k$  matrix indicating the frequency with which each k-mer mutates into each other k-mer. This type of higher-level event

is descriptive of how functionally synonymous each pair of k-mers are in the context of the underlying biological function. There are a few important known instances of this type of event. For protein-coding sequences, some codons encode amino acids of similar properties and are thus found to be more frequently exchanged. For motif sequences, such kmer exchange frequency can capture the degeneracy of protein-DNA recognition sequences of different transcription factors. Lastly, kmer exchange frequencies can discriminate sequences according to their k-mer content bias, such as for CpG-islands that play important regulatory roles in promoter regions.

**Correlated mutations.** Correlated mutations observed in a given pair of bases reveal a linked constraint on this pair of bases, indicating that the content of these bases relative to each other is important for the biological function of the genomic region. A great example of functional elements discriminated by this type of events are RNA genes for which the secondary structure is important, even when the primary sequence is free to change. In this case, we observe correlated mutations in paired bases which preserve the RNA secondary structure.

#### 1.1.3 Tertiary evolutionary events - The sentences

The events presented above capture specific types of constraint caused by types of biological function. The evidence from these secondary-level evolutionary events can be gathered together in tertiary-level events that can discriminate elements based on combinations of these events, specific to different biological constraints. We consider two ways to combine these secondary events.

Linear combination. The simplest way to combine those events is using a linear combination. Different secondary events can be given equal weight or the weights can be chosen for feature selection or to provide a normalization of the different secondary events or feature selection.

**Principal components.** This is a more sophisticated way to extract the most discriminative linear combinations from a set of uniform events. It is very useful, for instance, in extracting the most significant linear combination of the set of 3-mer substitution frequencies.

#### **1.2** Implementing this language of evolution

In order to use this concept of a language of evolution towards our goal of genome interpretation, we would like to capture these evolutionary events into concrete evolutionary metrics, implement them and use them in a discriminative framework.

Our goal is to implement a flexible system that takes as input a list of genomic regions and the corresponding aligned genomic sequence for each of the regions, and returns as output a row of numeric values representing the value of each of these metrics for each input element. The flexibility of this system allows us to easily introduce new metrics and to choose which metrics we would like to compute.

We chose to implement this system in Python for its fast development, objectoriented support and easy string handling capabilities. We achieve such flexibility by creating an abstract class for a metric, so that creating a new metric in our system is as simple as creating a new instance of this metric class and registering it in a factory class that knows about all the implemented metrics.

#### **1.2.1** Implementation of the specific metrics

We briefly describe below the implementation of some of these specific metrics. Note that these are all pairwise evolutionary metrics. In the next chapter we explain how we extend these pairwise metrics to work in a multi-species setting.

**Conservation rate.** This metric captures the conservation level and it is obtained by simply counting the number of conserved bases as a fraction of the number of aligned bases.

**Conservation profile.** In this metric, we divide the genomic region in 10 segments of equal length and we compute the conservation rate for each of this 10 segments, obtaining a profile of the conservation for this element.

Alignment success. This metric measures how much of the sequence was successfully aligned. Namely, this metric is simply the ratio of missing bases in the informant species that are aligned to a nucleotide in the other species.

Gap distribution This metric counts how many gaps of each particular length (up to a defined ceiling) are observed in the alignment.

**K-mer abundance.** This metric computes the frequency of each of the different  $4^k$  k-mers. Notice that this metric does not actually use the informant species sequences.

K-mer delta abundance in a strand. This metric computes the difference between the frequency of each of the different  $4^k$  k-mers in the plus strand minus its frequency in the minus strand of the sequence. This metric does not use the informant species sequences

Length. This metric is simply the length of the genomic region.

**Reading frame conservation.** Similarly to [9] this reading frame conservation metric is computed in the following way. We consider each of the 3 relative offsets in 3-periodicity between the two sequences. For each of these, we compute how many aligned bases are in the same offset relative to the total number of aligned bases. The metric is the highest value of the 3 different offsets.

**Compensatory substitutions.** This metric captures correlated mutations of bases that are predicted to be paired in an RNA secondary structure (we produce this prediction using RNAfold [8], using no alignment data besides the target species sequence). Precisely, we consider each pair of bases that undergo a mutation (either base or both). We then considered the pair of nucleotides that this pair of bases mutate to, and count how often each of the 16 different pair of nucleotides appear in this situation, obtaining a 16-element metric.

Substitution k-periodicity. This metric is obtained by computing the distance between consecutive mutations modulo k. The metric values are the frequency of each of the k possible values.

**Nucleotide substitution rate.** This metric is computed by counting the frequency with which each nucleotide mutates to each other nucleotide as a fraction of all mutations.

## Chapter 2

## **Multi-species setting**

The language of evolution based on evolutionary events discussed in the previous chapter was described in the context of comparison between two species. However, those evolutionary events can be observed through out evolution by comparing multiple species, and we would like to set ourselves to make our analysis with that in mind.

Previous work has shown that in comparative genomics analysis for functional element identification, using multiple species to look for these types of evolutionary events results in more powerful comparisons ([9], [17]). This can be intuitively understood as having more data available reduces the relative noise in a similar fashion to the law of large numbers.

In this work, we will focus on looking for evolutionary events as described in the previous chapter, but using comparisons across multiple species. To do so, we need to define a method to combine pair-wise comparisons into one multi-species comparison. The first benefit that we would like to get from this multi-species comparison is the increased amount of data and consequential decrease in noise, as mentioned above. We would like to go beyond, however, and use comparison across multiple species to be able to extract other evolution specific information such as non-uniformity of events across all the species, the directionality of events and its time frame. We discuss below a couple of methods to perform this multiple species comparison.

### 2.1 Aggregating

Perhaps the simplest way to use pair-wise comparison to produce a muti-species comparison is to perform several pair-wise comparisons and aggregate them. One way to aggregate the pair-wise comparisons is to perform an average or sum. This is very natural when the evolutionary events corresponds to direct counts, in which case the aggregated average corresponds to the total count of the events in all comparisons. When the events from the pair-wise comparisons are not direct counts, but some frequency or more complicated numeric value, the aggregated sum is not as natural but still gives a relevant summary of all the pair-wise metrics. The greatest advantage of this method is its simplicity. Other variant includes a voting scheme, where the events in each pair-wise comparison need to meet a threshold in order to count for a vote [9].

#### 2.2 Discriminating

In order to extract information from the comparisons such as directionality of events and non-uniformity of events across species, we cannot simply aggregate all pairwise comparisons, but rather collect these different pairwise comparisons and consider them all as distinct metrics. Using this method, we can, for example, have identification power of functional elements using simply conservation, distinguishing them sing the different time-scale of evolution.

The remaining question is which pairwise comparisons to use. In the context of multiple species comparison, where we have one species that we are most interested in, we can compare this species with each of the other informant species. In fact, this is a popular method used in multiple species comparison and its simplicity is a great advantage.

We would like to propose an alternate method that is aware of the phylogenetic tree of the species. Instead of using pairwise comparisons between species, we can reconstruct the ancestral sequences for each of the internal nodes of the tree, and then perform pairwise comparisons between each node and its ancestral sequence. This method has several advantages: it deals very well with missing sequences since ancestral sequences can be inferred with only a few of its child sequences available; comparisons are made across shorter branches, reducing the number of duplicated events; and the comparisons are as local in the species tree as possible, allowing us to observe phenomena such as non-uniformity and directionality of events.

This method could further be improved by, instead of inferring ancestral sequences, defining a probabilistic model and integrating over all possible ancestral sequences. This way we are not limiting ourselves to look at only the most likely ancestral sequence, which is possibly incorrect. However, this method adds a non-trivial amount of complexity and it is not clear that this complexity can be dealt with in the context of this project.

#### 2.3 Implementing these metrics

The implementation of the aggregating method for multi-species evolutionary metrics is built in the Python system described in the previous chapter, where pairwise evolutionary metrics are simply averaged across the pairs of each of the informant species and the target species.

To implement a discriminating strategy for using multi-species comparison as described above we choose to build a new system. The added complexity of doing parsimony reconstruction of the ancestral sequences is such that the performance of our Python system would make the problem infeasible.

We thus build a C++ system similar to the previous one, integrating a parsimony reconstruction method so that we can now compute pairwise metrics between each node of the phylogenetic tree and its parent.

#### 2.3.1 Parsimony

We briefly describe here the method of parsimony for ancestral sequence reconstruction [4]. For each base position in the alignment and a given tree topology, we would like to obtain the bases for each of the internal nodes in the tree that minimize the number of substitutions in the tree.

This problem can be solved using dynamic programming. We first order the nodes in an array so that each node is before its parent (for example we can use post-order). Then, for each internal tree node in this array and each of the four nucleotides we can compute the minimum number of substitutions in the subtree rooted at this node if the given nucleotide is used in the reconstruction. After computing this for the whole array, we can backtrack and obtain the nucleotides that correspond to the minimum number of substitutions.

Additionally, many of our metrics depend on the gap information in the sequence. Thus we would like to be able to infer gaps in the ancestral sequence as well as nucleotides. There are different cost models for indels to be considered. Many assume a non-linear cost of a gap, thus making parsimony reconstruction not a single-base problem. In fact, for many such models this problem is NP-complete [2]. We thus opt to use a linear cost model, which essentially means considering gaps as a fifth nucleotide using the parsimony method described above.

## Chapter 3

# A classification framework for feature selection

Having established a set of metrics designed to identify the different types of conserved elements, we would like to now evaluate those metrics in a classification setting before we move on to unsupervised clustering.

The difference between classification and unsupervised clustering is the use of a training data set. A classification framework will allow us to survey the power of each of the designed evolutionary signature metrics in a more controlled scenario.

We first perform single element classification, where the goal is, given a class of functional elements, classify the conserved elements as being of this class or not. Following, we try a multi-element classification setting, where there is still a training set, but now the evolutionary signatures identifying specific elements compete with each other.

#### 3.1 Single-element classification

We create classifiers for three specific classes of functional elements,

This task can be split in two: feature selection and classification algorithm. The step of feature selection is done based on the nature of each of the specific evolutionary signatures. Additionally, we experiment with using principal component analysis for feature selection and in the cases that it is successful we use the most significant principal components.

The classification approach used is a Support Vector Machine (SVM). This method is based on finding the best hyper plane that separates the training points of the two different classes, possibly considering the points under some map to a higher dimensional space, the kernel function. Because of the high-dimensional nature of the evolutionary metrics that we are using, we decide to use a linear kernel in order to minimize computation. This way we can train our SVM with as many examples as possible.

#### **3.1.1** Exons

We perform classification of protein-coding exon elements among the conserved elements in the human genome. For this classification, we use the following three metrics: Reading Frame Conservation, Conservation throughout the species and 3mer abundance. You can see the first two principal components of this data plotted in figure 3-1.

We use the gene annotation from the Ensembl database available through the UCSC genome browser, marking each conserved element as an exon if it has a nonzero intersection with any exon called by Ensembl. The results for the classification are shown in figure 3-2.

This result shows that our metrics present a substantial power to identify proteincoding signal, encouraging that we will be able to use this signal in our unsupervised clustering setting later. However, it does not perform nearly as well as specifically designed methods to identify protein-coding sequences that can perform above 95% sensitivity and specificity - our method was not designed with this specific purpose in mind, thus we do not expect it to perform as well. Many of these gene-finding methods use specific models and detailed sequence information to reliably identify boundaries of exons. In particular, most of the metrics used here are of a very general nature (which is necessary in order to capture novel evolutionary signatures), thus are expected to have less power in identifying any particular type of functional



Figure 3-1: Scatter plot of exons versus other elements under the first two principal components of the metrics used

element.

Nevertheless, this classification problem shows that a subset of our metrics have substantial power in identifying protein-coding signal and gives us some idea of which are these metrics.

#### 3.1.2 CpG promoters

We would like to now establish a classification method for CpG promoters. We use as a metric the 3-mer content of the sequences, extracting the two most significant principal components and then using an SVM. We show in figure 3-3 the conserved elements ploted with respect to these two principal components. You can note that these promoters are very few among all the conserved elements, so it is important to set a fair weight for positive versus negative training weight for the SVM.

The results of this classification are again encouraging and demonstrate the power of our metrics in identifying these promoters (figure 3-2).



Figure 3-2: Results of the classification framework for exons, promoters and RNAs, showing precision sensitivity and specificity

#### 3.1.3 RNA Genes

In order to perform classification of RNA genes among the conserved elements in the human genome, we select our most powerful feature for identifying RNA structure, the compensatory substitution metric.

Before using an SVM for classification, we first use Principal Component Analysis to identify the most discriminative aspects of these metrics data. The coefficients of these principal components show that the most discriminating mutations are the ones arriving on GT as well as the three corresponding pairs by reverse and complement, just as we would expect from intermediate mutations that preserve the RNA secondary structure by using the G-U wobble pair.

The results, shown in figure 3-2, were once again positive, demonstrating that our metrics are powerful enough to pickup RNA genes using the metric of compensatory substitutions.



Figure 3-3: Scatter plot of CpG islands versus other elements under the first two principle components of the metrics used

## 3.2 Multiple element type classification

We would like to know take a step further and test the power of our metrics in distinguishing between two classes of elements, still in a classification framework. Our motivation is that when distinguishing between different types of elements, our different metrics will be competing with each other and we would like to know the effect of that.

The classes of elements that we choose to be distinguished are exons versus RNA genes. In this setting, we filter the conserved elements in order to obtain only the elements that are marked as being either an exon or an RNA (eliminating those that are marked as neither or both). With those elements, we train an SVM in a small training set with an equal number of elements of each class. We then use the SVM to classify all the elements. The results, pictured in figure 3-4, show almost 90% sensitivity for the exon label and over 80% sensitivity for the RNA label, both with specificity above 95%, indicating that the competition of metrics will not be a problem

in the more general cluster setting.



Figure 3-4: Results of multiple element classification, showing the number of elements of each type among the elements given each of the two labels

## Chapter 4

# Unsupervised clustering based on evolutionary patterns

Having set a framework in which we can capture the evolutionary signatures of the conserved elements in a set of metrics, and further experimented with these metrics in a classification framework, we would like to now to apply these evolutionary signatures to analyze the conserved elements in a clustering framework.

Through this clustering analysis we would like to identify sets of similar elements that have evolved in a similar way, and further find a relation between this similarity of evolution and a possible similarity of biological function.

#### 4.1 Clustering

Clustering analysis consists of partitioning elements in sets that have similar features. There exist a diversity of clustering approaches. The two main popular classes are the agglomerative methods and the hierarchical methods.

Agglomerative clustering methods work by directly creating clusters according to some criteria (such as distance to some k fixed points) and then rearranging these clusters until they achieve some desired quality. Hierarchical methods work by starting with each element in its own cluster and progressively joining clusters or by starting with a single cluster and progressively splitting clusters. The main difference between these two classes of methods is that hierarchical methods produce more than a set of clusters of the elements, but a hierarchy of sets of clusters, one for each number of clusters, while agglomerative methods produce a set of cluster of the elements for a fixed number of clusters. Moreover, there is an essential difference between the two classes of methods in terms of computation. While hierarchical methods generally have a time complexity of at least  $O(n^2)$ , agglomerative methods can be substantially faster, like O(kn).

For computational reasons, we opt to use agglomerative methods in this clustering analysis. Namely, some of the clustering problems that we solve involve clustering up to millions of elements, which would certainly be infeasible to solve using an  $O(n^2)$ algorithm. Among the agglomerative methods, we choose to use the most popular one, K-means. K-means consists of initially choosing k random cluster centers among the points being clustered. At each step then, we proceed by assigning each point to the cluster whose center is the closest, followed by recentering each cluster at the point that is the mean of the members of that cluster. These steps are repeated until we obtain a stable set of k clusters.

The scalability of this algorithm is of central importance in determining whether or not we are able to perform clustering with all of the 3 million conserved elements. As a back of the envelope calculation, considering a runtime of O(kn), and a number of clusters k = 10, we expect to be able to have a feasible runtime with a few million of elements as longs as they all fit in memory. In practice, we observed runtimes of less than an hour, with the main factor being the number of metrics being used, which in some cases caused the size of the problem to be bigger than the available amount of memory. One solution we used in these cases was to perform k-means clustering with a 10% sample of the conserved elements and subsequently assigning each of the other 90% elements to the cluster with the closest center.

As we mentioned, when using agglomerative cluster methods such as k-means, we have to determine the number of clusters beforehand. We ran experiments with number of clusters varying from 2 to 10 clusters, being limited above by the amount of computation. At k = 10 we observed some evidence of over clustering by seeing pairs of similar clusters. This indicates that likely at k=10 we are not missing any significant cluster by having it merged with other non similar elements, so we chose this value for our analysis.

Another important parameter of the K-means algorithm is the distance metric used to compare how far each point is from the cluster centers. Some of the options are the Euclidean metric, Manhattan distance and Pearson correlation. It is important to consider the non-uniform nature of the metrics that we are considering, each metric having different numeric ranges and a different number of values. The use of a correlation metric would normalize for the range of each of the individual values of each of the metrics, but it would give each metric a weight proportional to its number of values.

To solve this problem of metrics normalization, we use the Euclidean metric for its mathematical simplicity. In comparing points using the Euclidean metrics, we can ensure that each of the different metrics have the same weight across the comparison of all of the elements by normalizing each of the metrics so that the sum of the variance of each of the values that comprise each metric is the same.

#### 4.2 Feature selection

The process of feature selection is done by first experimenting with the different metrics in the classification setting described in the previous chapter. From that analysis, we were able to observe specific metrics that were shown to have substantial power in identifying particular classes of functional elements.

After identifying these specific relevant metrics that presented successful in identifying specific classes of known functional elements, we try to experiment with our implemented metrics in a clustering setting, but looking specifically for the identification power of specific metrics for specific classes of functional elements. We show these results in figures 4-1, 4-2 and 4-3.

Once we have understood the power of the individual metrics in functional element identification both in a classification and in a clustering setting, we are ready to



Figure 4-1: Performance of clustering exons using the metric of conservation across each of the branches

combine our chosen metrics to perform clustering of the conserved elements into novel clustering. We discuss the results of this clustering approach in detail in chapter 6.

#### 4.3 Computational issues

As mentioned above, the clustering problems that we wish to solve approach the edge of being computationally feasible, so the choices we make regarding the infrastructure we use to perform this task are essential.

Initially we opted to use the K-means clustering package built-in to Matlab. This function uses a very standard algorithm for K-means and seemed very attractive given the ease of manipulating all the metrics in Matlab. However, the performance of this package turned out to not match our needs. The biggest size of problems we were able to solve using these problems were of 50,000 elements when using less than 100 metrics values, which took approximately 20 minutes. Bigger-sized problems simply caused Matlab to crash.



Figure 4-2: Performance of clustering CpG promoters using the metric of 3mer composition

After reaching this technical barrier, it was clear that a different solution for clustering was needed. We considered whether implementing our own infrastructure would be the right choice, but came across the Cluto software package that turned out to be well suited for our needs.

Cluto is a well-maintained software package developed in the University of Minnesota [18]. It has several interesting features such as the ability of perform clustering with a wide variety of algorithms and distance functions, as well as produce reports such as in-cluster similarity, clusters dissimilarity and most discriminative metric for each cluster. Most importantly, the Cluto package presented an excellent performance that caused us to choose it for our Clustering analysis.



Figure 4-3: Performance of clustering RNA genes using the metric of compensatory substitutions

## Chapter 5

## Measuring success

Before we describe the results of the unsupervised clustering of conserved elements in chapter 6, we would like to establish a method through which we can measure success. In measuring success we compare our results with two types of data - authoritative sequence annotation and experimental data.

#### 5.1 Authoritative sequence annotation data

In terms of authoritative sequence annotation, we are interested in looking at how the clusters resulted from our analysis partition the known types of functional elements. Some of the functional elements that we are interested in looking at are protein-coding genes, RNA genes, promoters, binding sites for transcription factors and for micro RNAs. For protein-coding genes annotation we use the Ensembl gene annotation of the human genome, available through the UCSC genome browser [7]. We mark conserved elements as being protein-coding if they intersect an exon from the Ensembl annotation. Additionally, we are interested in elements that are not protein-coding but are in specific positions relative to a gene structure. Namely, we use gene structures annotated in Ensembl for marking each conserved element as an intron, exon, 5'-UTR, 3'-UTR or within 100, 1k, or 5k upstream or downstream of the 5' end of the gene.

For RNA genes there is not such high-quality authoritative annotations as for

protein-coding genes, reflecting the fact that RNA genes are not nearly as well understood. Nevertheless, there is a diversity of prediction methods available, and we choose to use Evofold [13] predictions, a method that uses the energy of the predicted RNA secondary structure as well as the conservation of this secondary structure to infer annotation of RNA genes. This data is available through the UCSC genome browser and we also use non-zero intersection with Evofold to mark conserved elements as RNA genes.

For annotations of promoter regions, we take two different types of annotations. One is the annotation of CpG islands [5], available in the UCSC genome browser. CpG islands are regions of the genome that have a higher frequency of a nucleotide C followed by a nucleotide G. This regions are typically found upstream of the 5' end of genes and the abundance of CpG is associated with the fact that the CpG in these regions are unmethylated unlike most CpG found elsewhere in the genome, which is the reason why the rest of the genome is depleted of CpG. Our second approach for annotation of promoter regions is to consider the regions 1 kilobase upstream of the 5' end of all genes, using the gene annotation already described.

In terms of binding site annotation data, we adopt two distinct sources. The first one is the Transcription Factor Binding Site (TFBS) track provided in the UCSC browser [7]. This track is obtained by considering all motifs in the Transfac database and marking all instances of these motifs in the genome of human, mouse, rat and dog. Each genomic region that is identified as an instance for a motif for all these 4 species is annotated as a TFBS. This is a very liberal method of calling binding sites, but consistent, thus should be useful in the downstream analysis of our clustering results. Our second set of annotations uses a more sophisticated method by Kheradpour et al ([10]) for calling binding sites for both transcription factors (using motifs from [19]) and micro RNA. This method makes binding site calls to a much smaller number of regions and thus should provide us with an annotation of higher specificity..

#### 5.2 Experimental data

We recall that one of the main goals of this project is to identify novel classes of functional elements. In measuring our success towards this goal, using existing annotations is not enough. For this reason, we are interested in the distribution of different genome-wide experimental data through out the clusters we obtained. Specifically, we look at histone modification experimental data, DNAse hypersensitivity and Expressed Sequence Tags (EST).

#### 5.2.1 Histone modification data

Histones are small proteins that the chromosome folds around to form the compact chromatin structure [6]. After transcription, some modifications can happen to these histones that affect the way the histones interact with the chromosome and other chromatin proteins. These modifications have been shown to reflect information about the expression level of the near by genes and the activity of the corresponding regulatory region.

The data we use was produced by Barski et al [1] through Chromatin Immunoprecipitation using Sequencing (Chip-Seq). The generated data is a set of sequence reads of observed modifications of each type. We use this data by marking each of the conserved elements as having a specific histone modification if it contains any of the experimental reads for this particular modification. Our goal is that observing enrichment for specific types of histone modification in a certain cluster will give us an indication of specific biological function that is linked to this modification.

#### 5.2.2 DNAse Hypersensitivity

Regulatory genomic regions that operate by binding proteins usually require an open chromatin state so that the region is exposed, increasing the binding sensitivity. This open chromatin state is usually reached by cleavage by a DNAse enzyme.

DNAse hypersensitivity sites are genomic loci that show an increased sensitivity for the DNAse enzyme. This hypersensitivity is evidence of binding activity in the genomic region, indicating some type of regulatory function.

We use the genome-wide DNAse hypersensitivity data from Crawford et al [3] obtained form human CD4+ T cells and call conserved elements as DNAse hypersensitive if they contain any hypersensitive sites.

#### 5.2.3 EST data

Expressed Sequence Tags (EST) are fragments of transcripts observed in the cell, indicating transcription activity in the genomic region. Most of these EST regions correspond to genes. However, we are mostly interested in indentifying transcribed regions that are not genes, so we use the unspliced EST data from Genbank available through the UCSC genome browser and filter out the ones that correspond to genes.

#### 5.3 Measuring enrichment method

For both authoritative annotations and experimental data we would like to establish a method for quickly observing the over or underrepresentation of the specific type of elements in any of the produced clusters.

To do so, we measure the enrichment of elements of the specific type in each of the clusters by computing the ratio of the fraction of elements of that type inside the cluster over the fraction of elements of that type over all conserved elements.

This enrichment value gives us a way to quickly glance at the over or underrepresentation of each type of annotation or experimental data used. However, it does not take into account the size of the cluster, the number of elements of that type and the statistical effect of this numbers. In order to correctly capture these effects we compute the p-value for the enrichment or depletion of each of these types of elements using a hypergeometric distribution for sampling from a finite pool of elements of two classes.

Having this framework for measuring our success, we analyze the results of our clustering approach.

## Chapter 6

# **Empirical results**

Having set a framework for analysis and validation, we now finally move on to apply these methods to the human genome. Before we dive into the results, we briefly describe the settings.

#### 6.1 Species

In order to maximize our power for comparative genome interpretation for human, we use a large number of mammalian and vertebrate species. Namely, we work with the alignment of the genome sequence of 28 vertebrates (figure 6-1) available from UCSC genome browser.

These 28 vertebrates include 20 mammals, 4 of which are primates (human, chimp, rhesus and bush baby) and 2 of which are non-eutherian mammals (opossum and platypus), the 8 non-mammals having representative from birds, reptiles, amphibians and fishes. These selection of species provides us with both a wide variety of evolutionary distances to capture evolutionary events of distinct time-frame as well as a density of similar species more similar to human to provide more power in identifying more recent evolutionary events.



Figure 6-1: Phylogeny tree of the 28 vertebrates

#### 6.2 Conserved elements

As we mentioned before, we would like to perform this analysis restricted to conserved elements, accepting the paradigm that conservation is correlated with function. We use the popular phastCons [15] method for determining these conserved regions. PhastCons uses a generative model described by a phyloHMM that generates the sequences of all of the species using two distinct phylogenetic trees that are identical except for scaling, one for conserved and one for non-conserved regions.. Given this generative model, phastCons computes the conserved regions by finding the hidden states most likely to generate the observed sequences using the Viterbi algorithm [4].

One issue to consider with phastCons is that there is no explicit prior for the length distribution of the conserved elements, which in effect means that the prior distribution is exponential. Though we cannot say authoritatively, this is likely not realistic as specific classes of conserved elements have their specific length bias (exons, for example, are rarely shorter than 30bp). As a result, we observe many small conserved elements called by PhastCons. Some of these might not be more than a statistical fluctuation on the level of conservation, and even if they are not, there is very little signal in them for us to compute our evolutionary signatures. For these reasons, we opted to filter out of our analysis the elements having 10bp or less.

A second issue to consider is that, though the PhastCons elements' boundaries are defined to maximize the Viterbi probabilities, they often do not coincide with the boundaries of functional elements. These can happen for many reasons such as the region of functional elements extending beyond the conserved regions or two functional elements being close or intersecting each other.

#### 6.3 Performance of clustering

Having ran our clustering framework for the data described (figure 6-2), we now show the performance of the clustering method in separating functional elements of known classes into separate clusters. We have shown in chapter 3 and 4 the performance of the classification and clustering methods when using specific metrics with the goal of identifying just a single class of functional element. We perform a similar measure of performance here, except that the clusters are generated with the same set of metric with no particular goal of identifying a specific type of functional elements. Figures 6-3 and 6-4 show the power in identifying exons, RNA genes, and promoters. Note that in many of the cases the functional elements are grouped into more than one cluster and the performance resulting from using both clusters is described in the legend.

#### 6.4 Interpretation of results

More important than understanding our identification power is understanding the specific enrichment of each of the specific clusters with respect to the different type



Figure 6-2: Heat map of clustering algorithm ran on the conserved elements

of functional annotation and experimental data that we have available.

#### 6.5 Functional annotation

Regarding the enrichment with respect to functional annotation, you can see the results in figure 6-5. The most visible feature is the strong enrichment in clusters 3 and 7 for exon elements and depletion in every other cluster, reflecting what we have just seen in our analysis above. Just as strongly, we observe enrichment in cluster 9 for 5' regions of genes, CpG promoters and transcription factor binding sites found in promoter regions, giving a strong indication that this clusters is capturing promoter regions. We observe a similar trend, though not as strong, in cluster 1.

These results are interesting but somewhat expected from our previous analysis for identifying these types of functional elements. More surprising is the strong en-



Figure 6-3: Clusters sensitivity for known elements. Note that the elements of some classes are split in more than one cluster, so their sensitivity must be added

richment in cluster 10 for TFBS, Evofold predictions and micro RNA binding, as well as 3' upstream of gene regions. These enrichments indicate that cluster 10 is capturing a class of regulatory elements in the 3' end of the genes. Additionally, we observe strong enrichment in cluster 2 for TFBS, but no enrichment for any region nearby genes, provoking us to speculate that this cluster might be capturing enhancer-like regulatory elements that do not have to be nearby the regulated genes.

Last, we want to highlight that clusters 4, 5 and 8 show depletion for all types of functional annotation we use and therefore are potential candidates for novel classes of functional elements.



Figure 6-4: Clusters specificity for known elements

### 6.6 Experimental data

In order to provide us some power in understanding the clusters of potentially novel classes of functional elements, we turn ourselves to look at the enrichment of experimental data in each of the clusters (figures 6-6 and 6-7). We look specifically at the enrichment for of histone modifications, DNAse hypersensitivity and non-coding ESTs.

First we look at the clusters containing class of known elements. Cluster 3, which is strongly enriched for exons, show strong enrichment for all modifications except H3K9 di- and tri-methylated. Intersecting, cluster 7, which is the other cluster with enrichment for exons, show depletion for all of the modifications. One hypothesis is that these clusters are capturing distinct types of genes. Alternatively, this may be happening just because clustering 3 is having some contamination by regulatory regions when cluster 7 is not.



Figure 6-5: Enrichment of each of the clusters for known classes of functional elements, including the fold enrichment and p-values for depletion and enrichment

Cluster 9, in which we observed enrichment for promoter regions, show enrichment for H3K4 and H3K79 modifications as well as Pol II, while having depletion for H3K9 di- and tri-methylations. Additionally, we observe a strong enrichment for DNAse HS sites, providing additional evidence that cluster 9 captures regulatory promoter elements.

Cluster 10, which has indication of capturing 3' end regulatory elements, show enrichment for H3K79 modifications and H3K9 tri-methylated as well as Pol II, while showing depletion for the other H3K9 modifications and the H3K4 modifications.

Most interestingly is the distribution of this Chip-Seq data across the clusters of potentially novel elements. Clusters 4, 5, 6 and 8 all show strong depletion for all different functional annotations that we considered. Out of those, clusters 4 and 8



Figure 6-6: Enrichment for regions with H3K4, H3K27 and H3K36 modifications

both show very strong depletions for all of the histone modifications that we study, indicating that their function may not be regulatory or at least not directly linked to transcription. We speculate that these could represent some class of elements involved with regulation in a trans fashion, or RNA genes that do not have a conserved folding structure.

Clusters 5 and 6, however, both show enrichment for histone modifications. In particular, cluster 5 shows a much stronger enrichment for H3K9 tri-methylated while cluster 6 shows a much stronger enrichment for H3K79 tri-methylated. This is strong evidence that these clusters are capturing a group of regulatory elements, since the enrichment is very strong and select in the type of modifications. Since there is depletion in both of these clusters for promoter regions and even for all other regions close to gene structure, we suspect that these clusters might be capturing elements



Figure 6-7: Enrichment for regions with H3K9 and H3K79 modification, as well DNAse hypersensitive sites and non-coding ESTs.

with enhancer or silencing activity.

# Chapter 7

## Conclusion

In this work, we designed and implemented a variety of evolutionary signatures with the goal of describing a variety of evolutionary events that are present in different types of functional elements. We did this designed based both on literature knowledge and the goal of discriminating novel classes of functional genomic elements. Moreover, we introduced metrics computed across the phylogenetic tree that provide a way to distinguish the evolution in terms of time frame, uniformity and location.

We then applied these metrics to a classification framework with the goal of testing their power. The results were substantial power in classification of exons, RNA genes and CpG promoters. Most surprisingly, we found that our metric of conservation across the phylogenetic tree has substantial power in identifying protein-coding sequences.

After gaining some insight into the power of each of the metrics in discriminating the conserved elements, we used them in our unsupervised clustering framework, with the main goal of identifying novel classes of functional elements. In this analysis, we found a couple of clusters that were able to capture particular types of functional elements, some expected, such as exons and promoters, and others a bit surprising, such as the cluster with 3' regulatory elements.

Most interestingly, however, we observed 4 clusters that show depletion for all types of functional elements that we consider. Two of those show no evidence of direct transcription regulation since they present no histone modifications. The other two show a strong and specific histone modifications enrichment profile, making us believe that they might be capturing some specific classes of regulatory elements that are involved with transcription but are not nearby genes.

#### 7.1 Next steps

This project ends at a point where many attractive paths are open for further work. In terms of methods, improvements can certainly be made in designing metrics and introducing new ones, which has a great potential to impact final results. Moreover, a more rigorous method for doing feature selection, perhaps using a machine learning approach could be used for determining the weights of each of the implemented metrics could provide great improvement. In particular, the results of classification can be used in a more principled manner to aid in feature selection for clustering.

In the clustering front, the method could greatly benefit of a direct way of determining the most relevant set of features for each of the clusters. We have done this in a somewhat manual way in our analysis and it seems to be an important step in understanding the contents of the clusters.

Last, but not least, it remains to further understand the elements of clusters 4, 5, 6 and 8. We have showed enough reasons to convince the reader that these clusters are capturing classes of conserved elements with interesting and specific features and yet distinct from any of the known classes. We have not however, achieved a rigorous precision or sensitivity for identifying these elements, making individual experimental testing difficult. Therefore, the next steps in discovering what is in these clusters involve either an improvement in the computational side to find these elements with higher precision, at the same time understanding the features that bring these elements together, or high-throughput experiments with elements of the existing generated clusters to test them for enhancers or silencer activity and possibly trans regulatory activity.

We hope that this work will help shed some light into the question of what are the functional elements in the genome that we don't know about yet, where they are and what do they look like.

# Bibliography

- [1] Artem Barski, Suresh Cuddapah, Kairong Cui, Tae-Young Roh, Dustin E Schones, Zhibin Wang, Gang Wei, Iouri Chepelev, and Keji Zhao. Highresolution profiling of histone methylations in the human genome. *Cell*, 129(4):823–837, May 2007.
- [2] Leonid Chindelevitch, Zhentao Li, Eric Blais, and Mathieu Blanchette. On the inference of parsimonious indel evolutionary scenarios. J Bioinform Comput Biol, 4(3):721–744, Jun 2006.
- [3] Gregory E Crawford, Ingeborg E Holt, James Whittle, Bryn D Webb, Denise Tai, Sean Davis, Elliott H Margulies, YiDong Chen, John A Bernat, David Ginsburg, Daixing Zhou, Shujun Luo, Thomas J Vasicek, Mark J Daly, Tyra G Wolfsberg, and Francis S Collins. Genome-wide mapping of dnase hypersensitive sites using massively parallel signature sequencing (mpss). *Genome Res*, 16(1):123–131, Jan 2006.
- [4] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, 1998.
- [5] M. Gardiner-Garden and M. Frommer. Cpg islands in vertebrate genomes. J Mol Biol, 196(2):261-282, Jul 1987.
- [6] Leland Hartwell, Leroy Hood, Michael L. Goldberg, Lee M. Silver, Ruth C. Veres, and Ann Reynolds. *Genetics: From Genes to Genomes*. McGraw-Hill Higher Education, 2004.
- [7] A. S. Hinrichs, D. Karolchik, R. Baertsch, G. P. Barber, G. Bejerano, H. Clawson, M. Diekhans, T. S. Furey, R. A. Harte, F. Hsu, J. Hillman-Jackson, R. M. Kuhn, J. S. Pedersen, A. Pohl, B. J. Raney, K. R. Rosenbloom, A. Siepel, K. E. Smith, C. W. Sugnet, A. Sultan-Qurraie, D. J. Thomas, H. Trumbower, R. J. Weber, M. Weirauch, A. S. Zweig, D. Haussler, and W. J. Kent. The ucsc genome browser database: update 2006. Nucleic Acids Res, 34(Database issue):D590–D598, Jan 2006.
- [8] Ivo L. Hofacker, Walter Fontana, Peter F. Stadler, L. Sebastian Bonhoeffer, Manfred Tacker, and Peter Schuster. Fast folding and comparison of rna secondary structures. *Monatsh. Chem.*, 125:167–188, 1994.

- [9] Manolis Kellis, Nick Patterson, Matthew Endrizzi, Bruce Birren, and Eric S Lander. Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature*, 423(6937):241–254, May 2003.
- [10] Pouya Kheradpour, Alexander Stark, Sushmita Roy, and Manolis Kellis. Reliable prediction of regulator targets using 12 drosophila genomes. *Genome Res*, 17(12):1919–1931, Dec 2007.
- [11] Michael F Lin, Joseph W Carlson, Madeline A Crosby, Beverley B Matthews, Charles Yu, Soo Park, Kenneth H Wan, Andrew J Schroeder, L. Sian Gramates, Susan E St Pierre, Margaret Roark, Kenneth L Wiley, Rob J Kulathinal, Peili Zhang, Kyl V Myrick, Jerry V Antone, Susan E Celniker, William M Gelbart, and Manolis Kellis. Revisiting the protein-coding gene catalog of drosophila melanogaster using 12 fly genomes. *Genome Res*, 17(12):1823–1836, Dec 2007.
- [12] Michael F Lin, Ameya N Deoras, Matthew D Rasmussen, and Manolis Kellis. Performance and scalability of discriminative metrics for comparative gene identification in 12 drosophila genomes. *PLoS Comput Biol*, 4(4):e1000067, Apr 2008.
- [13] Jakob Skou Pedersen, Gill Bejerano, Adam Siepel, Kate Rosenbloom, Kerstin Lindblad-Toh, Eric S Lander, Jim Kent, Webb Miller, and David Haussler. Identification and classification of conserved rna secondary structures in the human genome. *PLoS Comput Biol*, 2(4):e33, Apr 2006.
- [14] Matthew D Rasmussen and Manolis Kellis. Accurate gene-tree reconstruction by learning gene- and species-specific substitution rates across multiple complete genomes. *Genome Res*, 17(12):1932–1942, Dec 2007.
- [15] Adam Siepel, Gill Bejerano, Jakob S Pedersen, Angie S Hinrichs, Minmei Hou, Kate Rosenbloom, Hiram Clawson, John Spieth, Ladeana W Hillier, Stephen Richards, George M Weinstock, Richard K Wilson, Richard A Gibbs, W. James Kent, Webb Miller, and David Haussler. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. *Genome Res*, 15(8):1034–1050, Aug 2005.
- [16] Alexander Stark, Pouya Kheradpour, Leopold Parts, Julius Brennecke, Emily Hodges, Gregory J Hannon, and Manolis Kellis. Systematic discovery and characterization of fly micrornas using 12 drosophila genomes. *Genome Res*, 17(12):1865–1879, Dec 2007.
- [17] Alexander Stark, Michael F Lin, Pouya Kheradpour, Jakob S Pedersen, Leopold Parts, Joseph W Carlson, Madeline A Crosby, Matthew D Rasmussen, Sushmita Roy, Ameya N Deoras, J. Graham Ruby, Julius Brennecke, Harvard FlyBase curators, Berkeley Drosophila Genome Project, Emily Hodges, Angie S Hinrichs, Anat Caspi, Benedict Paten, Seung-Won Park, Mira V Han, Morgan L Maeder, Benjamin J Polansky, Bryanne E Robson, Stein Aerts, Jacques van Helden,

Bassem Hassan, Donald G Gilbert, Deborah A Eastman, Michael Rice, Michael Weir, Matthew W Hahn, Yongkyu Park, Colin N Dewey, Lior Pachter, W. James Kent, David Haussler, Eric C Lai, David P Bartel, Gregory J Hannon, Thomas C Kaufman, Michael B Eisen, Andrew G Clark, Douglas Smith, Susan E Celniker, William M Gelbart, and Manolis Kellis. Discovery of functional elements in 12 drosophila genomes using evolutionary signatures. *Nature*, 450(7167):219–232, Nov 2007.

- [18] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. *KDD Workshop on Text Mining*, 2000.
- [19] Xiaohui Xie, Jun Lu, E. J. Kulbokas, Todd R Golub, Vamsi Mootha, Kerstin Lindblad-Toh, Eric S Lander, and Manolis Kellis. Systematic discovery of regulatory motifs in human promoters and 3' utrs by comparison of several mammals. *Nature*, 434(7031):338–345, Mar 2005.