Creation of a Relational Database for Identifying Potential Functional DNA Sequence Motifs in *A. thaliana* and Other Genomes

Submitted to the faculty in partial completion of the
requirements for Honors in Biology

Jonathan M. Carlson
Robert H. Gross, adviser

Dartmouth College
Hanover, NH
June 2003

Creation of a Relational Database for Identifying Potential Functional DNA Sequence Motifs in *A. thaliana* and Other Genomes

Submitted to the faculty in partial completion for the requirements for Honors in Biology

Jonathan M. Carlson

Thesis Committee:

Robert H. Gross, adviser

Thomas P. Jack

Javed A. Aslam

Acknowledgements

I would like to thank first and foremost my adviser Bob Gross, who not only let me start this project as a sophomore, but also gave me so much ownership in this, providing a vision and direction, but trusting me to figure out how to make it work. It has been an incredible experience.

My thesis committee was fantastic. Tom Jack and Jay Aslam both provided great support and suggestions for future research. Thank you for taking the time to support me in this.

Dan Simola ’03 has been great to work with. Coming onto the project at the end of fall term, he completely re-wrote the GenBank parsing, allowing us to quickly import new GenBank genomes as they become available. He and I have worked closely on our separate theses.

Joe Kimpel ’03 very graciously helped me generate some figures in MatLab. He saved me an enormous amount of time and I am very grateful.

Kathryn Adams ’04 was amazingly patient and supportive throughout this. Thank you.

Abstract

Understanding the molecular basis for gene expression regulation is an important problem in biology. A fundamental sub-problem is understanding how DNA sequence information allows for the molecular control of regulation. With the increasing availability of fully sequenced genomes, we can begin to look directly at DNA for the answers. We have created a database that catalogs the position of every 9-mer in close proximity to every gene in the *Arabidopsis thaliana* genome. This allows us to search for motifs that are non-randomly distributed throughout the genome and so may serve some biological function. We have also created a website to serve as an interface. Here we discuss the structure and design of the database and website and how they can enable a biologist to identify putative cis-acting regulatory motifs. We show specific methods that can be used to identify non-random motifs on the genomic level that are likely to be involved in basal regulation or the regulation of large sets of genes. We also describe methods that are specific for sets of co-regulated genes. Using our database and website, we have easily detected a number of known cis-acting regulatory motifs, as well as a number of motifs that may represent novel elements. Introns were also analyzed, and known splicing elements were easily found. A group of co-regulated phase-0 clock genes has been analyzed as well. Known regulatory motifs were analyzed with a genomic perspective, and a potentially novel motif was identified.

Availability: The website can be accessed at https://gargoyl.dartmouth.edu/genomes/login.html. Please contact Bob Gross at rhg@dartmouth.edu for a guest account.

Table of Contents

Introduction

What is bioinformatics?

The April 24th, 2003 issue of *Nature* was dedicated to the Genomic Revolution. 2003 marks not only the 50th anniversary of Watson and Crick's description of the double helix, it also marks the completion of the Human Genome Project, formally announced by the International Human Genome Sequencing Consortium this April, and the official declaration of the "Era of the Genome" by the Nation Human Genome Research Institute ("International Consortium", 2003). As the sequencing technology developed to meet the goals set forth for this project, a number of other smaller genomes were fully sequenced; among them, *Arabidopsis thaliana*, the model plant organism, finished in December of 2000 (Arabidopsis, 2000).

The vast amount of raw sequence information represents a largely untapped source of potential knowledge. An organism's genome contains the hereditary information that makes that organism what it is, and we now have this in the form of millions, or for some organisms, billions, of A's, C's, G's and T's. While the early stages of the Human Genome Project focused on efficient sequencing methods, the latter stages have begun to focus more on how we can make sense of this sequence data. Bioinformatics is the relatively young interdisciplinary field that combines biology with the computational sciences in an effort to extract knowledge from this raw information.

Much of the bioinformatics research thus far has focused on the prediction of protein-encoding genes from DNA and algorithms that take an unknown sequence and search

databases of known sequences for similarities. Other algorithms use the properties of known proteins to predict various properties of an unknown protein. While the search for identifying protein function has always been of much interest, the problem this thesis focuses on is a newer problem that is drawing much attention: How can we use genomic sequence information to learn about gene regulation?

Public databases

The basis of bioinformatics lies in the databases that store the sequenced genomes. Many of these genomes, generally sequenced by various consortiums, end up in the GenBank databases of the National Center for Biotechnology Information (NCBI), a division of the National Institutes of Health (NIH). The raw DNA sequences alone are of little use, so the consortiums annotate the DNA in a common format with whatever information they have available to them. This annotation generally focuses on genes that code for proteins, including exon and intron data, protein function, untranslated regions, and alternative splicing patterns. Many of the genes in these databases are predicted solely on the basis of various gene-finding algorithms and so are purely hypothetical and lack any information beyond putative exon and intron definitions. Sometimes hypothetical functions are proposed on the basis of primary amino acid sequence similarity to better characterized proteins. The rest of the proteins usually have been identified in various labs and so often have exons confirmed by cDNAs and may have defined functions.

Genomic databases are complimented by various other databases that hold functional information. These databases include various protein databases, transcription factor

databases, restriction enzyme databases, and many other protein databases that specialize in specific types of proteins. Of particular interest to the study of gene regulation are the relatively new microarray databases (Devaux et al, 2001; Planet et al, 2001). Microarray technology is a powerful technique that allows us to quickly identify co-regulated genes at the genomic level by identifying all of the genes that are expressed under certain conditions. The databases that store this information provide a valuable resource for identifying genes whose patterns of regulation are profoundly similar. The availability of both expression data and raw sequence information presents us with the unprecedented opportunity to study how genes are regulated at the sequence level.

What is known about gene regulation?

Promoters

Gene regulation begins with regulating transcription, the process by which a sequence of DNA is transcribed into RNA, which is later translated into a protein. The cell usually regulates transcription by the sequences that are immediately upstream of each gene. For the purposes of this paper, we will define these proximal upstream regions as promoters. While regulation can also be influenced by the action of distal enhancers, the wide variation in location of these elements, which have been demonstrated to act up to 10,000 bases away from the target gene, makes them extremely difficult to study computationally. Promoters work through the action of transcription factors (TFs), which bind to certain regions in promoters, interact with other proteins, and ultimately regulate transcription. Each promoter typically contains a number of transcription factor binding sites, a subset of which are recognized by the basal transcription factors that act on most

promoters. These transcription factors function in localizing RNA polymerase, thereby defining the start of transcription. The basal transcription factors and the RNA polymerase are collectively referred to as the basal machinery.

While basal transcription factors are necessary for the expression of most genes, they are not sufficient to regulate genes in response to changes in the internal and external environment. Other transcription factors provide the specificity required to selectively turn on specific sets of genes while leaving others off. These transcription factors may recruit polymerase or certain basal transcription factors, thereby increasing the probability that the basal machinery will begin transcribing the gene. Other transcription factors act as repressors and block the recruitment or action of the basal machinery.

The fact that transcription factors bind to DNA and simultaneously interact with other proteins places some physical constraints on where they can bind. While DNA is flexible enough to allow non-adjacent DNA-bound proteins to interact, if a transcription factor is bound too far away from the protein with which it interacts, the odds of interaction are greatly reduced. Similarly, if a transcription factor is too close to another protein, it may physically prevent the binding of that protein to the DNA. This is how some repressors work. Finally, the orientation of the transcription factor when it binds may or may not be important. If the transcription factor is asymmetric, binding the DNA such that the amino-terminal end is facing downstream may result in different protein-protein interactions than if the amino-terminal end is facing upstream.

Often the presence of one transcription factor is not enough to significantly increase the probability of basal machinery binding, but two or more transcription factors will work synergistically to initiate transcription. Thus, it is generally a combination of transcription factors working together that leads to transcription initiation. Coordinated regulation therefore depends largely on what sets of genes are acted upon by the same set of transcription factors. Further specificity may be achieved based on the relative binding locations of these transcription factors. This allows a cell to control which genes are turned on by regulating which transcription factors are present and active in the nucleus.

In order to allow this kind of gene-specific control, each gene's promoter must be able to specify exactly which transcription factors may bind to it and where. This is made possible by the binding specificity of the transcription factors, each of which typically binds with the highest affinity to a single consensus sequence. Transcription factors usually do not require exact matches, but will often bind preferentially to those sequences that are closer to the consensus. Often times the consensus will require that certain bases be exact matches, while the others can vary to some extent.

While some transcription factors and their consensus sequences have been identified, the majority are unknown. Because of this, the transcription factors involved in the regulation of expression of most protein coding genes are unknown. The ability to predict the binding sites of transcription factors is an important step towards understanding the control of specific subsets of genes as well as basal control at the genomic level.

Introns

The next stage in protein regulation through gene expression occurs after the gene has been transcribed into RNA. In eukaryotic organisms, including plants, the introns are spliced out of the RNA soon after transcription. While many genes have the same introns spliced out all the time, there are a number of genes that can be alternatively spliced, such that the same gene can encode different proteins. The molecular mechanisms for splicing are well understood, with specific ribonucleoprotein complexes recognizing certain sequences within the transcript that mark the ends of the introns. Two of these sequences, called splice sites, occur at the 5' and 3' ends of the introns, while the branchpoint occurs around 60bps upstream of the 3' splice site. While consensus sequences for each of these have been identified in *A. thaliana*, these sequences are extremely short and occur frequently throughout the genome (Brown et al, 1996). It is not clear how the important ones are differentiated from the background noise, nor is it clear whether or not the rest of the intron sequence serves a biological purpose. Some studies in other organisms have indicated that mutations in introns can have adverse affects, but the mechanisms for these are not well understood. As such, there is much research that still needs to be conducted on these regions.

How can we use what is known to predict the binding sites of regulatory proteins?

While DNA replication is generally very accurate, random errors do occur. Natural selection suggests that mutations which have a profoundly negative effect will be eliminated from the population. If a section of DNA has little biological impact, then

mutations will not be eliminated but will instead accumulate, eventually leading to random sequences in that section. This suggests that if a sequence within a generally random section is statistically non-random, then there is a high probability that it confers some selective advantage, and so likely serves some biological function. So how can we identify non-random sequences, or motifs, from promoters, which may or may not be random over all?

We can extract the promoter region of all known and predicted genes from the public GenBank database. Thus, for each gene we can find every motif that lies in its promoter. Furthermore, we can store each motif's position relative to the start of that gene. If we do this for the entire genome, then we can look at each motif individually and find every gene that it lies in front of as well as every position in which it lies. We can then analyze the distribution of each motif's positions and determine if that motif is randomly distributed throughout the promoter region.

This can also be done for the transcriptional units, the regions of DNA that are transcribed, as well as exons, introns and any other region that is stored in the public databases. We can therefore compute ratios that compare how often a motif occurs in two different regions. Similarly, we can compute ratios that compare how often a motif occurs in the promoter region in one set of coordinately regulated genes versus the rest of the genes. These ratios can be normalized against an expected ratio given a random distribution of motifs throughout the genome. This allows us to identify motifs that occur

in one region or set of genes far more often than would be expected given a random distribution.

The non-randomness of a motif suggests one of two causes:

(a) The motif serves a biological function that requires it to be in certain locations; or

(b) There are so many other motifs that have to be in a certain functional location that this useless motif was effectively pushed away from the functional region and into a region that does not matter. That is, the presence of the non-functional motif in the functional region is detrimental because it doesn't allow functional motifs in. The distribution would then give the appearance of being pushed away from the functional region, and so would be non-random. In this case, non-randomness would not imply functionality.

It is likely that both happen to some extent. However, the high number of possible motifs suggests that (b) is not a very likely scenario. For a sequence of n nucleotides, there are 4^n possible motifs of that length. Thus, for any functional region of significant length and complexity (that is, non-repetitive), it is highly unlikely that all the motifs in that region strictly serve a purpose and do not appear simply due to chance. It should be noted, however, that non-coding DNA is known to contain highly repetitive sequences. Long runs of low-complexity repeats will not only make that repeated motif appear less random, it will also skew the computed randomness of truly random motifs. It is not unreasonable to assume, however, that the size of the genome (approximately 28,000

genes in the January 2002 TIGR release of the Arabidopsis genome) will minimize this effect.

If (a) is the reason for a motif's non-randomness, then we can begin to infer some biological function for that motif. From what is known about DNA, functional regions generally serve in specifying either (i) chromatin structure, (ii) a protein binding site, or (iii) something involving codons. If the region we're examining is not in an exon, then the motif is likely to be involved either in chromatin structure or to serve as a protein binding site. If the regions we are analyzing are promoter regions, and the motif is a protein binding site, then it is likely a transcription factor binding site, and so may be directly involved in gene expression. Chromatin structure involvement can rarely be ruled out, however, especially given how little is known about the mechanisms of chromatin structure. But even if this is the case, chromatin structure often plays an important role in gene regulation (for example, see Riechmann, 2002), and so the non-random motif is still likely to be indirectly involved in gene expression and regulation. The generic term for DNA sequences that act in some way in gene regulation is cis-acting regulatory motifs. This term encompasses the fact that a motif may aid in gene regulation by binding a transcription factor, or may work in some other unidentified way. If the non-random motif lies primarily in intron regions, its putative function is not as clear. The protein that likely binds to it may be involved in splicing patterns, or may serve some other, as yet unknown function.

It is therefore possible for us to identify potential regulatory sites by identifying significantly non-random motifs in various regions. In the case of promoters, these motifs are likely to be transcription factor binding sites. This can only be confirmed through wet lab experiments. The value of bioinformatic approaches is in suggesting wet lab experiments and pointing researchers in directions that are likely to yield interesting results. If we can identify potential functional motifs quickly, we can significantly reduce the amount of time required for a lab to identify the regulatory sequences of their gene(s) of interest.

In the case of introns, it is not as clear what we will be identifying. A functional motif located in an intron may serve as a binding site for a distal enhancer or somehow enhance the efficiency of splicing. It may also serve some as yet unidentified role. In any case, a significantly non-random intron motif will lend itself to further study in a lab, with the goal of better understanding the roles and definitions of introns.

Current Algorithms

There are a number of current approaches to the problem of finding cis-acting regulatory motifs. Most of these are available for public use at The Arabidopsis Information Resource (TAIR) website (Huala et al, 2001).

Searching for known motifs

The simplest method available is to search an unknown upstream region for known transcription factor binding sites. This approach is implemented by plantCARE and PLACE, both linked to from TAIR. While it is intuitively pleasing to know that the sites

identified by this approach have been implicated in other genes, there are some major problems that limit the usefulness of these approaches.

Cis-acting regulatory motifs are typically short sequences, usually ranging in length from five to ten bases. If the upstream regions were truly random, we would expect any given 5-mer to occur once every $4^5 = 1024$ bases, or once every one or two promoters in the entire genome! When you factor in mismatches, the rate of occurrence jumps even more. To look for these motifs without knowing how they are distributed throughout the genome therefore leaves us with little useful information. As is discussed below, while these searches by themselves are not very useful, it is possible that combining these approaches with motif position information will yield informative results.

Multiple sequence alignments

Hertz and Stormo (1999) describe a method for finding consensus motifs out of a set of sequences using local multiple sequence alignments. Approaches like these allow a user to input a set of upstream regions that are likely to be coordinately regulated. A multiple sequence alignment is done to find conserved regions in all the sequences. The conserved regions are then statistically summarized as a weight matrix, which represents the frequency of occurrence for each base in each position in the conserved region. A p-value can be calculated from the weight matrix that essentially reflects the probability that the weight matrix is random.

This approach is well suited for finding consensus sequences for cis-acting elements and for quantifying the statistical accuracy of those sequences. This can be particularly useful

when a set of motifs is known to be functionally important and a best consensus sequence is needed. However, in the case where little is known about the upstream regions being compared, the multiple sequence alignment approach has some serious drawbacks. One such problem is that position information is not taken into account. While this is not necessarily relevant, as discussed above it is likely that motifs' functions are often dependant on position and even orientation, which is represented in the DNA by reverse-compliment sequences and is therefore entirely missed by alignment programs. Another problem is that this approach looks only at the subset of genes and ignores the rest of the genome. Thus, the "significant" motifs identified may not be unique to the set of genes at all. A motif that occurs in 10,000 genes is not likely to be of much functional relevance to a set of 20, even if it occurs in all of them. The exception to this may be if position is important or the set of other motifs it occurs with is important. But as we discussed, the alignment program detects neither of these. This issue of failing to look at genomic information is an important recurring problem in many of these approaches.

Searching for over-represented motifs

A number approaches currently exist that look for over-represented motifs in the upstream regions of a set of co-regulated genes as compared to a background set of either random genes or the entire genome (Ohler and Niemann, 2001). These methods have an advantage over sequence alignment methods in that they take into account the background noise that may produce false positives. This allows these approaches to eliminate common motifs such as TATA boxes and other motifs that serve as binding sites for the basal transcription factors. Often times, the background set takes into account base distributions. Since most organisms have an unequal number of each of the

four bases on the primary strand, the expected number of occurrences for each motif is not the same.

Over-represented motif searching algorithms can be divided into two classes (Thijs et al., 2001). Word counting techniques typically examine motifs of a set size, counting the number of occurrences of each motif in a subset of co-regulated genes. Over-representation is determined by comparing the observed frequency of each motif of a set size against the expected frequency. The second class is the probabilistic methods. In these algorithms, position probability matrices are created for different motif models and analyzed statistically, usually using either expectation maximization, as implemented by MEME, or Gibbs sampling (Bailey and Elkan, 1995; Thijs et al., 2001). Like word-finding, these methods usually choose a set size of motif to look for, as it is computationally impossible at this point to analyze all possible motif lengths.

These approaches have met with varying success, and often the best approach is to analyze unknown promoter sequences with a number of different methods. Nonetheless, there are some major shortcomings in all of these methods that lead to a high number of false positives as well as false negatives.

First, while over-representation methods address the genomic-scale issues of multiple sequence alignments, the models constructed are typically summaries of the genome that often miss the specifics. That is, while a given motif may not be statistically significant in regards to its base composition, it is still possible that the particular sequence of bases

is unique to a given set. Similarly, the opposite often occurs, in which a particular motif that is statistically non-random on the genomic level is not unique to the set of genes being analyzed. The non-random distribution on the genomic level still causes these motifs to be flagged as significant in the subset.

Second, over-representation methods suffer from a lack of position information. The position matrices used in the statistical methods refer to the position of bases within a motif, not the position of motifs within a promoter. As discussed above, the position and orientation of a motif may profoundly influence its ability to act as a cis-acting regulatory element. By excluding this information, a number of false positives may creep into the analysis. Motifs that only function in one area may also occur randomly outside that area. Thus, detecting "functional" motifs in positions that render them ineffective is not very useful. The reverse can also be true. A motif may occur frequently in the genome as a whole, but only selectively occur in a unique range of positions. In this case, the frequency of genomic occurrence will hide them from over-representation methods. Even if these methods identify such a motif, the problem will merely switch from that of false negatives to one of false positives.

Finally, these methods are not very good at identifying unique sets of motifs that work together to regulate a set of genes. This is a very difficult problem that no algorithm has adequately solved. The reason for this is the computational complexities of analyzing all the permutations of possible motif combinations. Unfortunately, since much of the specificity of control comes from the unique combinations of motifs, there is a lot of

background noise that can be generated. That is, a motif may be involved in the control of a set of genes, but also occur throughout the genome. The specificity in this case may be the other motifs that it occurs with.

A novel approach

We have designed and implemented a relational database and web-based user interface that attempts to resolve the issues related to position distribution of motifs. This includes the position information for all motifs in a number of different regions, including promoters, transcriptional units, exons, introns, and untranslated regions. This allows a new genomic approach that is based both on over-representation at the genomic regional level and positional information with respect to genes. That is, instead of focusing solely on what motifs are over-represented in the promoters of what genes, we can look for potential basal elements by finding motifs that are significantly over-represented in promoters versus transcriptional units.

We have chosen *Arabidopsis thaliana* as the prototype organism. This model plant has a workably small genome, with 28,000 genes in the January, 2002 version released by The Institute of Genomic Research (TIGR). Despite this modest size, it has moderate complexity that allows us to study both promoters and introns, which are often significantly shorter in the genes of lower eukaryotes. In general, plant genomics are not as well understood as yeast or flies, which have been intensively studied for a number of years. This means that co-regulated pathways and basal transcriptional elements are not as well defined in *Arabidopsis* as in these other organisms, leaving much room for

additional research and bioinformatics tools that can guide it. Unfortunately, this also means that a number of the genes annotated for *Arabidopsis* are not experimentally supported, making their definitions uncertain. This can potentially lead to more noise than would exist for organisms that are better understood.

We have chosen to focus on motifs of 9 nucleotides in length. With $4^9 = 262,144$ possible DNA 9-mers, each motif occurs often enough to compute meaningful statistics and rarely enough to minimize the background noise. For each motif, the database holds the gene, region, and position of every occurrence of that motif. This information can be quickly queried to yield various interesting information:

- Ratios can be computed of the number of times each motif occurs in a subset of genes versus the rest of the genes. When done in promoter regions, this approach is essentially a word-finding approach to identifying over-represented motifs. This database allows the extension of this to introns and exons. It is not yet clear what purpose over-represented intron motifs in a subset of genes serve. These are questions that may be further explored in a lab and may or may not yield useful information.
- Ratios can be computed at the genomic level, comparing the number of times each motif occurs in one region versus another. We have used this to compare promoter regions to transcriptional units and have identified novel motifs that appear to be involved in the control of large numbers of genes. We have also compared motifs in introns versus exons and have identified known splicing-

related sites, as well as unknown, non-repetitive motifs that prefer introns to exons. It is not clear what biological function these may serve.

- Position statistics have been computed at the genomic level for each region. This allows us to quickly mine the vast amounts of data for motifs that have a potentially non-random distribution. This approach has yielded several known splice sites in introns and transcription factor binding sites in promoters, as well as novel motifs in both regions.
- These same statistics can also be computed for subsets of genes and compared to the rest of the genome. That is, given a subset of genes, we can quickly find motifs that are significantly more localized in the subset than in the genome as a whole. This suggests that these motifs serve a specific function for the given subset.
- Individual 9-mers, sets of 9-mers describing consensus sequences, and any n-mer that's shorter than 9 can be graphed as a histogram and statistically analyzed. This can be done at the genomic level or for a subset of genes, and can include a specific region or a range or positions relative to the start of transcription. This allows us to quickly visualize the distribution of a motif or set of motifs. We can also use this to compare the distribution of a motif with its reverse-compliment, and so gain some insight into the necessity of orientation for that motif.

Thus, this database, and the corresponding user interface, represents a novel approach in identifying potential functional elements in *Arabidopsis thaliana*. These functional elements are likely to be regulatory elements of some kind, though the interpretation of

function depends on the circumstances in which the motif is found. Ultimately, function can only be confirmed by wet-lab approaches. As such, this is intended for use as a bioinformatics tool that can guide a researcher towards motifs of interest. The methods this database uses are by no means organism-specific. While we have chosen to use *Arabidopsis* as a prototype, other organisms have recently been loaded into the database as well and are currently being analyzed.

Methods

The purpose of this section is to provide a detailed account of how the database was created and the information therein generated. For a high level description of the contents of the database and website, as well as a discussion of the biological implications, please see Results.

Information source

The sequences and annotations were sequenced and compiled by The Institute for Genomic Research (TIGR). The five chromosomes were submitted to GenBank on January 10, 2002, under the version numbers NC_003070.2, NC_003071.1, NC_003074.2, NC_003075.1, and NC_003076.2, for chromosome 1 through 5, respectively. These GenBank flatfiles were parsed with bio-perl scripts, written by Dan Simola '03, into files that could be loaded directly into the database. These files were loaded into the following tables: Identifiers, Descriptions, BlockInfo, BlockSequence, and the various RegionInfo tables. This included the information for all regions except the upstream region. Please see Appendix A for a description of how we interpreted the GenBank flatfile tags, what was included in each of the files that were loaded into the database, and the definitions of the various regions. GeneIDs were created during the parsing that identified each unique gene. Model numbers identified different alternative splicing variations of the same gene.

Creating the tables

The database used was Sybase Adaptive Server Anywhere, version 8.0. The ASA database management system (DBMS) includes the 1.3.1 Java Virtual Machine and a limited version of the corresponding Java libraries. We used this embedded Java capability extensively in creating new tables.

Once loaded into the database, private and foreign keys were placed on each table to insure basic data integrity (see the database schema in figure 1). These keys make sure that no gene is represented twice and that all genes in the Description, Identifiers, BlockSequence, or RegionInfo tables also occur in the BlockInfo table. In this way, BlockInfo serves as the gene definition table central to gene-related integrity.

Once the initial integrity was established, RegionSequence views were created (one for each region) that used the RegionInfo start and stop positions as pointers into the sequences stored in the BlockSequence table. Each sequence was made 8 bps longer than the start and stop suggest, allowing for a 9-mer that starts in the last position. These views serve as a means of direct lookup for the sequences corresponding to any region of any gene.

With the sequence views in place, a number of manual integrity tests were performed. The sequences of all the regions for each gene were checked to make sure they constituted a continuous, non-overlapping block that matched the BlockSequence record for that gene. For the fixed-length regions up1500 and down500, every gene was

checked to make sure it was the appropriate length. Finally, for each region, the sequences of all the genes were compared to ensure there were no duplicate sequences that had different gene names. An exception was made for alternative splicing models that may have portions that are identical between models. This was accounted for by the different model numbers.

Once the integrity of the initial data was established, the upstream regions were created. These are defined as the region upstream of each gene that extends 1500 bases or to the previous transcriptional unit (TU), whichever is shorter. In the case of overlapping genes, the upstream region is defined as 1500 bases upstream of the gene start. The TU regions were defined as the continuous block of DNA extending from the 5' end of the most 5' exon to the 3' end of the most 3' exon. To determine the upstream regions, overlapping TU regions were grouped into continuous blocks of transcribed DNA (CBTs), and these blocks were sorted by the block starts. Each upstream stop site was then set to the corresponding up1500 stop site. By definition, this stop site must be 1 bp upstream of the TU region of the same gene. Thus, every upstream stop site was either in a CBT, or 1 bp upstream of a CBT.

To determine the upstream start sites, we had to look at the locations of the corresponding stop sites in relation to the CBTs. For each gene, if the upstream stop site was in a CBT, the upstream start site was set to 1500 bps upstream of the stop site. If not, the stop site must lay one base upstream of a CBT. In this case, the upstream start site is dependant on the relative location of the previous CBT. If the CBT immediately upstream of the

stop site is less than 1500 bases away, the upstream start site is set to be one base downstream of that CBTs end. Otherwise, it is set to 1500 bases upstream of the upstream stop site. In the case of genes that are on the complimentary strand, this logic must be reversed to take into account the fact that all positions refer to bases on the primary strand. The logic for this was carried out by embedded Java, which created an UpstreamInfo text file in the same format as the other RegionInfo files. This file was then uploaded into the database and a corresponding UpstreamSequence view was created.

The motif tables were created from the sequence views, with a separate motif table for each region. For each sequence, a sliding window of 9 bps took each motif and created a record that included the motif, the geneID and model number of the sequence's gene, and the position relative to the start or stop (depending on the region) of the sequence that the motif was found. This logic was carried out by embedded Java. Indexes were placed on the Motif and geneID fields to allow for fast lookups and grouping.

The statistics tables were generated by embedded Java from the motif tables. For each motif in a given region, all the instances of that motif were pulled out of the RegionMotifs table and ordered by position. The specific statistics were calculated from this ordered list and the results were compiled into a single entry in the statistics table. This was done for each motif. In the case of exons and introns, it was done twice, once for positions calculated from the 5' end of the sequence, and once for positions calculated from the 3' end of the sequence.

Statistical equations used

The following equations were used to calculate the statistics. For each statistic, let

n = number of instances of that motif in the given region, and

pos[1..n] = the ordered list of positions in the given region

Standard Deviation

The population standard deviation was used in accordance with the conventions of the microarray community. The reason for this is that the statistic is a population statistic, since we have all instances of a given motif in the given region. The equation used was:

$$\sigma = \mathrm{SQRT}(\sum(\mathrm{mean}(pos[1..n]) - pos[i])^2 / n)$$

Expected Distribution

The expected distribution was used to calculate both the χ^2 and the Kolmogorov-Smirnov statistics. In the former, it was further binned, while in the latter it was converted into a cumulative probability function.

For each region, let Len(seq) be the length of any sequence seq in that region, and Count(len) be the number of sequences in that region such that Len(seq) ≤ Count(len). Let MAX_LEN be the length of the longest sequence in the genome of that region. Then any motif can occur in any position in the interval [1..MAX_LEN]. The probability prob[i] that a motif will occur in position i is determined by the distribution of sequence lengths in the region and is given by the equation

Prob[i] = Count(i)/MAX_LEN.

This probability array is easily converted into binned probabilities and cumulative probabilities.

χ^2

χ^2 values were computed using a bin size of 50. For each motif, the position array was binned and compared to the binned expected probability array. For each motif, let bin_pos[i] be the number of positions that are in bin i, and bin_exp[i], be the expected number of positions in bin i. The χ^2 statistic is given by the equation

$$\chi^2 = \sum \frac{(\text{bin\_pos}[i] - \text{bin\_exp}[i])^2}{\text{bin\_exp}[i]}$$

The degrees of freedom associated with each χ^2 value is equal to ν-1, where

ν = number of bins
= $\lceil$ MAX_LEN / bin_size $\rceil$

Kolmogorov-Smirnov

The Kolmogorov-Smirnov (KS) statistic requires that the position and expected probability arrays be converted to cumulative distribution functions. In our case, the cumulative distribution function defines for each position *x* the percent of all *n* positions whose values are less than *x*. To convert the prob[] array into a cumulative probability array, we use the recursive equation

cum_prob[i] = cum_prob[i-1] + prob[i], for i between 1 and n and cum_prob[1] = 0.

The pos[] array is converted into an unbiased estimate of the cumulative distribution function by the equation

$$cum\_pos[i] = i/n$$

Note that these two arrays are not of the same size. Length(cum_prob[]) = MAX_LEN and Length(cum_pos) = n (see above for notation). Moreover, cum_pos[i] = the unbiased estimate of the actual probability that a given motif's position will be upstream of pos[i], and cum_prob[pos[i]] = the expected probability that a given motif's position will be upstream of pos[i]. Thus, given a random distribution, we expect

$$cum\_pos[i] = cum\_prob[pos[i]]$$

The KS statistic is defined to be $D = MAX(| cum\_pos[i] - cum\_prob[pos[i]] |)$ taken over $1 \leq i \leq n$.

To compute the significance of a D value, we use the equation

$$Q(\lambda) = 2\sum(-1)^{j-1} e^{-2j^2\lambda^2}$$, summed from 1 to infinity,

which has the limiting values of Q(0) = 1 and Q(1) = 0. From this, the p-value giving the significance of a D value is given by

$$p = probability(D > observed) = Q([\sqrt{n} + 0.12 + 0.11/\sqrt{n}] * D)$$

(KS equations taken from Press et al., 2002, pp 623 – 624).

Ratios

Group-to-other ratios were computed by comparing the motifs found in the given region in the group of genes versus all the other genes. The calculations were done using the same formulas as region-to-region ratios. We will therefore look only at group-to-other ratios.

For each motif in a given region, let count(motif, group) = the total number of instances of that motif in the group of genes, and count(motif, other) be the total number of instances of that motif in all other genes. The group-to-other ratio is defined to be

Ratio(motif) = count(motif, group) / count(motif, other).

The ratio can be hard to interpret, since there may be many more motifs in one set than in another. To normalize for this, we calculate the normalized ratio as

Norm_ratio(motif) = Ratio(motif) * Total(other)/Total(group),

where Total(group) = total number of motifs in group in the region of interest, and Total(other) = total number of motifs not in the group in the region of interest.

Website specifics

The website is powered by an Allaire® JRun webserver, which implements version 1.3.1 of the Java Virtual machine and can connect to the database using the Java Database Connectivity (JDBC) API. The website is written using Java Servlets, Java Server Pages (JSP) scripting, and standard Java objects. The general flow of information on the website begins with HTML forms that collect user input. This information is passed by the servlets to Java objects that process the information and generate SQL queries. These queries are submitted to the database over a JDBC connection. The results of the queries are sent back to the calling Java objects, which may further transform the data before sending it back to the servlets, where it is finally encoded in HTML and sent to the client's browser.

For a complete description of how to use the website, please see Appendix B. For a general overview of what can be done with the website, please see Results.

Results

This section is divided into three parts.

1. *The Database*: Describes the general setup of the database and how the information was obtained and generated. This is generally a high-level discussion intended to describe how the design of the database can be used to identify non-randomly distributed motifs. For a detailed description of how the database was built and tested, please refer to Methods.
2. *The Website*: Describes the general setup of the website, emphasizing how the website serves as an interface that adds functionality to the database and enables efficient identification and analysis of non-randomly distributed motifs
3. *Applications of the database and website*: Describes a number of approaches that use the database and website to identify non-randomly distributed motifs. This serves as both a demonstration of how the database and website can be used and a biological discussion of the significance of the motifs we identified. Both experimentally confirmed and novel motifs are identified and discussed.

Part 1: The Database

Source of genomic data

GenBank serves as a central location where various researchers and consortiums can deposit partial sequences and entire genomes. This database is provided free to the public, allowing anyone to download any genome that has been entered. These genomes have a standardized format that includes gene-specific information for each gene. This information includes the

start and stop positions for regions that are transcribed, called transcriptional units, or simply TU regions, the start and stop positions for exons and introns, various unique identifiers, and various descriptions of what is known about each gene. If a gene is known to be alternatively spliced, all of this information is stored for each method of splicing, which we refer to as a "model".

We have taken this GenBank information for *Arabidopsis thaliana* and loaded it directly into our Sybase® database in a relational format. This means that we have a number of two-dimensional tables that contain specific information. Each column of each table holds a certain type of information, and each row represents a separate record. While the number of columns and type of data allowed in each columns remains static for each table, the number of rows and specific information in each row can be easily changed. The power of a relational database lies in the way these tables are linked together. To link the records in the tables, we defined our own databaseID and modelNumber for each gene and model. Thus, every record in the database that has something to do with gene information has geneID and modelNumber columns. The relational format of the database allows for easy retrieval of both the original data and emergent information that reveals itself when the data is combined and grouped in clever ways. This is made possible by the querying capabilities of the Structured Query Language (SQL) and the efficient indexing and optimizing capabilities of Sybase.

The information in the database can be segregated into three general categories:

1. Gene-specific information from GenBank

2. Region-specific information from GenBank
3. Motif-related tables we generated from gene and region tables

We shall examine each in turn. Throughout this discussion, please refer to Figure 1, which shows the structure of the database in the form of an entity-relationship (E-R) diagram.

Gene-specific tables

Gene-specific tables store information that relates to entire genes and their flanking sequences. This information includes the raw sequence, which we call the blockSequence, that covers from 1.5kb upstream of the gene to 0.5kb downstream of the gene, as well as descriptions and identifiers for each gene. General gene location information is stored in the blockInfo table, which includes the chromosome the gene is found on, the number of alternative splicing models the gene has, and the 5' and 3' ends of the blockSequence. The positions of the 5' and 3' ends are computed relative to the 5' end of the chromosome.

The identifiers table stores all available identifiers for each gene and model combination and relates them to the gene and model that they represent. Each entry in the table has the ID, the type of ID, and the corresponding databaseID and modelNumber. For Arabidopsis, available IDs include the pub_locus ID, the GI number, and an "NM" Accession number, which is the GenBank accession for the representative entry of a given section of DNA. Since these identifiers do not overlap, a user need not identify what type of ID he is looking up to find the corresponding databaseID. This allows for a great deal of flexibility and for quick updates as new types of IDs become available.

We have taken a similar approach in storing all the available descriptors. Each description for each gene is stored as an entry in the descriptors table, along with the type of information GenBank identifies it as and the geneID and modelNumber that the description is about. Unfortunately, the information provided in GenBank is often sparse and is not peer-reviewed, making for often out of date or even wrong information. The lack of reliability in the gene descriptions is important to keep in mind as we begin to examine cis-regulatory motifs and the genes they are associated with. The format of this table does, however, allow for easy updates as more and better information becomes available.

Region-specific tables

The GenBank format lends itself to easy identification of discrete regions, as these are flagged by the various GenBank tags. For example, the GENE flag marks out the full transcriptional unit of a protein-coding gene, from the 5' end of the most 5' exon to the 3' end of the most 3' exon. mRNA tags identify the start and stop positions of the sequences that make up the final mRNA, as would be found in a cDNA, and so define exons and introns. CDS tags identify the coding sequences and differ from mRNA tags only in that the CDS constructs lack the untranslated regions (UTRs). Thus, the UTRs can be deduced by the difference between a CDS and mRNA construct. Finally, proximal promoters can be deduced by taking the regions immediately upstream of the mRNA constructs. For a complete description of how we interpreted the GenBank file format, please see Appendix A.

It is important to note that full-length cDNAs are difficult to generate and rarely include the entire 5' UTR. Furthermore, gene-finding programs are often best at finding open reading frames (ORFs), and define exons and introns based on these. This means that such programs may miss UTRs entirely. While recent advances in cDNA technology have allowed high throughput sequencing efforts to identify many full-length cDNAs, only 33.5% of the protein-coding genes in the *Arabidopsis* genome have defined UTRs. The problem is that the GenBank gene start definitions are supposed to be transcriptional start sites. In those genes that contain UTR info, this is probably the case. For the genes that lack UTR info, however, this cannot be true. More likely, the gene-start defined for these genes is probably the translation start site. This introduces a slight amount of noise in the position data, which tends to elongate the humps in the distributions that represent preferred motif locations. This problem is discussed in more detail below.

The actual size of promoters varies from gene to gene, so it is important that we take a large enough region that we include the entire proximal promoter of all genes. Various motif-finding programs define the size of these differently, but typically take 500 to 1000 bases upstream of the transcription start site. We have chosen to take 1500 bases upstream to try to ensure that we have the entire proximal promoter. This has the advantage of highlighting motifs that have biological preferences for positions in ranges that include several hundred bases, as the relative absence of such motifs in the other positions will make it more likely that these motifs are flagged as non-random. This also means that for a smaller genome like Arabidopsis, most of the intergenic region is included within 1500 bases of the start of the gene. We have called this region the up1500 region.

The downside of this is that, in the case where two neighboring genes are within 1500 bases of each other, the up1500 regions often extend into the previous gene. Since coding regions are known to have a different base composition, and so different expected numbers of each motif, than intergenic regions (indeed, this is how many gene finders work), this fact may skew statistics that assume uniform randomness. To get around this, we have defined "upstream" regions to include 1500 bases upstream of the transcription start site, or to the previous gene, whichever is closer. Approximately 50% of the up1500 regions were truncated in the upstream regions. Interestingly, there were 369 instances where two transcriptional units overlapped, and in every one of those cases, the overlap was caused by two genes that are transcribed from different strands of DNA. In these cases, the upstream region defaulted to 1500 bases.

For each of these regions, we keep a table that stores the start and stop (5' and 3') positions of that region for each gene and model, as well as the exon or intron number, if applicable. The start and stop positions are relative to the 5' end of the chromosome the gene is on. Since the blockInfo table includes the start and stop positions of the block sequence, the relative start and stop position of each region within the block can be computed. These numbers can then be used as indexes into the blockSequence table. We can therefore quickly extract the sequence of each region in each gene without storing the sequences redundantly. These lookup definitions are stored in database views, which act to the user as tables but contain only the logic necessary to lookup the desired information.

The motif tables

One of the major goals of the database is to enable fast analysis of motifs with respect to the regions they lie in, the genes they're associated with, and the positions in which they're found. To accomplish this, we have created motif tables, with a separate table for each region. Each table has fields (columns) that keep track of the 9-mer sequence, the geneID, modelNumber, and intron/exon number (if applicable), and the position in which it's found. There is a separate record (row) for each instance of every motif.

To generate the motif tables, we use the gene and region specific sequences. To do this, we simply scan along each sequence, using a sliding window of 9 bps, and insert each motif into the motif table. Each entry in the table consists of the motif, the databaseID and modelNumber of the gene, and the position relative to a given start site. In the upstream and TU motif tables, the start site is the gene start specified by GenBank, which is either the start of transcription or the start of translation. For exons and introns, we store the position of each motif relative to both the 5' end and the 3' end of the exon or intron. The reason for this is that motifs that favor one end of the exon/intron will not appear to have a preferred location due to the varying lengths of the given region. In each region, the start site is 1 and all positions upstream of that site are negative, starting with –1. Since the motif length is 9, each sequence has an extra 8 bases on the 3' end so that we can store the 9-mer that begins in the last position of the region.

For the *Arabidopsis thaliana* genome of 28,000 genes, these five motif tables store a total of 186.6 million records, ranging from 18.6 million records in the IntronMotifs table to 58.0

million records in the TUMotifs tables. Indexing allows for extremely fast lookups despite these large table sizes. Furthermore, the motifs are stored in alphabetical order based on their sequences, which allows for quick lookups of all instances of a motif, as well as all instances of alphabetically adjacent motifs. This means that a n-mer, where $n < 9$, can be looked up very quickly simply by looking up all 9-mers whose first n bases match the n-mer. For example, to look up a 8-mer, we need only look up the 4 adjacent 9-mers whose first 8 positions match that of the 8-mer. This allows us to analyze individual motifs that are shorter than 9 nts.

Statistical information

The purpose of running statistics on the motifs is to find non-random motifs. This involves first identifying potentially interesting motifs, and then statistically determining whether or not these motifs are randomly distributed. Identifying interesting motifs within a set of 262,000 is a data mining problem. Ideally, statistics should be run only under human control, first looking at the distribution of a motif, then determining the appropriate statistics to run, but this is obviously not practical in this case. The purpose of data mining then is to scan the entire search space and identify motifs that should be analyzed further. We do this with a number of statistical analyses that are run on each of the 262,000 motifs and identify those motifs whose distributions differ from genomic trends.

Number of Occurrence and Mean.

For each motif and each region, we first count how many times the motif occurs in that region, as well as the mean position. This information is always helpful for later analysis.

Standard Deviation

Standard deviation is a measure of how spread a normal distribution is around the mean. Unfortunately, we have no reason to expect a normal distribution. Rather, for the up1500 region, in which every sequence is the same length, the uniform distribution is expected, while the expected distributions of the other regions depend on the lengths of the component sequences. Nevertheless, comparing all standard deviations reveals a general trend towards some mean standard deviation as the number of occurrence increases. This mean standard deviation varies among regions due to the varied lengths, but in upstream regions it is 413.59 bps. We can therefore generate a scatterplot of standard deviation versus the number of positions for all motifs to get an estimate of the expected standard deviation (see Figure 6). The standard deviation data mining technique exposes those motifs whose standard deviation represents an outlier when compared to the rest of the motifs. Such motifs will have either unusually localized distributions (low standard deviations) or unusually spread out distributions (high SDs). Specific analyses of such outliers are discussed below.

χ^2 *test of significance.*

The χ^2 test is a common biological statistical test used to determine if an observed distribution is significantly different from the expected distribution. It has the advantage that no assumption of normality is required for accurate results. Conceptually, the χ^2 value is a measure of the total amount of area in the difference of two distributions.

This statistic requires the grouping of observations into bins and has a number of assumptions that must be met in order for it to be statistically valid. The most relevant of

these for our purposes is the requirement that there be a sufficient number of expected occurrences for each bin, as small numbers will artificially inflate the probability of non-randomness. This is a problem for us since the number of occurrences for each motif ranges from 0 to 43,000. Nevertheless, χ^2 analyses can be used as a data mining technique in much the same way as standard deviation. By creating a scatterplot of the entire set of motifs, comparing number of occurrence against χ^2 values, we can find outliers that can be further analyzed (see Figure 10). Furthermore, this statistic can be converted into a p-value used to determine with some confidence whether or not the motif distribution is random. This must be used with caution due to the assumptions noted above. Due to the computational complexity of computing the p-value, and the lack of robustness of this test, we have not yet implemented the automatic calculation of the p-value from the χ^2 statistic. If a particular motif is being analyzed, this value can be looked up in a table. At this point, the χ^2 statistics is used primarily for the identification of outliers.

Kolmogorov-Smirnov test

The Kolmogorov-Smirnov (KS) test is the most robust of the tests we perform. Like the χ^2 test, it is based on comparing an observed distribution with the expected distribution. To do this, a single observed distribution is converted into an unbiased estimate of the cumulative distribution function and compared the expected cumulative distribution function. In our case, the cumulative distribution function defines for each position *x* the percent of all *n* positions whose values are less than *x.* Since we are analyzing monotonically increasing functions, we do not need to bin the data and can therefore analyze the distributions in much finer detail. The KS statistic is defined to be the maximum separation between the observed

and expected cumulative distributions. This means that a distribution that is identical to the observed with the exception of a large spike will still be marked as non-random. A binned analysis like the χ^2 will often obscure such distributions. (Press et al., 2002, pp. 623 – 626).

Another advantage of the KS test is that the p-value associated with the KS statistic is dependant on the number of observations and is easily computable. This p-value can be directly compared between two distributions with very different counts. This allows us to sort all motifs of a given range for those with distributions that are least likely to be random according to the KS test. It also allows us to compute a KS-derived p-value for any combination of motifs, including motifs that are shorter than 9 bps. Finally, p-values are easy for a human to comprehend as they represent the probability that a distribution would have a higher KS statistic than what is observed, which means the p-value can be loosely defined as the probability that a given distribution is random.

Summary

The statistics discussed above were all computed for each region and stored in the statistics tables. These tables are related to the motif tables by the motifs in a many-to-one relationship. This means there is precisely one record in a region's statistics table that summarizes all the information of all the records for that motif in the motifs table, irregardless of the genes that motif occurs in. These statistics tables complete the schema of the database (Figure 1). The central component of the database is the blockInfo table, which has a single entry for all 28,129 genes and keeps track of basic information about each gene. The descriptors and identifiers tables may each have a number of references for each gene in

blockInfo, but cannot reference any gene that is not in blockInfo. The blockSequence table has exactly one entry for each gene consisting of the DNA sequence for the entire gene block, extending from 1500 bps upstream of the 5' most exon to 500 bps downstream of the 3' most exon. The info tables store the start and stop positions for each region in each gene, and so reference the blockInfo and blockSequence tables in a many-to-one relationship. Information from the blockSequence and info tables is used to determine the DNA sequences for each region. These are stored in the sequence views. The sequences referenced in these views are then scanned using a sliding window of 9 bps, and each motif encountered is entered into the motif table, along with the gene and position it was found in. Various statistics for the positions of each motif in the motif tables are computed and stored in the stat tables. In the case of introns and exons, this information is stored both for positions that count from the 5' end of the region, and for positions that count from the 3' end of the region.

This schema can easily be duplicated for new organisms. Each table is owned by the organism it is associated with, so as new organisms are loaded, we simply duplicate all the tables and make them owned by the new organism. When querying the database, the organism name is appended to the front of the table name. We have designed these tables in such a way that the same format can be used for all organisms that come from GenBank. This is particularly important for the descriptions and identifiers, as the format for these varies between organisms. Thus, by giving each gene our own ID, we can assure compatibility with all organisms, while keep the conventional identifiers in the identifiers table.

Part 2: The Website

The database itself is designed in such away that many different questions can be asked and quickly answered. However, typing SQL commands can be tedious and interesting queries often require complicated SQL commands. Not only that, but often times results are most meaningful if viewed as graphics as opposed to tables. This is especially true when we want to look at position information in the form of histograms. Therefore, to allow easy access to the database, we have designed a web interface that takes user input, generates and submits SQL queries, and then formats the results of the queries as either tables or graphics interpretable by the user.

While the ultimate goal of this website is to allow biologists to use the database to help identify regulatory motifs in their search space of interest, an important step in the development process is to allow for easy development and quick expansion. To this end, the website is powered by Java Servlets, the Java2 SDK, and Java Servlet Pages (JSP) scripting. This combination of scripting and object-oriented languages allows for easy abstraction and expansion. For example, information the user enters is stored in a Javabean, and so easily persists and is accessible to all pages on the site. The combination of this Javabean and the object that generates all SQL queries allows us to quickly add functionality to the website as new ideas and needs arise.

The layout of the website is optimized both for ease of use and easy expansion. This site has a Results page and a Charts page, but centers around a central Main Query Page, which

allows the user to specify a number of parameters that are common to most queries (Figure 2). Such parameters allow a user to specify specific motifs, genes, regions, and positions.

Basic parameters

The motifs parameter allows a user to specify a list of 9-mers, a consensus 9-mer in the form A-CG-T-AC, or any single motif that's less than 9 bps (Figure 2A). The user can also define a maximum number of mismatches allowed on lowercase characters. For each of these options, the reverse-compliment (that is, the equivalent on the other DNA strand) can also be included in the query. It is important to note that only 9-mers are stored in the database, and, with the exception of the charts that are created, none of these motifs are clustered. This means that the set of motifs the user specifies are expanded to all the 9-mers they represent, each of which is reported separately. If no motifs are specified, all motifs in the search space are reported.

The region and range parameters allow a user to specify which region should be analyzed, with the option of limiting the range of that region (Figure 2B). Alternatively, only a range of positions relative to the start of each gene can be specified. This combines the motifs from the up1500 region and the TU region and selects only those whose positions that fall in the specified range. The default range is wide enough to cover all motifs.

The geneList parameter allows the user to limit the search space to a set of genes (Figure 2C). These gene names may simply be pasted in, or can come from a number of saved lists. New lists can also be created at any time. The names of the genes match any ID that is

currently loaded in the database. At the time of this writing, the available *Arabidopsis* IDs are pub_locus, GI number, and Accession number. If no genes are specified, all genes are used as part of the search space.

Available Results

Once the basic parameters have been set, the user can retrieve a number of different types of results (Figure 3). With the exception of charts, all results are displayed in the Query Results page in a common format that includes the SQL query used to generate the results, a description of the results, and a table that contains the results (Figure 4A). A number of different types of results can be displayed in this format

Sequences and Descriptions

The most basic type of query is the retrieval of sequence and description information. This allows the user to retrieve sequences from any region, as well as any descriptions we have on the genes of that region. When looking for specific motifs, retrieving descriptions also includes each position that each motif is found in each gene, as well as the exon or intron number it's found in, if applicable. These types of queries are often useful in looking for gene similarities among genes that have a common motif. Due to the lack of reliability in the GenBank descriptions (discussed above), the descriptions must be taken with a word of caution. To help get around this problem we have linked the pub_locus identifiers, which are displayed when descriptions are retrieved, directly to the TAIR Locus Detail page on www.arabidopsis.org. This page displays the latest sequence, source, and description data available for the given pub_locus, as well as all known protein IDs.

Statistics

Pre-computed genomic-level statistics can quickly be retrieved and sorted in any order (Figure 3A). This allows for the quick lookup of those motifs that are least likely to be randomly distributed in a given range. If specific motifs are identified, then only the statistics for those motifs are retrieved. Thus, if a researcher has a certain motif of interest, she can quickly lookup the position statistics of that motif in the region of interest. Furthermore, if a set of co-regulated genes is being analyzed, the distribution statistics of each motif in only those genes can be computed and stored temporarily for quick retrieval. When these results are retrieved, the p-value of the KS statistic for each motif is compared against the genomic KS p-value. This allows the user to quickly find motifs whose distributions are significantly less likely to be randomly distributed in the specified group of genes than in the genome as a whole.

Ratios

Two different types of ratios can quickly be computed (Figure 3B). Region-to-Region ratios compare the number of times each motif occurs in one region versus the number of times it occurs in another. Group-to-Other ratios compare the number of times each motif occurs in a given region in the group of genes versus the number of times it occurs in that region in the rest of the genes. Since the search spaces on each side of the ratios are not necessarily the same size, and so likely contain different numbers of 9-mers, the straight ratios are presented with normalized ratios. These normalized ratios reflect a linear transformation of the straight ratios, and so function only in making the ratios more human-readable.

User-defined queries

The user can also define his own query to be processed (Figure 3C). While this is primarily a development tool, it can also be useful to the general user who has some SQL knowledge by allowing him to tweak other queries to his liking. By presenting each result set with the SQL that generated it, a user can easily modify the query to address a specific question not answerable with the current setup of the Main Query Page.

Graphs

Finally, a user can graph the motifs that correspond to the various basic parameters (Figure 4B). Two different graphs are displayed on the graphs page: a cumulative distribution plot and a binned histogram. The cumulative distribution plot is what's used to calculate the KS statistic (see above). Each point (x,y) represents the percent of the population (y) that lies to the left of (x). The expected distribution is plotted as a green line. The histogram is a binned representation of the distribution, where the bin size is specified by the user from the Main Query Page. In both these plots, the graphs are grouped into two datasets corresponding to the two motif parameters (see fig 2A). This grouping means that data representing consensus motifs, for example, truly represent the group of motifs, not the individual 9-mers. Finally, the KS statistic and associated p-value is displayed for each group of motifs.

Part 3: Applications of the database and website

With the database in place, and a user-friendly web interface to access the data, we were able to identify non-randomly distributed motifs that are likely to be involved in gene regulation. Since our primary focus is on gene regulation, most of the results in this section will focus on upstream regions. All of the approaches employed here should, however, work for introns and exons as well. Some results from introns are presented here, but we are only beginning to look in depth at these regions.

Identifying non-random motifs at the genomic level

We will first look at techniques for finding motifs that are non-randomly distributed at the genomic level. The motifs we find using these methods are likely to be involved either in basal transcription or in the regulation of large sets of genes.

Upstream : TU ratios

Those motifs which have the highest upstream:TU ratio are shown in Table 1. Only those motifs that occur at least 10 times in upstream regions are reported. Of the top 25 motifs, 20 are some variation of the 10-mer AGGCCCATTA. It is important to note that the sliding window creates some artifacts, in that every time a given 9-mer occurs, the first 8 bases will make up another 9-mer that includes one of four bases preceding the 8-mer. There are only four such 9-mers, and so these will often be flagged as non-randomly distributed as a result of the true 9-mer's non-randomness. If we look at these 20 motifs, we can see a general consensus of aGGCCCatta, where GGCCC is conserved in all but one of the motifs. Since

the database is designed to analyze motifs that are longer than 9bps, we will focus on GGCCCatta and its reverse compliment.

Figure 5 shows the distribution of GGCCCatta with one mismatch allowed in the lowercase part of the motif. The p-value as estimated by the KS-statistic is $1.8x10^{-125}$ and $9.4x10^{-78}$ for this motif and its reverse compliment, respectively. Both the motif and the reverse compliment have a nearly identical distribution (the difference in p-values is due to the number of occurrences of each motif), indicating that motif orientation probably does not affect the function of this motif. The motif shows a clear preference to being at least 60 bps upstream of the gene start and generally closer than about 200 bps. This motif is clearly not randomly distributed either by region or by position within promoter regions. A motif search on plantCARE revealed three possible matches to GGCCC. All three were involved with light responsiveness in other plants. The Chs-unit 1 motif included GGCCCAT as part of a 42 base exact sequence of a light-responsive element experimentally confirmed in one gene in *Zea mays*. It is possible that GGCCCatta is part of an *Arabidopsis* version of this light-responsive element, though the evidence is hardly sufficient to warrant any sort of conclusion, especially considering that none of the genes with GGCCCatta or its reverse compliment within the range of –200 to –60 have "light" as part of their description. No matches for the reverse compliment were founding plantCARE.

To look for a possible role for GGCCCatta in gene regulation, we queried the Descriptors table for words common to those genes with variations of this motif in their promoter regions. The motifs tested were all motifs included in aGGCCCatt, GGCCCatta, their

reverse compliments, and variations with 1 mismatch in a lowercase position. This affectively included all genes that have either one of the 9-mers or the 10-mer aGGCCCatta in either orientation, and with at most 1 mismatch, in their promoter region. Only genes with the motifs within the range –400 to –50 were included. The range corresponds to an approximation of the preferred location of GGCCCatta. The results showed that 91 of 218 (42%) ribosomal proteins had one of these motifs in the given range of their promoter regions. The other top classifications included RNA polymerase (30%), DNA polymerase (21%), splicing factors (15%), and helicases (14%). Approximately 11% of all genes in the genome have one of these motifs in their promoters. It is hard to draw solid conclusions from these data, but it appears aGGCCCatta may serve in regulating the basic gene-expression machinery.

Intron:Exon ratios

Table 2 shows those motifs that have the highest Intron:Exon ratio and occur at least 10 times in introns. Not surprisingly, the motif GTaAtG shows up as the most prevalent motif in the top 25 ratios. This motif is known to be the strong consensus of all 5' splice sites in eukaryotes, with near 100% conservation in the first two positions. A less conserved sequence is the branchpoint, usually located 18-60 nts upstream of the 3' splice site. The ACTAAc/tTA motif in the table matches the weak consensus for the Arabidopsis branchpoint. Similarly, the TGTAGGc/tTG motif is close to the 3' splice site consensus of TGCAGGT (Brown et al, 1996). Indeed, 322 of 401 (80%) instances of these two motifs in introns occurs at position –5, making the AG the last two bases in the introns. While none of these motifs are novel, it is reassuring to know that we have immediately identified all three

splicing-related motifs. The other motifs in this table are low complexity repeats, which are known to be common in introns.

Standard Deviation

As discussed above, the purpose of using standard deviation as a data mining technique is to identify motifs whose standard deviations represent outliers from the rest of the population. Figure 6 shows a scatterplot of all upstream motif standard deviation values versus the number of occurrences. There is clearly a preferred standard deviation that moves towards 413 as the number of occurrences increases. If we zoom in, however, we can see that a number of motifs lie outside the general trend. I have chosen two random outliers, along with one motif that has an average standard deviation, to evaluate further.

The motif AAAATGGAG has an abnormally large standard deviation. Surprisingly, the distribution is extremely localized (Figure 7). The high standard deviation is probably due to the effect the extreme localization has on the mean. The internal presence of ATG in this motif, along with its clear localization around positions –10 to –1, suggests that this motif is involved in translation initiation. Indeed, a search for this motif reveals that it is very similar to the consensus ATG codon context in dicots, which is aaA(A/C)aATGGCt (Joshi et al., 1997). This is not surprising, given that *Arabidopsis* is a dicot. Interestingly, the monocot context consensus of c(a/c)(A/G)(A/C)cATGGCG has the same distribution as AAAATGGAG, but occurs three times less (data not shown).

Motifs with abnormally low standard deviations may be interesting as well. The motif ATTACCCCA is one such motif (see Figure 6 for standard deviation relative to the

population). As can be seen from its distribution (Figure 8), this motif has a non-random preferred location of –800 to –400 (KS p-value = 5.7×10^{-4}). Examination of the 88 genes that contain this motif revealed that of the 37 genes for which at least a putative function was identified, 30 (81%) were retrovirus related, including 25 retroelements, 3 reverse transcriptases, and 2 transposons. The distribution of the reverse compliment is very different and randomly distributed (KS p-value = 0.13), suggesting that the orientation of this motif is important. This motif is not in the plantCARE database and is an apparently novel sequence.

As a control for the standard deviation data mining technique, we chose, an ATG-containing motif with a non-deviant standard deviation. Its distribution is shown in Figure 9. This distribution clearly shows no preferential location, and is actually more uniformly distributed than the expected distribution, as reflected in its KS p-value of 2.7×10^{-4}. This low p-value is due to the peak in bin –525. It is not clear if this is due to random events. It should be noted that with 262,000 motifs, apparently non-random distributions are likely to happen by chance. A certain amount of biological inference is therefore necessary when analyzing these distributions. This motif is not in the plantCARE database, and while its p-value is somewhat low, its generally uniform distribution suggests that it is not likely to serve a functional role.

χ^2 *Analysis*

Like standard deviation, χ^2 analysis is a useful data mining technique despite the assumptions many of the motif distributions violate. The usefulness is due to the fact that the population

of motifs shows clear trends in χ^2 values that depend on the number of occurrences. By analyzing a scatterplot of χ^2 versus number of occurrences, we can quickly see these trends and identify motifs that deviate from them. Figure 10 shows the population χ^2 values as well as a zoomed in look at where the majority of motifs lie. CTCTCTCTT and TTCTCTCTC represent clear outliers in this scatterplot. The distributions of both motifs are very similar, as are the sequences. Figure 11 therefore shows the combined distribution of these motifs, as well as their reverse compliments. Remarkably, the reverse-compliments, known as GAGA motifs, show a clear preference for being located within 40 bases of the gene start (KS p-value = 1.3×10^{-33}). Furthermore, 180 of the 313 (69.0%) distinct genes that contained one of the GAGA motifs within 40 bps of the start site did not have defined UTRs, indicating that the gene starts in these cases are likely translation start sites. Thus, it appears that the GAGA motif primarily lies in the 5' untranslated region and functions either in regulating translation initiation or mRNA stability. It may also serve as a transcriptional regulator that operate downstream of the basal machinery. While this motif is not a known plant cis-regulatory element, it is known to be involved in the transcriptional regulation of *Xenopus* (Li et al., 1998).

The distribution of a motif with a typical χ^2 value is shown in Figure 12. This distribution more closely reflects the expected random distribution, which is slightly skewed due to the variation in the lengths of the upstream regions. Low KS p-values of 2.4×10^{-5} and 8.7×10^{-4} reflect the apparent preference of these motifs to not be within 200 bps of the gene start. Not only are these motifs AT-rich, but all motifs whose count and χ^2 values are within 50 units of these motifs are also AT-rich. We randomly graphed the distribution of some of these, and

all were similar in shape to that of Figure 12. It is interesting that the apparently typical AT-rich distribution avoids the very region that is known to be AT-rich. This may indicate that very specific sequences are necessary in the range of –200 to –1 and the presence of random AT-rich sequences is therefore detrimental. Another possibility is that we are seeing an artifact of the mixing of transcription and translation start sites. The average length of defined 5' UTR regions in the database is approximately 100bps, and 66% of the gene start sites are translation start sites. The tendency for the typical AT-rich motif to avoid ranges immediately upstream of the gene starts may be due the fact that UTR regions are typically not as AT-rich as intergenic regions. The fact that the general slope of the histogram starts to change between –300 and –200, and then accelerates around –100, may suggest that it's a combination of both UTR regions and promoter specificity.

Another motif identified by its relatively high χ^2 value was TATAAATAC (Figure 10). This motif matches the *Arabidopsis* TATA-box consensus, given by plantCARE as TATAAATA. The remarkable distribution of the TATA-box (Figure 13) allows us to look more closely at the effects of translation and transcription start sites in the database. The TATA-box is used to position the basal machinery in the right location, so the position of this motif is essential. Most of the time, this motif is positioned approximately 30 bps upstream of the transcription start site. Note that positions included in the bins range (in negative numbers) from 0-9, 10-19, 20-30, and so on. The bin size used to generate Figure 13 was 10, which means the spike includes positions –39 to –30. This means that the TATA-box occurs closer than 30bps less often than it occurs 1000 bps away. This emphasizes the importance of precision in TATA-box location. But this distribution has not only a spike in the range –39 to –30, but also a

hump in the range –140 to –60. This hump clearly peaks around –80, then drops sharply around –50. When we look at the genes that have the TATA-box in the range –35 to –30, we see that 231 of 254 (90.9%) have defined UTR regions, compared to the 33.6% genomic average. Given the precision of the TATA-box placement, one would expect that the –140 to –60 hump represents genes whose upstream regions include UTRs. In fact, only 152 of 825 (18.4%) of genes with a TATA-box in this range have defined UTRs. This suggests that the distribution of UTR length is skewed, as the average UTR length of 100bps would predict the peak of the hump to be at –130. More likely, most unidentified UTRs are around 50bps in length (the center of the hump), with some longer ones skewing the average. Note that this observation places greater weight on the sequence specificity explanation for the typical AT-rich motif distributions discussed above, as the base composition argument is dependant on longer UTRs whose lengths correspond with the drop in the distribution.

Base composition

To further assess the hypothesis that these distributions are the result of UTRs in the first 100 bases, we plotted the base composition of all genes using the range –500 to +200 and a bin size of 10 (Figure 14). Remarkably, this showed that G and T compositions remain stable until the gene start, while A begins to fall around –150 and C begins to rise around –200. There is a clear disturbance at the gene start, and base compositions are very different from there on. This corresponds to the change in base composition seen between coding and non-coding regions that many gene-finding algorithms use. Interestingly, when we look at our example of a motif with a typical χ^2 value, we see that the motif is dominated by T's while the reverse compliment is dominated by A's. Indeed, few of the typical motifs have an even

number of A's and T's. What's remarkable is the fact that the motif and reverse compliment have nearly identical distributions, as was the case with nearly all the typical motifs we randomly chose to graph. This does not correspond to the base composition graph, which suggests that AT-rich motifs that are heavily dominated by T's should have distributions that remain random until perhaps the bases within –50 of the gene start. Furthermore, the drop we see in these distributions appears to be more that the drop in the composition of base A. This all suggests that perhaps the hypothesis that the non-random distributions of these typical motifs are caused by promoter sequence specificity and not UTR base composition is correct.

Kolmogorov-Smirnov

The final genomic-level statistical test to look at is the KS test. As discussed above, the p-values associated with the KS statistics normalize for the number of occurrences. This means that the genomic scatterplot of these values do not show obvious trends the way χ^2 and standard deviation scatterplots do (Figure 15). One trend that does appear, however, is the fact that most motifs that occur over 5000 times have low p-values. One motif that violates this observation, TTTCTTTTT, with a p-value of 0.22, serves as a good example of what the expected distribution is, though it should be noted that this distribution drops in the range –200 to 0, which is not expected (Figure 16). This may be related to the reasons discussed above. Nearly all of the other motifs that occur more than 5000 times are repetitive AT-rich sequences. The fact that these all have low p-values reflect the observation made above regarding the tendency for these motifs to avoid being within 200 bps of the gene start. Remarkably, when we graphed a number of motifs that occur more

than 5000 times, we found that those that had a high AT content avoided the first 100 bps, while those that had any number of C's or G's preferred the first 100 bps. A typical example is shown in Figure 17, which is the distribution for ATATATATA and CTCTCTCTC. This may reflect the genes whose start sites represent translation start sites. This is supported by the fact that the defined UTRs average around 100 bps in length and these distributions cross around –100. This is clearly more than a base composition issue, however, as the localization is much more severe than the base composition alone would warrant, and motifs with the same base composition as CTCTCTCTC, but with an average number of occurrences, do not necessarily show this UTR localization.

Identifying non-random motifs in co-regulated sets of genes

Another goal of the database and user interface is to allow researchers to identify new regulatory motifs in subsets of co-regulated genes, as well as to evaluate known regulatory motifs for group specificity. As an example, we have analyzed a set of 68 co-regulated genes. These genes are all known to be regulated in response to the circadian clock, and are classified as being activated in the first of four phases (Harmer et al. 2000). Fourty-seven of the 68 genes (69%) have defined UTRs, suggesting that their start sites are transcriptional start sites. At least three motifs are suspected to be involved in regulating transcription. These include CCA1 (aaaAATCT), which is known to be necessary for circadian regulation (Alabadia et al., 2001; Wang et al., 1997), and G-box (CACGTG) and the a G-box like element Hex (tgacgtgg), which are both suspected to be involved in circadian regulation (Borello et al., 1993; Schindler et al., 1992). In each of these, point mutations in uppercase positions are known to disrupt normal gene regulation. We will analyze each of these motifs

and look for novel motifs using the group-specific approaches outlined above. It is important to note that these genes represent 68 of 454 clock genes from Harmer. The advantage of this is that it focuses on genes that are more closely co-regulated, but it also may allow a certain amount of noise to creep in due to the fact that these previously identified motifs do not necessarily act only on phase 0 clock genes.

Table 3 shows the top 25 ratios that compare the upstream sequences of clock genes to the upstream sequences of the rest of the genome. The top two entries are clearly related, but are not useful for our purposes, since they represent a 10-mer repeat that occurs 5 times in one gene. Another interesting motif that is readily apparent from these ratios is CCACGTGTC. This motif occurs in 10 distinct clock genes, appearing as the 11-mer CGCCACGTGC in four of those genes. Interestingly, all motifs that are within one mismatch of this motif have normalized ratios of at least two, with a combined normalized ratio of 6.6, and 24 of the 68 (35%) phase-0 clock genes include at least one of these motifs in their upstream regions. The combined distribution of these motifs shows a clear localization that appears to be the same in both clock genes and the entire genome, with hundreds of genomic occurrences in the localized area (Figure 18). This observation, combined with the high ratio of occurrence, suggests that this motif is functional not only in phase-0 clock genes, but may also be involved in regulating a large pathway, or set of tissue-specific genes, that includes a number of genes that are also regulated by the circadian clock.

We can analyze known motifs as well as identify new ones. While CCA1, Hex, and G-box have all been shown to be involved in regulating these genes, they are not necessarily

specific to this set. Table 4 shows the relative frequencies of each of these motifs, with mismatches allowed only in lowercase positions.

CCA1 is half as likely to occur in front of phase-0 clock genes as any other gene, though its ratio increases as the number of mismatches increase. Its reverse compliment is actually more likely to occur in this set of genes than CCA1 is. Combined, they occur in 62 of the 68 genes with up to two mismatches in each, though the low ratios of occurrence suggest this is not much better than chance. This motif has a random distribution (KS p-value = 0.99 with two mismatches and 0.68 with one).

The Hex motif has a much higher ratio of occurrence. While the exact motif only occurs in four distinct clock genes, with up to two mismatches this motif occurs in 59 of the genes. The fact that this motif is 4.5 times more likely to occur in front of these genes than any other indicates this is not likely to be due to chance. The reverse compliment, while generally much more prevalent in the genome, also shows a high ratio of occurrence. Both the Hex motif and its reverse compliment have random distributions with any number of mismatches (KS p-value = 0.93 for Hex and 0.043 for reverse compliment, both with one mismatch). This is especially remarkable considering CCACGTGTC (discussed above) is within two mismatches of the reverse compliment and did show a significantly non-random distribution.

Finally, the G-Box motif was found in 37 of the 68 phase-0 clock genes, with a normalized ratio of 5.956. It occurred exactly once in front of each of those genes. This motif did show a localized distribution, which was very similar to its distribution in the genome as a whole

(Figure 19). As noted above, there is experimental support suggesting this motif is necessary for clock gene regulation (Borello et al., 1993). But while this motif is six times more likely to occur in front of these clock genes than other genes, the distribution suggests that the G-Box motif functions in other roles as well. These genes may be involved in other pathways that the G-Box controls, or G-Box may only function in these genes in conjunction with other phase-0 clock regulatory motifs.

Discussion

We have designed and implemented a database that catalogs the position of every 9-mer that's in close proximity to genes in the *Arabidopsis thaliana* genome. This has allowed us to identify motifs that are non-randomly distributed throughout the genome with respect to either the regions they're found in or their positions within those regions. Such motifs are likely to serve some biological function related to gene regulation, though only molecular genetic experiments that manipulate those motifs can definitively prove this.

Overview and strengths of the database

We have looked at a number of different approaches to identifying non-randomly distributed motifs using ratios and data mining statistical approaches. While none of the examples were completely thorough, they indicate the types of approaches that can be used with our database, as well as a taste of the knowledge that is available when questions are asked in the right way. Most importantly, they show the value in a database that uses position information for identifying potential functional cis-acting elements.

The database and website is particularly effective at the genomic level. For example, region-to-region ratios revealed a number of interesting motifs that warrant further study. In particular, the 10mer aGGCCCatta was found to occur far more often in upstream regions than transcriptional units, and appears to be localized to positions in the range –200 to –60 with respect to the gene start. A cursory look at gene functions indicate this motif may be involved in regulating a large set of genes in basic gene expression. Of particular interest is the fact that 42% of identified ribosomal proteins include this motif

in their promoter regions. Analyzing intron-to-exon ratios immediately revealed motifs that matched the consensus motifs of the 5' and 3' splice sites, as well as the branchpoint consensus motif.

Data mining techniques using standard deviation and χ^2 scatterplots revealed a number of outlier motifs. Analyzing a few randomly chosen outliers revealed some interesting results, including both experimentally known motifs (eg, TATA-box, translation start site context, and GAGA), and novel motifs (e.g., ATTACCCCA, which appears in front of a number of retroviral-related genes) that warrant further analysis. The large number of outliers observable in these scatterplots suggest that many more interesting motifs can quickly be discovered and analyzed, providing potential direction for wet-lab experiments. Furthermore, by analyzing some typical motifs in the χ^2 scatterplot and comparing them to the distribution of base composition, we have suggested a general approach to learning more about the genomic sequence information currently available. The typical motifs appear to be AT-rich, but do no appear very often within 200 bps of the gene starts. While this may be due in part to the majority presence of translation start sites in the database, the base composition distribution, as well as the TATA-box distribution, suggests the more likely cause is the importance of sequence specificity in this range. Clearly more can be learned from analyzing these motifs. The Kolmogorov-Smirnov test also proved to be valuable both as a data mining tool and in assessing the randomness of various distributions.

We also looked at techniques for analyzing sets of co-regulated genes, using phase-0 clock genes as a test set. Group-to-other ratios failed to reveal known motifs, but did suggest a novel motif that is similar to the reverse compliment of the known regulator Hex but has a very different group and genomic upstream distribution. The similarity of this motif in the group's upstream regions to the genomic upstream regions highlights the importance of context in motif functionality and may suggest that the subset of phase-0 clock genes that contained this motif may be involved in other pathways as well. Known clock-regulator motifs were also analyzed for their specificity to the phase-0 clock genes. Hex and G-Box showed a high preference for these genes, while CCA1 did not. This emphasizes the importance of context for CCA1. None of these motifs had a non-random distribution either in this set of genes or in the genome.

From our brief overview of the various techniques made available by the database and website, it is clear that much information can be gained by identifying non-random motifs based on their position and region localization. Our database is especially good at analyzing motifs at the genomic level, as the high motif counts make for valuable statistics and data mining. Unfortunately, 9-mers do not appear to lend themselves as readily to analysis of small sets of genes. The number of phase-0 clock genes was nearly too small to do meaningful statistics. While including mismatches helped define meaningful distributions, it is not helpful in the initial data mining. The genomic strengths of the database do, however, make a valuable asset in analyzing known or suspected motifs in the context of how their presence and distribution in a group differs from in the genome.

We have shown that position and region information can be used by our database and web front-end to identify known cis-acting regulatory motifs. We also have identified novel motifs that may serve in a cis-acting regulatory role. Though not ready for general public use, we have demonstrated the potential for our database and web front-end to compliment existing programs, none of which incorporate position and region information on a genomic level. While this in no way replaces these programs, it does provide a valuable piece that has been missing. For example, many non-randomly distributed motifs and related genes can be identified with our database and further analyzed with multiple sequence alignments or Gibbs free sampling programs to be further characterized. Once a good understanding of the functional locations and consensus sequences of these motifs has been determined, point-mutation experiments can be conducted to test for biological functionality. The scope of the database's functionality will grow considerably as we add more organisms, thereby allowing comparative genomics approaches that can identify motifs whose distributions have been conserved among organisms.

Future directions

Our various examples of how one might use this database bring to light some suggestions for future development. The scope of this project was such that most of the time and energy spent was on creating the database itself as the backbone of future projects. Once completed, we were able to begin making use of its power through the creation of the website. We must acknowledge that this is still in the early stages and a number of limitations do still exist. Some of these limitations are due to the fact that this is not

intended to be a comprehensive tool; others highlight approaches we should take in the future.

The first obvious development that would be helpful is the ability to statistically cluster motifs into consensus sequences. While the graphing capabilities allowed for plotting consensus motifs and determining KS p-values for their distributions, the initial data mining strategies did not. This was especially clear when analyzing the ratios. While the aGGCCCatta motif was observable to the human eye, this appeared to be more the exception than the rule. Implementing a clustering algorithm that would cluster motifs based both on their sequence and their statistics or ratios would be very informative. A number of clustering algorithms currently exist and could be used to cluster the top results of any table we generate. In addition to finding consensus sequences, this would also help the problem of not having enough instances of any given 9-mer in a small group of genes.

One inherent limitation is the fixed size of the 9-mer. Again, while charts can be made of the distributions of motifs that are shorter than 9, all other queries return all the 9-mers that make up the shorter motif. While it would be a small step to consolidate results for queries of individual motifs, the data mining approaches outlined above do not lend themselves to variable length motifs. This is an inherent problem in all word-based approaches to motif finding. This emphasizes the fact that this, like all motif-finding tools to date, is not a comprehensive tool and so is best used in conjunction with other tools whose strengths and limitations are different and complimentary.

Finally, the methods outlined above only hint at the issues of motif context and the relationship of multiple transcription factors working together. While we can infer that context is important for some motifs by comparing group ratios and distributions to genomic ones (for instance, the G-Box), we have not dealt with the problem of identifying combinations of motifs that are unique to sets of genes. The reason for this is the size of the search space. In the case of 9-mers, there are $262{,}000^2 = 6.9 \times 10^{10}$ possible pairs. While 2.6×10^5 is searchable by brute force methods, 6.9×10^{10} is not. The complexity of this is compounded by the issues of consensus sequences, the possible order in which the motifs occur, and the question of whether orientation matters or not, and the problem only gets more complex as larger sets of motifs are considered. This is the reason no programs have thus far been able to successfully search for groups of motifs on a genomic scale.

While we cannot approach this problem in a brute-force manner, we do have a database that allows for rapid querying of all 9-mers. The next major goal for this project is to find a method that will build on the resources of the database and allow us to solve this problem. One approach we will devote a significant amount of attention to is the use of a Genetic Algorithm (GA). The general approach of a GA is modeled after population genetics and evolution and is composed of a solution stated in the form of a string, or gene, and a population of these genes that are allowed to evolve. Each generation involves the selection of the "fittest" genes (i.e., genes that are closest to an optimal solution), crossover from the mating of two fit genes, and random mutation. The net

result is that the population navigates the search space towards optimal solutions. Hopefully this approach will allow us to search for sets of variable length motifs that occur in combinations, orders, and orientations unique to a given set of co-regulated genes. Initial prototypes have shown some potential, though there is a long way yet to go.

A more immediate goal for the project is add other genomes to the database. We are currently loading the *Drosophila melanogaster* genome and will soon be loading the *S. cerevisiae*, *C. elegans*, and human genomes as well. In addition to using the methods outlined in this paper on these organisms, we will be able to incorporate comparative genomics approaches as well, using the common assumption that important sequences are conserved to some degree between species.

Conclusion

The database and website described and demonstrated in this paper are a new tool that represents a novel approach to the problem of finding cis-acting regulatory motifs. We tested this in *Arabidopsis thaliana* and have shown it to work well on the genomic level and to show promise at the level of small sets of co-regulated genes. While we have primarily focused on promoter regions, a look at the intron-to-exon ratios suggests that these approaches hold promise in analyzing introns as well. While some major limitations exist, most serve as challenges for further progress. In the future, we hope this will become a publicly available tool that helps the molecular genetics community move towards a better understanding of the cis-acting elements involved in the regulation of gene expression.

Works Cited

Alabadi, D., Oyama, T., Yanovsky, M.J., Harmon, F.G., Mas, P., and Kay, S. A., (2001) "Reciprical Regulation Between TOC1 and LHY/CCA1 Within the Arabidopsis Circadian Clock" *Science* **293**:880-883.

Arabidopsis Genome Initiative. (Six related articles announcing completion). *Nature,* **408**:786-826.

Bailey, T.L. and Elkan, C. (1995) "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation maximization" *Machine Learning* **21**:51-83.

Borello, U., Ceccarelli, E., and Giuliano, G., "Constitutive, Light-Responsive and Circadian Clock-Responsive Factors Compete for Different I box Elements in Plant Light-Regulated Promoters", Plant Journal 4:611-619 (1993).

Brown, J.W.S., Smith, P., and C.G. Simpson (1996) "*Arabidopsis* consensus intron sequences. *Plant Mol. Biol.* **32**(3):531-535.

Devaux, F., Marc, P., Jacq, C. (2001) "Transcriptomes, transcription activators and microarrays" *FEBS Lett* **498**(2-3):140-144.

Harmer, S.L., Hogenesch, J.B., Straume, M., Chang, H.-S., Han, B., Zhu, T., Wang, X., Kreps, J.A., and Kay, S.A., (2000) "Orchestrated Transcription of Key Pathways in Arabidopsis by the Circadian Clock" *Science* **290**:2110-2113.

Hertz, G.Z. and Stormo, G.D. (1999) "Identifying DNA and Protein Patterns with Statistically Significant Alignments of Multiple Sequences" *Bioinformatics,* **15**: 563-577.

Huala, E; Dickerman, A; Garcia-Hernandez, M; Weems, D; Reiser, L; LaFond, F; Hanley, D; Kiphart, D; Zhuang, J; Huang, W; Mueller, L; Bhattacharyya, D; Bhaya,

D; Sobral, B; Beavis, B, Somerville, C; and Rhee, SY (2001)

The Arabidopsis Information Resource (TAIR): A comprehensive database and web-based information retrieval, analysis, and visualization system for a model plant. *Nucleic Acids Res.,* **29**(1):102-5.

"International Consortium Completes Human Genome Project" (2003). Available at http://www.ornl.gov/TechResources/Human_Genome/project/50yr/press4_2003.htm (3 May 2003).

Joshi CP, Zhou H, Huang X, Chiang VL (1997) Context sequences of translation initiation codon in plants. Plant Mol. Biol. **35**(6):993-1001.

Li J., Liang V. C. T., Sedgwick T., Wong J., Shi Y.-B. (1998) "Unique organization and involvement of GAGA factors in transcriptional regulation of the *Xenopus* stromelysin-3 gene" *Nucleic Acids Res.* **26**:3018-3025.

Ohler, U. and Niemann, H. (2001) Identification and analysis of eukaryotic promoters: recent computational approaches. *Trends in Genetics,* **17**:56-60.

Planet, P.J., DeSalle, R., Siddall, M., Bael, T., Sarkar, I.N., Stanley, S.E. (2001) "Systematic analysis of DNA microarray data: ordering and interpreting patterns of gene expression" *Genome Res* 11(7):1149-1155.

Press, W. H., S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing, 2nd ed.* (New York, Cambridge University Press, 2002).

Riechmann, J.L., Transcriptional Regulation: a genomic overview (Sept. 30, 2002), The Arabidopsis Book, eds. C.R. Somerville and E.M. Meyerowitz, American Society of

Plant Biologists, Rockville, MD, doi/10.1199/tab.0085, http://www.aspb.org/publications/arabidopsis/

Schindler, U., Beckmann, H., and Cashmore, A.R., "TGA1 and G-box Binding Factors: Two Distinct Classes of Arabidopsis Leucine Zipper Proteins Compete for the G-box-like Element TGACGTGG", Plant Cell **4**:1309-1319 (1992).

Thijs, G, Lescot, M, Marchal, K, Rombauts, S, DeMoor, B, Rouze, P, Moreau, Y (2001) "A higher-order background model improves the detection of promoter regulatory elements by Gibbs sampling" *Bioinformatics,* **17**(12):1113-1122.

Wang, Z.-Y., Kenigsbuch, D., Sun, L., Harel, E., Ong, M.S., and Tobin, E.M., (1997) "A Myb-Related Transcription Factor is Involved in the Phytochrome Regulation of an Arabidopsis Lhcb Gene" *Plant Cell* **9**:491-507.

Tables

motif	upstream Count	TU Count	ratio	normalized ratio
nnnnnnnnn	2722	196	13.888	25.6575898
ggcccatta	617	72	8.5694	15.8320253
ggcccaata	510	65	7.8462	14.4957479
gcccattaa	544	70	7.7714	14.3576932
aggcccata	377	49	7.6939	14.2144179
cacgcgccc	23	3	7.6667	14.1641458
aggcccaat	510	70	7.2857	13.4603374
gcccaataa	505	71	7.1127	13.1406497
aggcccatt	625	90	6.9444	12.8298423
tcggcccac	59	9	6.5556	12.1113711
taaggccca	402	62	6.4839	11.978934
taatgggcc	578	92	6.2826	11.6071025
aagcgcgcg	12	2	6	11.0849837
aataggccc	151	26	5.8077	10.7296958
taggcccaa	378	67	5.6418	10.4231936
agcccaata	439	78	5.6282	10.3980937
taggcccat	421	75	5.6133	10.3706181
ataggccca	319	57	5.5965	10.3395023
aaggcccat	525	94	5.5851	10.3184689
aatgggcct	607	109	5.5688	10.2883564
cgcgtagtc	11	2	5.5	10.1612351
attgggcct	464	86	5.3953	9.96789233
agcccatta	460	86	5.3488	9.88196222
ggcccatat	297	56	5.3036	9.79833381
cggcccatt	212	40	5.3	9.79173561

Table 1) upstream:TU ratio of occurrence. Top 25 motifs with at least 10 upstream occurrences shown. Green highlights motifs what are within two mismatches of agGCCCatta. Yellow highlights motifs that are reverse-compliments of this 10-mer.

motif	intron Count	exon Count	ratio	normalized ratio
gtaagtata	366	46	7.957	15.5752877
gtaagtttt	1020	133	7.669	15.0127881
atatatata	5639	749	7.529	14.737815
tatatatat	6126	823	7.443	14.5710209
gtaattaac	208	28	7.429	14.5417987
actaattag	163	22	7.409	14.5036646
gttagttag	249	34	7.324	14.3361737
gtaagtctt	456	63	7.238	14.1689321
gtaagtact	249	35	7.114	13.9265687
taactaact	300	43	6.977	13.6573244
attttttt	4001	575	6.958	13.6211423
ttttttttt	20952	3069	6.827	13.3641525
tttttttta	3370	498	6.767	13.2468733
ttttttta	2458	364	6.753	13.2188392
ttaaatagg	121	18	6.722	13.1590849
actaactaa	208	31	6.71	13.1345278
ttttgtagg	762	114	6.684	13.0846751
gtaagttc	672	101	6.653	13.0244899
taattaaat	525	79	6.646	13.0090336
tgtaggttg	287	44	6.523	12.7685636
ttaattagg	250	39	6.41	12.5483963
taattaact	344	54	6.37	12.4703174
aatttttt	1963	309	6.353	12.4358262
tgtaggctg	114	18	6.333	12.3978155
gtaagtaat	390	62	6.29	12.3136199

Table 2) Top 25 Intron:Exon ratios with >10 occurrences in introns. Green highlights 5' splice site; yellow highlights branchpoint motif; orange highlights 3' splice site.

motif	num In Group	num Not In Group	Ratio	normalized ratio
cgttttccc	6	84	0.071	29.5175303
gttttcccg	6	93	0.065	26.6609951
cgccaaaac	6	129	0.047	19.2207174
cacgtgtcc	6	146	0.041	16.9826887
acgttttcc	6	148	0.041	16.7531929
accgcaaaa	5	128	0.039	16.1423994
catttagtc	5	134	0.037	15.4196054
ccacgtgtc	11	306	0.036	14.8552277
cgccacgtg	5	142	0.035	14.5508952
tccgccaaa	5	146	0.034	14.1522406
gccacgtgt	8	261	0.031	12.6665264
ccgtaaaat	5	171	0.029	12.0831996
atctctaag	5	176	0.028	11.7399268
ggagaaatg	5	183	0.027	11.2908586
atttagtgg	5	190	0.026	10.8748796
ttttcccgc	8	316	0.025	10.4619095
ccgccaaaa	9	383	0.023	9.710728
tgacgtggc	5	214	0.023	9.65526693
cttcctttc	6	274	0.022	9.04916988
cccgccaaa	6	288	0.021	8.60927968
aggtttggt	5	241	0.021	8.57355653
atttcgtta	6	293	0.02	8.46236365
aagatcaca	5	245	0.02	8.4335801
cagaaacat	5	250	0.02	8.26490849

Table 3) Ratios comparing upstream motifs of clock genes versus other genes. Top 25 results with at least 5 clock gene occurrences are shown. Orange and yellow selections are discussed in the text.

Element		Exact Match				One Mismatch				Two Mismatches			
name	sequence	clock genes	other genes	distinct clock genes	normalized ratio (C:O)	clock genes	other genes	distinct clock genes	normalized ratio (C:O)	clock genes	other genes	distinct clock genes	normalized ratio (C:O)
CCA1	aaaAATCT	6	4491	6	0.563	46	17163	34	1.130	113	31470	50	1.514
	AGATTttt	10	3428	9	1.229	55	16509	37	1.405	124	30559	55	1.711
Hex	tgacgtgg	5	214	4	9.851	13	914	11	6.000	159	14726	59	4.552
	ccacgtca	4	424	4	3.977	28	2117	23	5.576	216	22212	61	4.100
G-Box	CACGTG	37	2619	37	5.956	---	---		---	---	---		---

Table 4) Motif ratios of occurrence for known clock-regulating motifs. Light green rows are published consensus sequences. Orange rows are reverse compliments. Mismatches were only allowed in lowercase positions. Note that G-Box is a palindrome with all uppercase letters, so no reverse compliment or mismatches were evaluated.

FIgures

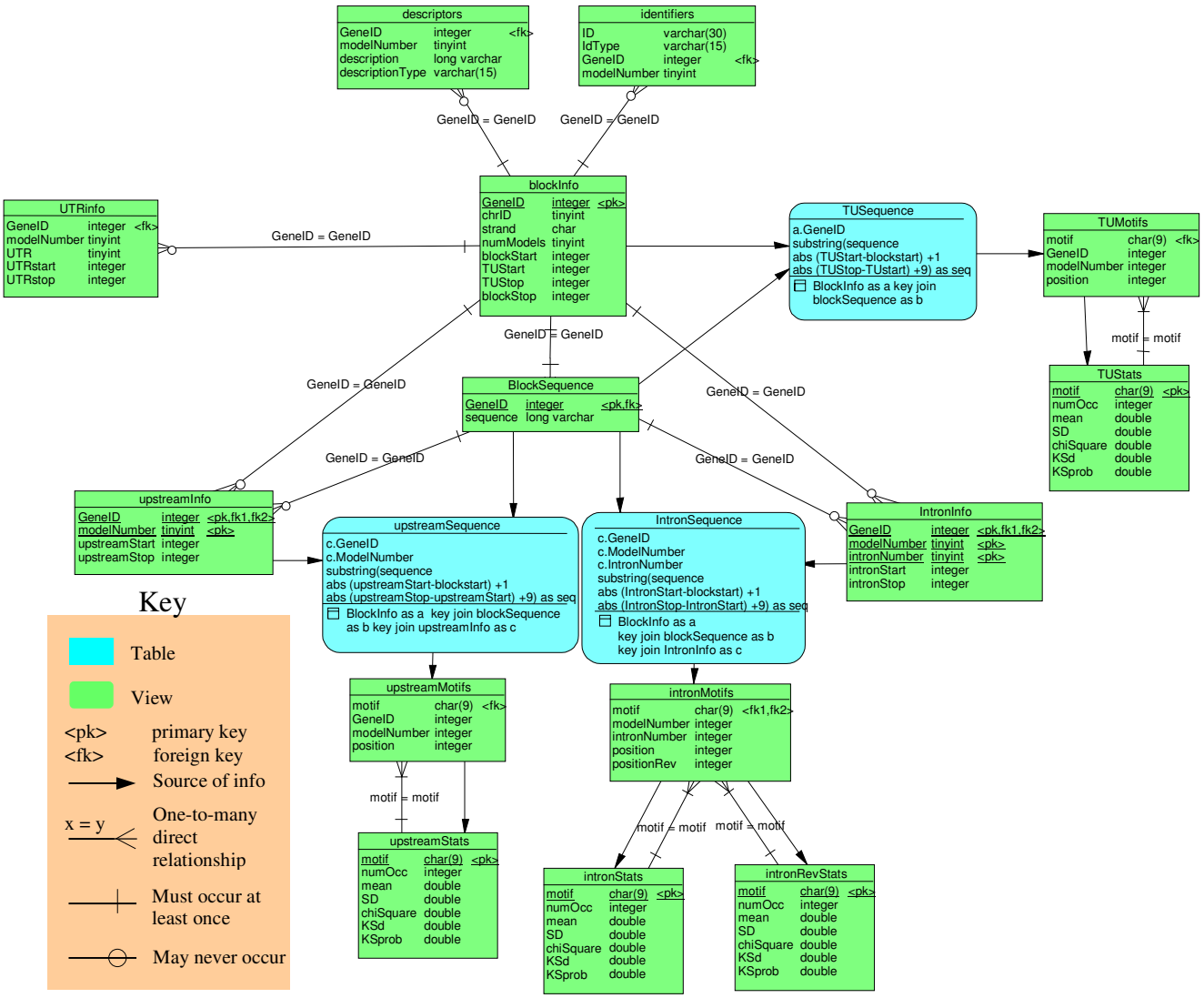


Figure 1) The database schema. This is a basic representation of the database schema, showing how all the tables relate to each other. Lines with equations represent foreign key relationships. Branching lines indicate many rows in that table can reference a single row in the linked table. Foreign keys are used to enforce integrity by making sure unrelated information is not present. They also indicate a direct relationship. Arrowed lines without equations show how the table the arrow points to was derived. That is, the sequence views are derived from the info and blockSequence tables, the motif tables are derived from the sequences, and the stat tables are derived from the motif tables. Only intron- and upstream-related tables are presented, as they are representative of exon- and up1500-related tables, respectively. The TUinfo is contained within the blockInfo table, since the definition of TU regions does not allow more than one entry per gene.

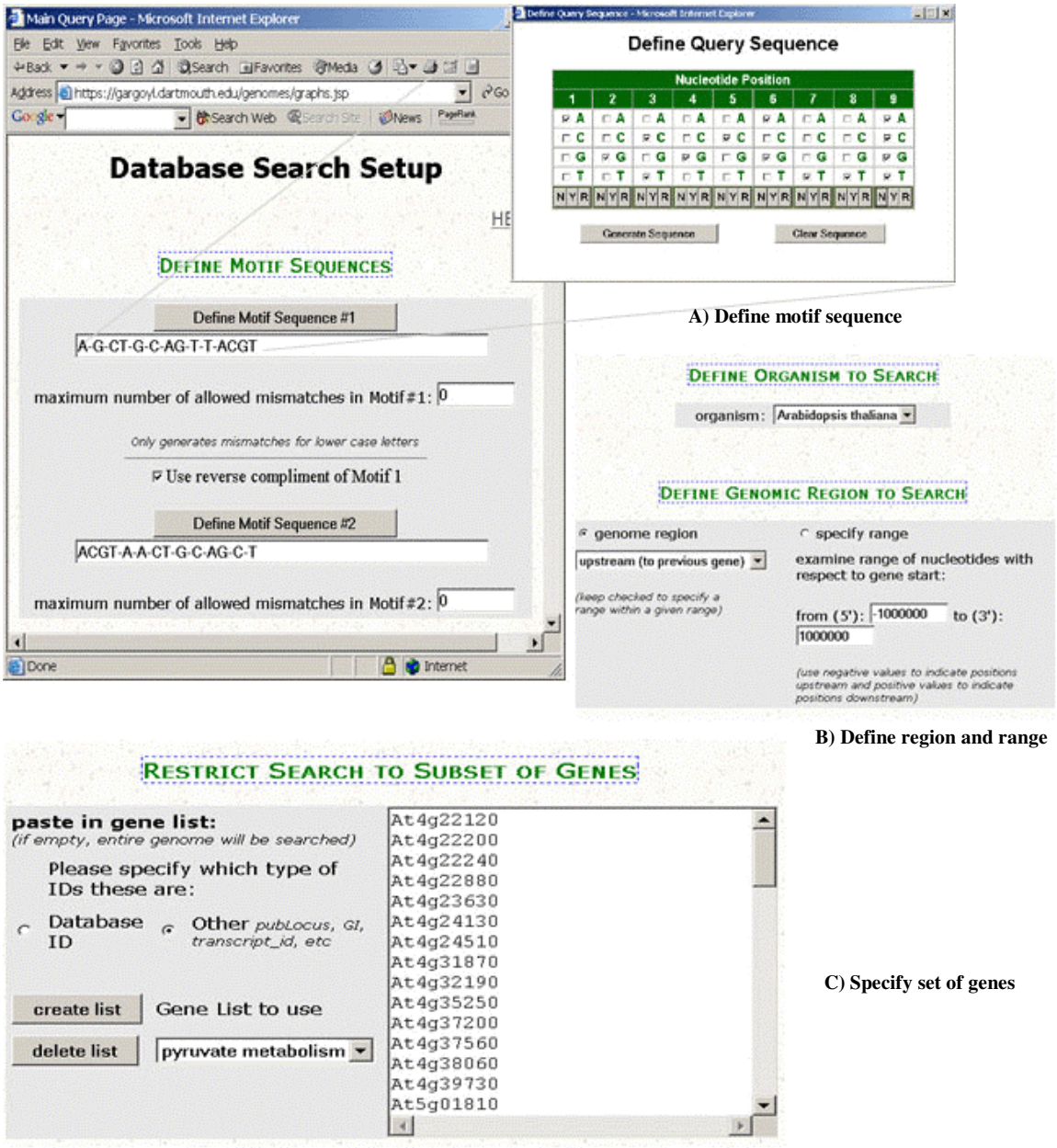


Figure 2) Defining basic parameters on the Main Query Page. (A) You can define the motifs you want to search, either by typing them in directly, or using the popup tool. Ambiguous consensus sequences are allowed, as are motifs < 9bps. Reverse compliment can be automatically generated, and minimum number of mismatches in lowercase letters can be specified. If left blank, all motifs will be searched. (B) Regions, ranges, or ranges within regions can be specified. (C) Sets of genes can be analyzed, saved, and retrieved. Gene names can be pub_locus ids, GI numbers, or Accession numbers. If left blank, all genes will be searched.

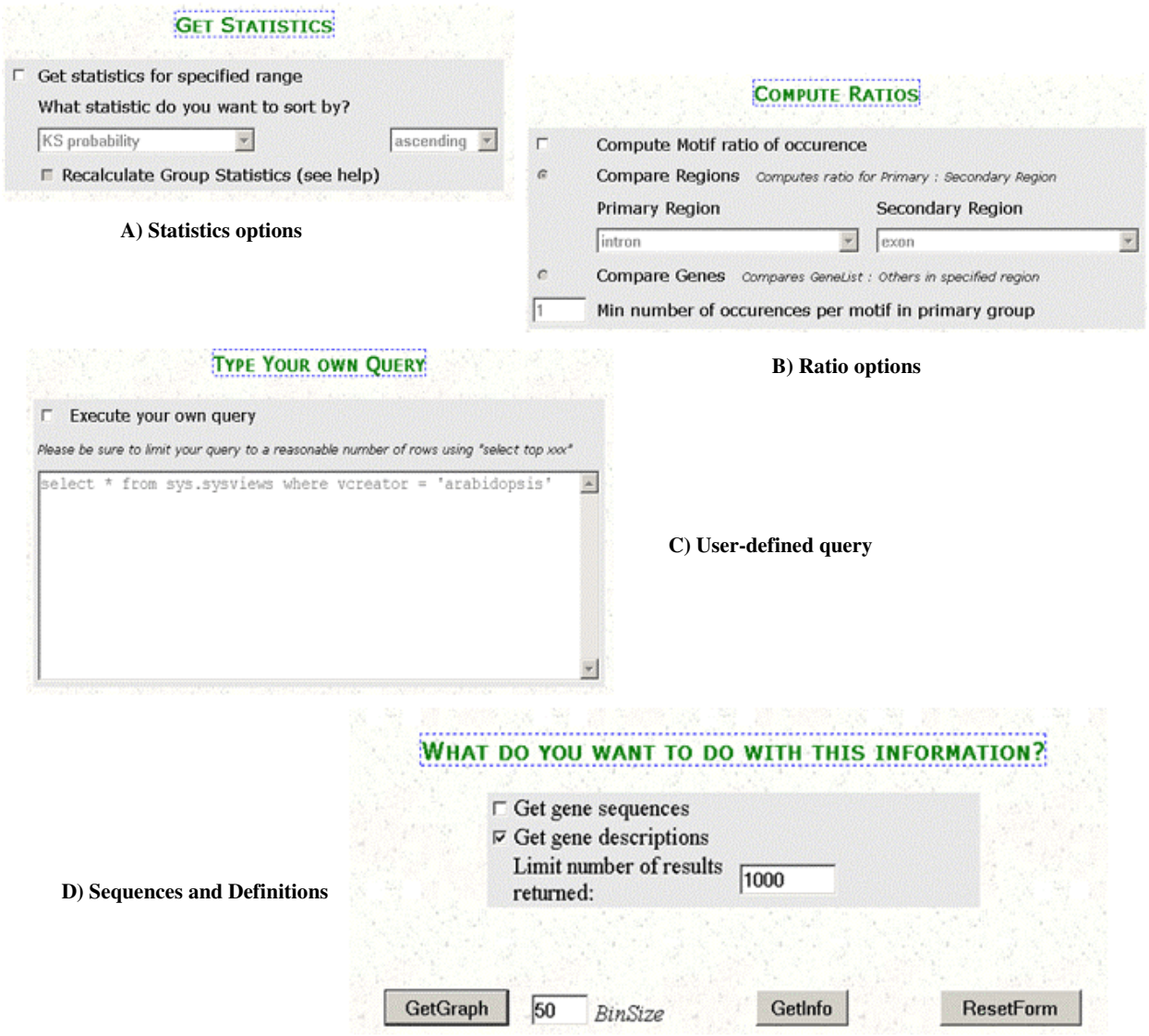
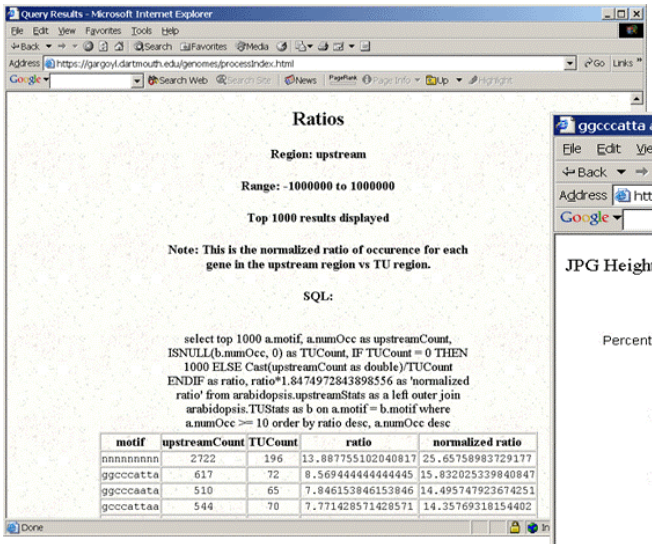
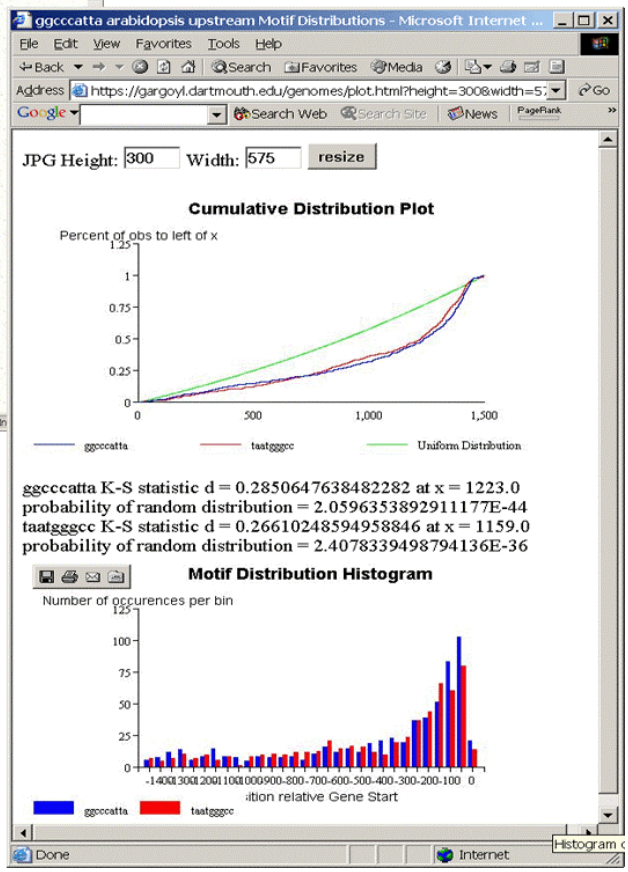


Figure 3) Specifying desired results on the Main Query Page. The last half of the Main Query Page lets the user specify what sort of output he or she would like to see. Most options have option-specific parameters as well. (A) If genes are specified in the genes list, the statistics for the subset are looked up, but they are only recalculated if the user requests for that to be done. (B) Either region-to-region or group-to-other ratios can be computed. Specify the minimum number of occurrences to weed out meaningless results. (C) If familiar with SQL, users can type their own queries. (D) Basic sequence information can also be retrieved. This section also allows you to limit the number of results returned. Users either press "GetInfo" to retrieve the results they have requested, or "GetGraph" to graph the motifs.



A) The Results Page



B) The Graphs Page

Figure 4) Results and Graphs pages. (A) The Query Results Page displays the results of all the queries requested from the Main Query Page. Each result includes a summary of the parameters, the SQL used to generate the results, and a table displaying the results. (B) The graphs page generates two charts that reflect the basic parameters specified from the Main Queries Page. Each chart will graph up to two motifs, each being a combination of all motifs included in one line (see figure 2A). The first chart is a cumulative distribution plot. The green line is the expected distribution. The second plot is a histogram with the bin size specified by the user on the Main Queries Page. The KS statistic and p-values are presented as well.

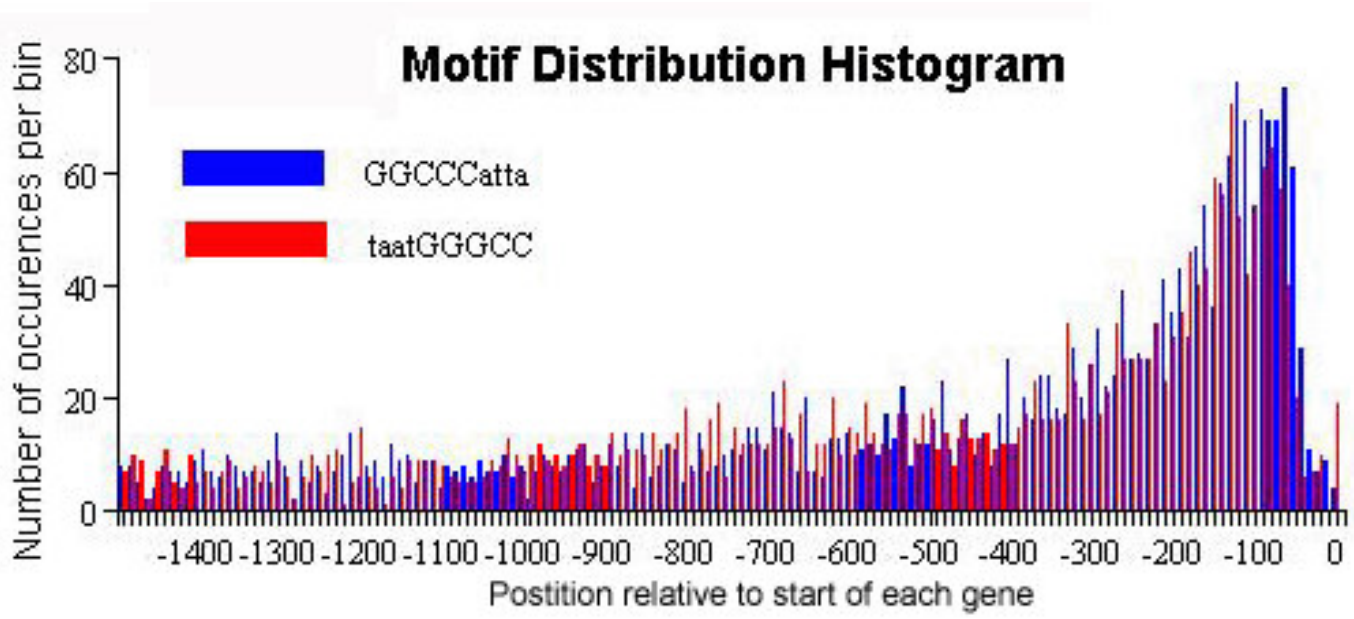


Figure 5) Distribution of GGCCCatta and reverse compliment with up to one mismatch in each. KS p-values are 1.8×10^{-125} and 9.4×10^{-7}, respectively.

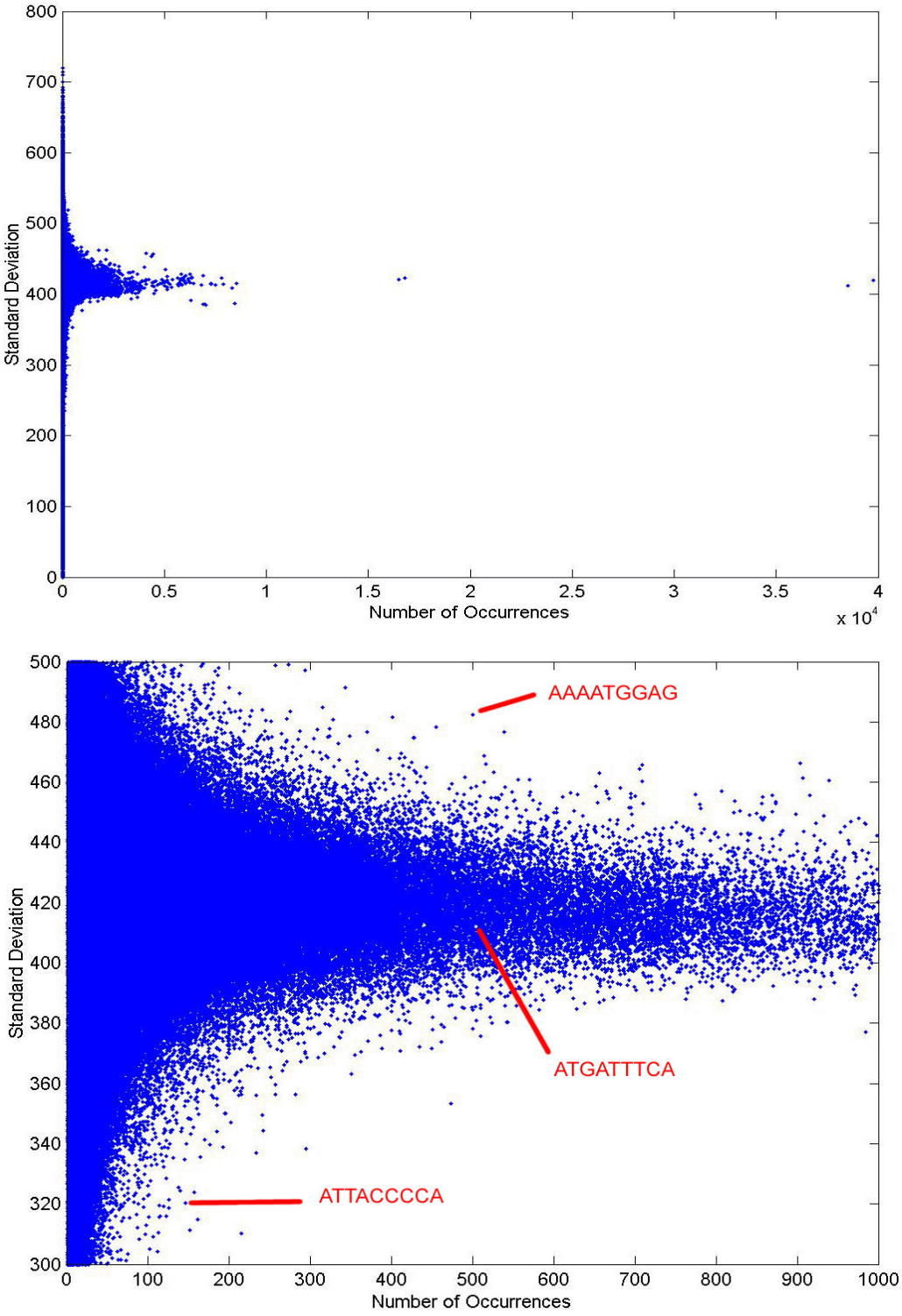


Figure 6) Scatterplot of Standard Deviation versus number of occurrence. TOP: all data points in the upstream regions. BOTTOM: Zoomed in to focus on motifs that occur no more than 1500 times. Motifs identified in red have been further analyzed.

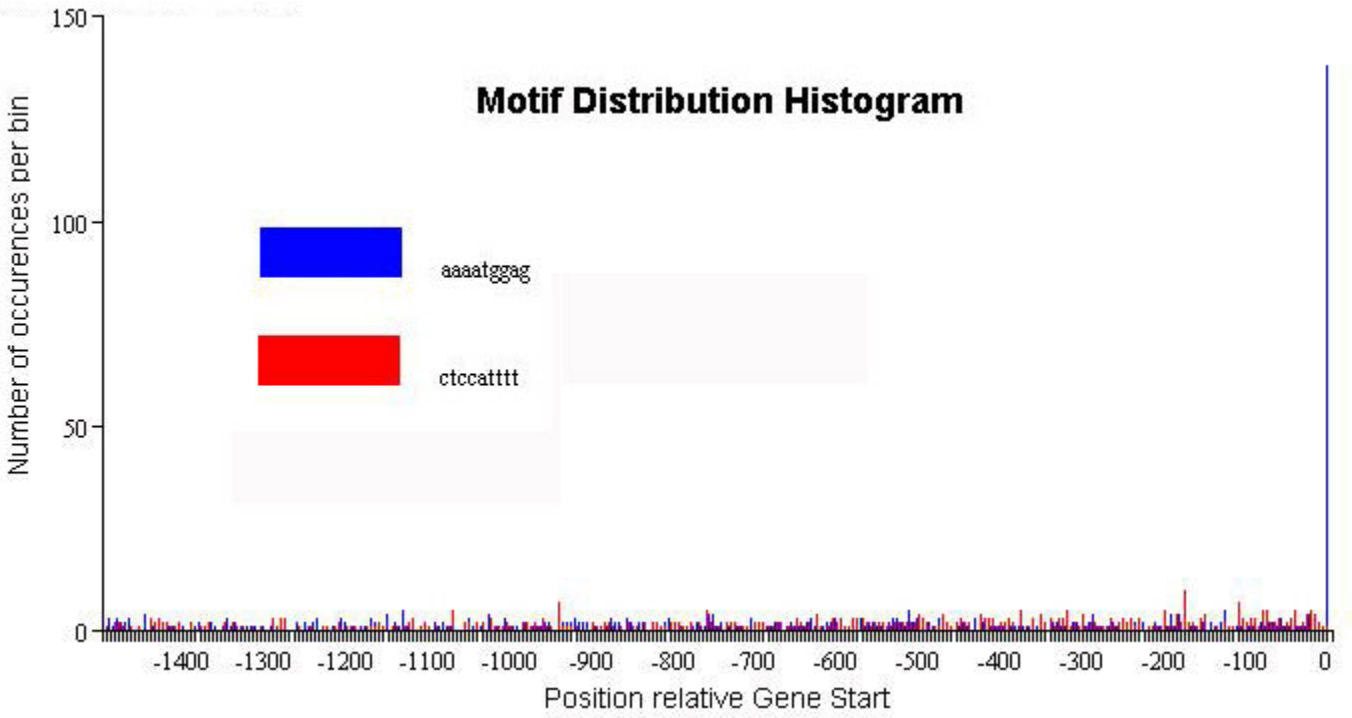


Figure 7) Distribution of AAAATGGAG shows clear localization to first 10 bases. Reverse compliment is randomly distributed. KS p-values are 2.2×10^{-33} and 0.02, respectively.

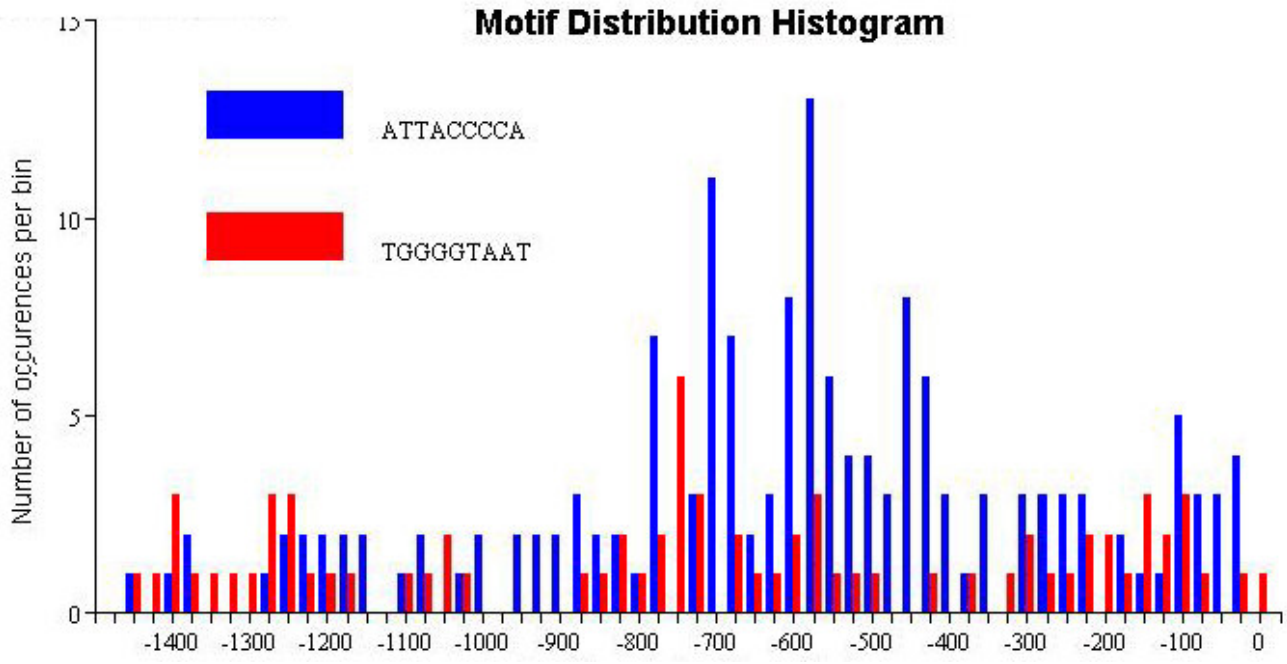


Figure 8) Distribution of ATTACCCCA shows localization to the range –800 to –400. Reverse compliment is randomly distributed. KS p-values are 5.7×10^{-4} and 0.13, respectively.

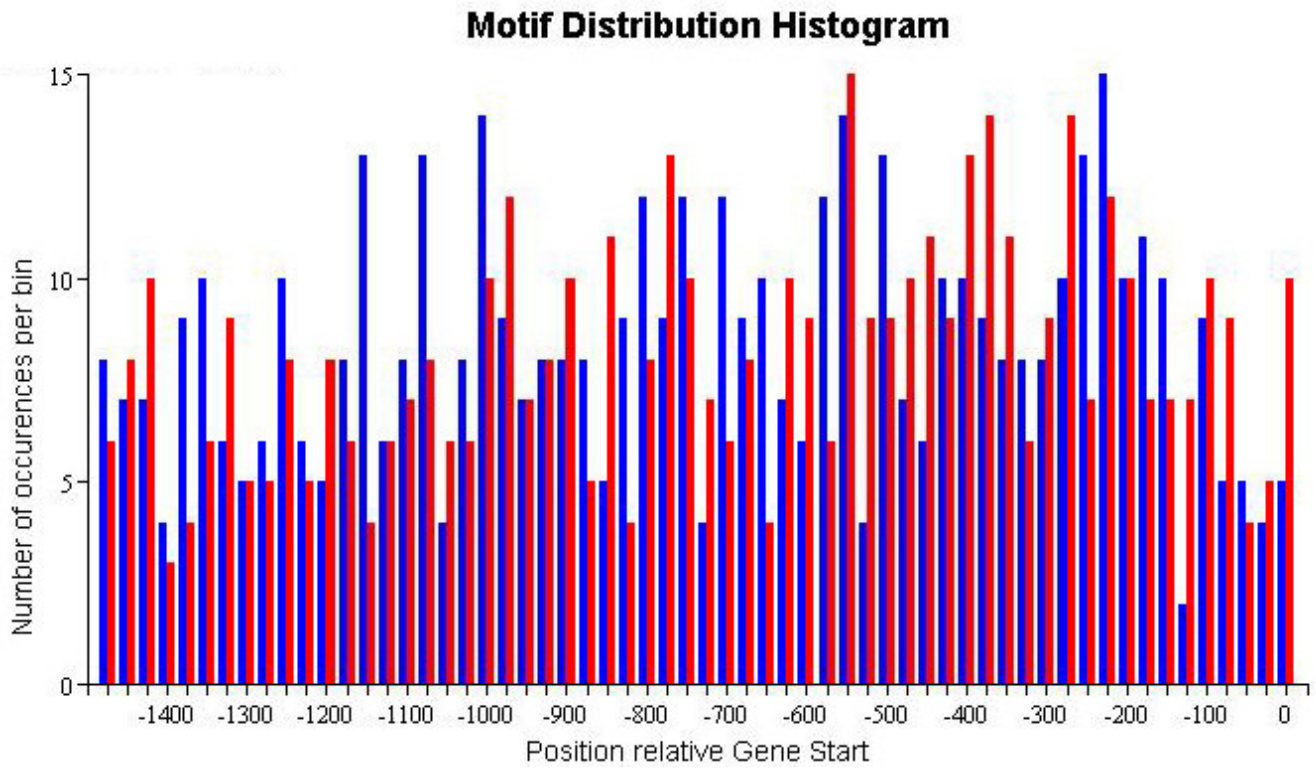


Figure 9) Distributions of ATGATTTCA and reverse compliment show no apparent preferential location. KS p-values are 2.7×10^{-4} and 0.04, respectively.

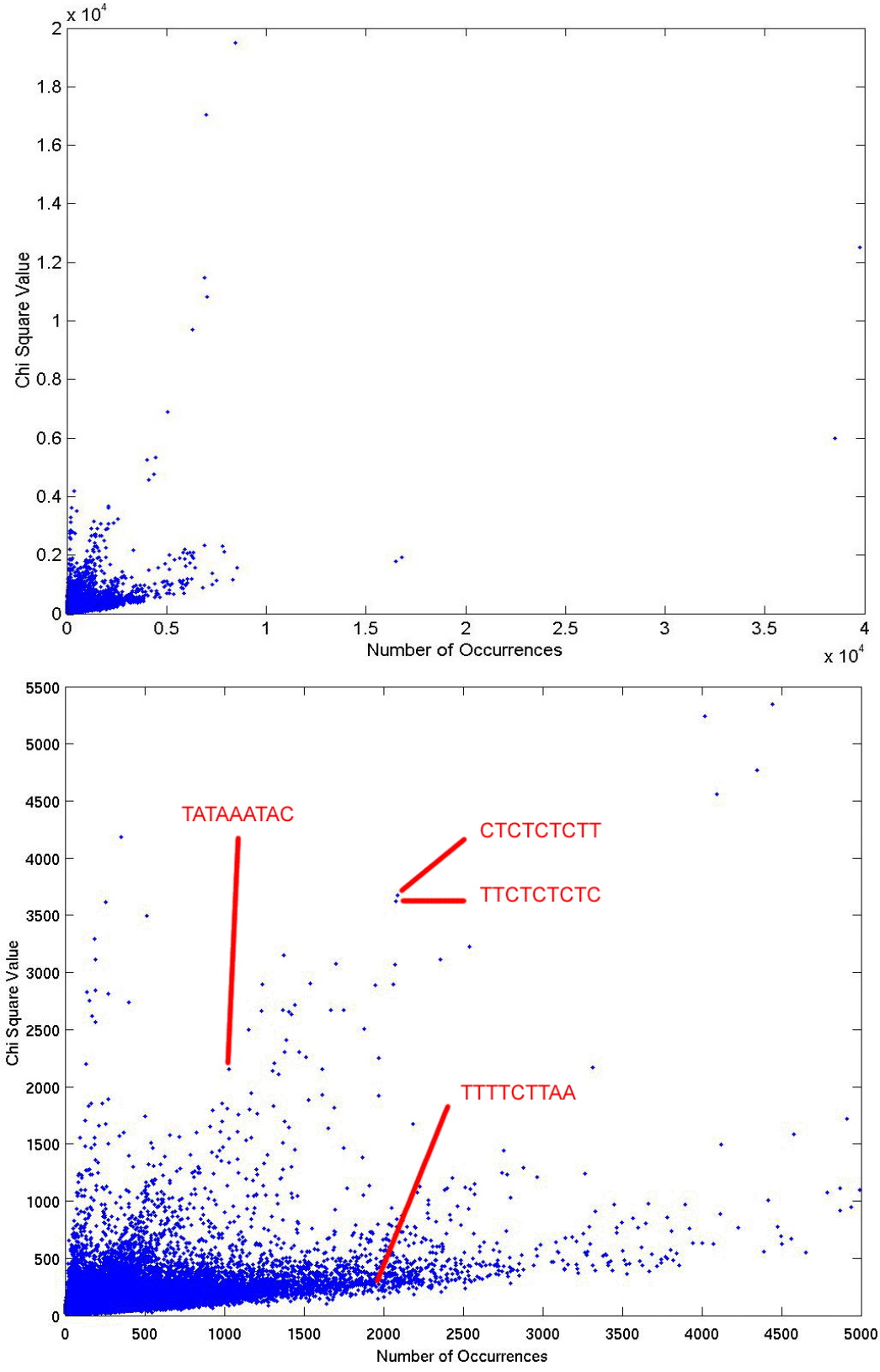


Figure 10) Scatterplot of χ^2 versus number of occurrence. TOP: all data points in the upstream regions. BOTTOM: Zoomed in to focus on motifs that occur no more than 5000 times. Motifs identified in red have been further analyzed.

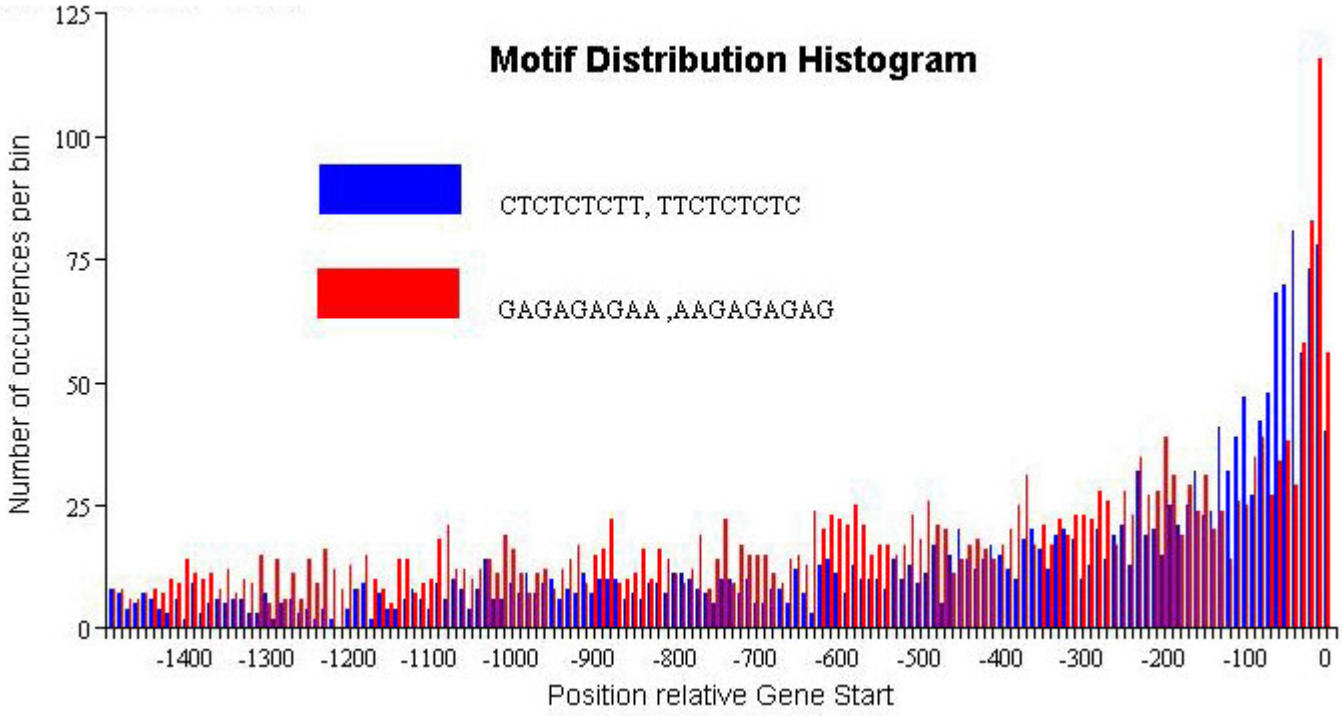


Figure 11) Combined distributions of CTCTCTCTT and TTCTCTCTC show a preferential location of < – 100 (KS p-value = 1.3×10^{-133}). Remarkably, the reverse-compliments show a preferential location of < – 40.

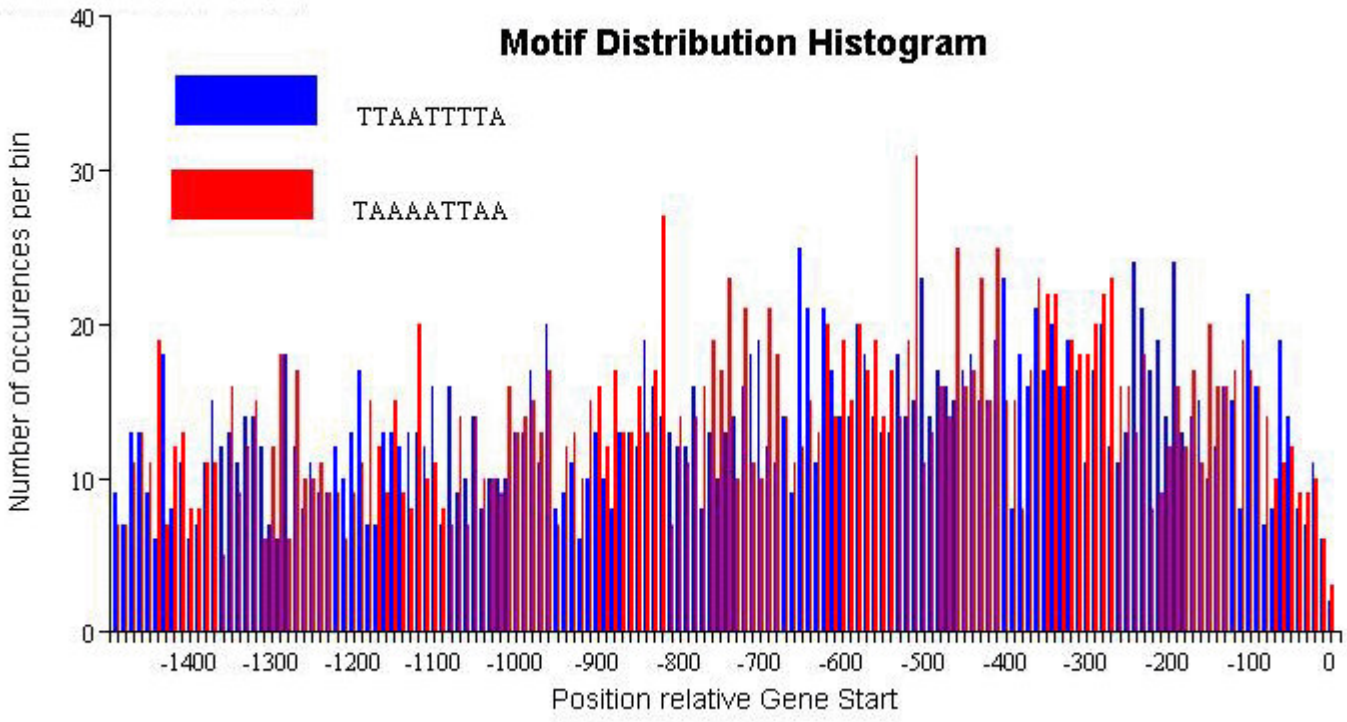


Figure 12) Distribution of TTAATTTTA and reverse-compliment, showing distribution of motifs with typical χ^2 value. KS p-values are 1.3×10^{-6} and 6.1×10^{-9}, respectively.

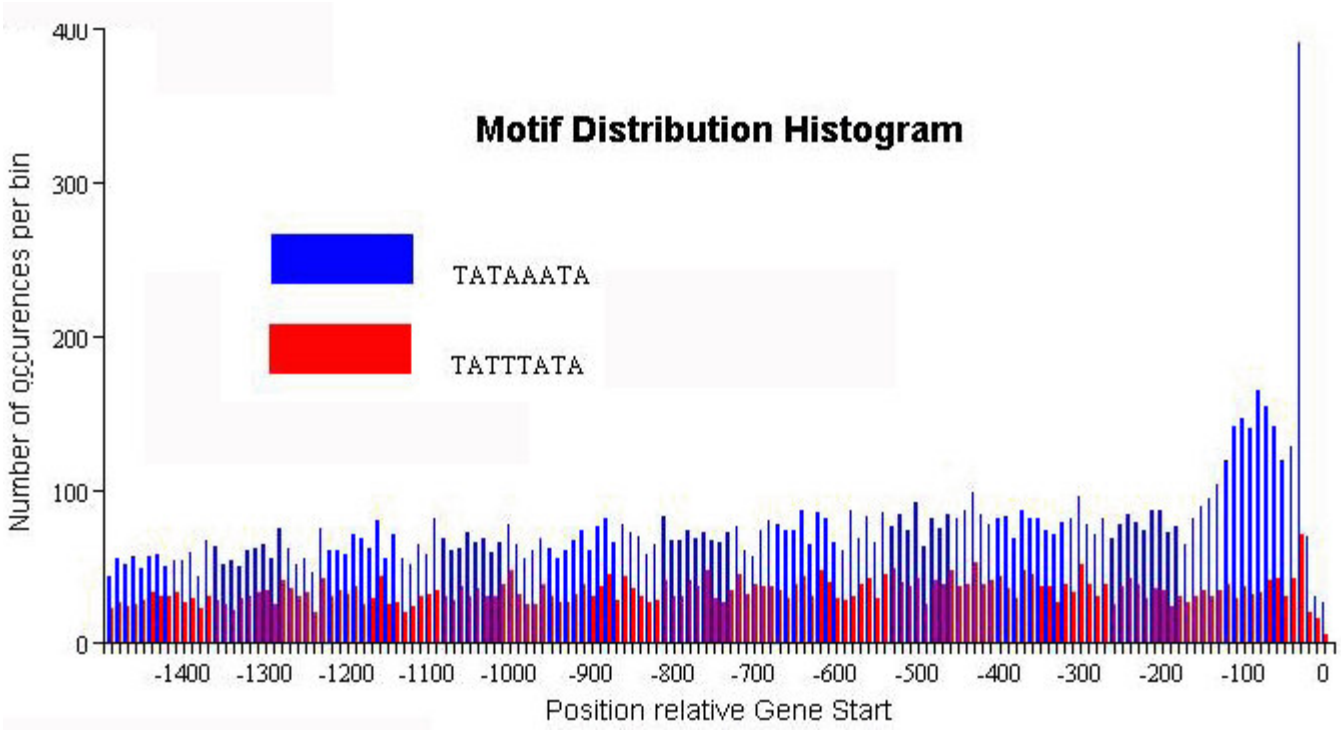


Figure 13) The distribution of the TATA-box and reverse complement. Illustrates precision location at –30 and effect of UTR regions in genes for which the gene start is the translation start site.

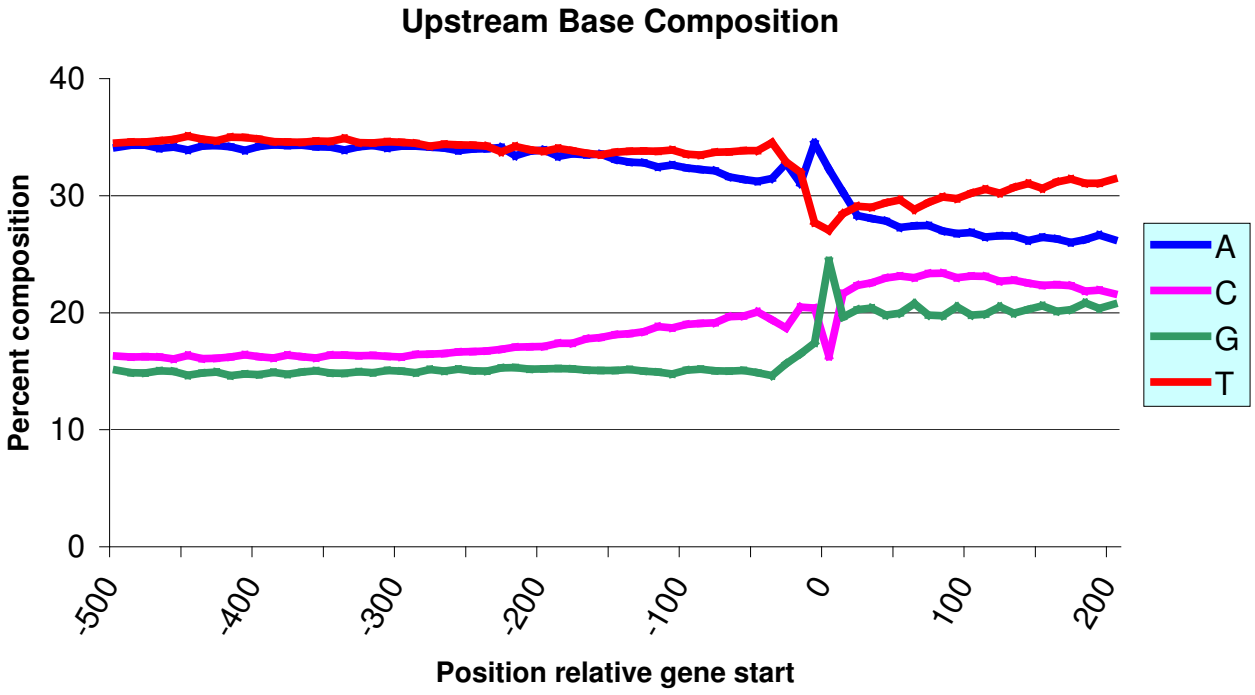


Figure 14) Base composition of upstream and TU regions in the range –500 to +200. Calculated from all genes using a bin size of 10.

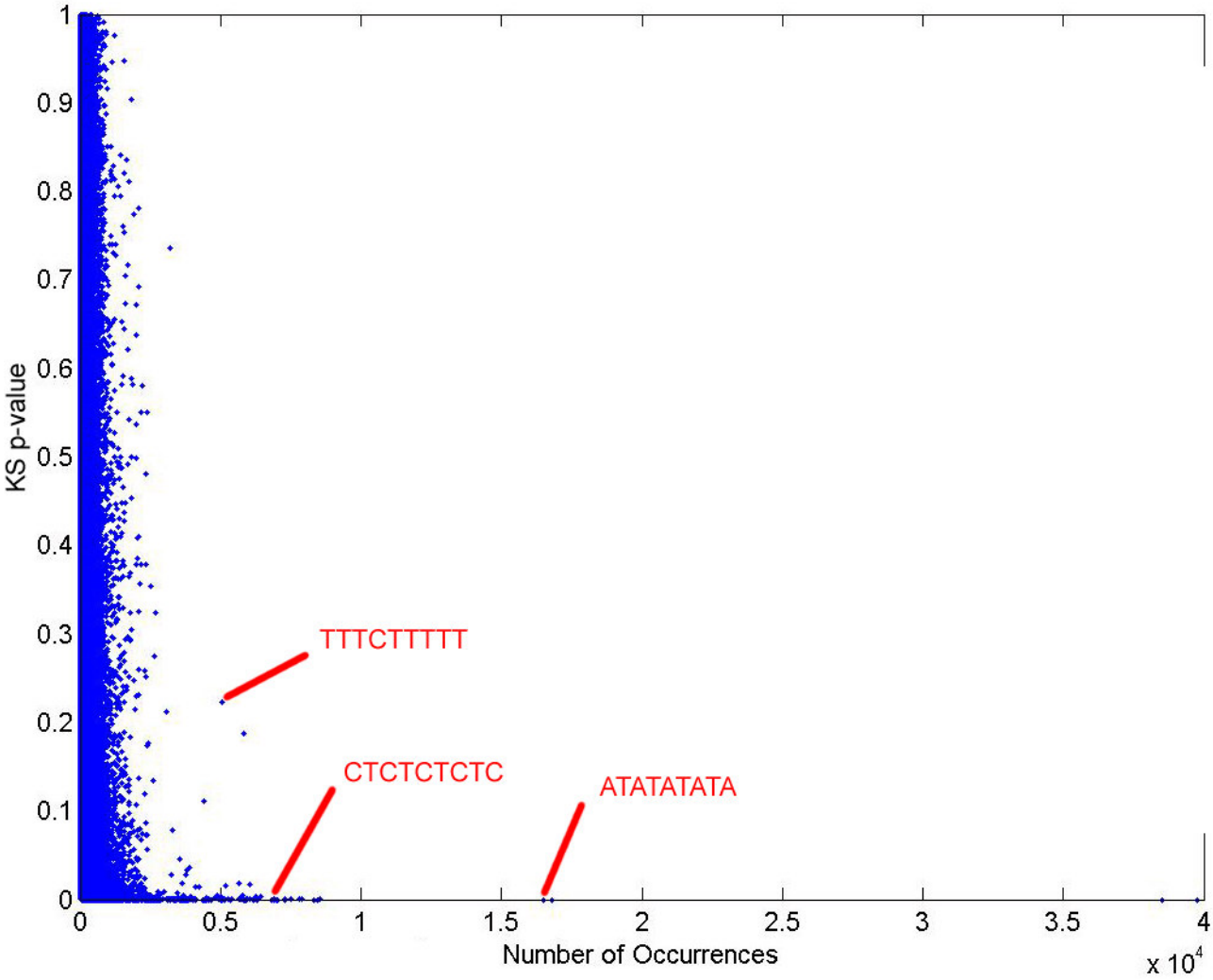


Figure 15) Scatterplot of KS p-value versus number of occurrences. Labeled motifs were analyzed further.

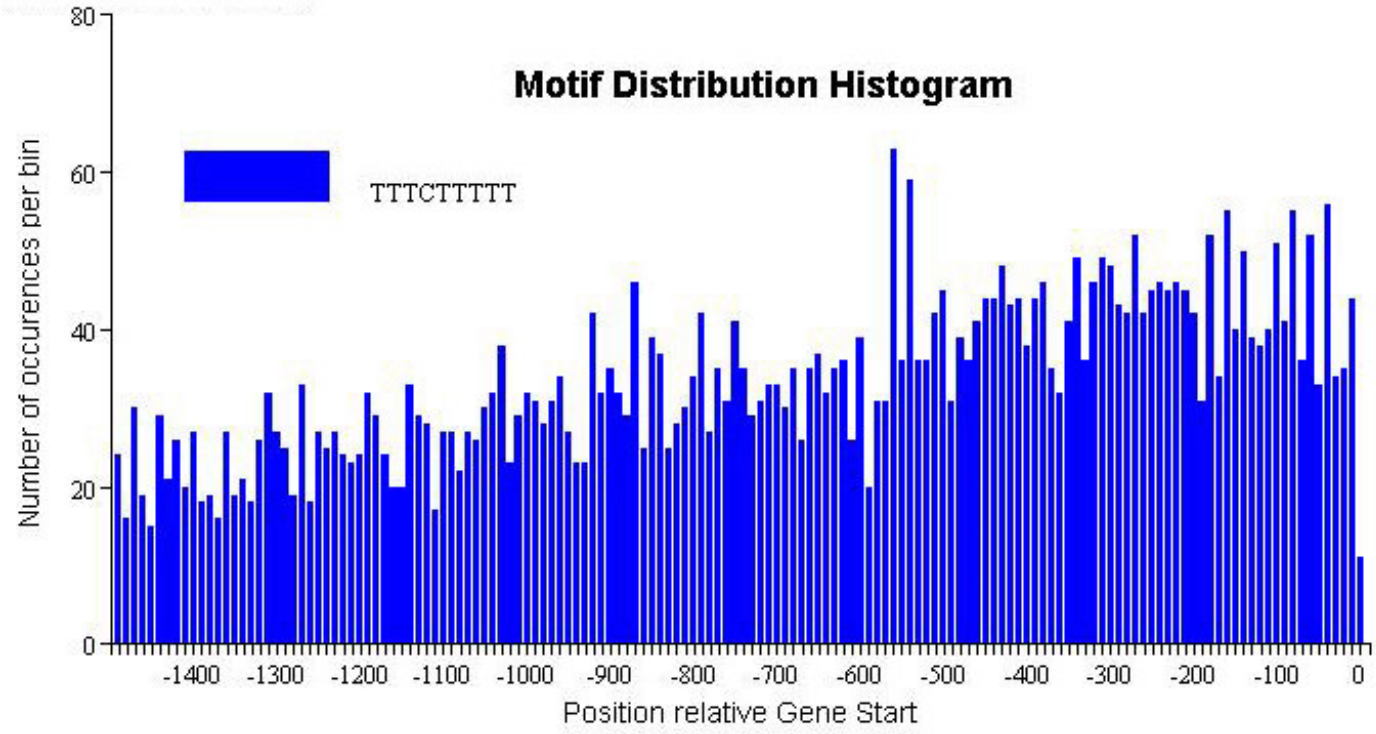


Figure 16) The distribution of TTTCTTTTT, KS p-value = 0.22, is very similar to the expected distribution.

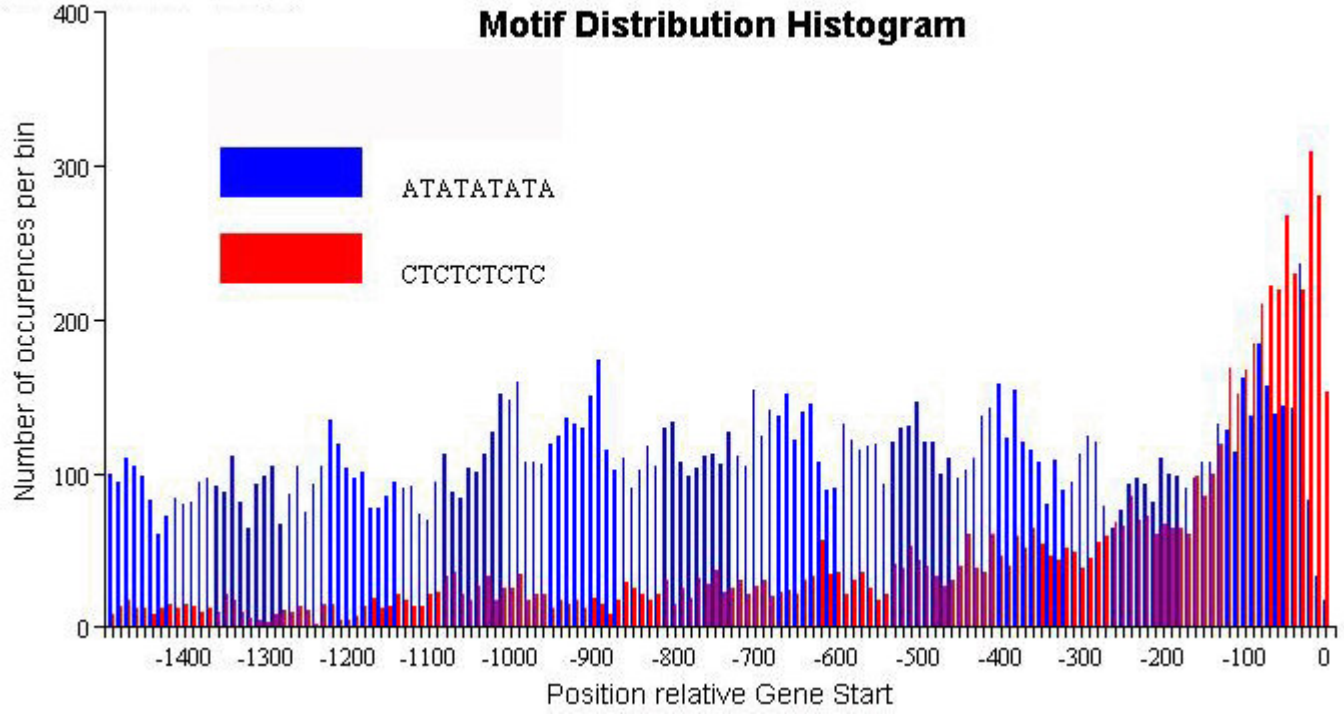


Figure 17) Distributions of an AT-rich motif versus a CT-rich motif. Both have small p-values (4.710^{-91} and 0.0) and high number of occurrences (16487 and 6978).

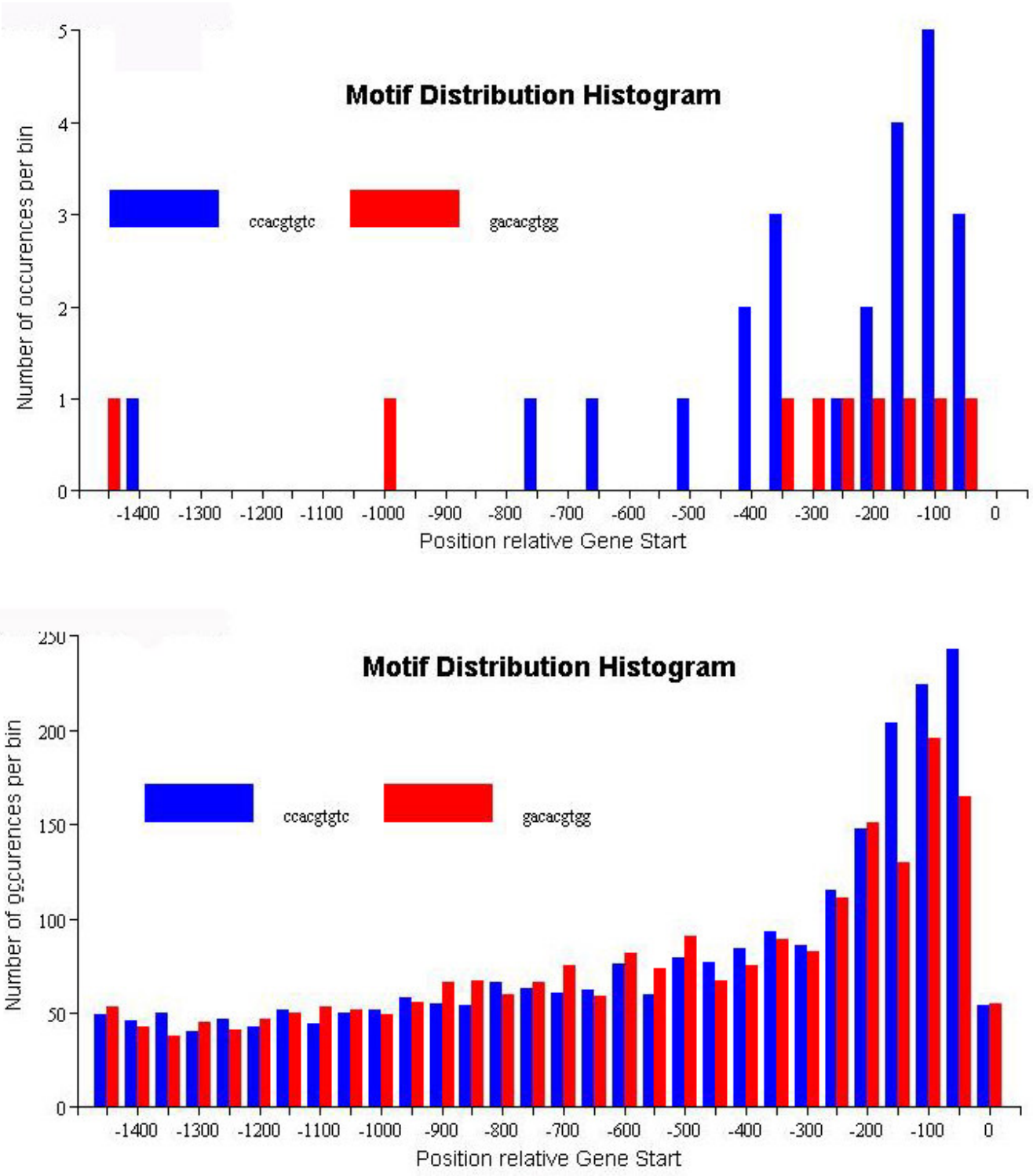


Figure 18) Motif distributions of ccacgtgtc and reverse compliment with one mismatch. TOP: distribution in phase-0 clock genes. . Primary motif KS p-value = 1.1×10^{-4}; reverse compliment KS p-value = 2.8×10^{-2}. BOTTOM: distribution in the entire genome. Primary motif KS p-value = 1.0×10^{-46}; reverse compliment KS p-value = 1.2×10^{-17}.

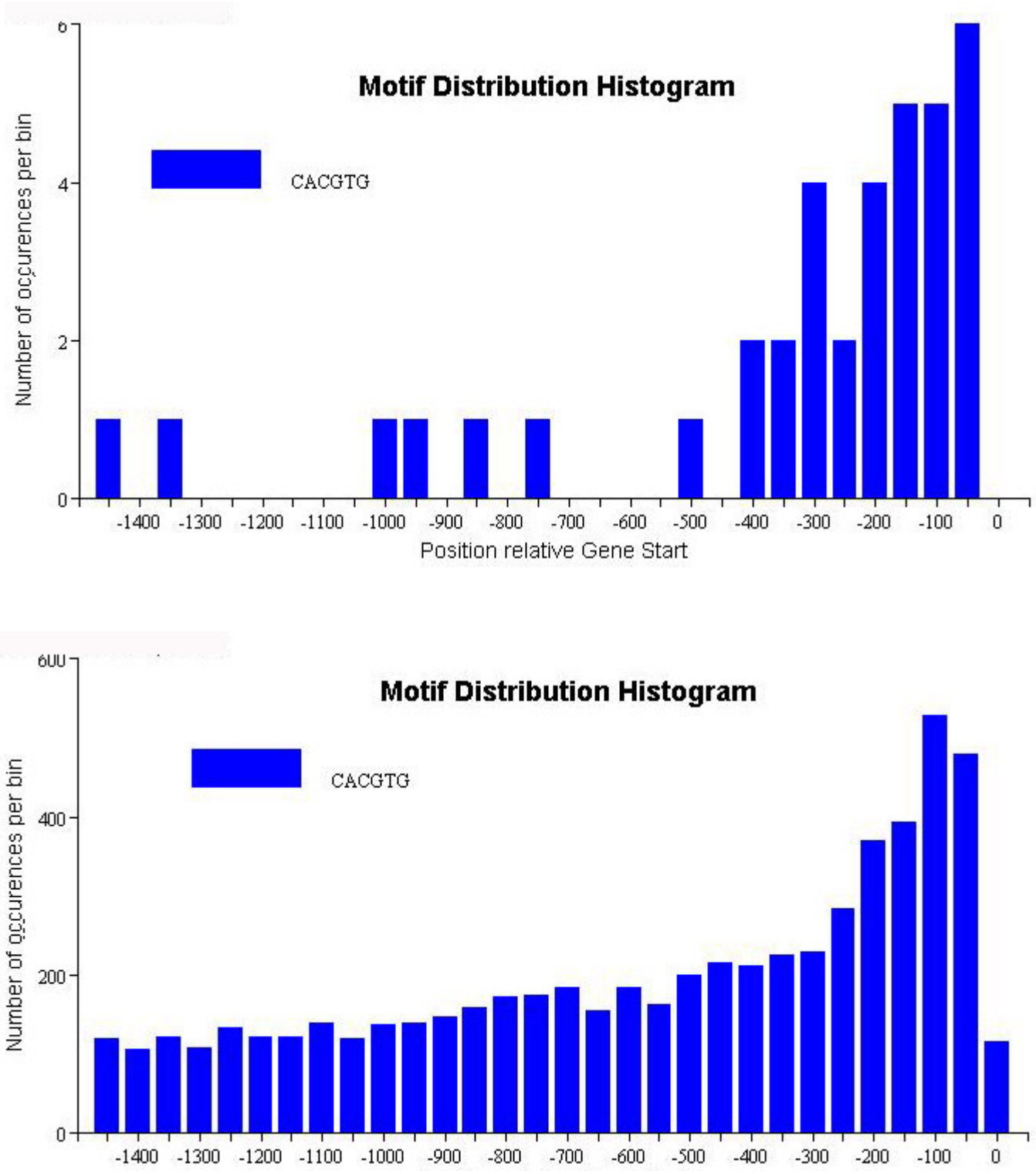


Figure 19) Upstream distributions of the G-Box motif in the phase-0 clock genes (top) and the entire genome (bottom). KS p-values are 2.7×10^{-6} and 1.5×10^{-57}, respectively.

Appendix A: GenBank file format, region definitions, and base table definitions

The contents of this appendix describes the standards we used in interpreting the GenBank flatfiles and the requirements for the results of parsing these files. It includes our interpretations of the GenBank tags, our definitions of terms used throughout the database, and a definition of the contents of the files that were loaded directly into the database. These files match the basic schema of all non-derived files (figure 1).

A. GBK format

Unfortunately, the "Standard" GenBank *.gbk flatfile format is hardly standard from genome to genome. It's standard only inasmuch as it allows different genomes to store whatever information they want in a way that is somewhat parse-able. Unfortunately, that means different genomes store different information, or the same information using different labels. This section standardizes the way we interpret the *.gbk files.

A.1 Structure

A.1.1 Heading

It appears to me that the .gbk "standard" of defining Accession, version, and GI numbers in part of a heading for every submitted sequence considers each chromosome to be a submitted sequence. Therefore, we cannot count on this information being available for every gene.

A.1.2 Features

This will include all the genes, as well as chromosome-wide info (base range, chromosome ID, organism, etc.). The major subentries of FEATURE are as follows:

A.1.2.1 Gene

This will include the range of the gene, which is either the transcription or translation start site, depending on the quality of the experimental data. As with all sequences *compliment*(x..y) means it's on the '-' strand.

A.1.2.2 mRNA

Of the form [compliment(]join([<]x1..y1, x2..y2, … , [>]xn..yn). This is defining all the exons of the gene. < implies the 5' end is incomplete. > implies the 3' end is incomplete. Theoretically, if complete, these exons will include the 5'UTR and 3'UTR

A.1.2.3 CDS

Of the form [compliment(]join(x1..y1, x2..y2, … , xn..yn). This is defining the coding sequence. Therefore, it should be identical to the corresponding mRNA definition, with the exception that the first and last exons will be shorter. If the mRNA definition includes > and <, the two definitions will likely be the same. The difference between mRNA and CDS defines the UTRs.

Note: GenBank encourages all genomes to define both CDS and mRNA for each gene, but there's no guarantee that they do.

A.2 Identifiers

Since all the standard identifiers of GI, Accession, and Version are only defined for sequence headings, and these appear to be used only once per chromosome, we're left with no standard for actual gene IDs. Thus, each organism seems to have come up with its own means of identifying the genes. Here are the ones I've seen in a couple organisms:

A.2.1 /gene="xxx"

This is related to the version number in some organisms, completely different in others.

This looks like the unique identifier that ties mRNA, CDS, and Gene records together. As such, you'll probably want to use this to keep track of the records, especially those that have multiple mRNA entries for one gene.

A.2.2 /note="xxx"

This is a wild card. In Arabidopsis, this appears to be where they store accession.version, but in one version of worms, it appeared to be where they store functional information in the form of whole sentences.

A.2.3 /db_exref="YY:xxx"

This is used to define equivalent IDs from other databases. The important thing is YY tells you what type of ID it is. Here's a couple examples I saw:

A.2.3.1 /db_exref="PID:xxx": Apparently a protein ID

A.2.3.2 /db_exref="GI:xxx": The GI number

A.2.4 /protein_id="xxx.y"

If present, this is supposed to be accession.version

A.2.5 /product="xxx"

The name of the protein this gene encodes. Unfortunately, for some organisms, this is an apparent identifier, while for others it's a multiple word semi-description. Probably depends on how well the proteins of the particular genome have been characterized.

There are probably many others as well, but the important thing to remember is an identifier is something that gives an apparently unique ID to the gene.

A.3 Descriptors

There are several other pieces of information that may be kept. This appears to be either functional information, or source information (ie, whether or not the gene is experimentally proved or not). Here's some I've seen:

A.3.1 /Function="xxx"

This will be a sentence describing what the protein does.

A.3.2 /Note="xxx"

Unfortunately, this is used both as an Identifier and as a Descriptor. We will need to change the definition of this based on the genome being parsed.

A.3.3 /experimental=[not_]experimental

Describes whether the protein definition was experimentally derived or predicted by a program.

A.3.4 /codon_start=n

This is presumably the position in the CDS that translation starts, though we have yet to find an instance where it n wasn't 1.

A.3.5 /product="xxx"

Unfortunately, sometimes this is something like "homologue of protein X", so we cannot classify it as an Identifier.

A.3.6 /translation="xxxxxxx"

This is just the translation from DNA into amino acids. We will ignore this.

There are others as well, but the point is, other information is everything that is not likely to be unique, and generally contains a highly variable length. The importance of the latter point comes into play with database efficiency issues.

Finally, notice that Identifiers and Descriptors are both specific to a Gene, mRNA, or CDS listing.

B. Our Definitions

Based on the "Standard" gbk file definitions, the purpose of this section if to clearly define certain items.

B.1 Models

The /gene="" identifier is what links CDS, mRNA, and Gene together. In some instances, there are multiple mRNA's with the same gene name. Therefore, we define a gene to be made up of one or more models where each model is an mRNA entry with the same ID as a corresponding Gene entry. The numbers are arbitrary. We'll give them 1, 2, 3, …

B.2 Transcription Unit (TU)

This is the total range of transcription for this gene. Thus, it extends from the 5' end of the most 5' exon of any of the models to the 3' end of the most 3' exon of any of the models.

B.3 UTRs

B.3.1 5'UTR

The range extending from the TU start to the start of the first exon of a given CDS.

B.3.2 3'UTR

The range extending from the end of the last exon of a given CDS to the TU stop site.

These are defined for a minority of genes (33% in *Arabidopsis thaliana*)

B.3 Promoter (up1500)

This is defined as **the region extending from 1500 base pairs upstream of the start of the first exon for a given model** (not CDS, we want to include UTRs) to 1 base pair upstream of this start site. Thus, if the first exon starts at 10,000 and is on the (+) strand, we define an up1500 region to be 8,500 – 9,999. Remember, models are defined based on the mRNA entries (if possible).

B.4 Trailer (down500)

This is the opposite of a promoter. That is, it's **the region extending from the last base of the last exon for a given model (not CDS, we want to include UTRs), plus 1, to 500 bases downstream.** That is, if the last exon ends with 10,000 and is on the + strand, we want the region 10,001 – 10,500.

B.5 Exons and Introns

All exons and introns are based on the mRNA entries, unless those do not exist, in which case they'll all be CDS entries. Now there are two possibilities: there is either only one exon, or there is more than one exon. Thus, you'll see either:

B.5.1 mRNA [compliment(]xx..yy

This is very unlikely to occur, but if it does, it's the exon. No introns exist.

B.5.2 mRNA [compliment(]join(xx1..yy1, xx2..yy2, …)

Most common. Each xx..yy pair is an exon, with xx and yy being included in the definition. The gaps between exons are introns, and don't include xx and yy. Thus, [xx1..yy1] is an exon, and (yy1..xx2) is an intron. There must therefore be one less intron than exon for each model.

B.6 Strand Identification

A sequence is assumed to be on the + strand unless it is enclosed by *complement()*, in which case it's on the – strand. Note that if it is on the – strand, you must take the *reverse compliment* of the sequence, where A→T, C→G, G→C, and T→A, and the sequence is in reverse. Also, all arithmetic operators are reversed. Thus, if you see the sequence *compliment(10000..5000),* the promoter region is 11,500 – 9,992.

B.7 Our unique ID

Thus, we finally come to the ID we'll be giving things. As is apparent by now, an ID is more model-specific than it is gene specific, since things like protein_ID, function, exons/intron definitions, UTRs, promoters, etc may vary by model, and are in fact stored in the *.gbk file under the various models. Therefore, the most intuitive way to define our ID is to be of the form GENE_ID.MODEL_NO; unfortunately, decimals make it difficult to deal with in the database, especially since many (most) "model-specific" information is really going to be redundant between models and the easiest way to get rid of duplicate info is using the UNIQUE command, which won't work well with this decimal format. Therefore, we are going to essentially divide the ID into two fields: GENE_ID | Model_Number. Therefore, **a unique integer ID for each gene encountered must be created and used whenever any aspect of any model of the gene is referred to. In addition, all models must be numbered and included in all**

model-specific files. These numbers are essentially arbitrary; ideally though, **geneID corresponded to the gene's relative position on the chromosome**. That is, gene i's furthest upstream base will be upstream of gene (i+1)'s furthest upstream base. The genes should already be in this order in the *.gbk file. If not, don't worry about it too much. Also, GENE_ID doesn't take into account chromosome number, so **be sure each GENE's ID is unique for all genes in all chromosomes.**

C. File Formats

Based off this understanding of how the information is currently stored, here's how we'll do the file formats.

C.1 File Standards

C.1.1 File Names:

Each subentry in section C.2 must be submitted as a separate file named

organism_subEntryName.txt

C.1.2 Format of File Contents

Each file must have attributes separated by tabs and entries separated by newline characters. Every line (including the last) must end with a newline character. Thus, translate anything here in the form attribute1 | attribute 2 | … | attributeN as being what each line in the file should look like, where '|' is replaced with a tab. **If an attribute is null for an entry, leave it null, but keep the surrounding tabs. Also, quotes are taken literally, so don't add any of your own quotes to any entry.**

C.2 File Contents

These are all the files each organism needs to submit, with the specified contents.

C.2.1 Identifiers

ID | ID_Type | Gene_ID | Model_Number

ID = the value of some Identifier from section A.2 (or something that is similar to those)

ID_Type = The type associated with that ID (See A.2). May be GI, PID, Protein_ID, Note, or anything else.

Gene_ID = the ID you give this gene

Model_Number = the model number you give this model

Notes: Some identifiers are found only in the CDS entry or mRNA entry, which is why they must be associated with both a Gene_ID and a Model_Number. You must therefore match each CDS to its corresponding mRNA and pull out all the IDs listed under both headings. If the specific ID is under the Gene heading, just make an entry for each model. Also, for ambiguous types such as "note=", only include it in ID if it looks like your organism uses Note to store ID. Your discretion, but **if there's spaces, it doesn't belong here**.

C.2.2 Descriptors

Gene_ID | Model_Number | Description | Description_Type

Description = Anything that looks like a description; that is, anything that fits in A.3 except translation info.

Description_Type = again, see A.3. Whatever type is associated with the description. May be Function, Note, Experimental, Product, Codon_Start, etc

Gene_ID and Model_Number are the same as always.

Notes: Once again, you'll have to glean this info from gene, mRNA, and CDS headings. This poses a serious problem of how to match the same mRNA model with CDS model, as there's no specific info matching the two together except the specific exon info, which may only differ in one exon (see example in section D) (/gene= might serve as a matching field, but that's not guaranteed). Somehow, you need to figure out a way to do this, because the info we want is in both mRNA and CDS entries.

C.2.3 GeneInfo

Gene_ID | Chromosome_Number | Strand | number_of_models | Block_Start | Gene_Start | Gene_Stop | Block_Stop

Gene_ID = your ID

Chromosome_Number = which chromosome is the gene on?

Strand = (+ or -) Assume + unless sequence includes *compliment()*

Number_of_models = number of unique models

Block_Start = the position of the start of the most 5' up1500 region (see B.3) of all the models.

Gene_Start/Gene_Stop = these correspond to the TU start and stop (See B.2)

Block_Stop = the position of the end of the most 3' down500 region (See B.4) of all the models.

Note: When determining if something is “most 5’ ” or “most 3’ ”, be sure to take into account which strand the gene is on. For (-) strand, most 5’ is the max position, for (+) strand, most 5’ is the min position, etc.

C.1.4 GeneBlock

Gene_ID | sequence

Sequence = The bases extending **from Block_Start to Block_Stop + 8**, as described in C.2.3. We need the plus 8 to catch the last 9-mers. Sequences of genes on the (-) strand must be reverse compliments of those stored in GenBank.

C.1.5 up1500

Gene_ID | Model_Number | up1500_start | up1500_stop

Up1500_start = position of the start of the up1500 region for this model

Up1500_stop = position of the end of the up1500 region, as described in B.3**.** Thus,

|up1500_stop – up1500_start| = 1499.

C.1.6 down500

Gene_ID | Model_Number | down500_Start | down500_stop

Down500_start = the start of the down500 region (ie, end of last exon+1).

Down500_stop = end of the down500 region as described in B.4.

|down500_stop-down500_start| = 499.

C.1.7 Exons

Gene_ID | Model_Number | Exon_Number | Exon_Start | Exon_Stop

Exon_Number = 1, 2, 3, etc, as determined by this exon's position in this model.

C.1.8 Introns

Gene_ID | Model_Number | Intron_Number | Intron_Start | Intron_Stop

(see exons)

C.1.9 UTRs

Gene_ID | Model_Number | UTR | UTR_Start | UTR_Stop

These are the Untranslated regions described in B.3

UTR = 5 or 3 [single digit], corresponding to which UTR this is.

Appendix B: Navigating the website

This appendix is the Help page from the website. It describes the specifics of how to use the Genomes website. Please refer to figures 2 – 4 for screenshots from the website.

Database Query Page

A. Define Motif Sequences

This section allows you to define which motifs will be searched in your query. If left blank, all motifs in the search space will be queried. There are two fields that allow you to type in distinct motifs. Each field has a corresponding number of mismatches to be included in the query.

A.1. Motif Fields

These two fields allow you to type in a list of motifs to be searched. The rules are as follows:

1) Any comma-separated list of 9-mers is valid. All will be searched.
2) Any 9-mer or comma-separated list of 9-mers in the form similar to AC-GTA-ATG-A-G-CG-AT-ACGT-G is valid. This format is interpreted as a consensus sequence, where any set of adjacent nucleotides between a set of dashes specifies the possible nucleotides that may be included in that position. Internally, this will be translated into a list of 9-mers that includes all possible motifs that meet the requirements of the strinF. In the case of this example, 2*3*3*1*1*2*2*4*1 = 288 motifs will be generated and searched. Therefore, you should be careful not to allow the search space to become too large. If it does, the query will take awhile to return. As long as you're patient, this is ok. One thing to note, if you

want to get a 7-mer, all you have to do is end with two ACGT nucleotides: A-C-G-G-C-T-C-ACGT-ACGT.

3) A single motif that is less than 9 nts lonF. This allows you to efficiently search a 6-mer, or even 2-mer without generating thousands of motifs. It is very efficient, but has some severe limitations:
    a. only one can be efficiently computed at a time. Indeed, only one per motif field can be entered. Others will be taken literally and therefore match no 9-mer in the database.
    b. while you can enter a motif that's less than 9 nts in both motif fields, or enter one, and in the other field specify a 9-mer using either of the methods in (1) or (2), to do so will nullify the indexes. A correct answer will return, but expect it to take a couple minutes. Unfortunately, this is true even if you have a 6-mer in the first field and want to take the reverse compliment in the second field. It's a problem with Sybase.
    c. Because of all this, the number of mismatches you specify will be ignored.
    d. Efficiency notes: That said, it is probably more efficient to use the method noted at the end of point (2) if you're looking for multiple 7-mers or 8-mers. But this method is very efficient if you have only one motif to search, and is absolutely preferred if you're looking for a very short motif (even a 3-mer returns in <15sec for a descriptions search).

A.2. Define Motif Sequence Buttons

These buttons launch a form that will generate a motif of the form specified in A.1.2. It includes buttons that will automatically generate n, y, and r. Using this will erase whatever you had in the motif field and replace it with the newly generated motiE.

A.3. Number of Mismatches

This allows you to specify the number of mismatches in the corresponding motif field. **Mismatches will only be generated for lowercase characters.** This allows you to specify which characters can be variable. Note that this also generates all matching 9-mers. A 9-mer with all lowercases and 1 mismatch will be transformed into 36 motifs; with 2 mismatches, 354. **So don't ask for more than 2 mismatches unless you're really patient or have several positions marked with uppercase letters.**

A.4. Reverse Compliment

Simply generates the reverse compliment of motif1 on the motif2 field, whether or not the motifs in field 1 are valid.

B. Define Organism

Simply defines which organism this query applies to.

C. Define Genomic Region to Search

Lets you specify what region to search in. There's two main parts of this: the genomic region, and the range.

C.1. The Radio Buttons

The radio buttons specify how the region and range information will be used. If "specify range" is selected, then only the range is used. Region information is ignored. The range

in this case is relative to the start of each gene. If "genome region" is selected, then both the region and the range are used.

C.2. The Genome Range Select Menu

Let's you specify the range the query applies to. Exon and intron use position information starting at the 5' end of the sequence. All positions are positive. "Exon from 3' end" and "intron from 3' end" use position information starting from the 3' end of the sequence. All positions are therefore negative. "Intron/Exon percent from 5' " is a calculation for each position that calculates the percent of positions in that exon that lie to the left of that position. Thus, for an exon of length 20, position 5 becomes 25(%). This is valid only for graphinF. Up1500 refers to the upstream regions that are all 1500 nts in length. Much of this information then includes overlaps with exon and intron information. Upstream region is the same, except those genes whose up1500 regions overlap with a transcriptional unit are truncated so that there is no overlap.

C.3. The Range Parameters

These fields allow you to specify an upper and lower bound to the region. They default to numbers that are way outside the range of any region, and so have no effect on your query. The fields allow you to specify, though, that you want to look at only a certain subset of the region.

D. Restrict Search to Subset of Genes

This allows you to narrow your search down to a subset of genes. If ratios are being computed, and the compare genes type is chose (see G), then the genes in this list are used as the primary set of genes. The genes can be specified in one of two ways: as a list of databaseIDs (integers) or a list of any of the Identifiers that are in the database for the

genome being searched. This should include GI numbers for all genomes. For Arabidopsis, it includes pubLocus (called "gene" in the database), GI, reference Accession number (called "transcript_id" in the database), and rna (which appears to be synonymous with transcript_id). As of now, searching for "other" will search for any match in the list of identifiers, except databaseID. For Arabidopsis, there are currently no non-unique identifiers. Those that aren't strictly unique refer to the same gene, and so are functionally unique. As of the time of this writing, there were no databaseIDs that were also GI numbers, or any other identifier type. This means if you specify "other" and enter geneIDs, an error will be returned.

Note: **Be sure the correct radio button is pushed**. If you enter non-integers and look for databaseIDs, you will receive an error. If you enter databaseIDs and look for "other", no results will be returned and you will receive and error.

Delimiters: Acceptable delimiters at this time include, space, comma, tab, and newline or any combination thereoE.

D.1. Using user-defined geneLists

If you find a list of genes that you like, you can save them using the "Create List" button. When you are through, you can delete it using the "Delete List" button. Lists are permanently stored in the database and can be accessed at anytime by any user, from any platform (including VXInsight). Only the lists that were created for the currently selected organism are displayed. Whichever gene type was specified upon creation of the list will always be remembered.

D.1.1. Selecting a pre-existing list

The select box under "Gene list to use" includes all the current lists for the selected organism, no matter who the creator is. When you change the selection, the page will be reloaded with the new list pasted into the geneList textarea. All form information will be saved.

D.1.2. Creating new gene lists

The Create List button allows you to save the current list of genes. It opens a prompt box and asks you to specify a name. Names can have spaces in them, but the total length must be less than 25 characters. If you submit a longer name, it will complain and ask you to enter a new name.

If the name you specify is the same as a name that already exists, you will be asked to confirm that you want to overwrite the existing list. Please note though: **you can only overwrite lists you created.** If you are not the creator of a list, you cannot overwrite it. In this case, you must choose a different name. Names are not case sensitive.

You will also be prompted for a description of the list. This is optional, may include quotes, and may be up to 500 characters, including spaces.

D.1.3. Deleting lists

When you press the "Delete List" button, the list name that is currently selected in the select box is deleted. You are asked to confirm that you want to do this. Please note: **you are not allowed to delete lists you didn't create.** While the webpage starts to act as

though it is deleting the list, when it discovers you're not the creator, it will let you know and stop the process.

E. Get Statistics

Click the checkbox to retrieve computed statistics for the specified motifs in the specified region. The select menus let you specify which field you would like to sort by, and in which order. If no genes have been entered in the geneList, this will return a sorted list of pre-computed, genome-wide statistics. If genes have been entered in the geneList, then statistics are generated for the motifs in that list.

E.1. With Gene List Empty

With the geneList empty, the statistics returned reflect those of the entire genome for the region you selected.

E.2. With Genes Specified

If the geneList is not empty (even if there's just non-whitespace non-sense characters), the statistics for those genes, using the given region, range, and motifs, are calculated and stored in a temporary table. The way Sybase implements global temporary tables, each user can see only the data they've they've entered. This ensures concurrency. Also, the data lasts only as long as the connection lasts. Thus, if you log out (or the server is restarted), you will lose that data and have to recalculate your statistics.

The statistics calculated are the same as those calculated for the genome, with the exception that for each motif, the KS probabilities from the geneList are compared to those of the entire genome by the formula LOG10(group.KSprob/genome.KSprob). The final result is sorted in user-defined order, but defaults to this LOG10 value. Thus, the

motifs at the top of the list are those that are less random in the set than they are in the whole genome.

E.3. Recalculate Statistics

This option is only available if you have genes listed in the geneList. Check this box if you want to recalculate the statistics for the given group. If left unchecked, the current table will simply be re-sorted based on the sorting parameters you provide. Since calculation takes a couple minutes, this allows you to play with the data quickly. If you change the genes in the geneList, you must check this box for the group statistics to reflect this.

F. Compute Ratios

Click the checkbox to compute ratios of occurrence, then specify which type of computation you would like to do.

F.1. Compare Regions

The default type compares the ratio of occurrence of each motif in the primary region vs that in the secondary region. The order matters only in how the results are displayed and sorted. Ratios are computed by taking each motif and dividing the number of times it occurs in the primary region by the number of times it occurs in the secondary region. Multiplying by T2/T1, where T2 = total number of motifs in secondary region, and T1 each the total in the primary region, gives us the normalized ratio. The data is organized in descending order by ratio (which is the same order as that by normalized ratio). If motifs are specified in *Define Motif Sequences*, only those motifs will be returned.

F.2. Compare Genes

Use this to compute the ratio of occurrence for each motif in the subset of genes versus all the other genes. The primary group is the genes specified in *Gene List*, the region and range from *Define Genomic Region* are used, and if motifs are specified in *Define Motif Sequences*, only those motifs are returned.

F.3. Min Number of Occurrences

Specifies the minimum number of occurrence in the primary group that will be returned. This is useful because often times the highest ratios will be motifs that occur only once or twice, which many not be biologically meaningful.

G. Type Your Own Query

Click the checkbox to have your own query computed. This obviously requires quite a bit of knowledge regarding the database. Its intended use is for minor modifications of automatically generated queries. Since every query you generate will include the SQL code on the results page, you can copy that code and modify it here. CBBC users have permissions to view all tables (except most system tables), and to create and destroy new tables and views. You cannot, however, modify existing tables in any way. Full access users can view any tables that are necessary for the website and can create and delete lists. They cannot, however, create new tables or views. Guest user accounts are limited further by not being allowed to create or delete lists.

G.1.Limit the number of results returned

Please be careful how many rows your query will return. A query can easily return millions of rows, which will never return to you. While I have hardcoded in a limit of 5000 rows, this number is still quite high for network traffic, so you should make a habit

of using the select top *xxx* clause. You may notice that all generated queries do this, and the default query starts out with a limit of 1000 rows.

G.2.Error Messages

If you're SQL code isn't perfect, you will receive an SQL exception on the results page. On the off chance that it's helpful, say a prayer of thanksgivinF. Otherwise, look hard at your code and hope for the best.

H. What do you want to do with this information?

H.1. Get Sequences

Takes the motif, geneList, region, and range information, and returns the sequences that contain motifs that match all the requirements.

H.2. Get Descriptions

Takes the motif, geneList, region, and range information, and returns descriptions of those genes that contain motifs that match all the requirements. If the regions are exons or introns, the exon/intron number, as well as the total number of exons/introns for that gene are returned. The motif list, position list, and exon/intron number list are all in matching order. So you can figure out what goes with what based on that.

If "include reference links" is selected, then the reference ID replaces the geneID. This is linked to a curated database of locus IDs and information. Currently, this is only implemented for *Arabidopsis*, which links to www.arabidopsis.org.

H.3. Limit Number of Results Returned

All queries but the user-defined query use this number to limit how many results are returned. 1000 should be more than enough. I've hardcoded a maximum of 5000 rows to

prevent you from crashing the system. If you want quicker results, make this number smaller.

I. The Buttons

I.1. Get Info

Processes all the queries you have indicated using the checkboxes. Directs you to the results page, which will display all results in order. This page includes the SQL query used for each calculation.

I.2. Get Graph

Launches a new window that will graph your specified motif, taking into account the gene list and the region and range information. BinSize specifies the size of the bins used in creating the histogram.

I.3. Reset Form

Simply resets the form back to its default settings.

The Graphs Page

A. The Graphs

The graph includes a Cumulative Distribution (CD) plot, a histogram, and some statistical information. The y value of the CDPlot shows the percentage of motifs that are expected to lie to the left of a given position, specified by the x axis. There are up to three separate lines: the expected line, which takes into account the varying lengths of sequences in the specified region, Motif1, and Motif2. The latter two are cumulative measures for all motifs entered in motif field 1 and two, respectively. Thus, you can graph a 6mer, a set of related motifs, or even completely random motifs together, and compare them against

their reverse compliments, for example. The same information is used to plot the histogram.

Note: All position information on the QQPlot is positive, even if it should be negative. This in an artifact of using a package I don't fully understand. To convert, simply take the maximum value shown on the graph, and subtract that from all x values. So when plotting up1500 regions, you should subtract 1500 from all positions on the x axis. I apologize for the inconvinience.

B. The Statistics

The statistical information includes three things: a d value, an x value, and a KSProb value. The d value is the maximum y-axis deviation between a graph and the expected graph. This will be between 0 and 1 and is a percentage. The x value is the position at which this d value occurred. Again, if this is a negative region (eg, upstream, 3' intron), you must subtract the maximum value from the x value to get the corresponding negative position. The KS probability is essentially the probability of getting that d value with the number of observations. You can interpret this as saying a really small KSprob value (eg <0.01) indicates this distribution is not likely to be random. It is not clear what appropriate cutoffs (α values) should be. Most distributions are pretty clear-cut though.

C. Width and Height Fields

These simply resize the jpegs. They actually recalculate and regenerate the graphs, so you'll get a different pixel count. If you need to copy this into photoshop, you may want to copy a large picture to give you better flexibility