DARM: DISTANCE-BASED ASSOCIATION RULE MINING

by

Aleksandar Icev

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

May 6, 2003

APPROVED:

_____

Professor Carolina Ruiz, Thesis Advisor

_____

Professor Elizabeth Ryder, Thesis Co-Advisor

_____

Professor Stanley Selkow, Thesis Reader

_____

Professor Micha Hofri, Head of Department

Abstract

The main goal of this thesis work was to develop, implement and evaluate an algorithm that enables mining association rules from datasets that contain quantified distance information among the items. This was accomplished by extending and enhancing the Apriori Algorithm, which is the standard algorithm to mine association rules. The Apriori algorithm is not able to mine association rules that contain distance information among the items that construct the rules. This thesis enhances the main Apriori property by requiring itemsets forming rules to "deviate properly" in addition to satisfying the minimal support threshold. We say that an itemset deviates properly if all combinations of pair-wise distances among the items are highly conserved in the dataset instances where these items occur. This thesis introduces the notion of proper deviation and provides the precise procedure and measures that characterize it. Integrating the notion of distance preserving frequent itemset and proper deviation into the standard Apriori algorithm leads to the construction of our Distance-Based Association Rule Mining (DARM) algorithm.

DARM can be applied in data mining and knowledge discovery from genetic, financial, retail, time sequence data, or any domain where the distance information between items is of importance. This thesis chose the area of gene expression and regulation in eukaryotic organisms as the application domain. The data from the domain was used to produce DARM rules. Sets of those rules were used for building predictive models. The accuracy of those models was tested. In addition, predictive accuracies of the models built with and without distance information were compared.

Acknowledgements

Without the patience and support of my advisor Professor Carolina Ruiz this thesis was going to be simply impossible. It was hard to survive the tough times of this thesis without hearing Carolina's words: "Oh, no problem, you should easily solve that using this and that". So, Carolina thank you from the bottom of my hart.

My co-advisor Professor Liz Ryder introduced me to the secrets of the DNA world. I can just imagine how hard was to do that with a computer engineer like me. Her incredible sensitivity of the work done for this thesis, kept my motivation at high levels all the time.

I would also like to thank Professor Stanley Selkow. His spirit is strong enough to vitalize the whole CS Department. So I was very happy that I was very often too close to his office to take from his energy.

My parents, even far from WPI, were full with the love and encouragement for my work on this project.

Katica, my fiancée, showed what a real love should look like when you should support your partner in a 24/7 working week.

CONTENTS

LIST OF FIGURES

1 Introduction

1.1 Context of the Problem

The Knowledge Discovery in Databases (KDD) field is concerned with the development of methods and techniques for making sense of data [FSS96]. Association rule mining [AIS93] identifies collections of data attributes that are statistically related in the underlying data. An association rule is an expression of the form X=>Y where X and Y are disjoint sets of items. In a dataset D, consisting of data instances where every instance is a set of items, the rule X=>Y has support **sup**, equal to the percentage of the instances of D that contain both X and Y. Support count **supcnt** is the number of instances of D that contain both X and Y. The confidence **conf** of the rule is the percentage of instances in D that contain Y among those that contain X.

Apriori [AS94] has become the standard algorithm for association rule mining. However, this algorithm is not able to mine association rules that contain distance information among the items that construct the rules. This thesis extended and enhanced the Apriori algorithm in order to extract important patterns from datasets that capture distance information among the items that construct the rules.

1.2 Application Domain

This thesis chose the area of gene expression and regulation in eukaryotic organisms as an application field. The DNA (DeoxyriboNucleic Acid) sequence of these organisms is being collected and stored in computer readable formats with an enormous rate of

progress in the last several years. Every cell in a single eukaryotic organism has the same DNA sequence, unique for that entity. Each DNA has a double strand helical structure. Each strand consists of a chain of nucleotide subunits. There are four nucleotides present in the DNA: adenine (A), guanine (G), thymine (T) and cytosine (C).

A gene is a part of the DNA, which when activated is responsible for the protein production in the cell. Different genes are active in different cells. There are two major factors that make the same genes in the same DNA in one type of cell become active (or be transcribed and then translated into protein) and in another kind of cell stay dormant. The first general factors are the so-called transcriptional proteins. They reside in the cell and interact with the binding sites of the DNA when the respective gene should be activated. Different types of transcriptional proteins are present in different types of cells. The second general factor is the combination of promoter subsequences (or motifs) of the DNA. The promoter is a part of the DNA that is located upstream of the gene and determines whether the gene is active ("on") or dormant ("off"). This process is illustrated in Figure 1**.** In this example, three motifs, M1,M2, and M3, which lie in the upstream region of gene X, are interacting with the transcriptional proteins in order to activate gene X.

Molecular biology experiments have shown that not only the existence of the appropriate combination of these motifs, but also the proper pairwise distances among them, are possible preconditions for a gene to be triggered [Whi01]. If the appropriate transcription factors are present in a particular cell, and the corresponding motifs are present on a particular gene, then the transcription factors will bind the motifs and turn

the gene “on” or “off”. We wanted to build association rules that would describe whether a gene is activated or not based on the presence of a certain mixture of motifs and distances among them in given cell type(s).

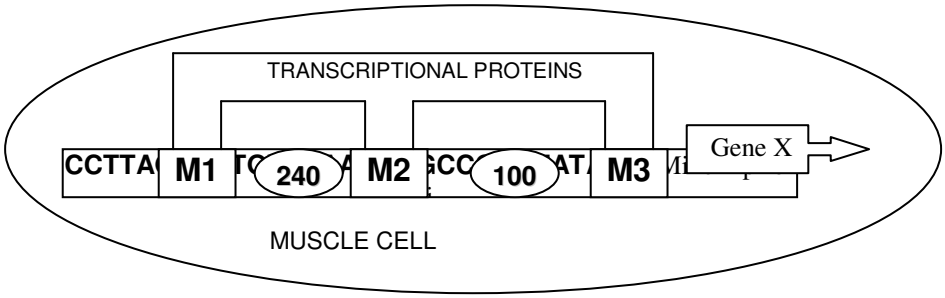


Figure 1. Gene expression

1.3 Problem Statement

The core problem of this thesis was to design and implement an algorithm to generate distance-based association rules. The parameters that measure the quality of the rule are the support (sup), the confidence (conf), and the maximal coefficient of variation of distances (cvd). This last parameter is introduced in this thesis to capture the clustering significance of all pairwise distances of motif (item) members of a rule.

In order to illustrate the meaning and the role of the cvd parameter we extract association rules from the sample data shown in Figure 2.

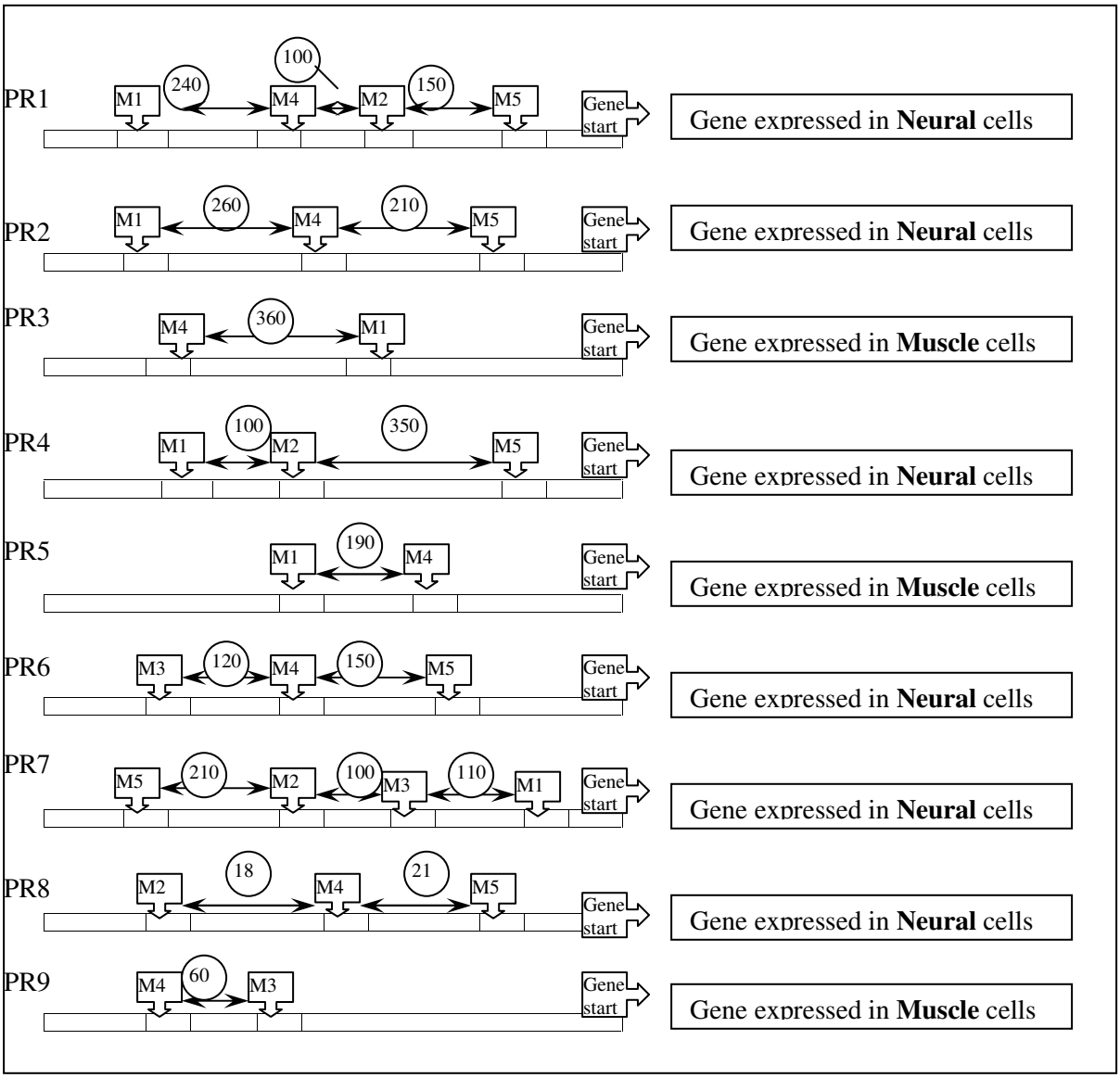


Figure 2. Data sample. PR (Type of the cell)) =Promoter region with gene being expressed. in a Neural cell or Muscle cell. Boxes=DNA sequences. (Mi)=motif(i). Numbers in the circles=distances between motifs**.**

This sample consists of 9 data sequences related to 9 different gene promoter regions (PR1-PR9). Each data instance consists of two attributes. The first one is a set-valued attribute containing the distinct motifs that are found present in the respective gene promoter region. The second one is the cell type(s) where this gene is expressed (Neural

or Muscle). Pairwise distances among the motifs are also given in Figure 2. These distances are measured in (DNA) basepair positions.

Let us assume that we want association rules that have three motifs in the antecedent and one type of cell in the consequent. If the support threshold is (2/9)*100%=22.2% and the confidence threshold is 100%, applying the standard Apriori [AS94] over this dataset will generate the rules presented in Figure 3.

R1: M1, M2, M5=>Neural (sup=33%), (conf=100%)

(M1, M2, M5 & Neural present in PR1, PR4, and PR7)

R2: M1, M4, M5=>Neural (sup =22%), (conf=100%)

(M1, M4, M5 & Neural present in PR1, PR2)

R3: M2, M4, M5=>Neural (sup =22%), (conf=100%)

(M2, M4, M5 & Neural present in PR1, PR8)

Figure 3. Rules obtained from standard Apriori

Rules R2 and R3 in Figure 3 have the same values for support and confidence. Based on those measures no distinction can be made between R2 and R3. However, comparing PR1 and PR2, we notice that M1, M4, and M5 are very similarly clustered with respect to their pairwise distances. In the promoter regions PR1 and PR8 supporting R3, we notice that M2, M4, and M5 are in a different order, and are further apart in PR1 than in PR8. After this small analysis it becomes clear that the second rule is likely to be more significant, from the biological point of view, than the third one. This significance will be

measured by the coefficient of variation of distances (cvd). This coefficient will enable the generation of distance-based association rules.

1.4 Related Work

Correlation measure and statistical dependence among the items that construct an association rule are introduced in [BMS97] and [SBM98] respectively. Collective strength [AY98] is used to measure if a group of attributes occur together in the data sequences. Our work relates to these approaches in that we have the same foundational principle of the importance of the statistical dependence among the items. But our work differs in that none of those approaches consider the variation of the distance among the items as a correlation measure.

Miller and Yang [MY97] introduced a type of distance-based association rules. Their work concentrates on datasets that contain numeric attributes. The values of the attributes are discretized into different numbers of bins using clustering. For example if there are six instances of data and the attribute age has values 7,20,22,50,51,53, respectively, they bin the attribute into three bins [7,7], [20,22], [50,53]. Binning is performed after the clusters related to the distances among the values are determined. After each numeric attribute is binned in the above manner, association rules are mined from the transformed dataset. Our approach differs from that in [MY97] in that we base our distance measures across different attributes, not within the values of each attribute.

Spatial association rules are explained in [Dun02]. According to their definition for this type of rules, either the antecedent or the consequent of the rule must contain

spatial predicates (such as *near*). An example of a rule with spatial antecedent and nonspatial consequent is: If a house is located near Central Park, it is expensive. Support and confidence for spatial association rules are calculated in the same manner as for regular association rules. The difference between spatial association rules and regular association rules is that in the former, the underlying database is not viewed as a set of transactions. Instead, it is a set of spatial objects [Dun02]. The spatial predicates that denote the topological relations are considered as given by the data mining query. This approach differs from ours in that it considers spatial relationships of the values of the same attributes across the dataset, while the relationships across the distinct attributes are not explicitly considered.

Previous work at WPI on motif and expression based classification of DNA (MEBCS)-[MPPT01] considered the significance of distances between the motifs that construct the rules in biological data, but only after the Apriori algorithm generated standard association rules. The MEBCS system established the sequential flow of tools and methods for constructing association rules out of DNA sequences. Generation of the association rules in this thesis follows this sequential flow.

1.5 Contribution of This Work

The main contributions of this thesis are the design, implementation, and evaluation of an algorithm for mining distance-based association rules. These rules should increase the significance of the patterns that are mined over data in domains in which distance information is of importance.

2 Background

2.1 Motif Elicitation: The Expectation-Maximization (EM) Algorithm

A preliminary point to this thesis was to obtain real genetic data that contain motifs and distance information among them. First, we needed an apparatus that will find motifs in a collection of DNA sequences. For this purpose the Multiple Expectation and Maximization for Motif Elicitation (MEME) tool was used. The MEME core algorithm extends the expectation maximization (EM) algorithm for identifying motifs in sequences [BE95]. Multiple motifs are found by fitting a two-component finite mixture model to the data [BE94]. Once the motifs are found for a series of sequences, the Motif Alignment and Search Tool (MAST) [BG98] is used to query each single sequence for particular motifs and respective distances between motifs. MEME/MAST can be set to find duplicate occurrences of the motifs as well. All motifs found are denoted by their p values- the probability of the random occurrence of each of the motifs.

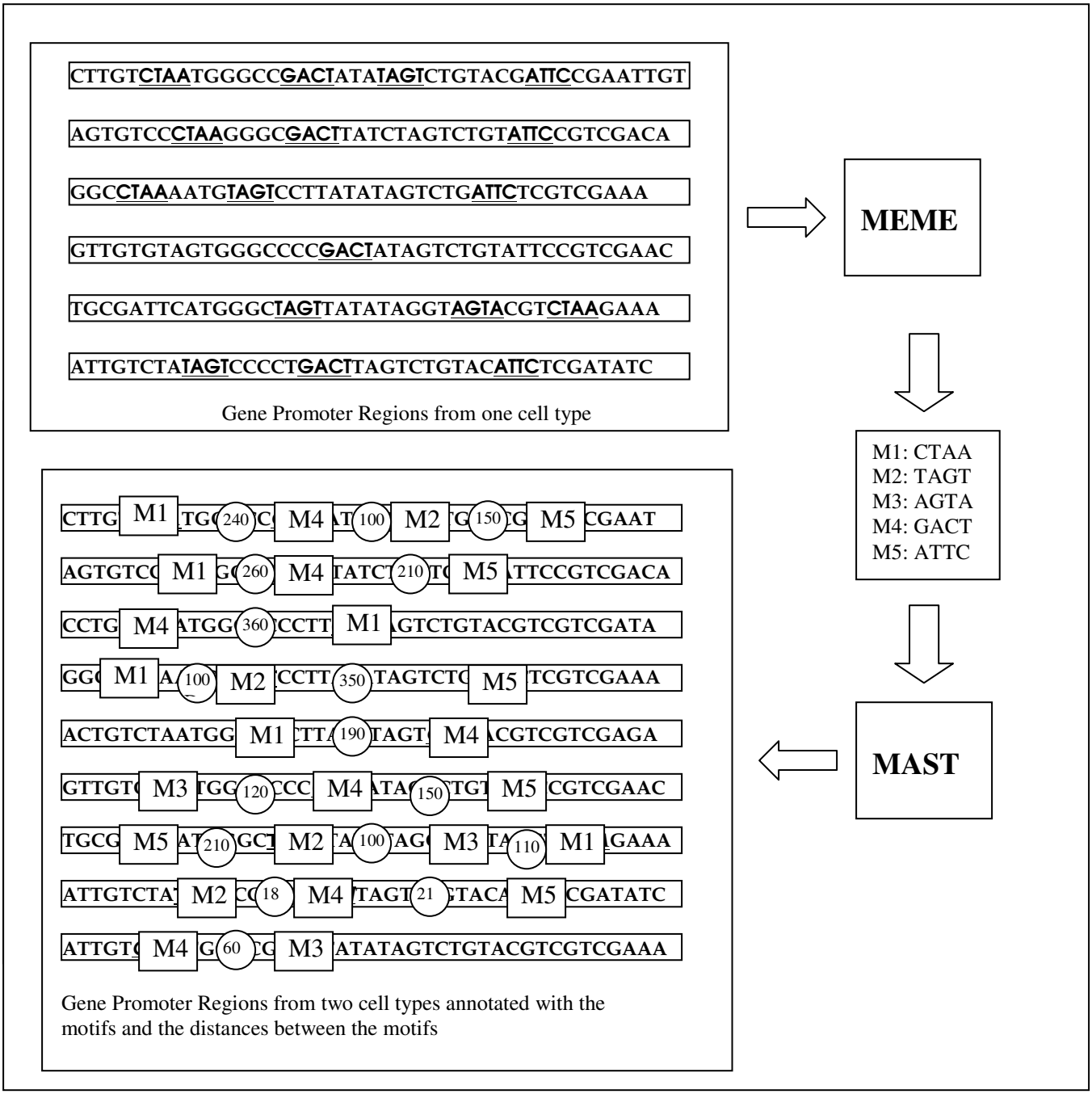


Figure 4. MEME-MAST MEME elicits motifs; MAST annotates sequences. Five sample motifs identified by MEME are shown in the input sequences

The lower the p-value of a motif is, the higher the authenticity of the motif. Annotated motifs and distances are produced as the output of the systems. Typically, MEME is

used to elicit motifs from promoter regions of genes that are all expressed in a particular cell type. MAST is then used to annotate a group of promoters of interest with these motifs (Figure 4).

2.2 The Apriori Algorithm

The Apriori algorithm to mine association rules was introduced in [AS94]. Given a dataset that contains sets of items (called instances), a minimal support threshold, and a minimal confidence threshold, Apriori mines all the association rules from the dataset that have support and confidence above the thresholds.

Apriori employs an iterative approach known as level-wise search, where k-itemsets (itemsets with k items) are used to construct (k+1)-itemsets [HK01]. An itemset is frequent if it has support greater than or equal to the user defined minimal support. The Apriori algorithm is based on the Apriori property: "All subsets of a frequent itemset must be frequent". The Apriori workflow is shown in Figure 5. In step (1), the set of frequent 1-itemsets is constructed. This set is denoted by L_1. In the next step the collection of candidate itemsets C_2 is generated from the frequent itemsets in L_1. In step (3) the itemsets in C_2 that are not frequent or that contain subsets that are not frequent (SP) are pruned, obtaining the frequent itemsets in L_2. In step (4) C_3 candidate sets are generated by joining L_2 with itself. In general, the candidate itemsets at level C_{K+1} are generated from the itemsets in L_K by joining L_K with itself as follows: If two itemsets $\{a_1,\ldots., a_K\}$ and $\{b_1,\ldots., b_K\}$ in L_K, whose items are sorted according to a given order,

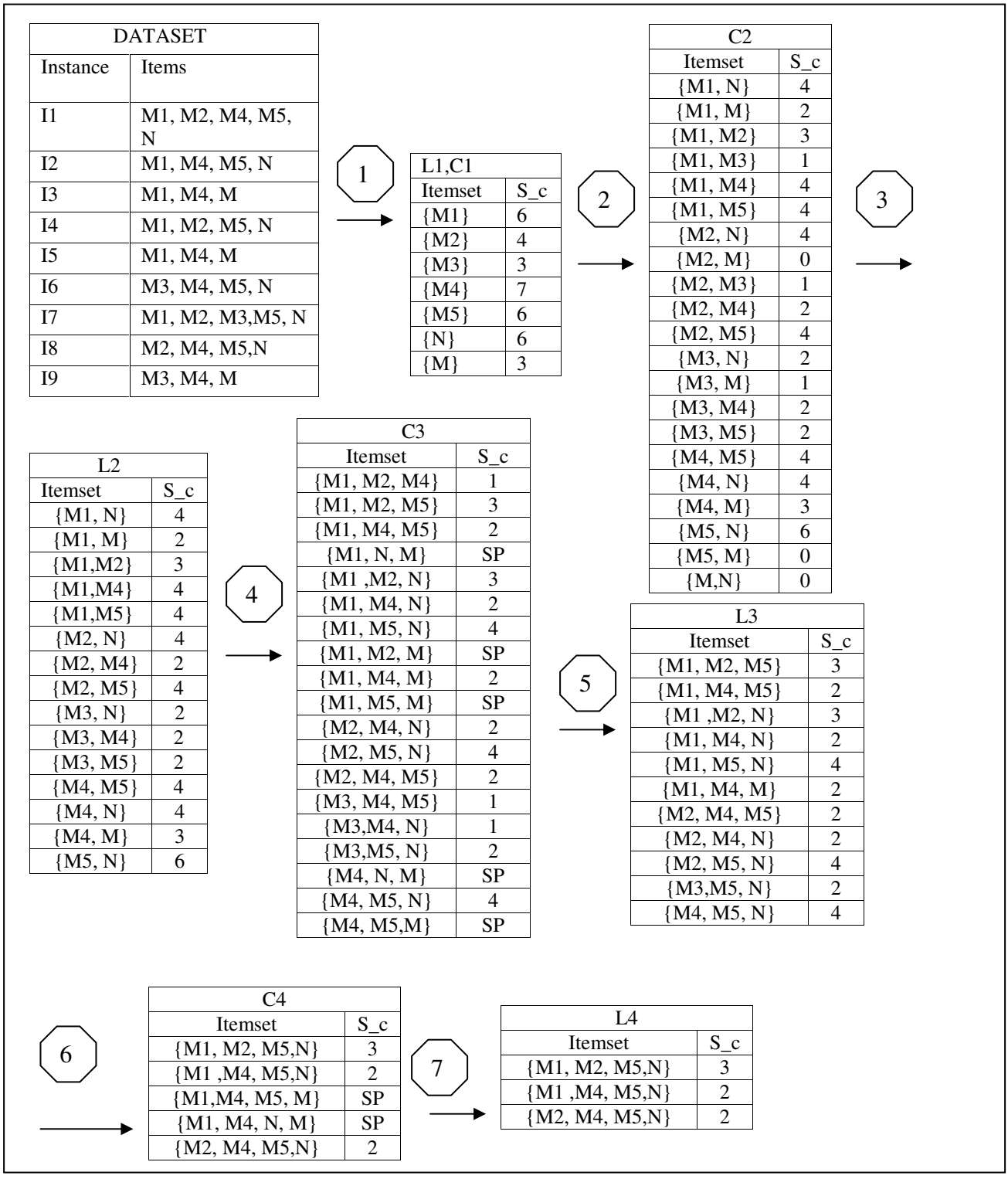


Figure 5. Regular Apriori Workflow. Minimal support count (msc) is 2, and confidence is 100%. Subset pruned (SP) Support count (S_c).

are such that $a_1 = b_1$, $a_2 = b_2$,…., and $a_{k-1} = b_{k-1}$, then the join of these two itemsets $\{a_1,\ldots., a_K , b_K \}$ is added to C_{K+1} . Subsequent steps repeat the procedure described by steps 2, 3 and 4 over the higher levels. This process stops when no further candidate itemsets can be generated. Once all the frequent itemsets have been constructed, association rules satisfying the minimal confidence condition are generated. This process is accomplished as follows [HK01]:

1. For each frequent itemset F, generate all nonempty subsets of F.
2. For every nonempty subset S of F, compute the confidence of the rule:

"S => (F\S)":

$$\text{confidence}(S \Rightarrow (F \setminus S)) = \frac{\text{support\_count}(F)}{\text{support\_count}(S)}$$

If confidence(S => (F\S)) >= minimal_confidence threshold, then output the rule.

Given the above example, Apriori will generate 27 rules and they will have the format given on Figure 6.

M5 ==> M	(conf: 1.0), (sup: 0.6666667)
M1 && M2 ==> M5 && N	(conf: 1.0), (sup: 0.33333334)
N ==> M5	(conf: 1.0), (sup: 0.6666667)
M ==> M4	(conf: 1.0), (sup: 33333334)
M1 && M ==> M4	(conf: 1.0), (sup: 0.22222222)
M3 && M5 ==>N	(conf: 1.0), (sup: 0.22222222)
……	

Figure 6. A subset of the rules generated by Apriori over the dataset in Figure 5.

2.3 Classification Model

Using our association rules, we wanted to predict whether or not a gene of interest will be expressed in a given type of cell. These rules will have the dependent attribute (gene expressed or not in the given type of cell) in the consequent. Motifs present in the antecedent will be predicting attributes. Association rules whose right-hand-sides are restricted to the classification class attribute are called class association rules (CARs) [LHM98]. A classification model is generated by selecting some (or possibly all) of the rules mined by Apriori. Several different criteria have been proposed for this selection, some of which are described in [LHM98]. If we test this model with a new dataset we can estimate its accuracy. The accuracy of a classification model is the proportion of correct predictions over the total number of predictions made.

3 Our Approach

3.1 Distance-Based Apriori

One of the main goals of this thesis was to build an accurate classification model consisting of association rules that capture distance information. We would expect variability of the distances among motifs to depend upon of the actual sizes of the distances. That is, longer distances would have bigger standard deviations than smaller distances. Thus, to determine whether distances represent similar clustering among promoters we used the coefficient of variation of distances (cvd) introduced in [Zar99]. The cvd of a pair of motifs with respect to an itemset I is the ratio between the standard deviation and the mean of the distances between the motifs taken from the promoter regions that contain I. For the itemset IR1= {M1, M2, M5} from the rule R1 in Figure 2 there are three cvd's: cvd_{IR1} (M1,M2), cvd_{IR1} (M1,M5) and cvd_{IR1} (M2,M5).

In order to calculate the cvd's, we first calculate the appropriate means and standard deviations. Each mean is calculated upon the distances in the promoter regions where all motif members of the itemset IR1 are present. The distance between say M1 and M2 in PR1 is denoted by d_{PR1} (M1,M2).

$$\mu_{IR1}(M1,M2)=\frac{d_{PR1}(M1,M2)+d_{PR4}(M1,M2)+d_{PR7}(M1,M2)}{3}=$$

$$=\frac{340+100+210}{3}=216.66$$

$$\sigma_{IR1}(M1,M2) = \sqrt{(\frac{1}{3-1}) * \sum_{i=1,4,7} (d_{PR_i}(M1,M2) - \mu_{IR1}(M1,M2))^2}$$
$$\approx 120$$

The coefficient of variation of distances (cvd) for the pair M1,M2 is:

$$cvd_{IR1}(M1,M2) = \frac{\sigma_{IR1}(M1,M2)}{\mu_{IR1}(M1,M2)} = 0.554$$

In the same manner we calculate the rest of the cvd's and we obtain the rules that are depicted in Figure 7.

R1: M1, M2, M5=>Neural (sup=33%, conf=100%)			
		M2	M5
M1	cvd	0.557	0.076
	mean	216.6	462.0
	sdev	120.0	35.0
M2	cvd		0.433
	mean		237.0
	sdev		103.0

R2: M1, M4, M5=>Neural (sup=22%, conf=100%)			
		M4	M5
M1	cvd	0.056	0.036
	mean	250.0	488.0
	sdev	14.0	18.0
M4	cvd		0.136
	mean		233.0
	sdev		31.68

R3: M2, M4, M5=>Neural (sup =22%, conf=100%)			
		M4	M5
M2	cvd	0.982	0.772
	mean	59.0	97.0
	sdev	58.0	75.0
M4	cvd		1.199
	mean		138.0
	sdev		165.0

Figure 7. Distance-Based Association Rules obtained from the dataset in Figure 2

Now we can illustrate what we want from the system: rules that satisfy the min support and min confidence thresholds, but also such that items in a rule preserve their distances in the dataset instances that support the rule; i.e. their cvd's are below some

maximal allowed threshold- maxcvd. The maxcvd is given by the user, and cvd's for each pair of items in the rule should be less than the maxcvd. So, for the rules given in Figure 7, if the user of the system sets maxcvd threshold to be maximum 0.15, rules 1 and 3 will be removed and only rule 2 will stay, since only for the rule 2 all pairwise cvd's are below the given maxcvd=0.15.

Inclusion of this parameter in the process of mining rules is accomplished by extending the Apriori Algorithm. First, the notion of a frequent itemset is enhanced to the notion of distance preserving frequent (DPF) itemset. In order to be DPF, itemsets need to satisfy the minimal support threshold and in addition to this, the itemset must *deviate properly*. In general, an itemset I deviates properly if it satisfies the following definition:

Given a minimal support count (msc), a maximum cvd (maxcvd), the set SI of instances that support I, that is the instances in the dataset that contain the itemset I, we say that I deviates properly if for each pair of items Mi, Mj in I there is a subset SIij of SI of cardinality msc for which the cvd(Mi, Mj) in SIij is less than or equal to the maxcvd.

This definition requires each pair of items in an itemset I to have a cvd less than the maxcvd in a subset SIij of SI. If an itemset does not satisfy this condition, it means that no matter what items are added to the itemset in higher levels of the Apriori process, the resulting superset either will fail to have the minimal support required or will contain a pair of items whose cvd is above the maximum cvd allowed. Hence no rules can be generated from this itemset (or any of its supersets) and so itemset can be removed from consideration.

If we look again at the example in Figure 3 we can notice that the itemset I= {M1, M4} is present in PR1, PR2, PR3 and PR5. If we calculate the cvd from all four sequences we obtain cvd_I(M1,M4)=0.27. But when R2 is generated we are only interested in the sequences that contain M1, M4, and M5 together, and those are PR1 and PR2. If we calculate cvd for itemset {M1, M4} only from PR1 and PR2 we obtain cvd_I(M1,M4)=0.0564. This is below the maxcvd and it is calculated from the same number of sequences (two) as the value of the minimal support count (two) for this example. Hence we say that this itemset deviates properly. Figure 8 presents our procedure to determine if a frequent itemset deviates properly. This procedure returns true if the input itemset I deviates properly, and returns false otherwise.

```
bool deviates_properly(Itemset I, msc, maxcvd)
//PRE: ItemSet I is nonempty; msc is the minimal support count, maxcvd is the maximum cvd and they
//are positive numbers.
//POST: returns true if the Itemset I deviates properly; returns false otherwise.
{
  numOfSeq= number of the instances in the dataset;

  for each pair of items in I do{

     {   If  the procedure combinations_deviate_properly (Pair, msc, numOfSeq, maxcvd) returns false
         return false }
   }
  return true;
}
```

Figure 8. Deviates_properly Procedure

A simple illustration of the deviates_properly workflow can be given for the itemset I={M1, M4} from our dataset. First, the function deviates_properly is called with required parameters {M1, M4}, msc=2 and maxcvd =0.15.

```
bool combinations_deviate_properly (Pair P, msc, numOfSeq, maxcvd) {

//PRE: P is a Pair of items, msc is the minimal support count, numOfSeq is the number
//of sequences, maxcvd is the maximum cvd and they are positive numbers.
//POST: returns true if there is a subset of instances in the dataset, with cardinality msc,
//such the cvd of the distances between the items of the pair P in those instances is less
//than the maxcvd

     howmanySubsets= numOfSeq - msc +1;

    //each subset will contain msc data instances
 Vector allSubsets=createTheSubsets (P, howmanySubsets, msc); (see figure 10)

for each subset from allSubsets do{
     calculate the subset's cvd ;
     if  (cvd < maxcvd)
     {  return true; }
 }

  return false;
}
```

Figure 9. Combinations_deviate_properly Procedure

Since there is only one pair of items in {M1,M4} we call the function combinations_deviate_properly with required parameters ({M1, M4}, 2, 4, 0.15). Since the number of data instances where this pair is found is four (PR1, PR2, PR3 and PR5) and msc is 2, there are three subsets of 2 promoter regions and each should be tested for their cvd's. Those three subsets are created using the procedure createTheSubsets. The createTheSubsets procedure first sorts the distances taken from the four promoter regions. In this example, the sorted array of distances will be 190, 240,260, and 360. Then according to the procedure three subsets of size two will be created: {190,240}, {240,260} and {260,360}. Only these three subsets need to be checked if their cvd's are below the maxcvd, since all other subsets of two that can be created from this array will have larger cvd's than at least one of these three pairs. Calculating the cvd(M1,M4) from

the subset {240,260} will give cvd(M1,M4)=0.056. Since 0.056<0.15 this is enough to conclude that the itemset I deviates properly.

```
Vector createTheSubsets (Pair P, int howmanySubsets, msc){

//PRE: P is a Pair of items, howmanySubsets is the number of subsets, and msc is the
//minimal support count and they are positive numbers.
//POST: Exactly (howmanySubsets) subsets are created, stored in the vector allSubsets
//and returned by this procedure

Calculate the distances between the two items, in the pair from
all promoter regions where the pair  is present
sort  these distances and store them in the array DistanceArray;
create new Vector called allSubsets that will hold all the subsets of
size msc of those    distances

    for(int i=0; i<howmanySubsets;i++){

            create the i-th subset from the members of the DistanceArray starting
            from the i-th index and ending at the (i+msc)-th index;

            store the created subset in the vector allSubsets;
    }
return allSubsets;
}
```

Figure 10. CreateThesubsets Procedure

The createTheSubsets procedure is based on the fact that the cvd between any two items members of the pair P will increase if we consider substituting some element of the distance array that is further apart from either the lower or the upper bound of the current member elements of the pair. For example if we consider the array subset {190,260} or {240, 360} instead of the subset {190,240}, they will have the bigger deviation and cvd as well. This observation is based on the fact that the standard deviation increases with the increase of the distance among the items. If we have distances between a pair of items measured in n regions, our approach needs to test only (n-msc+1) subsets to check if there is a cvd lower than the maximal one.

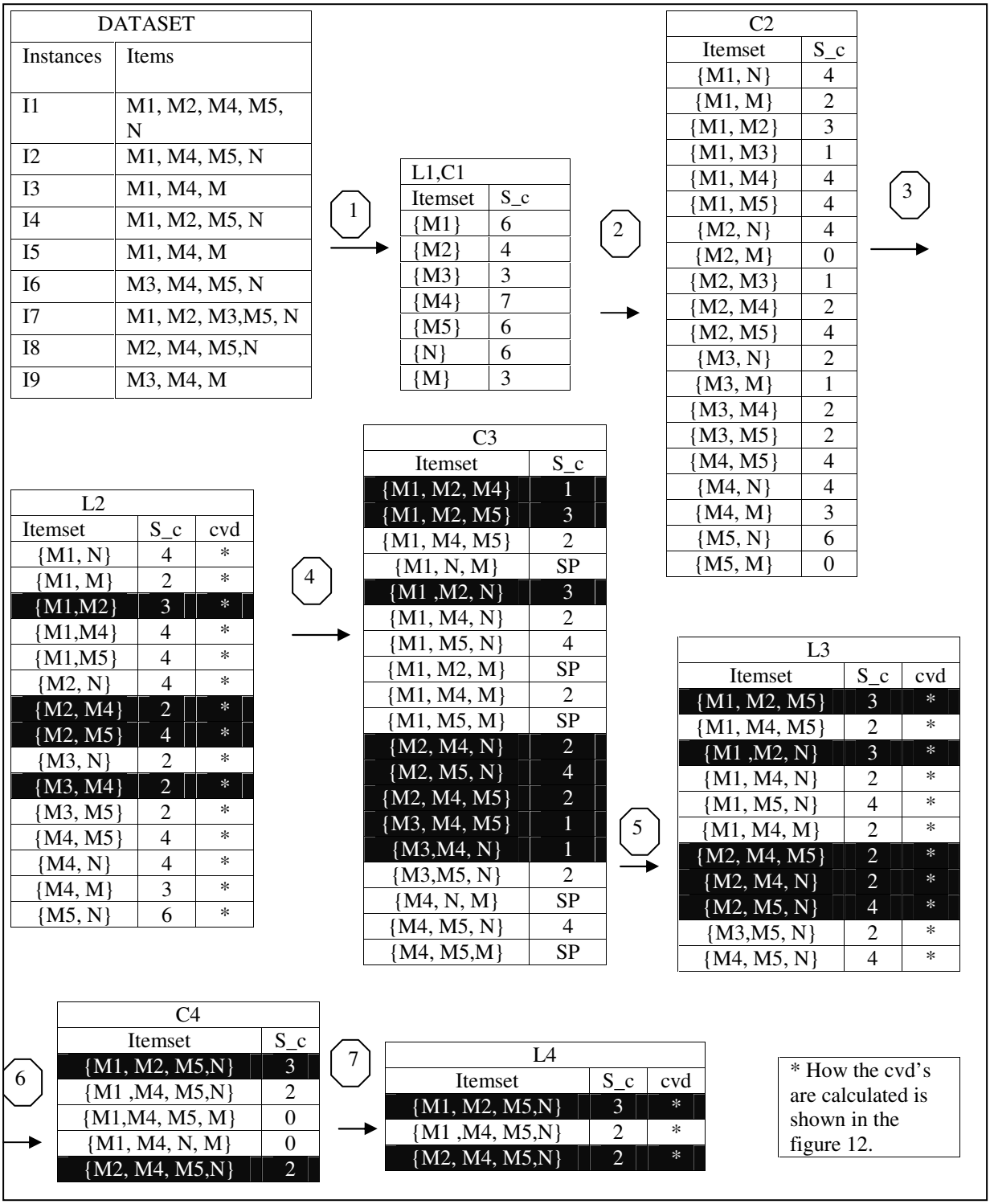


Figure 11. Deviates_properly procedure prunes the itemsets that are not DPF itemsets. Minimal support count (msc) is 2, and confidence is 100%. Subset pruned (SP) Support count (S_c).

This is a linear time O(n) procedure that will provide the information if the itemset is a DPF itemset. Figure 11 illustrates how our procedure for testing if an itemset is DPF changes the original Apriori algorithm's flow.

The notation in Figure 11 refers to our application domain: M1-M5 are the motifs found in the 9 promoter regions given in Figure 2. Item M denotes the expression in the muscle cells, while N denotes expression in the neural cells. The procedure deviates properly will test only pairs created from the itemset that contain the motifs. So the pair {M3, N} cannot be tested since there is not a notion of distance between the item that represents the Motif M3 and the item N that represents the expression Neural.

Using our approach on the Apriori example given in Figure 5 will lead to the process of pruning some of the itemsets because they are not DPF. In Figure 11 the itemsets that will be pruned away because they are not DPF itemsets (given the maxcvd is 0.15) are marked with black. From this figure we can notice that for example the itemset I={M1, M4} will not be pruned because we have shown above that this itemset deviates properly. Illustrative example for the itemset that will be pruned is given in the Figure 12. Figure 12 shows the pruning of the itemset {M2, M5} that belongs to the set of frequent itemsets L2 from Figure 11. The itemset {M2, M5} is present in four promoter regions PR1,PR4,PR7 and PR8 (given in Figure 2). The minimal support count for this example is 2 and the maxcvd is 0.15. The Combinations_deviate_properly procedure will create three subsets from the distances between M1 and M5. These distances are taken from the four promoter regions where M1 and M5 occur together and

they are 36, 150, 210, and 350. The subsets of cardinality equal to the msc are {36, 150}, {150,210} and {210,350}.

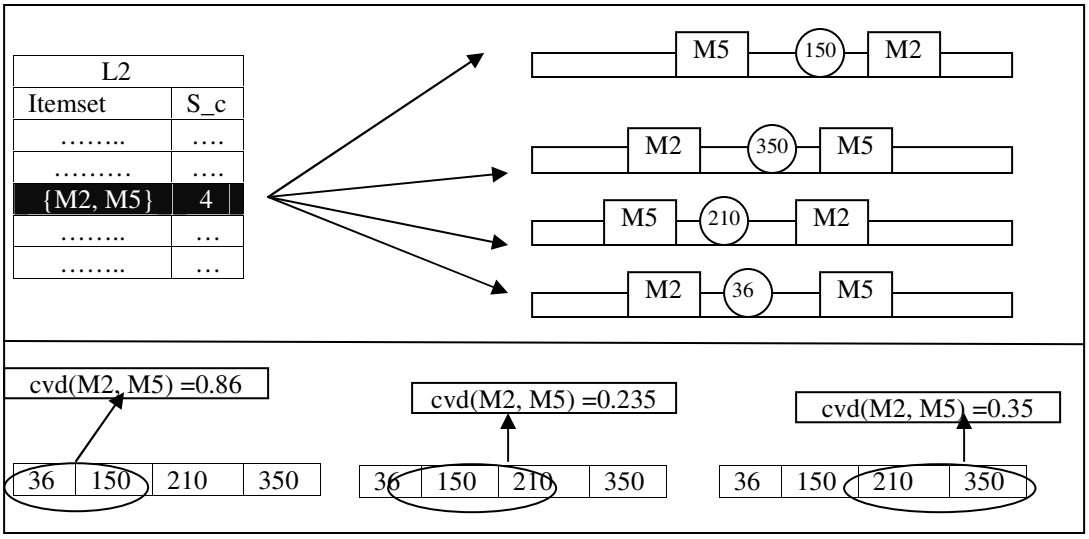


Figure 12. The itemset {M2,M5} does not deviate properly for maxcvd=0.15

The subsets of cardinality equal to the msc are {36, 150}, {150,210} and {210,350}. The cvd's calculated from these pairs, are (0.86), (0.235) and (0.35) respectively. Since none of them is less than the maxcvd (0.15) the itemset {M2,M5} does not deviate properly. All other itemsets marked with black in Figure 11 are pruned following the same course of action as we followed for the itemset {M2, M5}.

Our Distance Association Rule Mining (DARM) produces rules from *distance preserving frequent (DPF) itemsets* in same manner as regular Apriori produces rules from frequent itemsets. What we know for the DPF itemsets is that they are DPF when calculated from a number of instances equal to the minimal support count. If the support of the rule to be produced is larger than the minimal support count, we check again if the itemset is DPF,

but now calculated from all instances that support the rule. If the itemset is DPF, we record the statistics (each pairwise mean, standard deviation and cvd) and the rule is produced. For example, DARM will try to produce rules from the DPF {M1, M5, N} taken from set of frequent itemsets L3 from Figure 11. If the cvd for the pair {M1,M5} calculated from all the instances that contain the itemset {M1,M5,N} (in this case four) is below the cvd threshold, the rules will be produced.

So, our approach enhances the main Apriori property; now all nonempty subsets of a DPF itemset must also be DPF itemsets. The extension of this property and the encapsulation of the deviates_properly procedure in the Apriori algorithm build the skeleton of DARM and introduce a significant improvement, in terms of the frequent itemsets generated, over the approach not to use the deviates properly procedure and to mine distance-based rules from all frequent itemsets that the standard Apriori would produce.

3.2 Model Construction

We build our classification models over a training set of data. Our classification models consist of class association rules. Given a test data instance, a class association rule will classify it correctly if the antecedent of the rule is present in the test instance and the class predicted by the rule is the same as the class of the test instance.

Two different classification models are used for the experimental evaluation of this thesis work. The first one is called the All Rules classifier. This classifier consists of all class

association rules (rules that have the item that denotes the gene expression in the consequent of the rule) produced by DARM. The second follows the CBA model construction approach described in [LHM98]. The CBA model construction first sorts the rules by confidence, then by support. Then the association rules that classify correctly at least one instance from the training data are selected. Rules are added to the model one at a time in the order in which they occur after sorting them. Initially the first rule is included in the model. The resulting classifier is tested on the training instances for the error rate (the ratio of incorrect predictions over the training data).This process is repeated until exhausting the association rules or exhausting the training instances. The subset of the rules with lowest error rate is the final CBA model. This CBA model contains a default rule that is applied to test instances for which none of the other rules in the model apply. The default class is the majority class of the unclassified training instances. See [Pal03] for further details.

3.3 Implementation

The implementation of the DARM algorithm was done in the **W**aikato **E**nvironment for **K**nowledge **A**nalysis (**WEKA**) [FW99]. WEKA is an open source data mining and machine learning system from the University of Waikato, New Zealand. The DARM algorithm presented in this thesis has been developed with careful consideration of its feasibility within the WEKA environment. In the past several years the WEKA system has been improved by the students in the Knowledge Discovery in Databases

research group at WPI(KDDRG) [Pal03], [Pra03], [Sto02] and [Sho01]. This system is now called WPI-WEKA. DARM is now part of the WPI-WEKA system.

The sequence diagram of the DARM's main procedures with the present WPI-WEKA modules is given in the Figure 13. DARM's modules are rounded by thick frames. They are invoked by the existing module called ARMinerApriori. This module is responsible for the level-wise generation of the frequent itemsets. On each level before claiming the itemset as frequent the ARMinerApriori invokes the procedure deviates_properly in order to check if the itemset is a DPF itemset. The procedure deviates_properly interacts with the procedures Combinations_deviate_properly and createTheSubsets in the same manner as explained in Section 3.1.. If the itemset does not deviate properly, it is pruned from future consideration by the ARMinerApriori module.

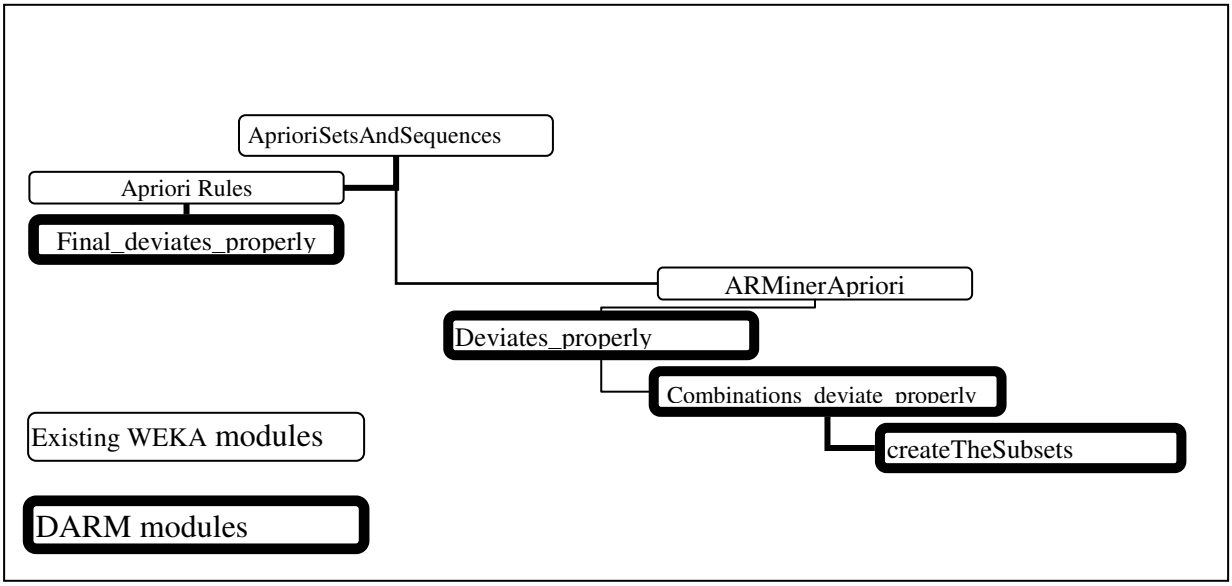


Figure 13. DARM's interaction with the WEKA-WPI modules for mining frequent itemsets

Once all the frequent itemsets have been determined by the ARMinerApriori, AprioriSetsAndSequences model calls the AprioriRules module to generate the

association rules satisfying the minimal confidence condition (Figure 13). AprioriRules sends the rules that satisfy the minimal confidence condition and have support bigger than the minimal support count to the DARM's module Final_deviates_properly. This module tests if the itemset that builds the rule is a DPF, but now calculated from all instances that support the rule. If the itemset is not a DPF this rule is pruned.

After the rules are produced they are outputted by the WPI-WEKA system (Figure 14). Rules contain all the distance statistics: means, standard deviations and cvd's for each pair of items members of the rule, presented in a format similar to the rules given in Figure 7.

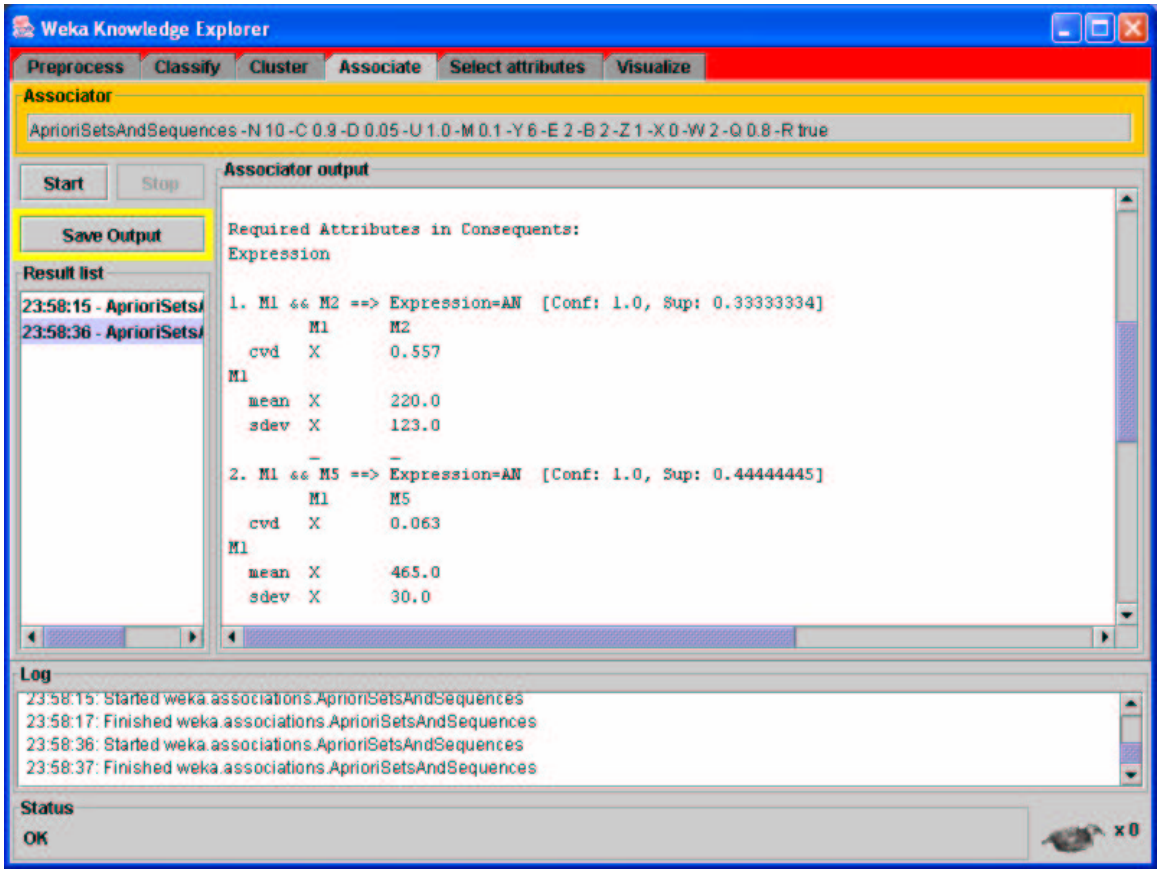


Figure 14. DARM Interface

4 Experimental Evaluation

In this section, we provide the results from testing our DARM system. First, the data are described, and the dataset construction process is explained. Then the evaluation metrics are defined. In the experimental protocol two phases of the experimental evaluation are described. The first phase shows the savings of the DARM during the process of mining association rules. The second phase shows the testing of the classification models consisting of DARM rules.

4.1 Data Description

We use two sets of data for our experiments. The first one contains genetic data for C.Briggsae and the second one contains data from C.Elegans. C.Elegans and C.Briggsae are soil nematodes. DNA of those organisms are completely sequenced and transformed in computer readable formats. Many of the genes of those organisms have been determined. These facts have made C.Elegans and C.Briggsae genomes the subject of many computational biology analysis and experiments.

The C.Briggsae data that we used for our experiments contained the promoter regions of 31 genes and the cell types where the genes are expressed. There are a total of five cell types in the dataset. The C.Elegans data that we used for our experiments contained the promoter regions of 57 genes and one cell type where the gene is expressed.

In order to obtain the motifs responsible for gene expression for one cell type, first MEME was run over the promoter sequences associated with the genes expressed in that cell type. MEME is run with an option to find multiple occurrences of each motif in a single promoter region. After the motifs are identified they are given as input to MAST, together with all the promoter regions. MAST annotates those promoter regions with the occurrences of the given motifs. The file outputted by MAST is used an input to the MASTToARFF module written for this thesis that takes the MAST file and transforms into the ARFF format. If there is more than one occurrence of a motif in a promoter region, our module selects the occurrence of the motif with the lowest p-value, i.e. the most significant occurrence of the motif according to MAST.

The ARFF format includes a data row for each gene, the cell expression of the gene, the motifs present in the promoter region of the gene, and the location of the most significant occurrence of each motif in terms of the starting and ending point of the motif (counted from the start of the gene). Appendix A contains the ARFF files used for our experiments. There are five ARFF files for C.Briggsae data, one for each cell type as described below. Appendix A also contains the ARFF file used for the experiments with C.Elegans data, although only the header of the ARFF file and a few illustrative data instances were included to save space.

The five cell types used were PanNeural, ASENeural, ASKNeural, OLLNeural, and BodyWall. PanNeural means that all neural cells express the gene; promoters in this class are thus a subset of those expressed in the particular neural cells ASE, ASK, and OLL. As described above, each dataset contains 31 promoter regions. MEME, MAST and

MASTToARFF are used to find the motifs for each cell type separately. The number of motifs and the number of genes expressed in each of the five data files are given in Figure 15.

C.Briggsae	Number of motifs	Number of Genes Expressed	Percentage of Genes Expressed
PanNeural	25	17	(17/31)*100=54%
ASENeural	25	21	(21/31)*100=67%
ASKNeural	28	24	(24/31)*100=77%
OLLNeural	24	19	(19/31)*100=61%
BodyWall	27	20	(20/31)*100=64%

Figure 15. C.Briggsae data statistics

Also the dataset from the PanNeural cell type from *C.Elegans* is used. This dataset is obtained using the same methodology as for *C.Briggsae*. The number of promoter regions for this dataset is 57. The statistics for this dataset are given in Figure 16.

C.Elegans	Number of motifs	Number of Genes Expressed	Percentage of Genes Expressed
PanNeural	28	17	(17/51)*100=29%

Figure 16. C.Elegans data statistics

4.2 Evaluation Metrics

We evaluate our classification models in terms of their accuracy. Accuracy is the proportion of correct predictions over the total number of predictions made.

A typical example of how to use a rule to predict the expression of a novel gene can be shown if we test Rule 2 from Figure 7. If the gene's promoter contains motifs M1, M4, and M5 and distances among them are within plus or minus one standard deviation from the respective pairwise means recorded in rule R2, it will be predicted that the gene

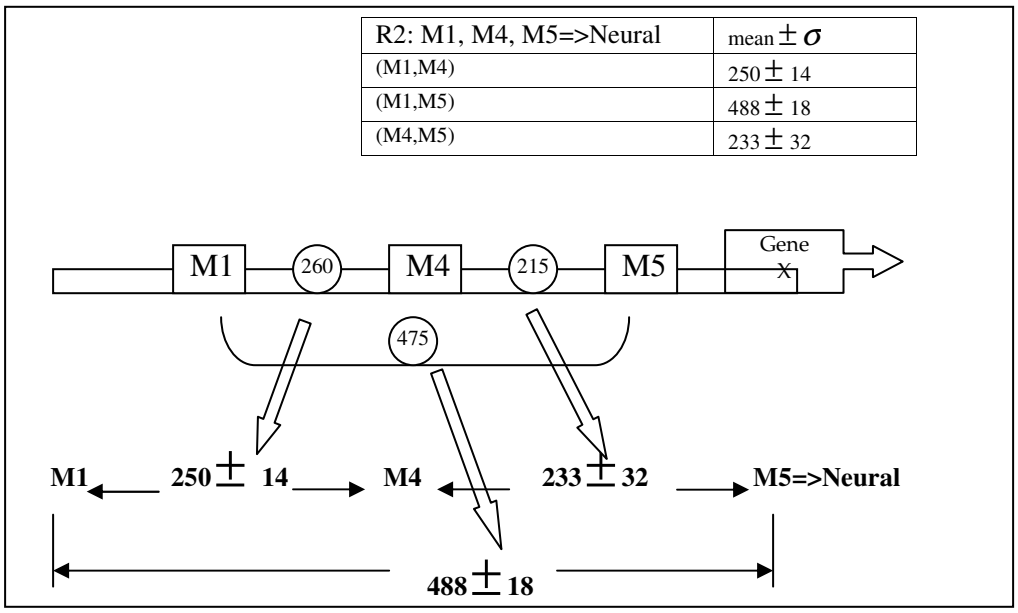


Figure 17. Rule used for prediction

will be expressed in neural cells. Figure 19 shows an example of how this rule is applied. Assume that a novel gene is given and that motifs M1, M4 and M5 are present in the promoter region of the gene. Assume also that the distances between M1 and M4, M1 and M5, and M4 and M5 are 260,475 and 215 respectively. Since those distances lie between the corresponding mean plus/minus one standard deviation, then the rule applies to this new gene and predicts that the gene is expressed in neural cells.

4.3 Experimental Results and Analysis

Our experimental protocol is divided into four phases. The first phase is to test the savings of the DARM during the process of mining association rules. The second phase is to test the classification accuracy of the models consisting of the DARM rules. The third

phase is to visualize the top rules in each of those modes. The forth and final phase is to compare the accuracies of models consisting of DARM rules against the accuracies of the models consisting of regular rules.

4.3.1 Comparison of Frequent Itemsets vs. DPF Itemsets

In order to assess the advantages of our enhancement of the Apriori algorithm, we compared this enhancement against a naïve approach to obtain distance-based association rules using regular Apriori. In this naïve approach, regular Apriori is used to generate frequent itemsets and rules from them. Then each rule is annotated with the cvd values of all the pairs of items in the rule, and finally only those rules that satisfy the maximal cvd threshold are kept. Both our enhanced Apriori and this naïve approach output the same set of rules.

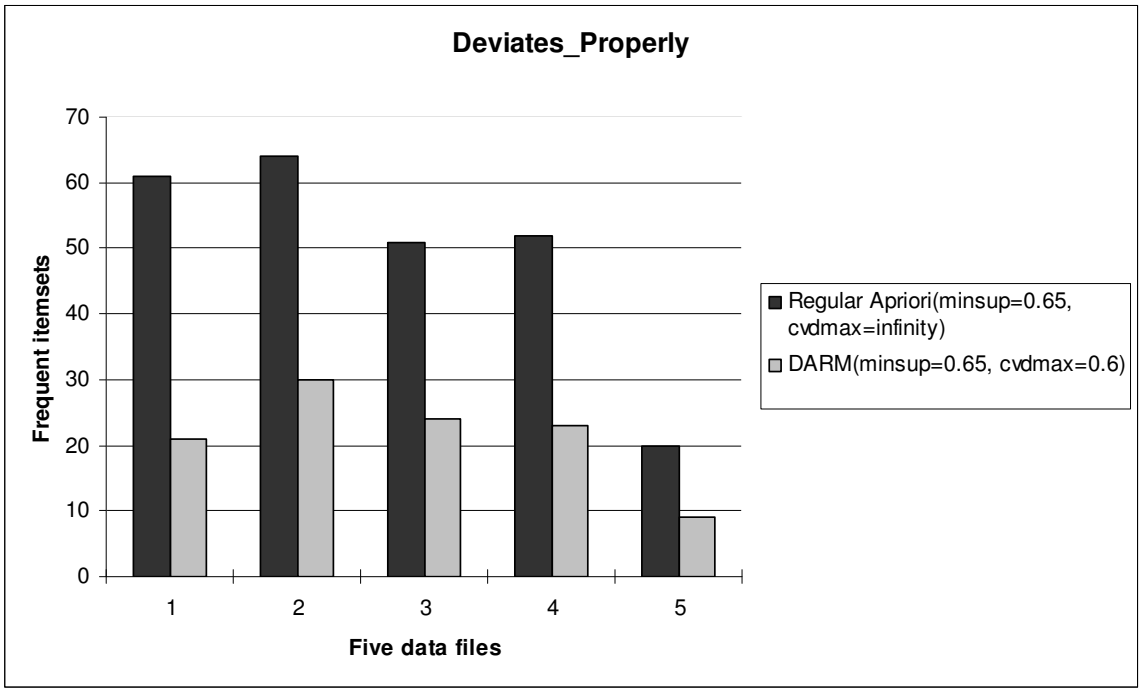


Figure 18. DARM Savings

However the naïve approach would consider many more unnecessary itemsets than our DARM approach would. Figure 18 summarizes the number of itemsets considered by both methods over the five C.Briggsae datasets. It is expected that the decrease of the number of the frequent itemsets yields savings during the mining process.

4.3.2 Distance-Based Models

The second phase of our experiments is to gauge the influence of the introduction of the cvd parameter in the process of mining association rules. In particular we observe the behavior of the mining algorithm and the accuracy of the resulting models as a function of the coefficient of the variation of distances. For the experiments shown in Figure 19, the same datasets are used for the model building and for the model testing. For the experimental results shown in Figure 20, 66% of the data was used for the model building and the rest of the data for the model testing. For both of the experimental approaches, for each of the five cell datasets, five CBA models are constructed and their accuracies are presented as well. CBA models contain only the rules that have cvd's less than 0.5. Comparison of those results can be made with experiments with the same dataset with ZeroR classifer (classifer that always predicts the majority class of the training instances). This classifier will give the results same as the percentages for gene expression in Figure 15 (since the gene expression is the majority class in all five cell types). For the experiments given in Figure 19 we can conclude that for the all five cell type obtained higher classification accuracy that of ZeroR.

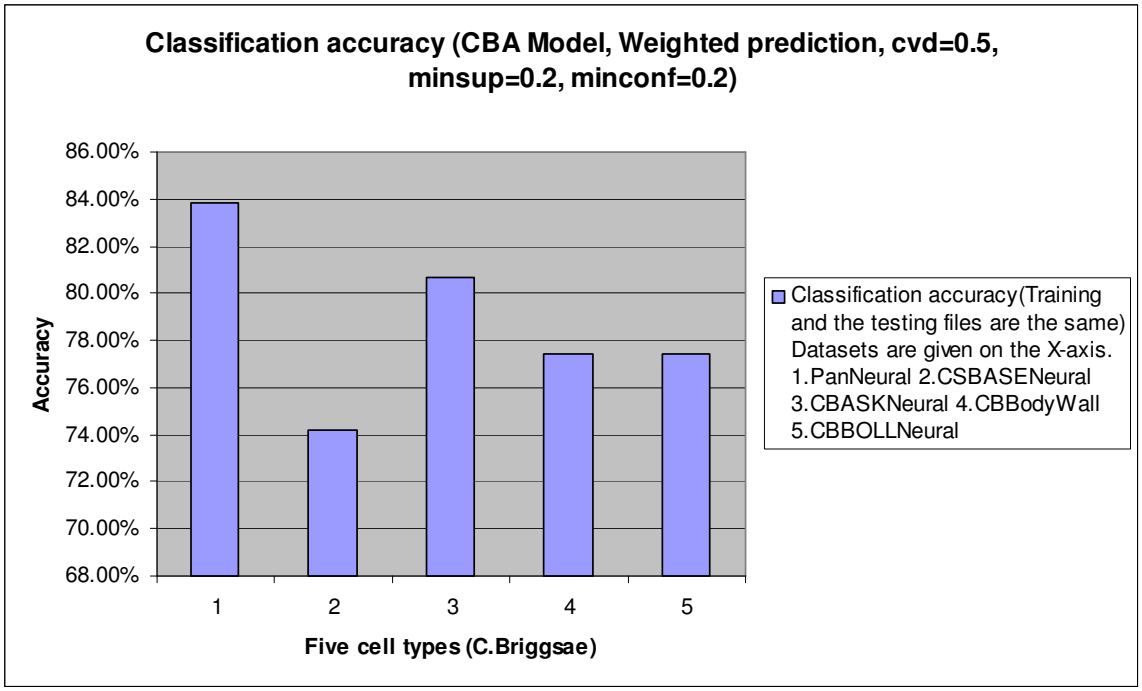


Figure 19. Classification accuracy (CBA models on C.Briggsae)

For the experiments given in Figure 20 we can conclude that the PanNeural and CBBodyWall models obtained higher classification accuracy, while the rest three obtained lower classification accuracy that of ZeroR..

Next, using the same experimental settings stratified 10-fold cross validation was performed over the five cell types from *C.Briggsae*. Tenfold cross-validation divides the data in ten parts, having the class attribute (cell expression in our case) distributed in each fold following the same distribution of the class attribute in the full dataset. The training model is constructed on nine folds, and testing is performed on the tenth fold. This process is repeated 10 times, having each of the folds as testing set. The overall classification accuracy for the 10 folds is obtained by averaging the accuracies obtained from the individual folds.

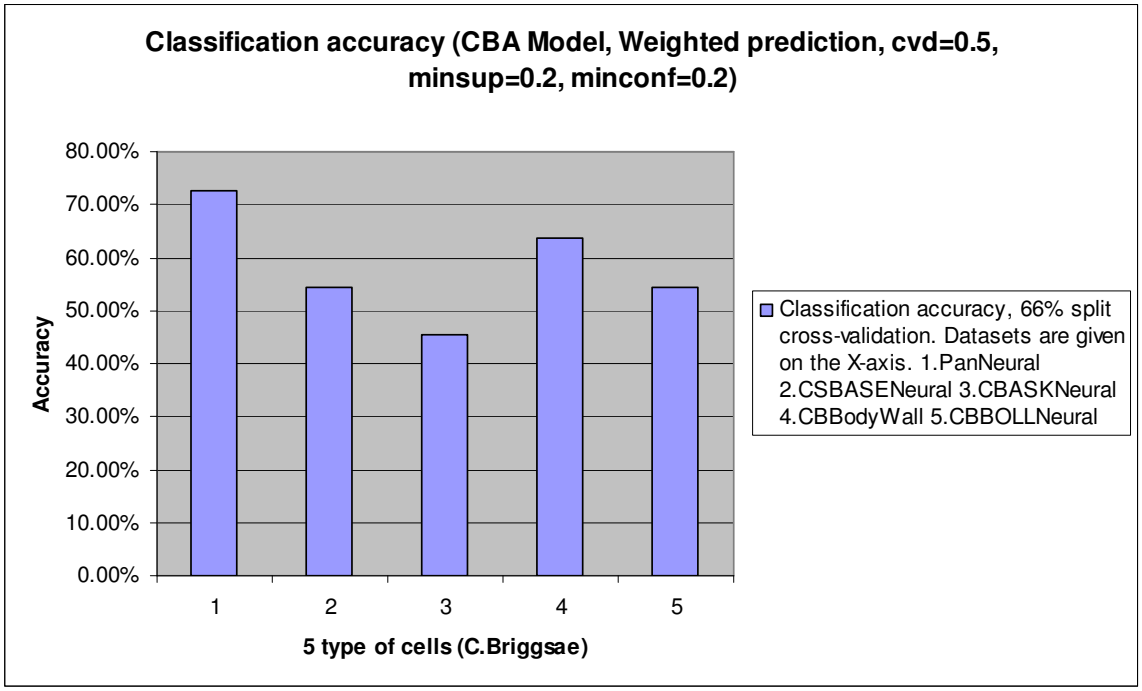


Figure 20. Cross-validation using 66% split of the datasets

For the 10-fold cross validation given in Figure 21 we can also compare the results with ZeroR and conclude that CBPanNeural, CBASKNeural, CBBodyWall and CBBollNeural models obtained higher classification accuracy than ZeroR.

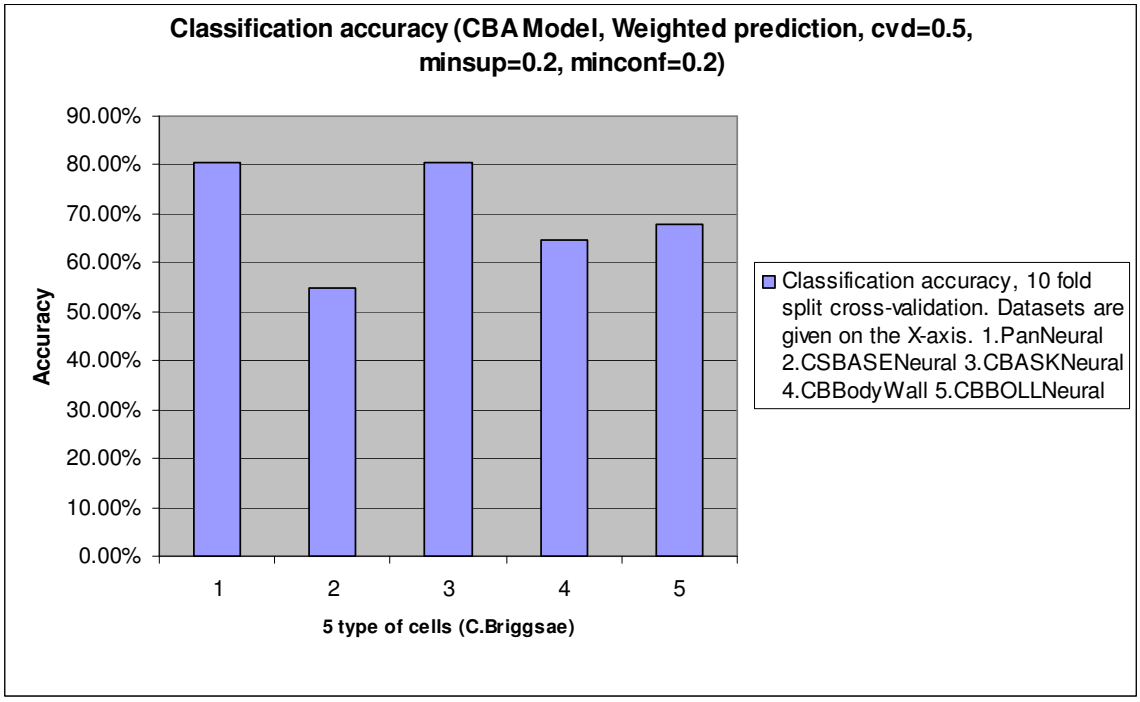
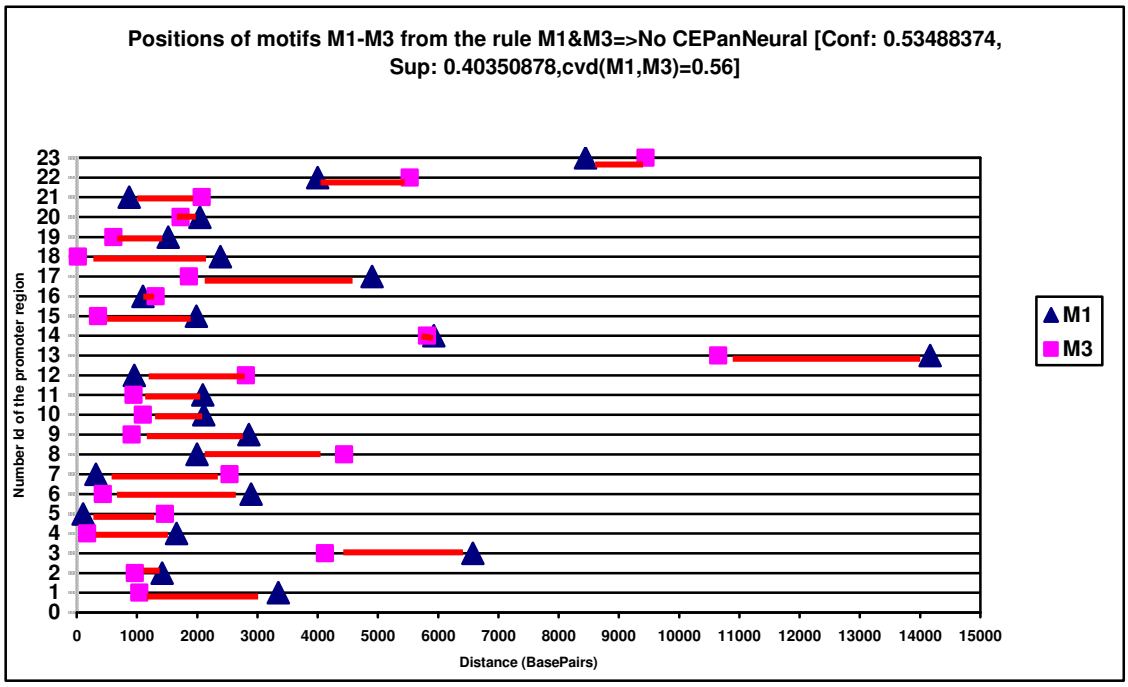


Figure 21. Cross-validation using 10 fold cross-validation

4.3.3 Visualizations of the Distance-Based Models

The best rules (sorted by confidence then support) for each of the CBA models over C.Briggsae and C.Elegans are visualized in Figures 22-27. From those visualizations we can see that the bigger the cvd, the bigger the deviations between the motif distances. The rule M9&M16=>CBASENeural in Figure 23, has the smallest cvd from all the rules given in these visualizations. So if we compare the variation of the distances between motifs from this rule and all the other rules we can visually notice that this variation is much smaller for this rule than for the others.



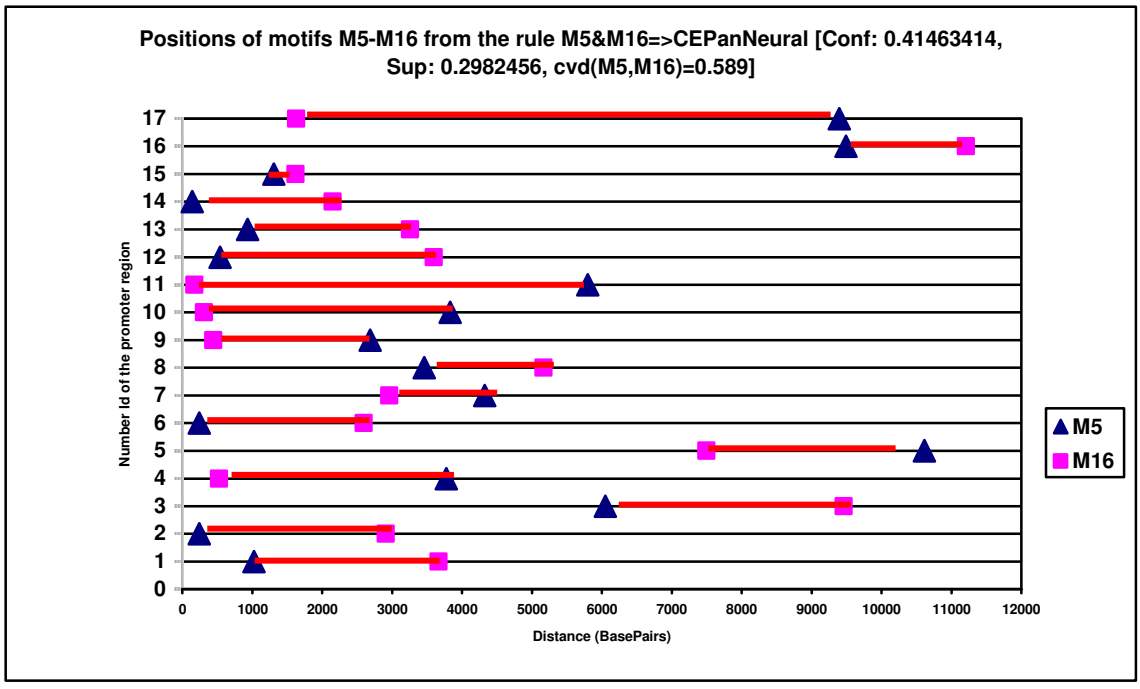


Figure 22. Visualizations of the first and second top rules from the CBA-C.Elegans-PanNerual cell model.

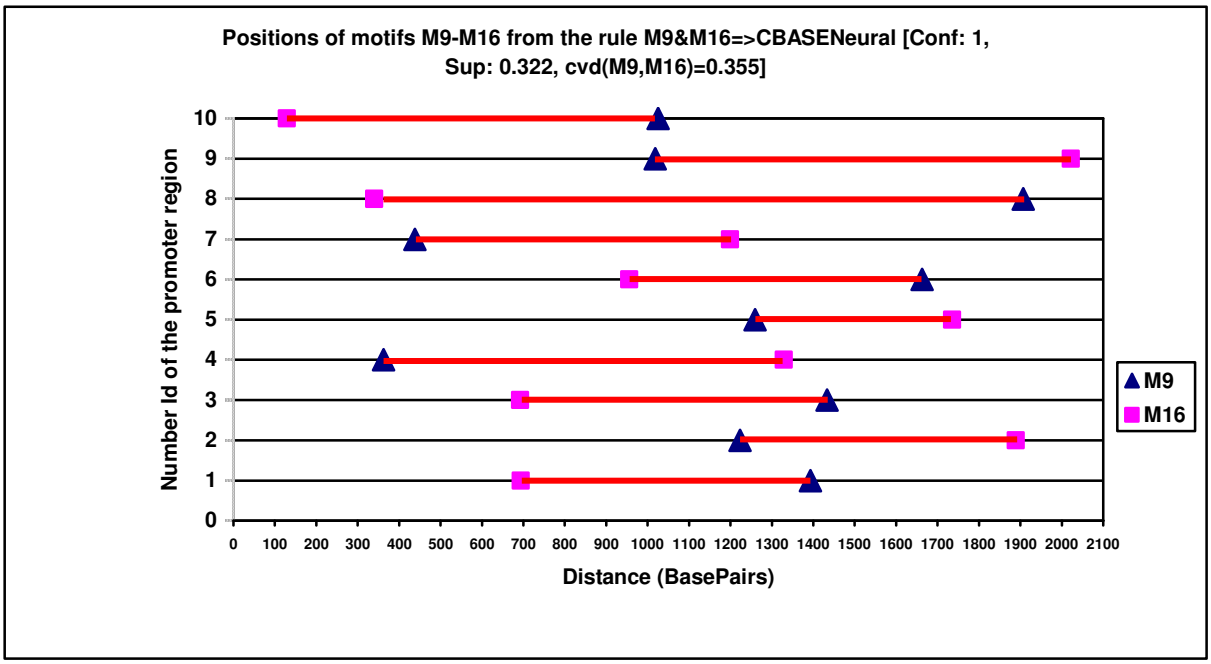


Figure 23. Visualization of the top rule from the CBA-C.Briggsae-ASENeural cell model .

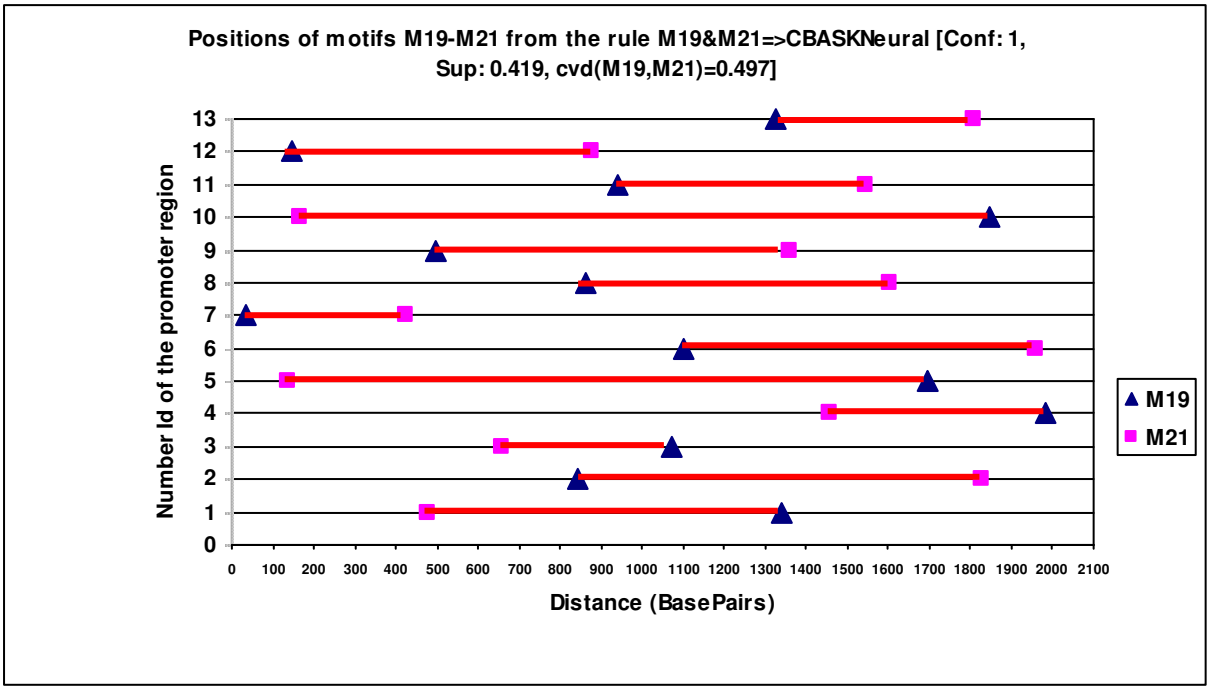


Figure 24. Visualization of the top rule from the CBA-C.Briggsae-ASKNeural cell model.

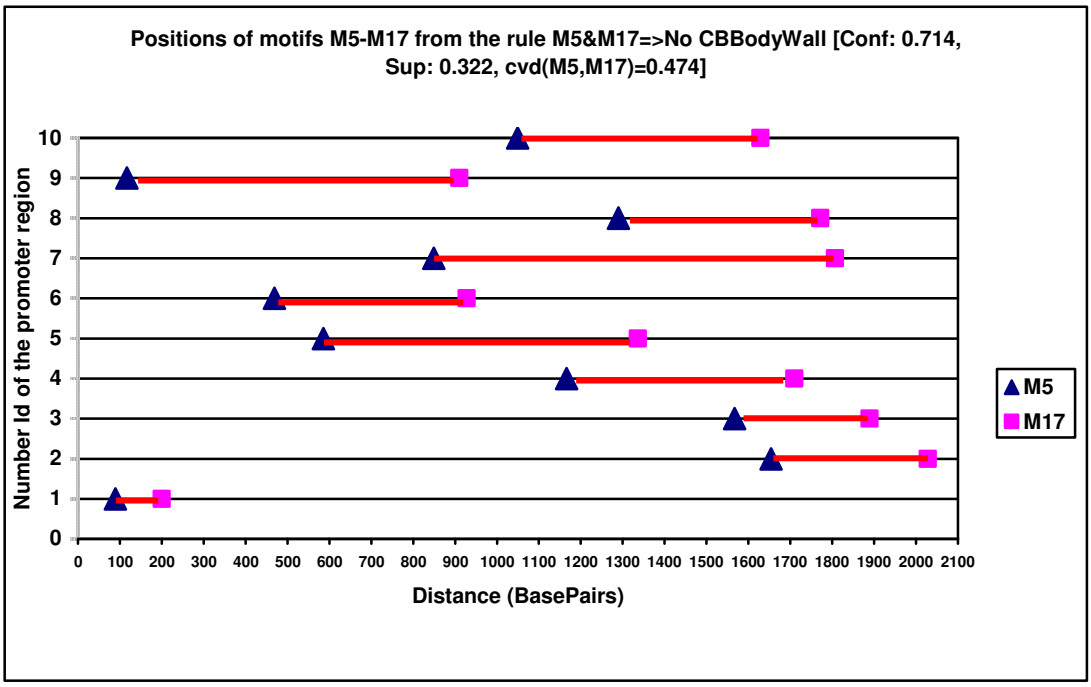


Figure 25. Visualization of the top rule from the CBA-C.Briggsae-BodyWall cell model.

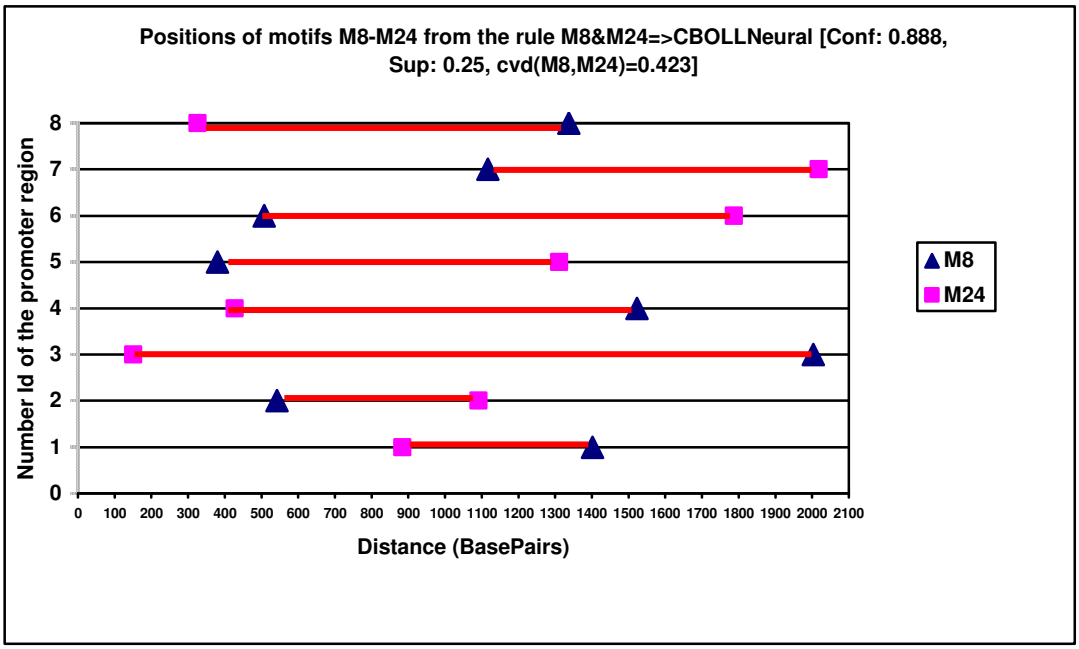


Figure 26. Visualization of the top rule from the CBA-C.Briggsae-OLLNeural cell model.

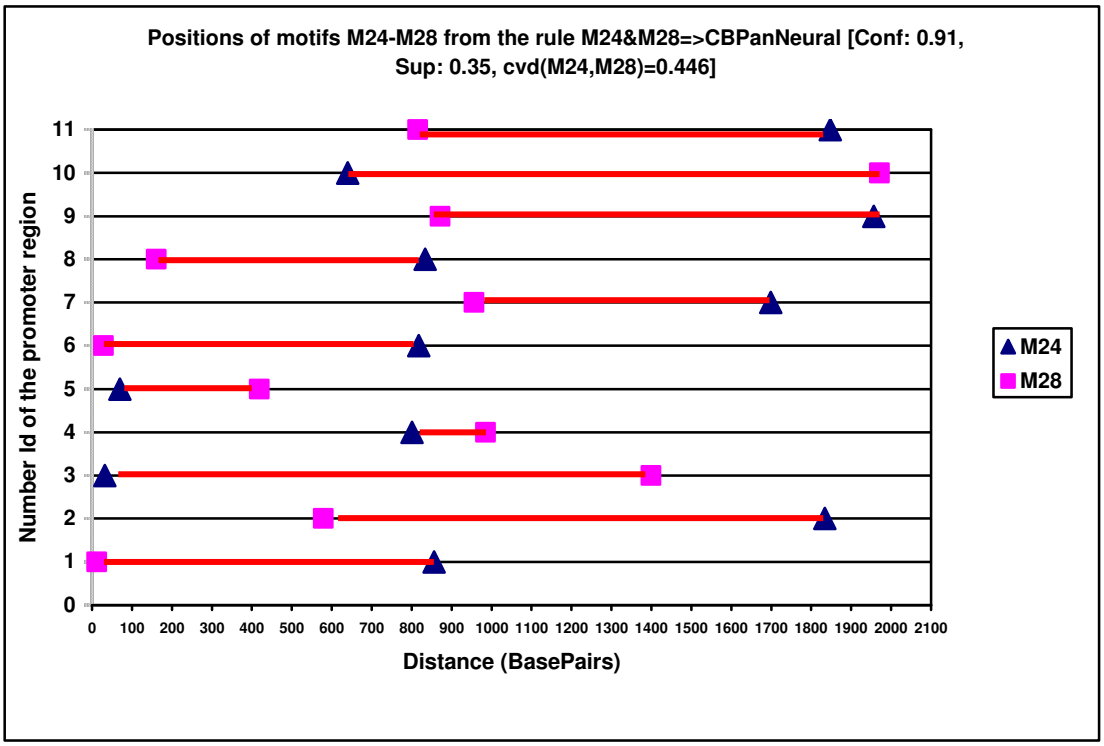


Figure 27. Visualization of the top rule from the CBA-C.Briggsae-PanNeural cell model.

4.3.4 Comparison of the DARM Model vs. Regular Models

One of the goals of this thesis work was to compare the classification models obtained from classification rules with and without the encapsulated notion of distance. For this experimental evaluation, models built over the PanNeural dataset were used, since this dataset contains near 60%-40% distribution of the class values. Classification accuracy for both CBA models is shown in Figure 28. for C.Briggsae, and in Figure 29 for C.Elegans.

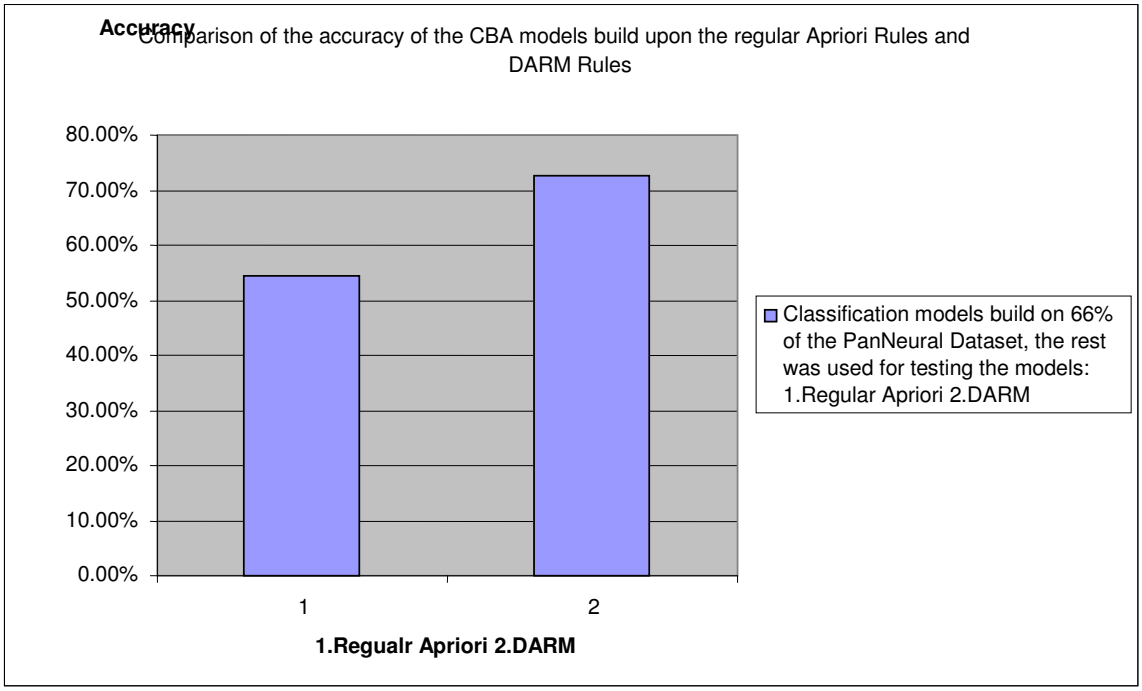


Figure 28. Comparison of the distance-based and regular models (C.Briggsae)

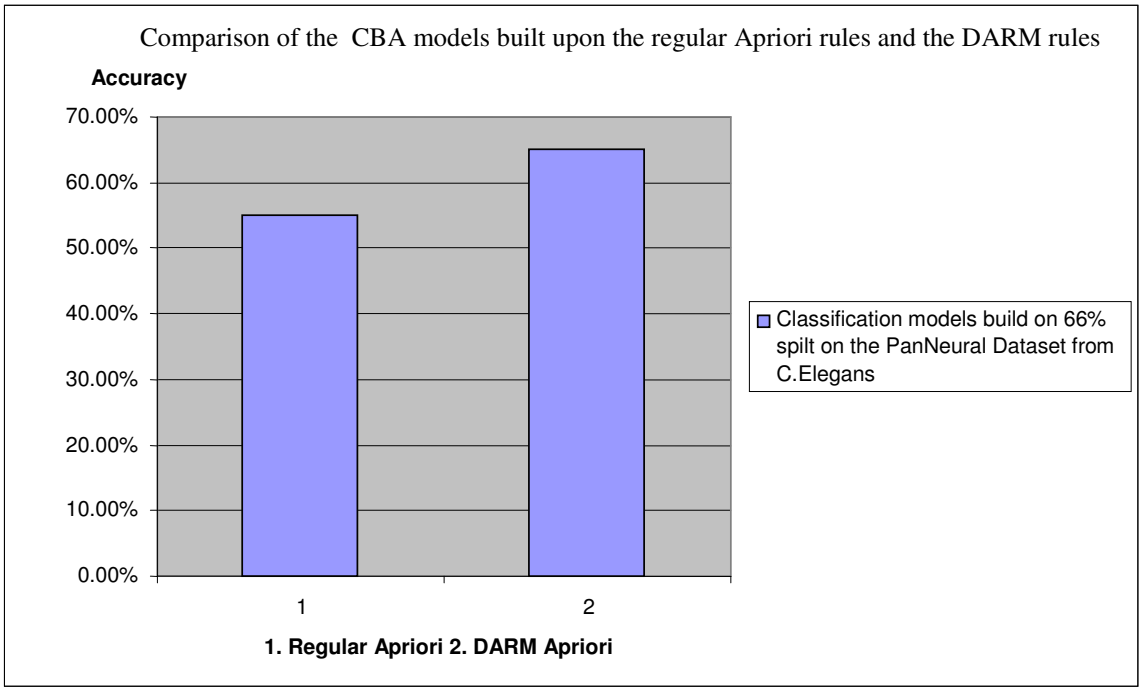


Figure 29. Comparison of the distance-based models and regular models (C. Elegans)

In addition to the experiments run with the CBA model, 10-fold cross validation was performed with the All Rules model (Figure 30). This classifier consists of all class association rules (rules that have the item that denotes the gene expression in the consequent of the rule) produced by DARM. This classifier does not have a default rule and hence instances to which none of the rules apply remain unclassified. The accuracy of the model is calculated only from the test instances that can be classified. Rules that are part of the AllRules models obtained with same parameter settings as the experiments in Figure 30 are given in Appendix B.

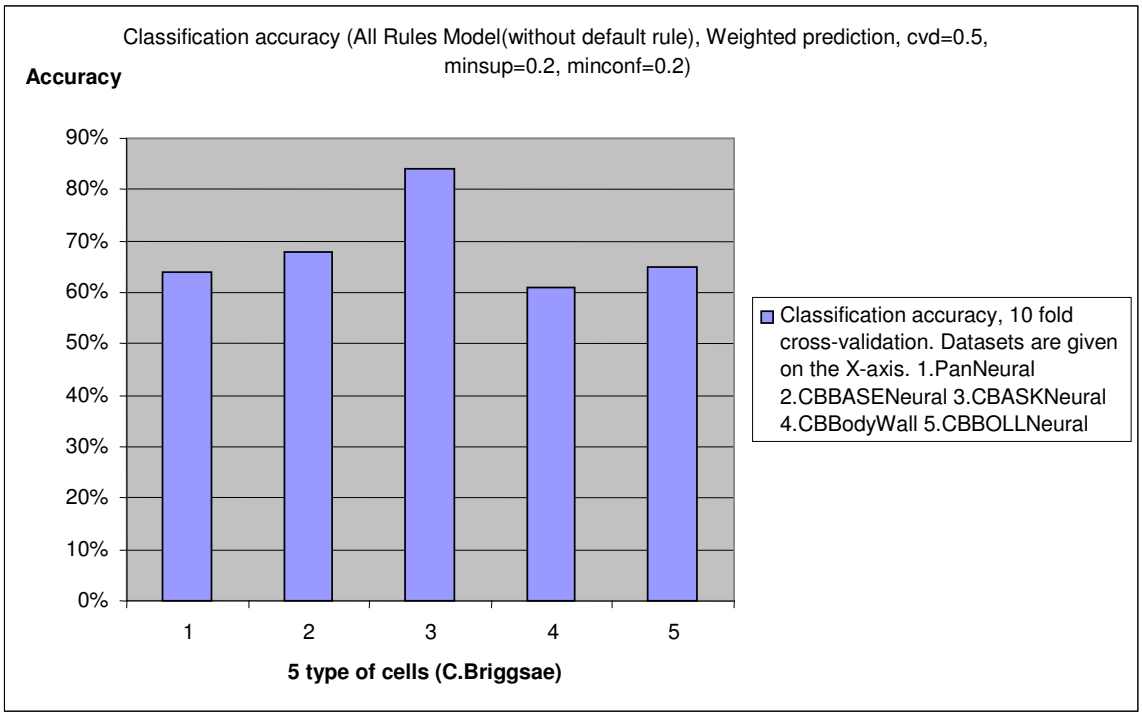


Figure 30. All Rules model (C. Briggsae)

In these experiments we used weighted prediction. If for a given test instance several rules in the model apply and make different predictions, then each predicted value is

weighted by the sum of the confidences of those rules that predict that value. The predicted value with the highest weight is chosen as the model's prediction for the test instance.

5 Conclusions and Future Work

The goal of this thesis was to design, implement, and evaluate an algorithm for mining distance-based association rules. We have accomplished this goal by extending and enhancing the Apriori algorithm. The Apriori algorithm is not able to mine association rules that contain distance information among the items that construct the rules. Our distance-based association rule mining algorithm (DARM) is able to use distance information during the mining process and is able to construct rules that make explicit both co-occurrences of items and distance-preservation patterns in the data.

The DARM implementation provided in this thesis produced significant savings over the regular Apriori process of mining frequent itemsets. DARM rules are capable of mining more significant patterns from the application domain data than the standard Apriori. Classification models built with DARM rules were shown to have better classification accuracy than the standard Apriori models.

This thesis presented an application of distance-based association rule mining to the area of gene expression. However the functionality of the DARM algorithm is independent from the application domain. The DARM algorithm developed for this thesis can be used for data mining and knowledge discovery from genetic, financial, retail, time sequence data, or any domain where the distance information between items is of importance. In order to manipulate data from other domains, the only requirement is that data instances contain numeric distances between the occurring items.

Our DARM algorithm restricts the number of occurrences of an item in a data instance to at most one. Future work on the DARM algorithm should include the ability to handle multiple occurrences of an item in a data instance and hence multiple distances between two items in an instance. This restriction was imposed in order to have a well-defined notion of distance between pairs of items. This future extension should yield a certain advantage in the significance and accuracies of the resulting rules in application domains where multiple occurrences of the items are of importance.

6 References

[AS94] R.Agrawal and R.Srikant. Fast Algorithms for Mining Association Rules. *In Proc. of the 20-th Very Large DataBases (VLDB) Conference*, pages 487-499, Santiago, Chile, 1994.

[AY98] C.C. Agrawal and P.S.Yu, A New Framework for Itemset Generation. Symposium on Principles of Database Systems(PODS) *'98, Seattle, WA*, pages 18-24, 1998.

[BE95] T.Bailey and C.Elkan. Unsupervised Learning of Multiple Motifs in Biopolymers using Expectation and Maximization. *Machine Learning Journal,21,* pages 51-83, *1995.*

[BG98] T.Bailey and M.Gribskov, "Combining Evidence using p-values: Application to Sequence Homology Searches", Bioinformatics, 14(48-54), 1998.

[BLT02] K. Blitsch, B.Lucas, S.Towey, Computational Analysis of Gene Expression, Undergraduate Graduation Project (MQP), WPI, 2002.

[BMS97] S.Brin, R.Motwani, and C.Silverstein. Beyond Market Baskets: Generalizing Association Rules to Correlations. *Conference of the Special Interest Group on Management of Data (SIGMOD),* pages *255-264, 1997.*

[Dun02] M.Dunham. Data Mining-Introductory and Advanced Topics. Prentice Hall, New Jersey, pages 233-235, 2003.

[FSP96] U.Fayyad, G.P. Shapiro and P.Smyth. From Data Mining to Knowledge Discovery in Databases., American Association for Artificial Intelligence, AI-Magazine, pages 37-54, 1996.

[FW99] E.Frank, I.Witten. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, 1999.

[HK01] J.Han, M.Kamber. Data Mining: Concepts and Techniques. *Morgan Kaufman Publishers, San Francisco,* pages 230-236, *2001.*

[LHM98] B.Liu, W.Hsu, Y.Ma. Integrating Classification and Association Rule Mining. Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining , pages 80-86, New York, 1998.

[MPPT01] B.Murphy, D.Phu, I.Pushee, F.Tan. Motif-And Expression-Based Classification of DNA., Undergraduate Graduation Project (MQP), WPI, 2001.

[MY97] R. Miller and Y. Yang. Association Rules Over Interval Data. In *ACM SIGMOD Intl. Conf. Management of Data*, pages 452-461, 1997.

[Pal03] S. K. Palanisamy. Classification and Adaptive Minimal Support Association Rule Mining. MS Thesis. Department of Computer Science, Worcester Polytechnic Institute. In preparation.

[Pra03] K.A. Pray. Mining Association Rules from Time Sequence Attributes. MS Thesis. Department of Computer Science, Worcester Polytechnic Institute. In preparation.

[PBCB99] A.G.Pedersen, P.Baldi, Y.Chauvin. S.Brunak. The Biology of Eukaryotic Promoter Prediction - A Review. *Computers&Chemistry 23,* pages 191-207, 1999.

[SBM98] C.Silverstein, S.Brin and R.Motwani. Beyond Market Baskets: Generalizing Association Rules to Dependence Rules. *Data Mining and Knowledge Discovery,2,* pages *39-68,1998.*

[Sho01] C. A. Shoemaker. Mining Association Rules from Set-Valued Data. MS Thesis. Department of Computer Science, Worcester Polytechnic Institute. May 2001.

[Sto02] Z. Stoecker-Sylvia. Merging the Association Rule Mining Modules of the Weka and ARMiner Data Mining Systems. Major Qualifying Project. Worcester Polytechnic Institute. April 2002.

[Whi01] R.White. Gene Transcription, Mechanisms and Control, *Blackwell science, 2001*.

[Zar99] J.Zar. Biostatistical Analysis. *Fourth Edition, Prentice hall*., page 40, 1999.

APPENDIX A -ARFF files for CBriggsae and C.Elegans

1.CBASENeural

```
%       Contains the following promoter regions (they appear as data instances in  order
%       of appearance from the first to the last data instance)
%
%      Expressed in
%
%       gsa-1 8.5e-40 2010
%       Snt-1 7.8e-33 2040
%       F25B3.3 1.4e-31 2039
%       unc-119 2e-28 2040
%       osm-3 5e-24 1983
%       unc-115 9.8e-24 2024
%       rgs-1 2e-22 2040
%       tax-4 2.1e-20 2040
%       syd-2 5.6e-18 2040
%       sng-1 1.4e-12 1920
%       eat-16 2e-12 2040
%       C41G11.3 6.7e-12 2043
%       C32E8.7 1.2e-11 2039
%       snb-1 5.5e-11 1998
%       aex-1 6.3e-11 2040
%       C06G1.4 2.6e-10 2100
%       B0272.2 2.4e-09 2013
%       che-3 7e-08 2040
%       egl-10 4e-07 2000
%       C01C4.1 8.5e-07 2045
%       ncs-1 1.1e-05 2042
%
%      Not expressed in
%
%       F44B9.2 1.7e-19 2056
%       eat-4 7.6e-08 2040
%       F07C3.4 6.3e-07 2040
%       unc-68 7.1e-07 2105
%       T28B4.1 5.9e-06 2040
%       gpa-14 2.1e-05 2040
%       unc-97 4.3e-05 2040
%       gpa-15 9.5e-05 2037
%       F10F2.1 0.0038 2014
%       odr-10 0.48 2036
%
%        MOTIF WIDTH  BEST POSSIBLE MATCH
%        ----- -----  -------------------
%          M1    15   CTCTTCCTCTTCTTC
%          M2    15   GAAGAGAGAGAGAGA
%          M3    15   GAAAAATACCAAAAA
%          M5    15   CGAATCTGGTTGGAA
%          M6    15   TTGTCAGCTGACAAA
%          M8    15   GAACCGAGATAATTG
%          M9    15   TATCTCGGTTCCTGT
%         M10    15   AAAAATTTCAATTTT
%         M11    12   TTTTTGATGTTT
%         M12    12   ATTTCTGAAAAA
%         M13    15   GTGGGCTTCTATTAG
%         M15    15   GGCCCCCCGAACTGA
%         M16    15   GTGCGCGGGGCGGTG
%         M17    15   CCTCTAATAGAAGGG
%         M18    15   GCTCGAAGTGCACGC
%         M19    15   TAACTTTGAGCCAAT
%         M20    11   GGCTCCACCCC
%         M21     9   AGGAGGTCC
```

```
%       M22    15   ATTCGGGGGTGCAAA
%       M23    11   CCCGGCGACCG
%       M24    14   CCACGGGGCGAGAA
%       M25    12   GGCACCGGTGCC
%       M27    15   CGCCGAGCACCCCAC
%       M29    12   CCCACCCATTCC
%       M30    15   GGACGACGACTCCCG


@relation CBASENeural

@attribute CBASENeural {yes,no}
@attribute M1 string
@attribute M2 string
@attribute M3 string
@attribute M5 string
@attribute M6 string
@attribute M8 string
@attribute M9 string
@attribute M10 string
@attribute M11 string
@attribute M12 string
@attribute M13 string
@attribute M15 string
@attribute M16 string
@attribute M17 string
@attribute M18 string
@attribute M19 string
@attribute M20 string
@attribute M21 string
@attribute M22 string
@attribute M23 string
@attribute M24 string
@attribute M25 string
@attribute M27 string
@attribute M29 string
@attribute M30 string

@data

yes,'','{465:480}','{1033:1048}','{219:234}','{1332:1347}','{1367:1382}','{1018:1033}','{
1952:1967}','{1535:1547}','{654:666}','{1268:1283}','{989:1004}','{122:137}','{1063:1078}
','{1428:1443}','','{175:186}','','','','{738:752}','{1394:1406}','{387:402}','{108:120}'
,'{1852:1867}'
yes,'{1717:1732}','{1388:1403}','{94:109}','','{110:125}','{185:200}','{79:94}','{1308:13
23}','{166:178}','{1810:1822}','{221:236}','{872:887}','','{722:737}','{201:216}','{63:78
}','{1924:1935}','','','','','','','',''
yes,'{333:348}','{198:213}','{1818:1833}','','{1042:1057}','{1116:1131}','{1011:1026}','{
68:83}','{1608:1620}','{1221:1233}','{1143:1158}','','{2015:2030}','{1492:1507}','{156:17
1}','{1519:1534}','','','','','','{431:445}','','{118:133}','{51:63}','{468:483}'
yes,'{471:486}','{1850:1865}','{1784:1799}','','{1079:1094}','','{1049:1064}','{202:217}'
,'{1128:1140}','{181:193}','{1714:1729}','','','{840:855}','{998:1013}','{860:875}','{328
:339}','','{95:110}','','','','','{422:437}','','',''
yes,'{191:206}','','{1687:1702}','','{1829:1844}','{1794:1809}','{1900:1915}','{890:905}'
,'{756:768}','{1114:1126}','','','{333:348}','{1755:1770}','','{1582:1597}','','','{1005:
1020}','','','','','','','{1941:1956}'
yes,'{127:142}','{1109:1124}','{446:461}','{1293:1308}','{462:477}','{392:407}','{431:446
}','{770:785}','{1858:1870}','{1653:1665}','{601:616}','','{1192:1207}','','{553:568}','',
'','','{1147:1162}','','','','','','{113:125}',''
yes,'{1556:1571}','{1573:1588}','{1314:1329}','','{883:898}','{917:932}','{1656:1671}','',
'{898:910}','{599:611}','{423:438}','','{949:964}','{768:783}','','','{1594:1605}','','',
'','{466:480}','{318:330}','','','{1538:1553}'
yes,'','','{1517:1532}','{1981:1996}','{1283:1298}','{183:198}','{1252:1267}','{1111:1126
}','{1339:1351}','','{219:234}','{10:25}','{1728:1743}','{1209:1224}','{199:214}','{61:76
}','','{1042:1051}','','','{659:673}','','','{927:939}','',''
yes,'{338:353}','{722:737}','{67:82}','','','{829:844}','{355:370}','{217:232}','{1205:12
17}','{11:23}','{1118:1133}','{597:612}','{1322:1337}','','','','','{2000:2009}','{621:63
6}','','','','{498:513}','','',''
```

```
yes,'{1798:1813}','{1678:1693}','{44:59}','','{520:535}','','','{324:339}','{225:237}','{
922:934}','{239:254}','','','{1379:1394}','','','{1365:1376}','{1453:1462}','{744:759}','
','','','','{1597:1609}',''
yes,'{761:776}','{1670:1685}','{253:268}','','','{1036:1051}','{1426:1441}','{1702:1717}'
,'{2006:2018}','{1235:1247}','','','{685:700}','{1487:1502}','','','{795:806}','','','',
'','{74:86}','{112:127}','','{871:886}'
yes,'{1643:1658}','{303:318}','{1260:1275}','{1008:1023}','','','{1588:1603}','{1142:1157
}','{42:54}','','','{614:629}','','','','','','','{1945:1960}','','{128:142}','','{388:40
3}','{289:301}',''
yes,'{1490:1505}','{1242:1257}','','{1674:1689}','','{1345:1360}','{979:994}','{1959:1974
}','{264:276}','{557:569}','','','','','','','{386:395}','','','{999:1013}','','','{17
02:1714}','{532:547}'
yes,'{1894:1909}','{133:148}','{1472:1487}','','','{1310:1325}','{1708:1723}','{1438:1453
}','{1497:1509}','{271:283}','','','','','{1759:1774}','','','','','{1544:1555}','','','
','{314:326}','{1983:1998}'
yes,'{1095:1110}','','{1437:1452}','{169:184}','','','','{1666:1681}','{1519:1531}','{83:
95}','','','','','','','','','','{1203:1217}','','','{933:945}',''
yes,'{360:375}','{1042:1057}','{1718:1733}','{1311:1326}','','{153:168}','','{452:467}','
{69:81}','','','{1398:1413}','','','','','','{632:641}','','','{901:915}','','','','{1604
:1619}'
yes,'{239:254}','{1318:1333}','{1957:1972}','','','','{1216:1231}','{1993:2008}','{924:93
6}','{1181:1193}','','','{1882:1897}','','{449:464}','','{700:711}','','','','','','','',
'{1902:1917}'
yes,'{778:793}','{1470:1485}','{109:124}','','','','{1386:1401}','{204:219}','{304:316}',
'{1993:2005}','{1517:1532}','','{687:702}','','','{339:354}','','','','','','','','{1326:
1338}',''
yes,'{480:495}','{1799:1814}','{690:705}','','','','{719:734}','{501:516}','','','{260:27
5}','','','','','','','{801:810}','{1083:1098}','{1216:1227}','','','','','',''
yes,'{396:411}','','{1732:1747}','','','','{235:250}','{1977:1992}','{586:598}','{695:707
}','','','','','','','{346:357}','{625:634}','','','','','','{902:914}',''
yes,'{1352:1367}','{1041:1056}','{643:658}','','{421:436}','{894:909}','{543:558}','{392:
407}','{1629:1641}','{141:153}','','','','{455:470}','','','','','','','','','{1079:1094}
','{1327:1339}',''
no,'{1370:1385}','{474:489}','{1684:1699}','{102:117}','{1740:1755}','{410:425}','{304:31
9}','{978:993}','{875:887}','{1815:1827}','{446:461}','','','{1959:1974}','{1649:1664}','
{1067:1082}','','','','','','','','','',''
no,'{85:100}','{1970:1985}','{1121:1136}','','{579:594}','{1031:1046}','{508:523}','{1218
:1233}','{244:256}','','{330:345}','{1242:1257}','','','','{492:507}','','','','','','','
','{1322:1334}',''
no,'{582:597}','{1930:1945}','','','{240:255}','{275:290}','{169:184}','{1752:1767}','{21
8:230}','{1737:1749}','','','','{314:329}','','{29:44}','','','','','','','{1348:1363}','
','',''
no,'{952:967}','{2080:2095}','{1949:1964}','','','{1114:1129}','{1631:1646}','','','{1803
:1815}','','{421:436}','','{1554:1569}','{93:108}','{1529:1544}','','','{1472:1487}','','
','','','','',''
no,'{791:806}','{1819:1834}','{521:536}','','{566:581}','','','{1397:1412}','{606:618}','
','','','{589:604}','{1281:1296}','','','','','','','','','','','',''
no,'{1420:1435}','{707:722}','{17:32}','','','','{1475:1490}','{1241:1256}','{417:429}','
{296:308}','','','','{1031:1046}','','','{1288:1299}','','{662:677}','','','','','{1630:1
642}',''
no,'{1652:1667}','{540:555}','','','{690:705}','{1322:1337}','{663:678}','{230:245}','{33
8:350}','','{411:426}','','','','','','','','','','','','','',''
no,'{1611:1626}','{1473:1488}','{624:639}','','{827:842}','','{389:404}','{330:345}','{54
:66}','','{302:317}','','','','','','{1914:1925}','','','','','','','','',''
no,'{635:650}','{1241:1256}','{736:751}','','','','{1280:1295}','{1702:1717}','{288:300}'
,'{1860:1872}','','','','{1174:1189}','','','','','','','','','','','',''
no,'','','{286:301}','','{442:457}','','','{101:116}','{1289:1301}','{1731:1743}','','{93
8:953}','','','','{706:721}','','','','','','','','{1452:1464}',''
```

2.CBASKNeural

```
%
%       CBASKNeural
%       Contains the following promoter regions (they appear as data instances   in order
%       of appearance from the first to the last data instance
%
```

```
%     Expressed in:
%
%      gsa-1 2.2e-47 2010
%      Snt-1 2e-31 2040
%      F25B3.3 8.9e-30 2039
%      unc-119 2e-29 2040
%      osm-3 6.6e-26 1983
%      tax-4 1.3e-23 2040
%      unc-115 1.7e-23 2024
%      rgs-1 4.6e-23 2040
%      syd-2 9.7e-20 2040
%      eat-4 1.8e-16 2040
%      eat-16 1.6e-13 2040
%      aex-1 5.1e-12 2040
%      gpa-14 7.2e-12 2040
%      sng-1 2.3e-10 1920
%      B0272.2 2.5e-09 2013
%      C06G1.4 2.6e-09 2037
%      snb-1 2.7e-08 1998
%      C32E8.7 5.3e-08 2039
%      che-3 8.5e-08 2040
%      C41G11.3 1.3e-07 2043
%      C01C4.1 3.7e-07 2045
%      gpa-15 4.3e-07 2037
%      egl-10 2.5e-06 2000
%      odr-10 0.0005 2036
%
%     Not Expressed in:
%
%      F44B9.2 1.4e-28 2056
%      F07C3.4 8.4e-13 2040
%      unc-68 3.3e-06 2105
%      unc-97 1.2e-05 2040
%      ncs-1 0.00083 2042
%      F10F2.1 0.012 2014
%      T28B4.1 0.019 2040
%
%
%        MOTIF WIDTH BEST POSSIBLE MATCH
%        ----- ----- -------------------
%          1    15   GAAGAAAGAGAGAGA
%          2    15   TTCTCCCTCTTCTTC
%          3    12   AAAAACTGAAAA
%          5    15   CGCCGCCGCCCCTGC
%          6    15   GAACCGAGATAATTG
%          7    15   CCTCTAATAGAAGGG
%          8    15   CGAATCTGGTTGGAA
%          9    15   AAAAAGTTGTCAACT
%         10    15   TTTTGCACATTTTCG
%         11    15   GAGCCCAATTATCTC
%         13    15   AAAATTTTCAATTTT
%         14    15   TCCTGTAAAAGATAT
%         15    12   ATTTCTGAAAAA
%         16    11   GTGGGCGGGAG
%         17    15   ACAGGTTTTACGGTA
%         18    15   GCTCAAAGTGCAAGC
%         19    15   TATCAGCAACATTTT
%         20    15   ATTCGGGGGTGCAAA
%         21    15   TCTCTTCTCTCACCT
%         22    15   GAGCGTGAAATTGAG
%         23    13   CTGGCGGTGGTGG
%         24    15   CGCCACGACGTCTTC
%         25    15   GAGGCAGCGGTGCCG
%         26    15   GGAAGGGGGCGGGCA
%         27    15   TGGCTGGTGTGGGGG
%         28    15   GGGGCAGGAGGTCCA
%         29    12   GAGCGCGCGCTT
%         30    12   GCGCGCGTGGAG
```

```
%


@relation 'CBASKNeural-Final'

@attribute CBASKNeural {yes,no}
@attribute M1 string
@attribute M2 string
@attribute M3 string
@attribute M5 string
@attribute M6 string
@attribute M7 string
@attribute M8 string
@attribute M9 string
@attribute M10 string
@attribute M11 string
@attribute M13 string
@attribute M14 string
@attribute M15 string
@attribute M16 string
@attribute M17 string
@attribute M18 string
@attribute M19 string
@attribute M20 string
@attribute M21 string
@attribute M22 string
@attribute M23 string
@attribute M24 string
@attribute M25 string
@attribute M26 string
@attribute M27 string
@attribute M28 string
@attribute M29 string
@attribute M30 string

@data

yes,'{465:480}','{282:297}','{1635:1647}','{709:724}','{1367:1382}','{1271:1286}','{219:2
34}','{1525:1540}','{1750:1765}','{1439:1454}','{1952:1967}','{1027:1042}','','{329:340}'
,'','','{1320:1335}','','{1800:1815}','{847:862}','','{1850:1865}','{1392:1407}','{373:38
8}','','','{1426:1438}','{122:134}'
yes,'{1388:1403}','{1348:1363}','{367:379}','','{480:495}','{722:737}','','{104:119}','{1
906:1921}','{70:85}','{1308:1323}','{88:103}','{922:934}','','{420:435}','{201:216}','{13
8:153}','','{870:885}','','','{1145:1160}','','','{577:592}','{1742:1757}','',''
yes,'{198:213}','{333:348}','{1348:1360}','','{1116:1131}','{1492:1507}','','{1036:1051}'
,'{578:593}','{1002:1017}','{785:800}','{683:698}','{1221:1233}','','{958:973}','{156:171
}','{1070:1085}','','','','{1978:1991}','{466:481}','','{112:127}','{543:558}','','',''
yes,'{1850:1865}','{471:486}','','','','{840:855}','','{1073:1088}','{123:138}','{1040:10
55}','{1924:1939}','{1058:1073}','{181:193}','','{1196:1211}','{998:1013}','{935:950}','{
95:110}','{1537:1552}','','','','','','','','',''
yes,'','{191:206}','{428:440}','{1544:1559}','{1904:1919}','{1755:1770}','','{1875:1890}'
,'{977:992}','{1789:1804}','{889:904}','{1693:1708}','{1245:1257}','','','','{1841:1856}'
,'{1005:1020}','{160:175}','{1025:1040}','','{30:45}','','','{1285:1300}','','','{333:345
}'
yes,'','{1519:1531}','{1705:1720}','{183:198}','{1209:1224}','{1981:1996}','{1277:1292
}','{154:169}','{68:83}','{1625:1640}','{1261:1276}','','','','{199:214}','{1311:1326}','
','','','{1734:1747}','','','','{1368:1383}','{1646:1661}','',''
yes,'{1109:1124}','{127:142}','{691:703}','{1202:1217}','{537:552}','{1142:1157}','{1293:
1308}','{241:256}','{216:231}','{422:437}','{770:785}','{440:455}','{1653:1665}','','','{
553:568}','{490:505}','','{1354:1369}','','','','','{1237:1252}','','','{1802:1814}',''
yes,'','{1556:1571}','{624:636}','','{917:932}','{768:783}','','{836:851}','{1310:1325}',
'{802:817}','{1061:1076}','','{599:611}','{949:960}','','','{854:869}','','{1598:1613}','
{425:440}','','{1774:1789}','{316:331}','','{2021:2036}','','','{1628:1640}'
yes,'{722:737}','{263:278}','{966:978}','{559:574}','{829:844}','{1121:1136}','{1432:1447
}','','{1388:1403}','','{217:232}','{1493:1508}','{11:23}','','','','{27:42}','{621:636}'
,'{418:433}','{1331:1346}','','','','','{1723:1738}','{1995:2010}','',''
```

yes,’{1970:1985}’,’’,’{243:255}’,’{1845:1860}’,’{1140:1155}’,’’,’’,’{1060:1075}’,’{532:547}’,’{1026:1041}’,’{1276:1291}’,’{517:532}’,’’,’’,’’,’’,’{1093:1108}’,’’,’{1950:1965}’,’’,’’,’{1901:1916}’,’’,’’,’’,’{1824:1839}’,’’,’’
yes,’{1670:1685}’,’{761:776}’,’{2007:2019}’,’{556:571}’,’’,’’,’’,’’,’’,’{119:134}’,’{1702:1717}’,’’,’{1235:1247}’,’{361:372}’,’’,’’,’’,’’,’{823:838}’,’{454:469}’,’’,’{869:884}’,’’,’’,’{343:358}’,’{1951:1966}’,’’,’{399:411}’
yes,’{1438:1453}’,’{1093:1108}’,’{524:536}’,’’,’’,’’,’{169:184}’,’’,’{775:790}’,’’,’{1666:1681}’,’’,’{83:95}’,’’,’’,’’,’’,’’,’{959:974}’,’{1554:1569}’,’’,’{923:938}’,’’,’’,’{1199:1214}’,’’,’{1359:1371}’,’’
yes,’{707:722}’,’{247:262}’,’{18:30}’,’{1991:2006}’,’’,’{1031:1046}’,’’,’{886:901}’,’{139:154}’,’{1694:1709}’,’{1241:1256}’,’’,’{296:308}’,’’,’{1331:1346}’,’’,’’,’{662:677}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
yes,’’,’{1798:1813}’,’{37:49}’,’’,’’,’{1379:1394}’,’’,’’,’{149:164}’,’’,’{324:339}’,’’,’{921:933}’,’’,’’,’’,’’,’{744:759}’,’{1112:1127}’,’{1885:1900}’,’’,’{1321:1336}’,’’,’’,’{1647:1662}’,’’,’’,’’
yes,’{1318:1333}’,’{239:254}’,’’,’{788:803}’,’{346:361}’,’{669:684}’,’’,’’,’{1945:1960}’,’’,’{1470:1485}’,’’,’{1181:1193}’,’{1802:1813}’,’’,’{449:464}’,’’,’’,’{162:177}’,’{1397:1412}’,’’,’’,’’,’’,’{529:544}’,’{1351:1366}’,’{1536:1548}’,’’
yes,’’,’{681:696}’,’’,’{928:943}’,’{51:66}’,’{472:487}’,’’,’’,’{607:622}’,’’,’{1711:1726}’,’{1457:1472}’,’’,’’,’’,’’,’{1690:1705}’,’’,’{128:143}’,’{1008:1023}’,’{1564:1577}’,’{1862:1877}’,’{946:961}’,’{795:810}’,’{1319:1334}’,’’,’’,’’
yes,’{133:148}’,’{1894:1909}’,’{556:568}’,’’,’{1310:1325}’,’’,’’,’’,’{1451:1466}’,’{1531:1546}’,’’,’{1235:1250}’,’{271:283}’,’’,’’,’’,’’,’’,’’,’{934:949}’,’{244:257}’,’{16:31}’,’’,’{1165:1180}’,’’,’’,’{1755:1767}’,’’
yes,’{1242:1257}’,’{1490:1505}’,’{1184:1196}’,’’,’{860:875}’,’’,’{1674:1689}’,’{662:677}’,’{773:788}’,’’,’{1959:1974}’,’’,’{557:569}’,’{636:647}’,’{1718:1733}’,’’,’{205:220}’,’’,’’,’{345:360}’,’’,’{527:542}’,’’,’’,’{2024:2039}’,’’,’’,’’
yes,’{1470:1485}’,’{1134:1149}’,’{194:206}’,’’,’’,’’,’’,’{1759:1774}’,’{44:59}’,’’,’{1923:1938}’,’’,’{1993:2005}’,’{687:698}’,’’,’{393:408}’,’{1975:1990}’,’’,’{1451:1466}’,’’,’’,’’,’’,’’,’’,’{332:347}’,’’,’’,’’
yes,’{1992:2007}’,’{1643:1658}’,’{1628:1640}’,’’,’’,’’,’{1008:1023}’,’’,’{1131:1146}’,’’,’{75:90}’,’’,’{1536:1548}’,’’,’’,’’,’{117:132}’,’{1945:1960}’,’’,’’,’’,’{429:444}’,’’,’’,’’,’{607:622}’,’{1909:1921}’,’’
yes,’’,’{396:411}’,’{1144:1156}’,’{1884:1899}’,’’,’’,’’,’’,’{8:23}’,’{872:887}’,’{1977:1992}’,’’,’{695:707}’,’’,’’,’’,’{1066:1081}’,’’,’{651:666}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
yes,’{1598:1613}’,’{1473:1488}’,’{1367:1379}’,’’,’’,’’,’’,’{811:826}’,’{1880:1895}’,’{1721:1736}’,’{1993:2008}’,’{689:704}’,’’,’’,’’,’’,’{838:853}’,’’,’{1818:1833}’,’{704:719}’,’’,’’,’’,’’,’’,’’,’{1920:1932}’,’’
yes,’{1799:1814}’,’{480:495}’,’{304:316}’,’’,’’,’’,’’,’’,’{1355:1370}’,’’,’{502:517}’,’’,’’,’{1688:1699}’,’’,’’,’’,’{1083:1098}’,’{758:773}’,’’,’{814:827}’,’’,’’,’’,’{911:926}’,’’,’{997:1009}’,’’
yes,’’,’’,’{1871:1883}’,’’,’’,’’,’{220:235}’,’{447:462}’,’{1777:1792}’,’{713:728}’,’{101:116}’,’’,’{1731:1743}’,’’,’’,’’,’{1331:1346}’,’’,’{470:485}’,’’,’’,’’,’’,’’,’{1351:1366}’,’’,’’,’’
no,’{1474:1489}’,’{1370:1385}’,’{484:496}’,’’,’{410:425}’,’{1040:1055}’,’{102:117}’,’{1746:1761}’,’{855:870}’,’{295:310}’,’{978:993}’,’{1678:1693}’,’{1983:1995}’,’’,’{1949:1964}’,’{1649:1664}’,’{363:378}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
no,’’,’{582:597}’,’{889:901}’,’’,’{275:290}’,’’,’’,’{246:261}’,’{979:994}’,’{160:175}’,’{1752:1767}’,’{178:193}’,’{1737:1749}’,’’,’{324:339}’,’’,’{212:227}’,’’,’’,’{1925:1940}’,’{1229:1242}’,’’,’’,’’,’{1273:1288}’,’’,’’,’’
no,’{2080:2095}’,’{952:967}’,’{1110:1122}’,’{775:790}’,’’,’{1554:1569}’,’’,’{451:466}’,’{436:451}’,’’,’’,’’,’{1406:1421}’,’{1803:1815}’,’’,’{634:649}’,’’,’{226:241}’,’{1472:1487}’,’’,’’,’{57:70}’,’’,’’,’’,’’,’’,’’,’{1977:1989}’,’’
no,’{1655:1670}’,’{958:973}’,’{931:943}’,’’,’{1366:1381}’,’’,’{1285:1300}’,’{348:363}’,’{1014:1029}’,’’,’{230:245}’,’{1244:1259}’,’’,’’,’{1411:1426}’,’’,’’,’’,’’,’’,’’,’{1566:1581}’,’’,’’,’’,’{1698:1713}’,’’,’{857:869}’,’{729:741}’
no,’{1352:1367}’,’{1234:1249}’,’{1597:1609}’,’’,’{1727:1742}’,’’,’’,’{943:958}’,’{202:217}’,’’,’{392:407}’,’’,’{141:153}’,’{1325:1336}’,’{329:344}’,’’,’’,’’,’{1101:1116}’,’’,’’,’’,’’,’’,’’,’{1118:1133}’,’’,’{960:972}’,’’
no,’{1589:1604}’,’{635:650}’,’{288:300}’,’’,’{976:991}’,’’,’’,’’,’{1772:1787}’,’’,’{733:748}’,’’,’{1860:1872}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
no,’{1819:1834}’,’{1773:1788}’,’{237:249}’,’’,’’,’{1281:1296}’,’’,’{52:67}’,’{1100:1115}’,’’,’{1397:1412}’,’’,’’,’’,’{1998:2013}’,’’,’{271:286}’,’’,’’,’’,’’,’’,’{361:376}’,’’,’{1669:1684}’,’’,’’,’’

3.CBBodyWall

```
%
%       CBBodyWall
%       Contains the following promoter regions (they appear as data instances   in order
%       of appearance from the first to the last data instance
%
%       Expressed in:
%
%       F07C3.4 1.3e-30 2040
%       gsa-1 8.6e-26 2010
%       C06G1.4 3.1e-25 2037
%       F44B9.2 7.4e-17 2056
%       B0272.2: 2.2e-15 2013
%       unc-97 1e-13 2040
%       syd-2 3e-12 2040
%       egl-10 8.4e-12 2000
%       eat-16 2.8e-10 2040
%       unc-68 7.5e-09 2105
%       T28B4.1 2.2e-08 2040
%
%
%       Not Expressed in:
%
%       F25B3.3 8.1e-13 2039
%       Snt-1 2.7e-11 2040
%       tax-4 1e-07 2040
%       unc-119 3.5e-07 2040
%       osm-3 1.4e-06 1983
%       rgs-1 1.2e-05 2040
%       unc-115 0.0041 2024
%       eat-4 0.011 2040
%       gpa-14 0.02 2040
%       gpa-15 0.034 2037
%       snb-1 0.04 1998
%       che-3 0.04 2040
%       C41G11.3 0.058 2043
%       aex-1 0.36 2040
%       C32E8.7 0.58 2039
%       sng-1 0.88 1920
%       ncs-1 1.4 2042
%       F10F2.1 5.3 2014
%       C01C4.1 6 2045
%       odr-10 8.8 2036
%
%       MOTIF WIDTH BEST POSSIBLE MATCH
%       ----- ----- --------------------
%         1    15   AAGAAAAAGAGAGAG
%         2    15   CCTTTCTCTTCTTCT
%         3    15   TCTGAAAATCTGAAA
%         4    15   CCACCCCCGCCACCG
%         5    15   TTTATCAGTTGACAA
%         6    15   GGCCTTCTATTAGAG
%         7    12   GTTGTTTCCTCC
%         8    11   GAAATAAAAAA
%         9    15   GGAACCGAGATAATT
%        10    15   TGCAGGTGCGCGCGG
%        11    15   TTTCTCGGTTGCTGT
%        12    15   GTTCGAAAAGTTTTC
%        13    15   ATTTTATGATAAACA
%        14    15   CGCCGCCTCCCCTGC
%        15    12   CCCTCCCGATCG
%        16    15   CCGCAATCGGACTCG
%        17    15   AGTGGGAATGGGAAT
%        18     9   TTTTCGTGG
%        19     8   GCACGTCG
```

```
%        21     8   CCGGCCCG
%        23    15   GGCATGAGAGGGCGC
%        25    12   CTTCCCACACTA
%        26    12   GGAGGTCCTCCC
%        27    15   GTCCGGTCGCCAATG
%        28    14   GGTTGGTCTCCTCG
%        29    15   TGCTGCTTGCTCCCT
%        30    15   TGGCTCGGTGGCATC
%
%
%
@relation 'CBBodyWall-Final'

@attribute CBBodyWallExpr {yes,no}
@attribute M1 string
@attribute M2 string
@attribute M3 string
@attribute M4 string
@attribute M5 string
@attribute M6 string
@attribute M7 string
@attribute M8 string
@attribute M9 string
@attribute M10 string
@attribute M11 string
@attribute M12 string
@attribute M13 string
@attribute M14 string
@attribute M15 string
@attribute M16 string
@attribute M17 string
@attribute M18 string
@attribute M19 string
@attribute M21 string
@attribute M23 string
@attribute M25 string
@attribute M26 string
@attribute M27 string
@attribute M28 string
@attribute M29 string
@attribute M30 string

@data

yes,'{1175:1190}','{1771:1786}','{912:927}','{1229:1244}','{240:255}','{313:328}','{1443:
1455}','{1580:1591}','{274:289}','{1277:1292}','{169:184}','{571:586}','{224:239}','','{5
90:602}','{425:440}','{1630:1645}','{989:998}','{1838:1846}','','{378:393}','{529:541}','
','','','{1660:1675}',''
yes,'{504:519}','{613:628}','{1708:1723}','{705:720}','{1332:1347}','{1270:1285}','','{13
81:1392}','{1553:1568}','{1421:1436}','{958:973}','{1750:1765}','{1316:1331}','{164:179}'
,'{991:1003}','','','','{816:824}','','','','','{89:104}','','{376:391}',''
yes,'','{374:389}','','{924:939}','{466:481}','','{1166:1178}','','','','','{83:98}','','','
{943:958}','{297:309}','{657:672}','{337:352}','','{1582:1590}','','','','{990:1002}','',
'{233:247}','{792:807}','{1659:1674}'
yes,'{1882:1897}','{1430:1445}','{911:926}','','{375:390}','{1960:1975}','{265:277}','{16
88:1699}','{409:424}','','{1669:1684}','{739:754}','{1724:1739}','','{235:247}','','','',
'','','','{1338:1350}','','','','','',''
yes,'{1319:1334}','{1824:1839}','{1704:1719}','','{1359:1374}','','','{730:741}','','{187
7:1892}','{1955:1970}','{916:931}','','','{1841:1853}','{1021:1036}','{44:59}','{22:31}',
'','{792:800}','{1531:1546}','','','','','','',''
yes,'{541:556}','{665:680}','{1151:1166}','{1573:1588}','{452:467}','','{97:109}','','','
','{1459:1474}','{1013:1028}','{1335:1350}','','{1506:1518}','{861:876}','{1540:1555}','{
775:784}','','','{722:737}','','','','','','',''
yes,'{740:755}','{353:368}','{14:29}','{1783:1798}','','','{162:174}','{325:336}','','{70
8:723}','{635:650}','{1369:1384}','{890:905}','{559:574}','{693:705}','{1687:1702}','{175
9:1774}','','','','{618:633}','','','','{498:512}','',''
```

```
yes,’{1800:1815}’,’{1294:1309}’,’{1560:1575}’,’{814:829}’,’’,’{262:277}’,’{1855:1867}’,’{1274:1285}’,’’,’’,’{906:921}’,’{1634:1649}’,’’,’{1541:1556}’,’’,’’,’{1039:1054}’,’’,’{1317:1325}’,’’,’’,’’,’{802:814}’,’’,’’,’’,’{1258:1273}’
yes,’{1673:1688}’,’{760:775}’,’{2008:2023}’,’{7:22}’,’{222:237}’,’’,’’,’{1781:1792}’,’{1451:1466}’,’{347:362}’,’’,’{1888:1903}’,’’,’{556:571}’,’{305:317}’,’{489:504}’,’’,’’,’’,’’,’{531:546}’,’’,’’,’’,’’,’’,’’
yes,’{1714:1729}’,’{2014:2029}’,’{1454:1469}’,’’,’’,’{1553:1568}’,’{50:62}’,’{1754:1765}’,’’,’{612:627}’,’’,’{227:242}’,’’,’’,’{779:791}’,’{1499:1514}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
yes,’{1773:1788}’,’{786:801}’,’{1980:1995}’,’’,’{566:581}’,’{1280:1295}’,’’,’{196:207}’,’{982:997}’,’{94:109}’,’{277:292}’,’’,’{329:344}’,’’,’’,’{965:980}’,’’,’{590:599}’,’’,’’,’’,’’,’’,’{494:509}’,’’,’’,’{1671:1686}’
no,’{1786:1801}’,’{447:462}’,’{57:72}’,’{2011:2026}’,’{1042:1057}’,’{1493:1508}’,’’,’{1097:1108}’,’{1115:1130}’,’’,’{408:423}’,’{1228:1243}’,’{1058:1073}’,’’,’{508:520}’,’’,’{1622:1637}’,’’,’’,’’,’’,’’,’’,’’,’{121:135}’,’’,’’
no,’{1346:1361}’,’{1715:1730}’,’{301:316}’,’’,’{110:125}’,’{721:736}’,’’,’{1223:1234}’,’{76:91}’,’’,’{448:463}’,’{421:436}’,’{126:141}’,’’,’’,’’,’{903:918}’,’’,’’,’’,’’,’{1012:1024}’,’’,’’,’’,’’,’’
no,’’,’{1914:1929}’,’{1186:1201}’,’’,’{1283:1298}’,’{1210:1225}’,’’,’{1271:1282}’,’{182:197}’,’’,’{77:92}’,’{1053:1068}’,’{123:138}’,’’,’{1977:1989}’,’’,’{1765:1780}’,’’,’{1886:1894}’,’’,’{775:790}’,’’,’{1652:1664}’,’’,’{5:19}’,’’,’’
no,’{1990:2005}’,’{1626:1641}’,’{1431:1446}’,’’,’{1079:1094}’,’{841:856}’,’’,’{1935:1946}’,’{873:888}’,’’,’{1049:1064}’,’{1340:1355}’,’{931:946}’,’’,’’,’’,’’,’’,’’,’’,’{92:107}’,’’,’’,’’,’’,’{682:697}’,’’
no,’{107:122}’,’{192:207}’,’{429:444}’,’’,’{1829:1844}’,’{1756:1771}’,’’,’{279:290}’,’{1903:1918}’,’’,’{1599:1614}’,’’,’{1853:1868}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’{213:227}’,’’,’’
no,’{1655:1670}’,’{1557:1572}’,’’,’{1713:1728}’,’{842:857}’,’{769:784}’,’’,’{1831:1842}’,’{916:931}’,’{1272:1287}’,’{204:219}’,’{708:723}’,’’,’’,’{1684:1696}’,’’,’{1800:1815}’,’’,’{638:646}’,’’,’’,’’,’’,’’,’’,’{1626:1641}’,’{227:242}’
no,’{1166:1181}’,’{1279:1294}’,’{285:300}’,’’,’{462:477}’,’’,’{1607:1619}’,’{865:876}’,’{536:551}’,’’,’{1047:1062}’,’{837:852}’,’’,’{1202:1217}’,’{1360:1372}’,’’,’{921:936}’,’’,’{1453:1461}’,’’,’’,’’,’’,’’,’’,’{192:207}’,’’
no,’{102:117}’,’{1940:1955}’,’{311:326}’,’{471:486}’,’{579:594}’,’’,’’,’{1650:1661}’,’{1032:1047}’,’’,’{1293:1308}’,’’,’{338:353}’,’’,’’,’’,’{1329:1344}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
no,’{1420:1435}’,’{248:263}’,’{545:560}’,’’,’{1159:1174}’,’{1032:1047}’,’’,’{24:35}’,’’,’’,’{1778:1793}’,’{1120:1135}’,’’,’’,’’,’{811:826}’,’{1703:1718}’,’’,’’,’’,’’,’’,’’,’’,’{1987:2001}’,’’,’’
no,’{224:239}’,’{1604:1619}’,’{126:141}’,’’,’{827:842}’,’’,’’,’{1267:1278}’,’’,’’,’{360:375}’,’{624:639}’,’’,’’,’’,’’,’’,’’,’’,’’,’{34:49}’,’{1715:1727}’,’’,’{293:308}’,’’,’’,’’
no,’{693:708}’,’{1893:1908}’,’{844:859}’,’’,’’,’’,’’,’{1197:1208}’,’’,’’,’{1499:1514}’,’’,’’,’’,’’,’’,’’,’’,’{325:333}’,’’,’’,’{351:363}’,’’,’’,’’,’’,’’
no,’{1437:1452}’,’{1466:1481}’,’{905:920}’,’’,’{359:374}’,’’,’{1486:1498}’,’{221:232}’,’’,’{1213:1228}’,’{1594:1609}’,’{25:40}’,’{1765:1780}’,’’,’’,’’,’’,’{1108:1117}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
no,’{302:317}’,’{1641:1656}’,’{1667:1682}’,’’,’’,’’,’’,’{68:79}’,’’,’’,’{34:49}’,’{512:527}’,’{1932:1947}’,’{388:403}’,’’,’’,’’,’’,’’,’’,’’,’{1949:1961}’,’’,’’,’’,’’,’’
no,’{1091:1106}’,’{1436:1451}’,’{98:113}’,’’,’’,’’,’’,’{1659:1670}’,’{1750:1765}’,’’,’{1371:1386}’,’’,’’,’’,’{1721:1733}’,’’,’’,’{426:435}’,’’,’’,’’,’{1645:1657}’,’’,’’,’’,’’,’’
no,’{1491:1506}’,’{1242:1257}’,’{774:789}’,’{1303:1318}’,’{1560:1575}’,’’,’’,’{99:110}’,’{1185:1200}’,’’,’{1695:1710}’,’’,’{719:734}’,’{1628:1643}’,’’,’’,’{1882:1897}’,’’,’’,’’,’’,’’,’’,’’,’{1534:1548}’,’{385:400}’,’’
no,’{1189:1204}’,’{1796:1811}’,’{567:582}’,’{1369:1384}’,’{375:390}’,’’,’’,’{1503:1514}’,’{1853:1868}’,’’,’{958:973}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’{676:690}’,’’,’’
no,’{1099:1114}’,’’,’{630:645}’,’’,’{1647:1662}’,’’,’’,’{1805:1816}’,’’,’’,’{601:616}’,’{313:328}’,’’,’’,’’,’’,’{2021:2036}’,’’,’’,’’,’{176:191}’,’{34:46}’,’{473:485}’,’’,’’,’’,’’
no,’’,’{634:649}’,’’,’’,’’,’’,’’,’{403:414}’,’’,’’,’{1465:1480}’,’{130:145}’,’{1248:1263}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
no,’{213:228}’,’’,’{20:35}’,’{1932:1947}’,’’,’’,’’,’{1329:1340}’,’’,’’,’{81:96}’,’’,’’,’{1884:1899}’,’’,’’,’{1305:1320}’,’’,’’,’’,’’,’’,’’,’’,’’,’{338:353}’,’’
no,’{466:481}’,’{1933:1948}’,’{810:825}’,’{1981:1996}’,’{82:97}’,’’,’’,’{1773:1784}’,’{719:734}’,’’,’’,’{1721:1736}’,’’,’’,’’,’’,’{193:208}’,’’,’’,’’,’’,’’,’’,’’,’’,’’,’’
```

4.CBOLLNeural

```
%
%       CBOLLNeural(whole)
%       Contains the following promoter regions (they appear as data instances   in order
%       of appearance from the first to the last data instance
%
%     Expressed in:
%
%       1.gsa-1 4.5e-46 2010
%       2.F25B3.3 1.3e-34 2039
%       3.rgs-1 1.7e-32 2040
%       4.Snt-1 2.8e-31 2040
%       5.unc-119 1.5e-29 2040
%       6.unc-115 1.3e-24 2024
%       7.syd-2 1.2e-20 2040
%       8.eat-4 3.2e-20 2040
%       9.sng-1 5.8e-18 1920
%       10.snb-1 7.3e-16 1998
%       11.C32E8.7 4.5e-15 2039
%       12.eat-16 5.4e-15 2040
%       13.che-3 2.7e-14 2040
%       14.egl-10 2.4e-13 2000
%       15.aex-1 6.2e-13 2040
%       16.C06G1.4 6.9e-13 2100
%       17.C41G11.3 1.1e-12 2043
%       18.B0272.2 2.2e-11 2013
%       19.C01C4.1 9.8e-08 2045
%
%
%     Not Expressed in:
%
%       F44B9.2 1.1e-25 2056
%       osm-3 1.2e-17 1983
%       F07C3.4 1.5e-13 2040
%       unc-68 3.3e-12 2105
%       gpa-14 9.7e-09 2040
%       T28B4.1 1.2e-08 2040
%       tax-4 2.3e-08 2040
%       unc-97 3.3e-07 2040
%       gpa-15 1.5e-05 2037
%       ncs-1 1.6e-05 2042
%       F10F2.1 0.00099 2014
%       odr-10 0.13 2036
%
%       MOTIF WIDTH BEST POSSIBLE MATCH
%       ----- ----- -------------------
%         1    15   GAAGAGAGAGAGAGA
%         2    15   CTCTTCCTCTTCTTC
%         6    15   CGAATCTGGTTGGAA
%         7    14   TTATCTCGGTTTCT
%         8    15   ATTTTATCAGTTGAC
%         9    15   GAACCGAGATAATTG
%        12    15   GTTTTCCAGAAAAAT
%        13    15   TGGAAAATTGAAAAA
%        14    15   GGCCTTCTATTAGAG
%        15    15   CTCTAATAGAAGGCC
%        16    12   CCGCTCTCCGCT
%        17    15   ATTAGGGGGTGCAAA
%        18    15   TTGTCAACTAAAAAA
%        19    12   CCCACCCATTCC
%        20    12   GCGCGGGGCGGC
%        21    15   GAGGCACCGGTGCCC
%        22    15   GAAAAGTTTTGAGAG
%        24    15   ATGAGTGTGTGTGCG
%        25    15   TGTACCCCCGAATTG
%        26    15   TCCCGTCGCGACGTC
```

```
%       27    15   CATTTTCATTTCATC
%       28    15   GTTCATCATAATATC
%       29    11   CCCGGCGACCG
%       30    15   CCACGGGGCGAGAAA
%


@relation 'CBOLLNeural-Final'

@attribute CBOLLNeural {yes,no}
@attribute M1 string
@attribute M2 string
@attribute M6 string
@attribute M7 string
@attribute M8 string
@attribute M9 string
@attribute M12 string
@attribute M13 string
@attribute M14 string
@attribute M15 string
@attribute M16 string
@attribute M17 string
@attribute M18 string
@attribute M19 string
@attribute M20 string
@attribute M21 string
@attribute M22 string
@attribute M24 string
@attribute M25 string
@attribute M26 string
@attribute M27 string
@attribute M28 string
@attribute M29 string
@attribute M30 string

@data

yes,'{465:480}','{282:297}','{219:234}','{1017:1031}','{1330:1345}','{1367:1382}','{1652:
1667}','{1710:1725}','{1270:1285}','{978:993}','{1408:1420}','','{1519:1534}','{108:120}'
,'{124:136}','{1392:1407}','{861:876}','{319:334}','','{341:356}','{263:278}','{1496:1511
}','','{738:753}'
yes,'{198:213}','{333:348}','','{1010:1024}','{1080:1095}','{1116:1131}','{832:847}','{66
2:677}','','{1493:1508}','{634:646}','','{1042:1057}','','{1657:1669}','','{572:587}','',
'','','{862:877}','{1059:1074}','','{431:446}'
yes,'{1933:1948}','{1556:1571}','','','','{917:932}','{597:612}','{1300:1315}','{425:440}
','{769:784}','','','{842:857}','{152:164}','{947:959}','{316:331}','','{1271:1286}','','
{1771:1786}','{1887:1902}','','',''
yes,'{1388:1403}','{1348:1363}','{1275:1290}','{78:92}','{516:531}','{185:200}','{1559:15
74}','{784:799}','{738:753}','{721:736}','','{411:426}','{110:125}','','','','','','','',
'','{127:142}','',''
yes,'{507:522}','{471:486}','','{1048:1062}','{1110:1125}','','{180:195}','','{1185:1200}
','{841:856}','{1847:1859}','{95:110}','{1079:1094}','','','{1824:1839}','','{2011:2026}'
,'','','{726:741}','{930:945}','',''
yes,'{1109:1124}','{1875:1890}','{1293:1308}','{430:444}','{500:515}','{537:552}','{1712:
1727}','{681:696}','','','{1412:1424}','','{462:477}','{113:125}','{1194:1206}','{1209:12
24}','{28:43}','{1780:1795}','','','{231:246}','{813:828}','','',''
yes,'{722:737}','{263:278}','','','','{829:844}','{123:138}','{1258:1273}','','','{692:70
4}','{621:636}','','','{1326:1338}','','{1923:1938}','','{596:611}','','{1596:1611}','{88
9:904}','','',''
yes,'{1761:1776}','{1970:1985}','','{507:521}','{1103:1118}','{1031:1046}','{1475:1490}',
'{487:502}','','','{1381:1393}','','{1066:1081}','','{1265:1277}','','','','','{1898:1913
}','{1333:1348}','{337:352}','','',''
yes,'{1760:1775}','{1104:1119}','','{1003:1017}','{373:388}','','{404:419}','{35:50}','',
'','','{744:759}','{1211:1226}','{1597:1609}','','','','{1304:1319}','{792:807}','','{499
:514}','','','',''
yes,'{133:148}','{1891:1906}','','{1712:1726}','','{1310:1325}','','{1871:1886}','','','',''
,'','','{314:326}','{1676:1688}','{1781:1796}','','','','','','','{1544:1555}',''
```

```
yes,'','{1014:1029}','{1674:1689}','{981:995}','{815:830}','{1345:1360}','{1764:1779}','{
1182:1197}','{1939:1954}','','','{518:533}','{1969:1984}','{1702:1714}','{693:705}','',''
,'','','','','','','{999:1014}'
yes,'{1670:1685}','{761:776}','','{605:619}','{1516:1531}','','{1233:1248}','{2009:2024}'
,'','','{1644:1656}','','','','{854:866}','','','{420:435}','','','','','','',''
yes,'{1470:1485}','{1486:1501}','','','','','{1958:1973}','{761:776}','','','','','','{13
26:1338}','','','{29:44}','','','','{1637:1652}','','',''
yes,'{700:715}','{480:495}','','','','','','{302:317}','','{611:626}','{1013:1025}','{108
3:1098}','','','','','','{1028:1043}','','','{272:287}','{418:433}','','{1216:1227}','{230:2
45}'
yes,'{1799:1814}','{1093:1108}','{169:184}','','{1996:2011}','','{798:813}','{6:21}','','
','','{420:435}','{475:490}','{933:945}','','','','{144:159}','','','{1732:1747}','','','
{1203:1218}'
yes,'','{359:374}','{1311:1326}','{690:704}','{535:550}','{153:168}','{409:424}','{1979:1
994}','','','{927:939}','','','','','','{1461:1476}','{1084:1099}','{1435:1450}','{780:79
5}','{259:274}','','','{901:916}'
yes,'{303:318}','{1643:1658}','{1008:1023}','{1568:1582}','{1395:1410}','{1698:1713}','{1
65:180}','{78:93}','','','','{1945:1960}','','{289:301}','{387:399}','','{975:990}','{877
:892}','','{1443:1458}','','','{332:343}',''
yes,'{1318:1333}','{239:254}','{1386:1401}','{1720:1734}','{289:304}','','{756:771}','{14
89:1504}','','','','','{1244:1259}','','{1884:1896}','{539:554}','{1409:1424}','','','{21
2:227}','','','','',''
yes,'{1755:1770}','{396:411}','','{50:64}','','','{91:106}','{18:33}','','','','','','{90
2:914}','','','{1561:1576}','{67:82}','','','','','','',''
no,'{1474:1489}','{1370:1385}','','{303:317}','{373:388}','{410:425}','{752:767}','{912:9
27}','{1960:1975}','{1627:1642}','','','{335:350}','','','{580:595}','{611:626}','','','',
'','{352:367}','','',''
no,'{170:185}','{191:206}','','{1902:1916}','{1831:1846}','{1794:1809}','{1507:1522}','{1
16:131}','','{1756:1771}','{1544:1556}','{1005:1020}','{1869:1884}','','','','','','','{2
28:243}','','{1852:1867}','','',''
no,'','{582:597}','','{168:182}','','{275:290}','','{913:928}','{313:328}','{127:142}','',
'','{240:255}','','','','','{1378:1393}','','','{1632:1647}','{503:518}','{223:238}','','',''
no,'{2080:2095}','{952:967}','','{1669:1683}','','','{2036:2051}','{1650:1665}','{1553:15
68}','','','','','{237:252}','{204:216}','','','','','{1754:1769}','','','','','','',''
no,'{707:722}','{1750:1765}','','','','','','','{546:561}','{1032:1047}','{1303:1318}','','','{
1568:1583}','','{1630:1642}','','','{1214:1229}','{831:846}','','','{1244:1259}','','','',''
no,'{1840:1855}','{1773:1788}','','','{564:579}','','{274:289}','{1399:1414}','','{1280:1
295}','','','','{1198:1213}','','','{361:376}','','{1547:1562}','','{495:510}','','{1316:133
1}','','',''
no,'','','{1981:1996}','{1251:1265}','{1285:1300}','{183:198}','','{1185:1200}','{1210:12
25}','','','','{147:162}','{927:939}','','','{1049:1064}','','{9:24}','','','{1300:1315}'
,'','',''
no,'{1655:1670}','{958:973}','','{883:897}','','{1366:1381}','{1334:1349}','','','{52:67}
','{835:847}','','{327:342}','','','','{1168:1183}','','{1035:1050}','','{1764:1779}','',
'','',''
no,'{1598:1613}','{1473:1488}','','{362:376}','','','{1732:1747}','{1365:1380}','','','',
'','{1756:1771}','','','','','{484:499}','{1941:1956}','','{1057:1072}','{535:550}','','',''
no,'{1100:1115}','{1352:1367}','','{1254:1268}','','{1727:1742}','{1910:1925}','{671:686}
','','','','','','','{1327:1339}','','','','{1409:1424}','','','{1013:1028}','','','',''
no,'{1433:1448}','{635:650}','','{975:989}','{319:334}','','','{16:31}','','','','','','{363
:378}','','','','{1852:1867}','','','','','','','',''
no,'','','','','','','','','','{811:826}','','','','','','{82:97}','{1452:1464}','','','','','','{939
:954}','','{1250:1265}','{154:169}','','',''
```

5. CBPanNeural

```
%
%       CBPPanNeural
%       Contains the following promoter regions (they appear as data instances   in order
%       of appearance from the first to the last data instance
%
%     Expressed in:
%
%       gsa-1 5.3e-41 2010
%       F25B3.3 1.3e-31 2039
```

```
%       unc-119 3.4e-29 2040
%       rgs-1 4.2e-27 2040
%       Snt-1 4.9e-27 2040
%       unc-115 9e-27 2024
%       egl-10 2.6e-23 2000
%       sng-1 2.7e-18 1920
%       eat-16 3.2e-18 2040
%       syd-2 6.4e-18 2040
%       snb-1 7.5e-15 1998
%       C41G11.3 1e-13 2043
%       C06G1.4 1.6e-13 2100
%       aex-1 1.6e-11 2040
%       B0272.2 9.1e-11 2013
%       C32E8.7 7.8e-09 2039
%       C01C4.1 1.1e-08 2045
%
%     NotExpressed in:
%
%       F44B9.2 1e-21 2056
%       osm-3 1.3e-18 1983
%       eat-4 1.9e-12 2040
%       F07C3.4 9.1e-12 2040
%       unc-68 9.3e-08 2105
%       gpa-15 5.2e-07 2037
%       tax-4 5.8e-07 2040
%       T28B4.1 9.9e-07 2040
%       unc-97 1.2e-06 2040
%       gpa-14 3.7e-06 2040
%       che-3 0.0003 2040
%       F10F2.1 0.00052 2014
%       ncs-1 0.024 2042
%       odr-10 0.94 2036
%
%         MOTIF WIDTH BEST POSSIBLE MATCH
%         ----- ----- -------------------
%           1    15   CTCTCCCTCTTCTTC
%           2    15   GAAGAAAGAGAGAGA
%           5    15   CGAATCTGGTTGGAA
%           6    15   ATTTTATCAGTTGAC
%           7    15   GTGGGCTTCTATTAG
%           9    15   CGAGATAATTGAGCT
%          12    12   TTTTTGATTTTT
%          13    14   TTATCCCAGTTTCT
%          14    15   CTCTAATAGAAGGCC
%          15    15   ATTCGGGGGTGCAAA
%          16    15   TTGTCCACTAATAAA
%          17    12   ACCGCTCTCCGC
%          18    15   AGCCGAGCGGCACAC
%          19    15   GTTTATCATAATATC
%          20    12   GTGCGGGGGGCG
%          21    15   TACTTGTTCCCTTGC
%          22    11   CCCGGCGACCG
%          23    15   CCCCTCTCTACCCCC
%          24    15   CCGGGATACCCGAAC
%          25    12   ACAAGTTTTCGG
%          26    15   CTCACCCCCAGACCC
%          27    12   CCCTCTTCTCTC
%          28    15   TCCTTTTGACACCTC
%          29    12   GGCACCGGTGCC
%          30    12   GGGTTACTGTAG
%
%
%
%
%

@relation 'CBPPanNeural-Final'
```

```
@attribute CBPPanNeuralExpr {yes,no}
@attribute M1 string
@attribute M2 string
@attribute M5 string
@attribute M6 string
@attribute M7 string
@attribute M9 string
@attribute M12 string
@attribute M13 string
@attribute M14 string
@attribute M15 string
@attribute M16 string
@attribute M17 string
@attribute M18 string
@attribute M19 string
@attribute M20 string
@attribute M21 string
@attribute M22 string
@attribute M23 string
@attribute M24 string
@attribute M25 string
@attribute M26 string
@attribute M27 string
@attribute M28 string
@attribute M29 string
@attribute M30 string

@data

yes,'{282:297}','{501:516}','{219:234}','{1517:1532}','{1268:1283}','{1558:1573}','{1535:
1547}','{1017:1031}','{1064:1079}','','{1479:1494}','{1407:1419}','{387:402}','{1496:1511
}','{113:125}','','','','{1841:1856}','{1649:1661}','{169:184}','{1799:1811}','{808:823}'
,'{1394:1406}',''
yes,'{333:348}','{198:213}','','{1080:1095}','{1143:1158}','{1003:1018}','{1608:1620}','{
1534:1548}','{1493:1508}','','{1042:1057}','','{115:130}','{1059:1074}','{1657:1669}','',
'','','{633:648}','{1229:1241}','{2009:2024}','','{1964:1979}','',''
yes,'{471:486}','{531:546}','','{1110:1125}','{1714:1729}','','{1128:1140}','{875:889}','
{841:856}','{95:110}','{1079:1094}','','{422:437}','{930:945}','','','','{664:679}','','{
965:977}','','{568:580}','','',''
yes,'{1556:1571}','{1209:1224}','','','{423:438}','{921:936}','{898:910}','','{769:784}',
'','{842:857}','{1706:1718}','','','','{380:395}','','','','{287:299}','','{1684:1696}','
','{318:330}',''
yes,'{1348:1363}','{1388:1403}','{1275:1290}','{516:531}','{221:236}','','{466:478}','{78
:92}','{721:736}','','{110:125}','','','{127:142}','{583:595}','','','','{1950:1965}','',
'{1522:1537}','','{864:879}','',''
yes,'{1875:1890}','{1109:1124}','','{500:515}','{601:616}','{541:556}','{449:461}','{430:
444}','','','{462:477}','{1411:1423}','','','{1192:1204}','','','','{827:842}','{741:753}
','','{1348:1360}','{153:168}','','{630:642}'
yes,'{1799:1814}','{861:876}','','','{260:275}','{307:322}','{1644:1656}','{749:763}','{6
11:626}','{1083:1098}','','{1012:1024}','','','','{1541:1556}','{1216:1227}','','{1692:17
07}','{1029:1041}','','{218:230}','{949:964}','','{1928:1940}'
yes,'{1798:1813}','{1753:1768}','','{373:388}','{1135:1150}','','{571:583}','{1003:1017}'
,'{1635:1650}','{794:809}','{1211:1226}','','','{970:985}','','','','{748:763}','{811:826
}','{323:335}','{1367:1382}','','{22:37}','',''
yes,'{761:776}','{1689:1704}','','{1516:1531}','{430:445}','{1040:1055}','{1783:1795}','',
'','','','{1645:1657}','{112:127}','{744:759}','{356:368}','{1428:1443}','','{529:544}',
'','','','{403:415}','','{74:86}','{34:46}'
yes,'{263:278}','{722:737}','','','{1643:1658}','','{1205:1217}','','','{621:636}','','{6
91:703}','','{889:904}','','{1985:2000}','','','{1443:1458}','{911:923}','{505:520}','','
','',''
yes,'{1894:1909}','{133:148}','{1050:1065}','','','','{1497:1509}','{739:753}','','','{88
4:899}','','{321:336}','','{657:669}','','{1544:1555}','','{63:78}','{836:848}','','{1781
:1793}','{412:427}','',''
yes,'{1643:1658}','{303:318}','{1008:1023}','{1216:1231}','{1532:1547}','','{42:54}','{15
53:1567}','','','','{1842:1854}','{388:403}','','','{880:895}','{332:343}','','','{74:86}
','','{1437:1449}','{1341:1356}','',''
```

```
yes,'{835:850}','{90:105}','{1311:1326}','{535:550}','{2060:2075}','','{69:81}','','','',
'{1595:1610}','{928:940}','','','','','','{963:978}','{794:809}','{219:231}','{898:913}',
'{1022:1034}','{978:993}','',''
yes,'{1093:1108}','{1232:1247}','{169:184}','{1996:2011}','','','{1519:1531}','','','','{
475:490}','','','{981:996}','','{1304:1319}','','','{25:40}','{1603:1615}','','{1115:1127
}','{1392:1407}','',''
yes,'{239:254}','{1318:1333}','','{289:304}','','','{1207:1219}','','','','{1244:1259}','
','{786:801}','','{1882:1894}','{219:234}','','','','{1410:1422}','{810:825}','','','','{
719:731}'
yes,'{1014:1029}','{1242:1257}','{1674:1689}','{1110:1125}','','','{1804:1816}','{1343:13
57}','','','{1969:1984}','','','','','','','','{1827:1842}','{778:790}','','','{572:587}'
,'','{655:667}'
yes,'{396:411}','{447:462}','','','{1703:1718}','{1781:1796}','{586:598}','{50:64}','','',
'','','','','','','','{1870:1885}','{849:864}','{22:34}','','','{5:20}','','{1467:1479}'
no,'{1370:1385}','{474:489}','','{373:388}','{503:518}','{414:429}','{1554:1566}','{303:3
17}','{1960:1975}','{811:826}','','','','{352:367}','','','','','','','{913:925}','','{588:6
00}','{1194:1209}','',''
no,'{191:206}','{1218:1233}','','{1831:1846}','','{1908:1923}','{1816:1828}','{1704:1718}
','{1756:1771}','{1005:1020}','{1869:1884}','','','{1852:1867}','','','','','','{1300:131
2}','{76:91}','{370:382}','{1423:1438}','',''
no,'{1489:1504}','{1970:1985}','','{1103:1118}','','{1027:1042}','{1286:1298}','{507:521}
','','{406:421}','{1066:1081}','','','{337:352}','','','','','','{320:332}','','{1949:196
1}','{110:125}','',''
no,'','{1930:1945}','','{202:217}','','{161:176}','{1774:1786}','{273:287}','{313:328}','
','{240:255}','','{1348:1363}','{223:238}','{1241:1253}','','','','','{1379:1391}','','{5
85:597}','{964:979}','','{617:629}'
no,'{952:967}','{2080:2095}','{466:481}','','','{1110:1125}','','','{1553:1568}','','{237
:252}','','','{894:909}','','','','','','{1782:1794}','','{1970:1982}','','',''
no,'{1611:1626}','{1:16}','','','{302:317}','{163:178}','{1639:1651}','{1266:1280}','','{
1130:1145}','{1756:1771}','','','{793:808}','','','','{715:730}','','{626:638}','','{523:
535}','{1427:1442}','','{1739:1751}'
no,'','','{926:941}','{1285:1300}','{219:234}','{187:202}','{1270:1282}','{76:90}','{1210
:1225}','{705:720}','{850:865}','','{1704:1719}','{1300:1315}','','','','','','{1051:1063
}','','','','',''
no,'{1773:1788}','{1819:1834}','','{45:60}','','','{237:249}','','{1280:1295}','','{1198:
1213}','','','{1316:1331}','','','','','{123:138}','{1353:1365}','','{425:437}','','',''
no,'{958:973}','{1655:1670}','','{1130:1145}','','{1181:1196}','{930:942}','{1676:1690}',
'{52:67}','','{324:339}','{725:737}','{1506:1521}','','','','','','','{702:714}','','{819
:831}','','',''
no,'{706:721}','{1820:1835}','','{1601:1616}','','','{469:481}','{1281:1295}','{1032:1047
}','{662:677}','{1159:1174}','','','{314:329}','','','','','','{538:550}','','{1697:1709}
','{1789:1804}','',''
no,'{1473:1488}','','','','','{9:24}','{110:122}','{1307:1321}','','','{1718:1733}','','',
'','','','','','{1216:1231}','{30:42}','','','{1367:1382}','','{1012:1024}'
no,'{635:650}','{1223:1238}','','{1202:1217}','','','{288:300}','{975:989}','','','','','
','','','{1918:1933}','','','','{939:951}','','','{557:572}','','{1169:1181}'
no,'{1040:1055}','{1352:1367}','','','','','{1629:1641}','','','','{487:502}','{1860:1872
}','','','','{2025:2040}','','','','{1492:1504}','','','{1990:2005}','','{1312:1324}'
no,'{466:481}','{1871:1886}','{220:235}','{1628:1643}','','','{1404:1416}','{721:735}','',
'','{82:97}','','','{154:169}','','','','','','{401:413}','','','','',''
```

6. CElegansPanNeural

```
%
%       CEPanNeural
%       Contains the following promoter regions (they appear as data instances   in order
%       of appearance from the first to the last data instance
%
%
%
%
%       goa-1 3.5e-29 9717 Expression:yes
%       C06G1.4 1.4e-28 3258 Expression:yes
%       C41G11.3 1.5e-23 11702 Expression:yes
%       unc-11 1.1e-19 1918 Expression:yes
```

```
%       C04E12.7 1.1e-19 6439 Expression:yes
%       unc-64 3.2e-19 5002 Expression:yes
%       eat-16 4.3e-18 4721 Expression:yes
%       Y105C5B.19 1.9e-17 5998 Expression:yes
%       aex-3 5.5e-15 1320 Expression:yes
%       F25B3.3 8.8e-15 4019 Expression:yes
%       snb-1 1.2e-14 3329 Expression:yes
%       BO464.5 2.5e-13 6196 Expression:yes
%       syd-2 2.6e-13 6003 Expression:yes
%       unc-119 5.9e-13 3283 Expression:yes
%       egl-10 5.8e-12 11970 Expression:yes
%       unc-51 1.7e-11 3980 Expression:yes
%       nhr-74 1.1e-10 4548 Expression:no
%       C32E8.7 1.7e-10 1200 Expression:yes
%       jnk-1 1.9e-10 14208 Expression:yes
%       elg-10 2.8e-09 8943 Expression:no
%       rab-3 5.5e-09 3064 Expression:yes
%       rgs-1 6.8e-09 2400 Expression:yes
%       jkk-1 6.8e-09 3601 Expression:yes
%       F42A10.3 4.1e-08 6626 Expression:no
%       Snt-1 6.6e-08 6205 Expression:yes
%       unc-54 1e-07 1894 Expression:no
%       gpa-14 1.2e-07 3000 Expression:no
%       osm-3 2.6e-07 1889 Expression:no
%       myo-3 7.4e-07 3751 Expression:no
%       eat-4 1.6e-06 2340 Expression:no
%       tax-2a 2.4e-06 6409 Expression:no
%       T28B4.1 1.8e-05 5272 Expression:no
%       W05B10.4 2.8e-05 6020 Expression:no
%       sng-1 0.00024 4998 Expression:yes
%       unc-97 0.00032 2175 Expression:no
%       unc-112 0.00036 2947 Expression:no
%       nhr-89 0.00041 2010 Expression:no
%       tax-2c 0.00043 1129 Expression:no
%       gpa-3 0.00059 6001 Expression:no
%       tax-4 0.0012 17056 Expression:no
%       gpa-15 0.0063 3000 Expression:no
%       che-3 0.0074 2600 Expression:no
%       Flp-6 0.008 2951 Expression:no
%       kin-8a 0.0081 5883 Expression:no
%       F54C9.7 0.014 4927 Expression:no
%       nhr-81 0.022 3269 Expression:no
%       Sra-9 0.027 4000 Expression:no
%       ncs-1 0.03 3000 Expression:no
%       nhr-73 0.043 2536 Expression:no
%       nhr-82 0.061 3688 Expression:no
%       ceh-22 0.13 4000 Expression:no
%       Sra-7 0.15 7685 Expression:no
%       F44B9.2 0.17 1745 Expression:no
%       nhr-75 0.63 3421 Expression:no
%       F07C3.4 1.6 6956 Expression:no
%       nhr-72 2.1 3096 Expression:no
%       odr-10 2.2 1000 Expression:no
%


@relation 'CEPanNeural-weka.filters.AttributeFilter-R5,23'

@attribute Expression {yes,no}
@attribute M1 string
@attribute M2 string
@attribute M3 string
@attribute M5 string
@attribute M6 string
@attribute M7 string
@attribute M8 string
@attribute M9 string
```

```
@attribute M10 string
@attribute M11 string
@attribute M12 string
@attribute M13 string
@attribute M14 string
@attribute M15 string
@attribute M16 string
@attribute M17 string
@attribute M18 string
@attribute M19 string
@attribute M20 string
@attribute M21 string
@attribute M23 string
@attribute M24 string
@attribute M25 string
@attribute M26 string
@attribute M27 string
@attribute M28 string
@attribute M29 string
@attribute M30 string

@data

yes,'{3462:3478}','{8816:8832}.....
.....
```

APPENDIX B –Rules part of the AllRules models (minsup=0.2, minconf=0.2, cvd=0.5) for CBriggsae.

1.CBASENeural

```
1. M2 && M11 ==> CBASENeural=yes  [Conf: 0.6666667, Sup: 0.516129]
        M2      M11
  cvd   X       0.463
M2
  mean  X       852.0
  sdev  X       395.0
        _       _
2. M8 && M10 ==> CBASENeural=yes  [Conf: 0.75, Sup: 0.38709676]
        M8      M10
  cvd   X       0.481
M8
  mean  X       634.0
  sdev  X       305.0
        _       _
3. M1 && M13 ==> CBASENeural=yes  [Conf: 0.6923077, Sup: 0.29032257]
        M1      M13
  cvd   X       0.489
M1
  mean  X       924.0
  sdev  X       453.0
        _       _
4. M3 && M29 ==> CBASENeural=yes  [Conf: 0.78571427, Sup: 0.3548387]
        M3      M29
  cvd   X       0.465
M3
  mean  X       945.0
  sdev  X       440.0
        _       _
5. M9 && M16 ==> CBASENeural=yes  [Conf: 1.0, Sup: 0.32258064]
        M9      M16
  cvd   X       0.355
M9
  mean  X       833.0
  sdev  X       296.0
        _       _
6. M1 && M17 ==> CBASENeural=yes  [Conf: 0.5714286, Sup: 0.2580645]
        M1      M17
  cvd   X       0.458
M1
  mean  X       850.0
  sdev  X       390.0
        _       _
7. M2 && M6 ==> CBASENeural=yes  [Conf: 0.5714286, Sup: 0.2580645]
        M2      M6
  cvd   X       0.284
M2
  mean  X       844.0
  sdev  X       240.0
        _       _
8. M3 && M20 ==> CBASENeural=yes  [Conf: 0.8, Sup: 0.2580645]
        M3      M20
  cvd   X       0.467
M3
  mean  X       1103.0
  sdev  X       515.0
        _       _
9. M12 && M16 ==> CBASENeural=yes  [Conf: 1.0, Sup: 0.29032257]
        M12     M16
  cvd   X       0.468
M12
```

```
  mean  X      740.0
  sdev  X      347.0
```

2.CBASKNeural

```
1.  M1 && M6 ==> CBASKNeural=yes  [Conf: 0.6923077, Sup: 0.29032257]
        M1      M6
  cvd   X      0.453
M1
  mean  X      737.0
  sdev  X      334.0
        _       _
2. M7 && M10 ==> CBASKNeural=yes  [Conf: 0.8125, Sup: 0.41935483]
        M7      M10
  cvd   X      0.460
M7
  mean  X      785.0
  sdev  X      361.0
        _       _
3. M1 && M9 ==> CBASKNeural=yes  [Conf: 0.6666667, Sup: 0.32258064]
        M1      M9
  cvd   X      0.448
M1
  mean  X      742.0
  sdev  X      333.0
        _       _
4. M1 && M11 ==> CBASKNeural=yes  [Conf: 0.90909094, Sup: 0.32258064]
        M1      M11
  cvd   X      0.431
M1
  mean  X      945.0
  sdev  X      407.0
        _       _
5. M1 && M14 ==> CBASKNeural=yes  [Conf: 0.75, Sup: 0.29032257]
        M1      M14
  cvd   X      0.376
M1
  mean  X      879.0
  sdev  X      331.0
        _       _
6. M7 && M13 ==> CBASKNeural=yes  [Conf: 0.8666667, Sup: 0.41935483]
        M7      M13
  cvd   X      0.465
M7
  mean  X      694.0
  sdev  X      323.0
        _       _
7. M5 && M10 ==> CBASKNeural=yes  [Conf: 0.90909094, Sup: 0.32258064]
        M5      M10
  cvd   X      0.452
M5
  mean  X      1134.0
  sdev  X      514.0
        _       _
8. M5 && M19 ==> CBASKNeural=yes  [Conf: 0.8888889, Sup: 0.2580645]
        M5      M19
  cvd   X      0.316
M5
  mean  X      595.0
  sdev  X      188.0
        _       _
9. M5 && M10 && M19 ==> CBASKNeural=yes  [Conf: 0.8888889, Sup: 0.2580645]
        M5      M10     M19
  cvd   X      0.487   0.316
M5
  mean  X      1046.0  595.0
```

```
  sdev  X      510.0   188.0
        _      _       _
  cvd   X      X       0.463
M10
  mean  X      X       834.0
  sdev  X      X       386.0
        _      _       _
10. M19 && M21 ==> CBASKNeural=yes  [Conf: 1.0, Sup: 0.41935483]
       M19     M21
  cvd   X      0.497
M19
  mean  X      808.0
  sdev  X      402.0
        _      _
11. M21 && M22 ==> CBASKNeural=yes  [Conf: 1.0, Sup: 0.32258064]
       M21     M22
  cvd   X      0.302
M21
  mean  X      872.0
  sdev  X      264.0
        _      _
12. M2 && M19 ==> CBASKNeural=yes  [Conf: 0.7777778, Sup: 0.4516129]
       M2      M19
  cvd   X      0.488
M2
  mean  X      868.0
  sdev  X      424.0
        _      _
13. M11 && M24 ==> CBASKNeural=yes  [Conf: 1.0, Sup: 0.2580645]
       M11     M24
  cvd   X      0.474
M11
  mean  X      972.0
  sdev  X      461.0
        _      _
14. M1 && M19 ==> CBASKNeural=yes  [Conf: 0.78571427, Sup: 0.3548387]
       M1      M19
  cvd   X      0.404
M1
  mean  X      918.0
  sdev  X      372.0
        _      _
15. M1 && M27 ==> CBASKNeural=yes  [Conf: 0.75, Sup: 0.29032257]
       M1      M27
  cvd   X      0.434
M1
  mean  X      798.0
  sdev  X      347.0
        _      _
16. M6 && M22 ==> CBASKNeural=yes  [Conf: 0.8888889, Sup: 0.2580645]
       M6      M22
  cvd   X      0.398
M6
  mean  X      647.0
  sdev  X      257.0
        _      _
17. M2 && M14 ==> CBASKNeural=yes  [Conf: 0.71428573, Sup: 0.32258064]
       M2      M14
  cvd   X      0.488
M2
  mean  X      806.0
  sdev  X      394.0
        _      _
18. M14 && M21 ==> CBASKNeural=yes  [Conf: 1.0, Sup: 0.29032257]
       M14     M21
  cvd   X      0.334
M14
  mean  X      1035.0
```

```
  sdev  X      346.0
        _      _
19. M22 && M24 ==> CBASKNeural=yes  [Conf: 1.0, Sup: 0.29032257]
        M22    M24
  cvd   X      0.470
M22
  mean  X      753.0
  sdev  X      354.0
        _      _
```

3.CBBodyWall

```
1. M5 && M17 ==> CBBodyWallExpr=no  [Conf: 0.71428573, Sup: 0.32258064]
        M5     M17
  cvd   X      0.474
M5
  mean  X      522.0
  sdev  X      248.0
        _      _
2. M2 && M12 ==> CBBodyWallExpr=yes  [Conf: 0.4285714, Sup: 0.29032257]
        M2     M12
  cvd   X      0.483
M2
  mean  X      936.0
  sdev  X      452.0
        _      _
3. M2 && M5 ==> CBBodyWallExpr=no  [Conf: 0.6363636, Sup: 0.4516129]
        M2     M5
  cvd   X      0.473
M2
  mean  X      1006.0
  sdev  X      477.0
        _      _
4. M2 && M12 ==> CBBodyWallExpr=no  [Conf: 0.57142854, Sup: 0.38709676]
        M2     M12
  cvd   X      0.488
M2
  mean  X      789.0
  sdev  X      385.0
        _      _
5. M3 && M17 ==> CBBodyWallExpr=no  [Conf: 0.6666667, Sup: 0.32258064]
        M3     M17
  cvd   X      0.373
M3
  mean  X      981.0
  sdev  X      366.0
        _      _
6. M8 && M13 ==> CBBodyWallExpr=no  [Conf: 0.6666667, Sup: 0.32258064]
        M8     M13
  cvd   X      0.481
M8
  mean  X      1092.0
  sdev  X      526.0
```

4. CBOLLNeural

```
1. M1 && M7 ==> CBOLLNeural=yes  [Conf: 0.6315789, Sup: 0.38709676]
        M1     M7
  cvd   X      0.439
M1
  mean  X      979.0
```

```
  sdev  X      430.0
        _      _
2. M1 && M9 ==> CBOLLNeural=yes  [Conf: 0.6923077, Sup: 0.29032257]
        M1     M9
  cvd   X      0.440
M1
  mean  X      876.0
  sdev  X      386.0
        _      _
3. M1 && M8 ==> CBOLLNeural=yes  [Conf: 0.73333335, Sup: 0.3548387]
        M1     M8
 cvd    X      0.495
M1
  mean  X      744.0
  sdev  X      368.0

        __     __
4. M1 && M18 ==> CBOLLNeural=yes  [Conf: 0.58823526, Sup: 0.32258064]
        M1     M18
  cvd   X      0.481
M1
  mean  X      798.0
  sdev  X      384.0
        _      _
5. M6 && M8 ==> CBOLLNeural=yes  [Conf: 0.8888889, Sup: 0.2580645]
        M6     M8
  cvd   X      0.448
M6
  mean  X      936.0
  sdev  X      420.0
        _      _
6. M8 && M24 ==> CBOLLNeural=yes  [Conf: 0.8888889, Sup: 0.2580645]
        M8     M24
  cvd   X      0.423
M8
  mean  X      1002.0
  sdev  X      425.0
        _      _
7. M2 && M17 ==> CBOLLNeural=yes  [Conf: 0.8, Sup: 0.2580645]
        M2     M17
  cvd   X      0.432
M2
  mean  X      498.0
  sdev  X      215.0
        _      _
8. M2 && M18 ==> CBOLLNeural=yes  [Conf: 0.57894737, Sup: 0.3548387]
        M2     M18
  cvd   X      0.433
M2
  mean  X      849.0
  sdev  X      368.0
```

5. CBPanNeural

```
1. M1 && M7 ==> CBPPanNeuralExpr=yes  [Conf: 0.8666667, Sup: 0.41935483]
        M1     M7
  cvd   X      0.423
M1
  mean  X      995.0
  sdev  X      422.0
        _      _
2. M1 && M12 ==> CBPPanNeuralExpr=yes  [Conf: 0.60714287, Sup: 0.5483871]
        M1     M12
  cvd   X      0.495
M1
```

```
  mean  X      847.0
  sdev  X      420.0
        _      _
3. M2 && M7 ==> CBPPanNeuralExpr=yes  [Conf: 0.8666667, Sup: 0.41935483]
        M2     M7
  cvd   X      0.391
M2
  mean  X      1001.0
  sdev  X      392.0
        _      _
4. M1 && M25 ==> CBPPanNeuralExpr=yes  [Conf: 0.5555555, Sup: 0.48387095]
        M1     M25
  cvd   X      0.471
M1
  mean  X      893.0
  sdev  X      421.0
        _      _
5. M6 && M28 ==> CBPPanNeuralExpr=yes  [Conf: 0.6, Sup: 0.29032257]
        M6     M28
  cvd   X      0.485
M6
  mean  X      468.0
  sdev  X      227.0
        _      _
6. M24 && M28 ==> CBPPanNeuralExpr=yes  [Conf: 0.9166667, Sup: 0.3548387]
        M24    M28
  cvd   X      0.446
M24
  mean  X      863.0
  sdev  X      385.0
        _      _
7. M1 && M16 ==> CBPPanNeuralExpr=yes  [Conf: 0.54545456, Sup: 0.38709676]
        M1     M16
  cvd   X      0.312
M1
  mean  X      886.0
  sdev  X      277.0
        _      _
8. M16 && M28 ==> CBPPanNeuralExpr=yes  [Conf: 0.5625, Sup: 0.29032257]
        M16    M28
  cvd   X      0.433
M16
  mean  X      790.0
  sdev  X      342.0
        _      _
9. M2 && M16 ==> CBPPanNeuralExpr=no  [Conf: 0.45454547, Sup: 0.32258064]
        M2     M16
  cvd   X      0.436
M2
  mean  X      1196.0
  sdev  X      522.0
        _      _
10. M12 && M13 ==> CBPPanNeuralExpr=no  [Conf: 0.5, Sup: 0.3548387]
        M12    M13
  cvd   X      0.488
M12
  mean  X      835.0
  sdev  X      408.0
        _      _
```