Discovering Geometric Patterns in Genomic Data

Wenxuan Gao
Department of Computer Science
University of Illinois at Chicago
wgao5@uic.edu

Christopher Brown
Institute for Genomics & Systems Biology
caseybrown@uchicago.edu

Robert L. Grossman
Institute for Genomics & Systems Biology
robert.grossman@uchicago.edu

Lijia Ma
Institute for Genomics & Systems Biology
ljma@uchicago.edu

Matthew Slattery
Institute for Genomics & Systems Biology
mgslattery@uchicago.edu

Kevin P. White
Institute for Genomics & Systems Biology
kpwhite@uchicago.edu

Philip S. Yu
Dept. of Computer Science
University of Illinois at Chicago
psyu@cs.uic.edu

ABSTRACT

ChIP-chip and ChIP-seq are techniques for the isolation and identification of the binding sites of DNA-associated proteins along the genome. Both techniques produce genome-wide location data. The geometric arrangements of these binding sites can provide valuable information about biological function, such as the activation or repression of genes.

In this paper, we formalize this problem and propose a novel graph based algorithm called Patterns of Marks (PoM) to discover efficiently these types of geometric patterns in genomic data. We also describe how we validate the algorithm using experimental data.

Categories and Subject Descriptors

J.3 [**Life and Medical Science**]: Biology and genetics; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Algorithms

Keywords

geometric pattern, DNA binding sites, graph mining

1. INTRODUCTION

The genome has been sequenced for some time and a fundamental biological challenge now is to understand how genomic sequences code biological function. Biological function is determined not just by genes, but also by genomic sequences that code repressors, activators, and other regulatory structures, such as chromatin regulators, that determine how genes are transcribed into RNA. There are laboratory techniques, such as ChIP-chip and ChIP-seq technology, that can help identify these types of regulatory structures by attaching certain proteins to the genome at what are called binding sites. The geometric combination of these binding sites provide valuable information about biological function and finding out such genomic patterns (A precise definition is given in Section 3.2) can offer new insight into the mechanism of regulation.

In this paper, 1) we abstract and formalize this problem as the discovery of specific types of geometric patterns in genomic data (Geometric Pattern Discovery); 2) we propose an algorithm called PoM for efficiently discovering these types of geometric patterns in genomic data based upon frequent subgraphs; 3) we evaluate the PoM on experimental data to validate its usefulness.

Marks along the genome. There are several different types of proteins that bind to certain regions along the genome. In this paper, we use the term *chromatin factor* to refer to transcription factors and chromatin regulators, both of which bind along the genome at binding sites. You can think of these binding sites occurring in *Regions of Interest (ROI)* along the genome. For simplicity, in this paper, we abstract this simply as a *mark* along the genome associated with a *factor*. A mark is not a single point along the genome but rather an interval or region along the genome. It is important to note there are many types of marks, depending upon the specific protein that binds to the genome, and in general, each protein binds in *multiple places* along the genome. This is contrary to many of the genomic patterns studied previously in the KDD community involving gene-gene interactions, or gene-protein interactions, in which a gene occurs in *one* region along the genome.

Biological significance. Genome-wide protein-DNA binding site data are now available for transcription factors and

ACM-BCB '12, October 7-10, 2012, Orlando, FL, USA

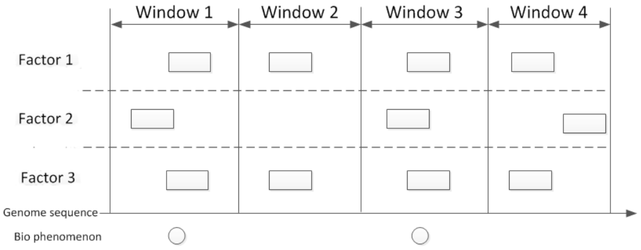

Figure 1: Geometric Pattern Example.

chromatin regulators for many species. There are also databases that summarize what is known about marks along the genome for certain species, such as the fly [11]. A number of studies have shown that patterns of marks can offer novel insight into the mechanisms of regulation[8][19][10]. In general, pairs of marks have been studied, but not triples of marks, or more complex structures involving marks. Part of the motivation in this paper is introduction a technique for studying more complex geometric patterns involving marks along the genome, not just pairs of marks.

As Figure 1 shows, the horizontal axis stands for the whole genome sequence. The top three rows indicate the binding sites of the three chromatin factors on the genome, and each box is a mark of a given chromatin factor. The bottom row indicates the sites (identified by circles) where a particular biological phenomenon happens. In this figure, there are four windows, and each is a region taken from the genome. In this hypothetical example, the biological phenomenon (identified by the circle) is present when Factors 1, 2 and 3 are present, when the binding sites of Factors 1 and 3 are downstream of the binding site of Factor 2, and when the binding sites of Factors 1 and 3 partially overlap the binding site of Factor 2. In this example, if Factor 2 binds downstream of Factors 1 and 3, the biological phenomenon does not occur. In this particular example, the binding sites of Factor 1 and Factor 3 completely overlap each other and we may or may not want to make this relationship a property of the geometric pattern.

We now briefly motivate why geometric patterns of marks are important in understanding biological function. Recall that a gene is called *active* if it codes RNA and the term *enriched* is used for a phenomenon that occurs more frequently compared to another phenomenon.

As a first example, if a geometric pattern of several chromatin factors is enriched near active genes and rarely observed around inactive genes, then presumably this pattern is associated with the activation of genes.

Recall that an *enhancer* is a short region of DNA that binds to certain proteins to regulate the transcription levels of genes. Traditionally, identifying enhancers is done through expensive and time consuming laboratory work. On the other hand, ChIP-chip experiments can relatively inexpensively identify a collection of marks and their locations along the genomes. As a second example, given the location of sufficient number of enhancers identified with direct experiments, an algorithm for identifying geometric patterns along the genome that are enriched near enhancers can be used to *predict* the location of other enhancers from ChIP-chip data alone, without doing expensive experiments.

Order matters. The order of marks is an important feature and cannot be simply ignored. As an example, whether a regulatory region is upstream or downstream of a gene is important. For another example, look at the difference between window 1 and 4 again in Figure 1. We are interested in the following problem: *given marks along a genome and regions of biological interest, identify the geometric patterns of marks that are associated with regions of biological interest.* We give a formal definition below.

If the order of marks was not important, then this problem could be solved using standard association rules [5], [6]. However, order does matter, and we need a new algorithm to identify relevant geometric patterns efficiently.

In this paper, we abstract the problem of finding geometric patterns of marks that are associated with a specific biological function. We then introduce a new algorithm called Pattern of Marks or PoM that uses graph mining to identify geometric patterns that tend to be associated with the biological function. We show that the PoM algorithm can be used to predict the specific biological function (instead of merely identifying it). Finally, we perform several experimental studies to validate the PoM algorithm.

In summary, our main contributions in this paper can be summarized as the following:

1. A new concept, geometric patterns of marks, is proposed to study the combinatorial properties of multiple chromatin factors. For the first time, we formalize this problem.

2. We introduce a graph representation for the marks and their geometric relations associated with chromatin factors.

3. We introduce an algorithm called Patterns of Marks (PoM) for identifying geometric patterns of marks that are associated with a desired biological phenomenon.

4. As an extension, we build classifiers based on patterns of marks for known promoters of Drosophila melanogaster and show that the pattern of marks can predict promoters with good accuracy.

The rest of the paper is organized as follows. We first formalize the problem in Section 2. In section 3, we present the novel graph representation for marks associated with chromatin factors and motivate the framework to solve the problem. In Section 4, we show how sliding windows along the genome lead naturally to patterns of marks. In section 5, we define a qualitative measure and show how to use it to find the patterns of marks that are highly associated with a specific biological phenomenon. In Section 6, we extend our work for promoter prediction. Section 7 contains an experimental study to validate our approach and an analysis of the results. Section 8 is a summary.

2. PROBLEM FORMALIZATION

Our goal is to find out the geometric patterns that can reveal the correlation between multiple chromatin factors and a given biological phenomenon. We formalize this problem in this section. We first describe and define the related elements and then we formally define the problem.

The biological phenomenon could be any biological structure or process, such as the presence of enhancers or promoters along the genome. In general, the phenomenon occurs at many locations across the genome. We denote the set of sites where that biological phenomenon happens as $S = \{s_1, s_2, s_3, ...\}$, where each s_i is one such site. The site s_i is usually given as a point. However, sometimes it is given as a region, with the start point v_i and the end point w_i of the region specified. In that case, we take s_i as the midpoint $s_i = (v_i + w_i)/2$.

We denote the (chromatin) factors as M_1, M_2, M_3,..., and M_n. Usually the binding sites of chromatin factors are given as regions, which have start positions and stop positions. We call the regions marks or Regions of Interest(ROIs). For each chromatin factor M_i, the set of ROIs is denoted as R_i= $\{R_i^1$,R_i^2,... $R_i^{k_i}\}$, where k_i is the number of ROIs for M_i. Each ROI R_i^j has a start position denoted by $Start(R_i^j)$ and a stop position denoted by $Stop(R_i^j)$ on the genome.

In this paper, from the binding sites data of several chromatin factors, we want to identify the geometric patterns that are enriched[1] around the sites of the given biological phenomenon while being rarely observed at a random position across the whole genome. We denote such a set of patterns by $P = \{p_1, p_2, ...\}$, where each p_i is a particular geometric combination of ROIs of the involved chromatin factors. In summary, the problem can be formalized as the following:

Definition 1. **Geometric Pattern Discovery.** Assume there are n factors M_1, M_2, M_3,..., and M_n and that each factor has a set of binding sites along the genome that we call marks or Regions of Interest (ROIs). Denote the ROIs of factor M_i by $\{R_i^1$,R_i^2,... $R_i^{k_i}\}$, where k_i is the number of ROIs for M_i. Each R_i^j has a start position denoted by $Start(R_i^j)$ and a stop position denoted by $Stop(R_i^j)$. Also assume that across the genome a specific biological phenomenon happens at a set of sites $S = \{s_1, s_2, s_3, ...\}$. The goal is to identify geometric patterns $P = \{p_1, p_2, ...\}$ that occur nearby S with a much higher probability[2] than that of a random position on the genome, where p_i is a geometric combination of ROIs of the involved factors.

Figure 1 contains an example. In this paper, we always assume that patterns of marks are local in the sense that are contained in a window. A simple example of a pattern is to require that Factor 2 must occur before (upstream) of Factor 1. This pattern occurs twice in this figure.

3. GRAPH THEORETIC FRAMEWORK

In this section, we introduce a graph theoretic framework for representing geometric patterns of marks in genomic data.

3.1 Basic Graph Concepts

In this subsection, we introduce some basic graph concepts related to our work.

Definition 2. **Graph**. A graph is denoted by $g = (V, E)$, where V is a set of nodes and E is a set of edges connecting the nodes. Both nodes and edges may have labels. In a graph, each node has a unique ID.

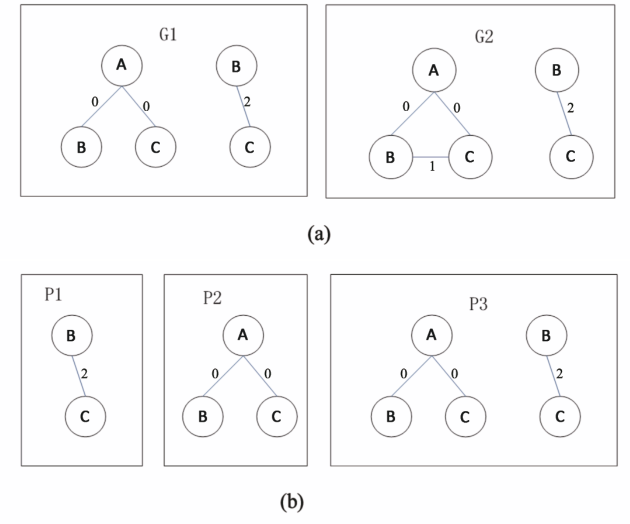

Figure 2: Graph and Subgraph Example.

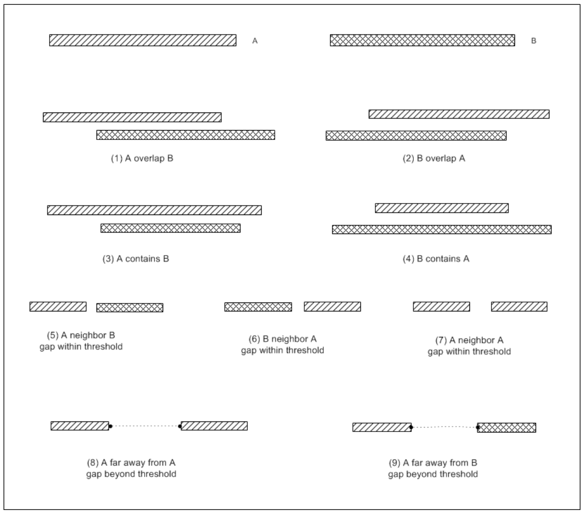

Figure 3: Geometric Relation of ROIs.

For example, in Figure 2(a), there are two graphs G1 and G2. The text in each node is the node label, and the text on each edge is the edge label. Two nodes in a graph may have the same label, but their IDs are different.

Definition 3. **Subgraph Isomorphism**. The label of a node u is denoted as $l(u)$ and the label of an edge (u, v) is denoted as $l((u, v))$. Given two graphs $g' = (V', E')$ and $g = (V, E)$, g' is a subgraph of g, if there is an injective function $f : g' \to g$ such that
1. $\forall v \in V', f(v) \in V$ such that $l(u) = l(f(v))$.
2. $\forall (u, v) \in E', (f(u), f(v)) \in E$ such that $l((u, v)) = l((f(u), f(v)))$.

For example, in Figure 2(b), P1, P2 and P3 are subgraphs of G1 and G2 in Figure 2(a).

3.2 Geometric Relation of ROIs

In this subsection, we examine the geometric relations among ROIs of multiple factors. In general, the geometric

[1] Biologists commonly use the term enriched to apply to a phenomenon that is more highly present than would be expected from random behavior.

[2] We are also interested in geometric patterns that occur with a much lower probability than would be expected if there were no relationship between the marks and the sites S.

relations of two ROIs of different factors can be categorized into three categories: overlapping, containing and neighboring. More precisely, given two ROIs from two factors A and B, we have 9 cases as Figure 3 shows. In the following paragraphs, we let A and B denote two factors, and R_A and R_B denote their respective ROIs.

Definition 4. **Overlapping**. Given two ROIs of different factors R_A and R_B, they have some overlapping area but none of them completely contains the other one.

Figure 3 parts (1) and (2) shows two examples of overlaps. In Case (1), we have R_A is upstream of R_B, with $(Start(R_A) < Start(R_B)) \wedge (Stop(R_A) > Start(R_B)) \wedge (Stop(R_A) < Stop(R_B))$. In Case (2), we have R_B is upstream of R_A.

Definition 5. **Containing**. Given two ROIs of different factors R_A and R_B, we say that R_B contains R_A if R_A is completely inside the region R_B.

If R_A contains R_B, then $(Start(R_A) \leq Start(R_B)) \wedge (Stop(R_A) \geq Stop(R_B))$. Parts (3) and (4) of Figure 3 shows two examples of containing.

Definition 6. **Neighboring**. Given a threshold θ, two ROIs R_A and R_B of factors A and B are called neighbors if they are within a distance of θ of each other.

In Figure 3, parts (5) and (6) are examples of neighbors. In (5), we have $Stop(R_A) < Start(R_B)$ and $Start(R_B) \leq Stop(R_A) + \theta$. In (6), we have $Stop(R_B) < Start(R_A)$ and $Start(R_A) \leq Stop(R_B) + \theta$.

Cases (7) and (8) show two possibilities for the geometric relations of two ROIs of the same factor. In one case, but not the other, they are close enough to be neighbors.

Finally, when two ROIs from two different chromatin factors are far away from each other, they are considered not to have a (local) relation. That is the case (9) in Figure 3.

3.3 Graph representation

In this section, we describe a method for coding geometric patterns of marks as graphs. The method uses a fixed length window that slides along the genome. As Table 1 shows, we assign a unique node label for each factor so that different factors have different labels. In a fixed window, there can be several ROIs of the same factor. In that case, the corresponding nodes in the graph have different node IDs but share the same node label.

Table 1: Node Labels

chromatic factor	*node label*
cbp	0
h3k27ac	1
h3k27me3	2
h3k4me1	3
h3k4me3	4
h3k9ac	5
h3k9me3	6
polII	7

The geometric relation between two ROIs is represented by the edge label. We label each edge with a key from 0 to 7 using Table 2. Note that if we choose to use undirected graphs, then for two ROIs of different factors (two nodes of different labels), we cannot distinguish the case A contains B from the case B contains A. In this paper, we consider A as the ROI with the smaller node label value and B as the one with the larger label value.

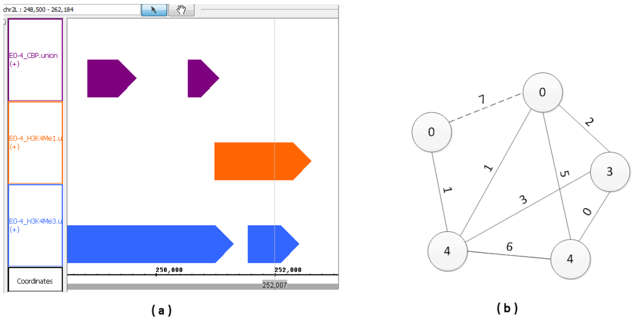

Figure 4: A Conversion Example.

Generally speaking, we assume there is no relation between two ROIs when the distance between them is greater than the threshold θ. It is useful though, as we will see below, to add a special edge in this case, which we call a virtual edge.

When we code geometric patterns of marks in this way, we can use frequent subgraph mining [21], [18] and [13] to extract frequent patterns. For our studies below, we use the gSpan [21] algorithm to extract frequent subgraphs due to its good computational efficiency.

The gSpan algorithm can only find patterns that lie within a connected component. For example, in Figure 2(a), suppose the graph is the union of G1 and G2. If we perform gSpan with a support of 2, the frequent subgraphs returned by gSpan will be Patterns P1 and P2 in Figure 2(b). P3 will not be found because it is not a connected subgraph.

However, if a geometric pattern of marks occurs more often than would be expected from a random placement of marks, it may represent an interesting biological phenomenon. Thus, there is a need to discover such patterns. Existing Apriori-based mining solutions [14], [16] can find such patterns but face significant overheads when they join two existing graph patterns to form larger candidate sets [20].

In the paper, we solve this problem by using virtual edges. Since the virtual edge has a distinct edge label, it can be easily distinguished from the other labeled edges. Note that adding virtual edges makes the nodes from the same factor a connected clique, and so gSpan can be used to pull out frequent subgraphs even if they span multiple components.

Figure 4 contains an example of coding geometric patterns

Table 2: Edge Labels

geometry relation	*edge label*
A contains B	0
B contains A	1
A overlap B	2
B overlap A	3
A neighbor B	4
B neighbor A	5
A neighbor A	6
A faraway A	7

of marks as graphs. Figure 4(a) shows a view of 3 chromatin factors between 248500 and 252000 on fly chromosome 2L using visualization tool GeneBrowser [3]. In this figure, each row gives the binding sites for a chromatin factor: row 1 for the factor cbp, row 2 for h3k4me1, and row 3 for h3k4me3. Their node labels are given in Table 1. Using the edge labels given in Table 2, we get Figure 4(b) as the graph representation for the window from 248500 to 252000 in Figure 4(a). In Figure 4(a), there are 2 ROIs of cbp, 1 ROI of h3k4me1, and 2 ROIs of h3k4me3 in the window. This gives 5 nodes in the graph representation. There is one virtual edge in the graph and its edge label is 7. Note that it is not necessary for an ROI to be completely included in the window to be considered as a node. As long as part of the ROI is in the window, we create a node. Although h3k4me1 and h3k4me3 are cut off by the boundary of the window, they are still represented in the graph in Figure 4(b).

The representation is quite flexible and we can simply change the definition of node labels and edge labels to adapt to various applications. If we would like to include more factors for the application, we can define some new node labels. If we do not want to differentiate between overlapping and containing, we can set their edge labels to the same value.

3.4 Framework of proposed approach

Once the graph is formed, it is straightforward to extract the geometric patterns of marks. Here are the steps:

1. Step 0: Generate the genome-wide background graph dataset G_1 from the marks.
2. Step 1: Add the locations of the phenomenon of biological interest to create application-specific target graph dataset G_2.
3. Step 2: Given a support s, find all frequent patterns in G_2, denoted by $P = \{p_1, p_2, ...\}$.
4. Step 3: For each p_i in P, calculate its frequency in G_1 as well as its log ratio score and sort P in descending order.
5. Step 4: Analyze the patterns having high log ratio scores.

4. GRAPH GENERATION ALGORITHM

In this section, we describe the PoM Algorithm for generating graph datasets from the ROIs data of different chromatin factors across the genome.

Algorithm 1 describes the Graph Generating Algorithm. Lines 2 ∼ 8 generate the genome-wide background dataset and lines 9 ∼ 15 generate the phenomenon-oriented target graph dataset. In the algorithm, the size and step of the sliding window are controllable and can be adjusted by the user for different needs.

Given the start position s and an end position d, the GenerateOneSlidingGraph Algorithm generates the graph representation for the factors falling within the region $[s, d]$. Lines 2 ∼ 10 add all the factors within$[s, d]$ as nodes in the graph, and lines 11 ∼ 18 check every pair of nodes and add the corresponding edges. The algorithm returns *null* for empty graphs. The GetEdge function checks the geometric relation of two nodes and returns the corresponding edge with the proper label. It returns *null* for two faraway ROIs of different chromatin factors.

Algorithm 1 PoM Graph Generation Algorithm.

```
Require:
    size: The length of a sliding window;
    step: How far to slide the moving window;
    n: The length of the whole genome;
    $\theta$: Neighboring threshold;
    R: The set of ROIs of all the chromatin factors;
    S: The set of sites where the biological phenomenon happens;
Ensure:
    $G_1$: Genome-wide background graph dataset;
    $G_2$: application-specific target graph dataset;
 1: $G_1 = \{\}$, $G_2 = \{\}$
 2: Sorting R in the ascending order of the start position.
 3: for $i = 0$; $i * step < n$; $i++$ do
 4:     $g_i = GenerateOneSlidingGraph(i * step, i * step + size, \theta, R)$;
 5:     if $g_i$ is not null then
 6:         $G_1 = G_1 \bigcup g_i$
 7:     end if
 8: end for
 9: for all $p \in S$ do
10:     $c$ = The centroid of p;
11:     $g' = GenerateOneSlidingGraph(c - size/2, c + size/2, \theta, R)$;
12:     if $g'$ is not null then
13:         $G_2 = G_2 \bigcup g'$;
14:     end if
15: end for
16: return $G_1$, $G_2$;
```

5. SIGNIFICANT PATTERNS MINING

In this section, we describe how to find out the significant geometric patterns after generating the graph. The method is presented in algorithm 3.

It is important to sort by significance the frequent subgraphs produced. Since our interest is in geometric patterns of marks that occur relatively frequently over biological phenomena of interest, but relatively rarely otherwise, we introduce a score to quantify this. The score function is defined based on the pattern's positive frequency and background frequency.

Definition 7. **Positive frequency**. The positive frequency $f_p(p)$ of a subgraph pattern p is the ratio of the number of target graphs containing p to the total number of target graphs.

Definition 8. **Background frequency**. The background frequency $f_b(p)$ of a subgraph pattern p is the ratio of the number of background graphs (i.e. graphs that are *not* over a region representing a phenomenon of biological interest) containing p to the total number of background graphs.

Definition 9. **Log ratio score**. The log ratio score of a subgraph pattern p is the log ratio of the positive frequency to the background frequency of p and is defined as:

$$\text{log ratio score of pattern } p = log\frac{f_p(p)}{f_b(p)}$$

The log ratio score is used to estimate the interest of the pattern p_i. A high log ratio score means the pattern has a much higher chance to be present over phenomena of biological interest compared to a random place along the genome.

```
Algorithm 2 $GenerateOneSlidingGraph(s, d, \theta, R)$
Require:
    s: The start position of the current window;
    d: The end position of the current window;
    $\theta$: Neighboring threshold;
    R = $R_1$,$R_2$,...: The list of ROIs of all the chromatin factors;
Ensure:
    $g = (V, E)$: the graph representation of the current sliding window;
$V = \{\}$, $E = \{\}$
for $i = 0$; $i < R.size$; $i++$ do
    if $R_i < s$ then
        continue;
    end if
    if $R_i > d$ then
        break;
    end if
    $V = V \bigcup v(R_i)$;
end for
for $i = 0$; $i < V.size$; $i++$ do
    for $j = i + 1$; $j < V.size$; $j++$ do
        $e = GetEdge(V_i, V_j, \theta)$;
        if $e$ is not $null$ then
            $E = E \bigcup e$;
        end if
    end for
end for
return $g = (V, E)$;
```

The higher the score is, the more interesting the pattern is.

In algorithm 3, lines 2 $\sim$ 7 calculate the log ratio score for all the discovered frequent subgraph patterns and discard those with low scores. A set of patterns with a high score in the descending order of the log ratio score is returned for further study.

6. PREDICTION MODEL

As an application of our PoM Algorithm, we extend our work to predict promoters.

Theoretically, any significant pattern discovered in the previous section could be used for prediction. However, the recall of a single pattern is sometimes poor in practice, and we have been able to build more powerful classifiers by using multiple patterns. In order to improve the recall, we need to select a set of patterns and combine them together to build the prediction model. This can be done in a number of ways. We can choose the top k highest ranking patterns, top ranking patterns after eliminating some patterns, or use interesting patterns suggested by a domain expert.

To take advantage of a selected set of significant patterns to build the prediction model, we use feature vectors as follows. Given n selected patterns, we create an n dimensional feature vector, with each dimension of the vector corresponding to a pattern. Then each graph instance can be converted into a vector V. The value of $V[i]$ (the ith dimension) is 1 if the ith pattern is a subgraph of the corresponding graph instance; otherwise, $V[i]$ is 0. After converting all graph data into n dimensional data points, we use a SVM algorithm[9] to train the prediction model.

```
Algorithm 3 Pattern Mining Algorithm .
Require:
    $\theta$: The minimum support rate to be frequent;
    $\delta$: The score threshold of being a discriminative pattern;
    $G_1$: The genome-wide background graph dataset;
    $G_2$: The application-specific target graph dataset;
Ensure:
    $P$: The discriminative patterns;
1: Finding all frequent subgraphs patterns in $G_2$ with support $\theta$, denote the set of patterns as $P$;
2: for all $p \in P$ do
3:     Calculate frequency of $p$ in $G_1$ and then the log ratio score of $p$;
4:     if the log ratio score of $p < \delta$ then
5:         remove $p$ from $P$;
6:     end if
7: end for
8: Sorting P in descending order of the log ratio score.
9: return $P$;
```

Figure 5 shows the entire framework for the construction of the model. The same method could be applied to other biological phenomenon of interest, such as enhancers.

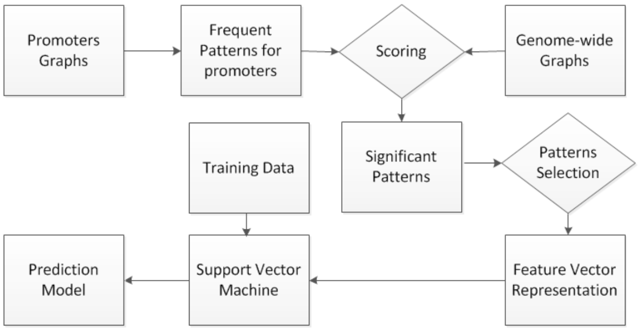

Figure 5: Promoter Prediction Model Construction.

7. EXPERIMENTAL STUDIES

In this section, we describe some of the experimental studies that we performed.

7.1 Dataset and Parameter Settings

We used genome-wide transcription factor binding data from the Encyclopedia of DNA Elements, or modENCODE, project[12].

Chromatin factors data. For chromatin factors, we selected the binding site data (ROIs) of 8 chromatin factors from the fly ChIP datasets generated by the Solexa platform[1] and Agilent platform. The data contains the binding sites of the 8 chromatin factors and is available here[2]. We mainly used data from Solexa platform, except when it was not available, in which case we used data from the Agilent platform. The data contained time for 12 time courses, from the fly embryo to the adult. The geometry of the factors is quite different over the different time periods. In the experiments, we analyze two time periods: 0 $\sim$ 4 hours (E0-4) and 8 $\sim$ 12 hours (E8-12).

Gene Activity Data. We take the gene data from flybase[4], and the list of active and inactive gene data from

[12]. For the embryo 0 $\sim$ 4 hours, we have 9657 active genes and 9169 inactive genes.

Table 3: Promoter Graph Data

time course	*promoter graphs*	*genomewide graphs*
E0-4	5484	49056
E8-12	5245	40384

Table 4: Promoter Classification Data

E0-4	*Positive*	*Negative*
Training data	3630	4200
Testing data	1854	2202
E8-12	*Positive*	*Negative*
Training data	3470	4000
Testing data	1765	2000

Promoter Data. The analysis of the promoter dataset is shown in Tables 3 and 4. Table 3 shows the graph statistics and Table 4 gives the data used for classification. For the purpose of classification, we need both positive cases and negative cases. We assume that the sites where promoters occur are positive cases and the sites without promoters are negative cases. We extract the promoters from the Transcription Start Site (TSS) class annotation at FDR 0.05, in the Supplementary Table 6 in [12]. This data gives the location information of the known promoters. We take the active promoters marked with "TP" and "FN" in that table. For each active promoter, we simply generate a graph based on its location. Then we randomly split the data into two groups, one for training and the other for testing.

Negative cases are generated by random drawing on a genome-wide basis. We randomly pick a region and use it as a negative instance if there is no TSS within 2000 base pairs. The negative cases are divided into two groups as well.

We have to choose carefully the size of the sliding window. On the one hand, if the window size is too small, it will not capture the geometry of different factors. On the other hand, if it is too large, the graph produced will be too complicated. The size of ROIs varies a lot for different chromatin factors. Some tend to have ROIs larger than 10,000 base pairs, while others are more likely to have ROIs of several hundred base pairs. In our experiments, we set the window size as 2000 base pairs. The sliding step is set to 1000 base pairs so that the two neighboring windows have 1000 base pairs of overlapping area.

In the experiments, to simplify the analysis, we use the same edge label for cases 1 to 4 in Figure 3. Table 1 is used to label the nodes. When mining the frequent patterns for the gene activity graphs, we set the minimum support to be 5%. For promoter prediction, we tried multiple support values. We selected a threshold log ratio score of 0.7.

7.2 Gene Activity

In this set of experiments, we take the chromatin factors data of embryo 0 $\sim$ 4 hours. We performed two sets of experiments on the above dataset. In the first set of experiments, we take the graphs of the active genes as the target dataset and graphs of the inactive genes as the background dataset. The top four discriminative patterns are shown in Figure 6, and their scores are shown in Table 5.

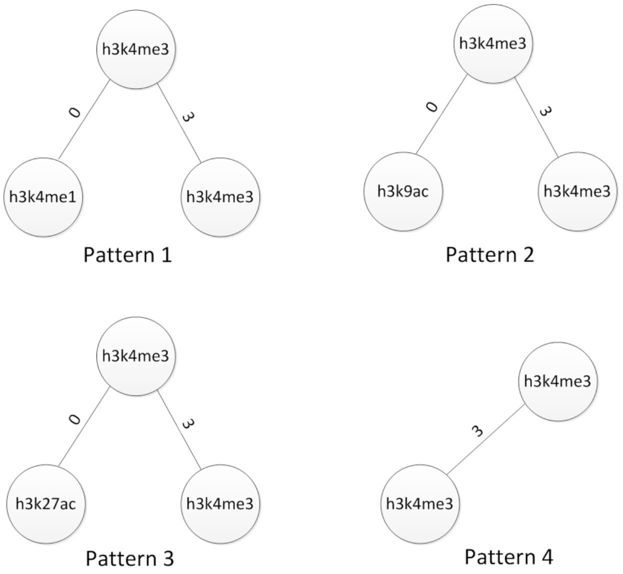

Figure 6: Active Patterns.

We can see that two h3k4me3s occur in all those 4 patterns, which indicates that the h3k4me3 has a positive impact on the gene expression. This is confirmed by the biological observations: h3k4me3 is an important epigenetic landmark for active transcription [15] [12].

In the second set of experiments, we take the graphs of the inactive genes as the target dataset and graphs of the active genes as the background dataset. The most discriminative pattern found is shown in Figure 7, and its score is shown in Table 6.

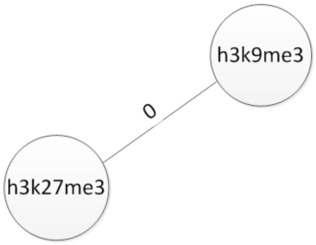

Figure 7: Inactive Pattern.

Table 5: Statistics of Active Patterns

ID	*# in G_2*	*# in G_1*	f_p	f_b	*ratio*	*score*
1	512	13	7.32%	0.91%	8.07	0.907
2	471	12	6.74%	0.84%	8.05	0.905
3	390	10	5.58%	0.70%	8.0	0.902
4	761	22	10.89%	1.54%	7.09	0.851

The result indicates that h3k9me3 and h3k27me3 together can repress the gene expression. This is in accordance with that h3k9me3 and h3k27me3 are often found associated with repressed genes for multiple species [17] [7].

7.3 Promoter Prediction

Table 6: Statistics of Inactive Pattern

ID	*# in G_2*	*# in G_1*	f_p	f_b	*ratio*	*score*
1	382	122	26.7%	1.75%	15.3	1.184

We did two sets of experiments for the promoter prediction, one for embryo during the period 0 $\sim$ 4 hours and the other for the embryo during the period 8 $\sim$ 12 hours. In each set of experiments, we reduced the top k highest ranked patterns into a smaller pattern set of size n such that no pattern is a subgraph of another pattern in this set. We constructed the feature vectors and built a predictive model based on those n patterns.

Table 7: Embryo 0-4 hours Classification Results

support	*n*	*k*	*precision*	*recall*	*accuracy*
15%	15	72	93.53%	83.44%	89.79%
10%	17	197	93.47%	84.20%	90.09%
5%	23	893	93.62%	83.87%	90.01%

Table 8: Embryo 8-12 hours Classification Results

support	*n*	*k*	*precision*	*recall*	*accuracy*
15%	10	84	97.12%	84.02%	91.34%
10%	13	313	97.49%	83.74%	91.37%
5%	26	1908	97.33%	84.65%	91.71%

The support is a critical parameter when mining the frequent patterns. We did three sets of experiments by setting the support as 5%, 10% and 15%. The results are given in Tables 7 and 8.

From the tables, we can see the precision is very high for both time courses. According to the results, if a new promoter is predicted by the prediction model, it has a high probability of being a true promoter. When we apply the trained model to the whole genome, 1859 new promoters are predicted. To fully validate these predictions, requires expensive and time consuming wet lab experiments. As a rough estimate of the validity of our predictions, we note that promoters are generally close to TSS. For known promoters in our dataset, 99% of them are within 2000 bp of a TSS. Among the new predicted promoters, 91% of them are within 2000 bp of a TSS, which is highly encouraging. Furthermore, even 9% that do not meet their 2000 bp criteria may represent previously unrecognized novel promoters.

8. CONCLUSION

In this paper, we have abstracted and studied a new problem: finding significant geometric patterns of marks along a genome. We have shown how to code geometric information about patterns of marks as a graph and introduced an algorithm called PoM for identifying patterns of marks of interest. To the best of our knowledge, this is the first time that the geometry of binding sites has been studied using frequent graph algorithms.

We did experimental studies using the PoM algorithm to study the fly's chromatin factors and active genes. We also showed the method could be adapted to predict promoter regions for the fly. The PoM algorithm can be immediately applied without change to other organisms and to study other phenomena, such as enhancers and hotspots.

9. REFERENCES

[1] http://genepool.bio.ed.ac.uk/illumina/index.html.

[2] http://www.bionimbus.org:8080/bionimbus/.

[3] http://www.bioviz.org/igb/.

[4] http://www.flybase.org/.

[5] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22:207–216, June 1993.

[6] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. VLDB '94.

[7] S. Bilodeau, M. H. Kagey, G. M. Frampton, P. B. Rahl, and R. A. Young. Setdb1 contributes to repression of genes encoding developmental regulators and maintenance of es cell state. *Genes and Dev.*, 23(21):2484–2489, 2009.

[8] D. Chen, A. S. Belmont, and S. Huang. Upstream binding factor association induces large-scale chromatin decondensation. *Proc Natl Acad Sci USA*, 101:15106–15111, 2004.

[9] C. Cortes and V. Vapnik. Support-vector networks. *Support-Vector Networks*, 20, 1995.

[10] E. B. et al. Regulation by transcription factors in bacteria: beyond description. *FEMS Microbiology reviews*, 33:133–151, 2009.

[11] F. T. et al. Flynet: a genomic resource for drosophila melanogaster transcriptional regulatory networks. *Bioinformatics*, 25:3001–3004, 2009.

[12] N. N. et al. A cis-regulatory map of the drosophila genome. *Nature*, 471:527–531, 2011.

[13] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. *ICDM*, 2003.

[14] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *ECML PKDD*, 2000.

[15] H. H. Kavi and J. A. Birchler. Drosophila kdm2 is a h3k4me3 demethylase regulating nucleolar organization. *BMC Research Notes*, 2009.

[16] M. Kuramochi and G. Karypis. Frequent subgraph discovery. *ICDM*, 2001.

[17] L. C. Lindeman, C. L. Winata, H. Aanes, S. Mathavan, P. Aleström, and P. Collas. Chromatin states of developmentally-regulated genes revealed by dna and histone methylation patterns in zebrafish embryos. *Int. J. Dev. Biol.*, 54:803 – 813, 2010.

[18] S. Nijssen and J. N. Kok. A quickstart in frequent structure mining can make a difference. *KDD*, 2004.

[19] H. Willenbrock and D. W. Ussery. Chromatin architecture and gene expression in escherichia coli. *Genome Biology*, 5:252, 2004.

[20] X. Yan. *Mining, indexing and similarity search in large graph data sets.* PhD thesis, Champaign, IL, USA, 2006. AAI3243031.

[21] X. Yan and J. Han. Gspan: Graph-based substructure pattern mining. *ICDM*, 2002.