Statistics and Its Interface Volume 5 (2012) 61–73

# Permutation methods for testing the significance of phosphorylation motifs[*]

Haipeng Gong and Zengyou He[†]

Phosphorylation motifs represent common patterns around the phosphorylation site. As the discovery of such kinds of motifs reveals the underlying regulation mechanism and facilitates the prediction of unknown phosphorylation events, some phosphorylation motif discovery methods are proposed. Existing methods include Motif-X, MoDL, and Motif-All. Each of these methods can find a certain number of motifs, however, there are still no theoretically guided measures to select true phosphorylation motifs from false ones. Since it is very expensive and time-consuming to perform the biological validation on all reported motifs, the use of effective statistical methods as a preliminary filter to remove non-significant motifs is actually needed. To solve this problem, we use permutation to calculate $p$-values of identified motifs and thus their statistical significance can be assessed accurately. We suggest to utilize three permutation methods: the Standard Permutation (SP), the Adaptive Marginal Effect Permutation (AMEP) and the Modified Adaptive Marginal Effect Permutation (MAMEP). We conduct comprehensive experimental studies to demonstrate the effectiveness of our methods. Experimental results on real data and simulation studies show that all permutation methods are capable of removing potential false positives. Particularly, both AMEP and MAMEP are of practical use and can satisfy different requirements of biological researchers.

Keywords and phrases: Phosphorylation motif, Frequent-pattern mining, Permutation test.

## 1. INTRODUCTION

The discovery of phosphorylation motifs reveals the underlying regulation mechanism and facilitates the prediction of unknown phosphorylation events. Recent advances in high-throughput methods such as tandem mass spectrometry enable rapid and direct discovery of hundreds of phosphorylation sites in a single experiment [1]. Both the biological significance and the availability of a large number of phosphorylation sites motivated the development of phosphorylation motif discovery methods, such as Motif-X [2], MoDL [3] and Moif-All [4]. Among the three motif discovery methods, both Motif-X and MoDL can only find a small subset of phosphorylation motifs, while Motif-All can discover all statistically significant phosphorylation motifs under a given parameter setting [4]. Experimental results have shown that Motif-All outperforms Motif-X and MoDL [4]. Motif-All applies Apriori algorithm [5] (a typical frequent pattern mining algorithm) first to find frequent motifs from the set of phosphorylated peptides $P$. From a statistical perspective, most of true phosphorylation motifs should be frequent, so almost all true positives belong to the set of frequent motifs. Since the number of frequent motifs is very large, Motif-All uses $p$-value derived from a normal distribution to measure the significance of each candidate motif. However, no multiple testing correction is carried out in Motif-All, leading to an inaccurate motif significance assessment. As a result, most false positives are still present in the result set.

In order to assess the significance of phosphorylation motifs more accurately, we use the permutation test to calculate the $p$-value of each motif. In this paper, we suggest three kinds of permutations: the Standard Permutation (SP), the Adaptive Marginal Effect Permutation (AMEP) [8], and the Modified Adaptive Marginal Effect Permutation (MAMEP). We conduct experimental studies to test the performance of the three permutation methods using the PhosPhAt database 3.0 of Arabidopsis phosphorylation sites [6, 7] and simulated data.

The rest of the paper is organized as follows: Section 2 presents the details of the three permutation methods. Section 3 and section 4 show the experimental results on real data and simulated data, respectively. Section 5 concludes the paper.

## 2. METHODS

### 2.1 Basic terminology

Phosphorylation motif is a consensus sequence that consists of conserved positions and wild-card positions that can match any residue. Here residues symbolize amino acids. For example, (S.......G$\underline{\text{Y}}$......A...) is a phosphorylation motif. It contains a single phosphorylated residue, which is denoted with an underlined character $\underline{\text{Y}}$, 3 conserved positions (S, G and A) and 17 wild-card positions (.). Also, it can be writ-

[*] We thank Dr. Waltraud X. Schulze for providing us the PhosPhAt 3.0 database. We thank Dr. Can Yang, Dr. Li Ma and Dr. Xiaodan Fan for giving us valuable suggestions about permutation. This work was partially supported by the Natural Science Foundation of China under Grant No. 61003176.

[†] Corresponding author.

ten as a combination of attribute-values, (R0 = S, R9 = G, R17 = A) in which the attribute symbolizes the position of residue and value symbolizes certain amino acid. In this form, we only record its conserved positions, and define the size of a phosphorylation motif as the number of conserved positions it contains. For simplicity, motifs with size $k$ are called $k$-motifs. We define "attribute combination" as the set of attributes involved in the conserved positions of a motif. For instance, (R0, R9, R17) is the corresponding attribute combination of the example motif. For each motif, we define $f$-counter as the number of matching peptides in the set of phosphorylated peptides $P$ and $b$-counter as the number of matching peptides in the set of unphosphorylated peptides $N$. Finally, the support of a motif is defined as the relative frequency of the motif among the set of phosphorylated peptides $P$. For example, if there are 1,000 peptides in the set of phosphorylation peptides $P$, and 40 of them match the motif $m$, then the support of $m$ is 4%.

## 2.2 Odds ratio and $z$-value

The methods of calculating odds ratio and $z$-value are the same as those applied in Motif-All [4]. The odds of an event is defined as the probability that this event occurs divided by the probability that it does not occur. The odds ratio is defined as the ratio of the odds of an event in one group to the odds in the complementary group.

In the context of phosphorylation motif discovery, the first group corresponds to the set of phosphorylated peptides $P$ and the second group is the set of unphosphorylated peptides $N$. For a given motif $m$, we can construct a contingency table as shown in Tab. 1. In this table, $c_{00}$, $c_{01}$, $c_{10}$ and $c_{11}$ are non-negative "cell counts" and $\overline{m}$ denotes that the motif $m$ doesn't exist. Then, the calculation of odds ratio becomes:

$$OR(m) = \frac{c_{00}c_{11}}{c_{10}c_{01}}. \tag{1}$$

An odds ratio of 1 means that the target motif is equally likely to be present in both $P$ and $N$. An odds ratio greater than 1 indicates that this motif is more likely to appear in the set $P$.

To conduct statistical inference, one approach is to use large sample approximations to the sampling distribution of the log odds ratio. More precisely, the sample log odds ratio is:

$$LOR(m) = \log\left(\frac{c_{00}c_{11}}{c_{10}c_{01}}\right), \tag{2}$$

and the standard error for the log odds ratio is approximately:

$$SE(m) = \sqrt{\frac{1}{c_{00}} + \frac{1}{c_{01}} + \frac{1}{c_{10}} + \frac{1}{c_{11}}}. \tag{3}$$

*Table 1. A contingency table for a phosphorylation motif*

| | $m$ | $\overline{m}$ |
|---|---|---|
| $P$ | $c_{00}$ | $c_{01}$ |
| $N$ | $c_{10}$ | $c_{11}$ |

Then, $z$-value

$$Z(m) = LOR(m)/SE(m), \tag{4}$$

is used as the statistic in the calculation of $p$-value in our permutation methods. The respective null and alternative hypotheses of the statistical inference are as follows:

- Null hypothesis: the distributions of motifs in foreground data and background data are the same.
- Alternative hypothesis: the distributions of motifs in foreground data and background data are different.

## 2.3 Modified FP-growth

Frequent patterns are patterns (such as itemsets, subsequences, or substructures) that appear in a data set frequently. Here we apply the term "itemsets" as patterns to explain the basic definitions. Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of items. Let $D$, the task-relevant data, be a set of database transactions where each transaction $T$ is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID. Let $E$ be a set of items. A transaction $T$ is said to contain $E$ if and only if $E \subseteq T$. The itemset $E$ holds in the transaction set $D$ with support $s$, where $s$ is the percentage of transactions in $D$ that contain $E$. $E$ can be claimed as a frequent itemset if its support is larger than a given threshold. Frequent subsequences or substructures are similar in definition and they are all called frequent patterns. In this paper, we need to discover frequent motifs from phosphorylated peptide set $P$, and these frequent motifs are also in the scope of frequent patterns.

To discover frequent motifs, Motif-All applied Apriori [5], which is one typical algorithm for frequent pattern mining. Apriori employs an iterative approach known as a level-wise search, where patterns of size $k$ are used to explore patterns of size $k+1$. First, the set comprising patterns of size 1 are found by scanning the database and collecting those ones that satisfy minimum support. The resulting set is denoted by $L_1$. Next, $L_1$ is used to find $L_2$, the set comprising frequent patterns of size 2, which is used to find $L_3$, and so on, until no more frequent patterns of size $k$ can be found. The discovery of each $L_k$ requires one full scan of the database. To improve the efficiency of the level-wise generation of frequent patterns, Apriori further uses an important property "All nonempty subsets of a frequent pattern must also be frequent" to reduce the search space.

Apriori is efficient in frequent pattern mining, however, it has some limitations [9]:

- It is likely to generate a large number of candidates. For example, if there are $10^4$ frequent patterns of size 1, the Apriori algorithm will need to generate more than $10^7$ candidate patterns of size 2. Moreover, to discover a frequent pattern of size $k$, it has to generate at least $2^k - 1$ candidates in total.

- It needs to repeatedly scan the database and check a large set of candidates by pattern matching.

In order to mine the complete set of frequent patterns without candidate generation, Han et al. proposed the frequent-pattern growth algorithm or simply FP-growth [11], which adopts a *divide-and-conquer* strategy as follows. First, it compresses the database representing frequent items into a frequent-pattern tree, or FP-tree. It then divides the compressed database into a set of conditional databases (a special kind of projected database), each associated with one frequent item or "pattern fragment", and mines each such database separately.

The FP-growth method transforms the problem of finding long frequent patterns to searching for shorter ones recursively and then concatenating the suffix. It uses the least frequent items as a suffix, offering good selectivity. This method substantially reduces the search costs. FP-growth is efficient and scalable for mining both long and short frequent patterns, and is about an order of magnitude faster than the Apriori algorithm. Hence, it is very appropriate to be used for frequent motif discovery.

Due to the specific format of our data, we cannot use FP-growth directly. So we first transform peptide sequences into transaction format. For example, the peptide (PTGAQIIYSKYAGTEVEFNDV) will be transformed into a transaction {1P, 2T, 3G, 4A, 5Q, 6I, 7I, 8Y, 9S, 10K, 11A, 12G, 13T, 14E, 15V, 16E, 17F, 18N, 19D, 20V}. In this case, the same amino acid in different positions of the peptide will be regarded as different items in the transformation. After the transformation, we apply FP-growth algorithm to find all the frequent patterns, and then we recover the frequent patterns to frequent motifs for later usage. The whole process for discovering frequent motifs is named "Modified FP-growth" in this paper.

## 2.4 Overview of standard permutation

In statistics, permutation is used to generate an appropriate null distribution, from which $p$-values can be calculated accurately. In general, we need three steps [10]:

1. Compute the statistic for the original data.
2. Choose permutation resamples from the data without replacement in a way that is consistent with the null hypothesis of the test and with the study design. Construct the permutation distribution of the statistic from its values of a large number of resamples.
3. Calculate the $p$-value by locating the original statistic in a large number of resamples.

In the permutation test, the use of more permutations will lead to more accurate $p$-value calculation; Meanwhile, it will become more time-consuming. In practice, we typically use 1,000 permutations in the test.

## 2.5 Standard permutation

Standard permutation has the following null and alternative hypotheses:

- Null hypothesis: the distributions of motifs in foreground data and background data are the same.
- Alternative hypothesis: the distributions of motifs in foreground data and background data are different.

### 2.5.1 Direct standard permutation

Let $Z$ be the set of all $z$-values obtained from the permutation test and let $|Z|$ denote the size of $Z$. We can divide Direct Standard Permutation (DSP) into three stages:

1. Run Modified FP-growth on the original set of phosphorylated peptides $P$ and then compute $z$-values of the identified frequent motifs.
2. Generate independent data sets by randomly permuting the class labels. Here we randomly select peptides from the set of phosphorylated peptides $P$ and the set of unphosphorylated peptides $N$ to build up a new set $P$ as well as a new set $N$. Then, we run Modified FP-growth on the new data set to get frequent motifs and calculate their $z$-values. Repeat the step for $L$ times, and add all $z$-values during all $L$ runs to $Z$.
3. Obtain $p$-value from $Z$. For each identified motif $m$ found in step 1, we define $T$ as the number of $z$-values which are larger than the $z$-value of $m$, then the final $p$-value of motif $m$ is $T/|Z|$. The motifs whose $p$-values are smaller than the threshold are returned to user.

The algorithm description is shown in Algorithm 1.

### 2.5.2 Improved Standard Permutation (ISP)

DSP applies the standard permutation method directly and is easy to use, however, it is time-consuming as it needs to update $b$-counter through traversing the set of unphosphorylated peptides $N$ in each permutation. In DSP, for every frequent motif in each permutation, a traversal of $P$ set and $N$ set is needed. In order to improve the efficiency, it is necessary to decrease the number of traversals. As the size of $N$ set is definitely large, most time is wasted on the process of updating $b$-counter. For each motif, $f$-counter and $b$-counter are different in different permutations, but their sum is always the same. In other words, permutation changes the distribution of peptides in $P$ set and $N$ set, however, the support of each motif in the whole data never changes. So when performing stage 1, we record the information of frequent motifs including $f$-counter and $b$-counter through a data structure named MotifRecords. When performing permutations in stage 2, after updating $f$-counter, we first look for the motif in MotifRecords. If it exists, we update its $b$-counter as the result of a recorded sum subtracting $f$-counter; If it is not found, we update $b$-counter through counting the number of matching peptides from $N$ set, and then add information of the motif to MotifRecords.

**Algorithm 1** Direct Standard Permutation (DSP)

```
1. Initialization: Let $F$ be the set of frequent motifs found from the original data set. Let $Z$ be the set of all $z$-values obtained from the permutation test and $|Z|$ denote its size.
2. Search all motifs whose support is above a threshold $s$ by applying Modified FP-growth. This produces $F$.
3. For each motif $m \in F$, compute its $z$-value using its $f$-counter and $b$-counter.
4. For $j = 1, 2, \ldots, L$ ($j$ is the permutation index and $L$ is the number of permutations),
    (a) Generate independent data sets by randomly permuting the class labels.
    (b) Search and count all motifs whose support is above a threshold $s$ using Modified FP-growth. This produces a set of frequent motifs $G_j$.
    (c) For each motif $m \in G_j$, compute its $z$-value using its $f$-counter and $b$-counter.
    (d) Add $z$-value of each motif $m \in G_j$ to $Z$.
5. For each motif $m \in F$, get the number of $z$-values which are larger than the $z$-value of $m$ as $T$, then the permutation $p$-value, $P^*(m)$, is given by $P^*(m) = T/|Z|$.
6. Obtain the set of significant motifs $G$ as $\{m \in F$: $P^*(m) < \alpha\}$, where $\alpha$ is the significance threshold for the $p$-values.
```

**Algorithm 2** Improved Standard Permutation (ISP)

```
1. Initialization: Let $F$ be the set of frequent motifs found from the original data set. Let $Z$ be the set of all $z$-values obtained from the permutation test and $|Z|$ denote its size.
2. Search all motifs whose support is above a threshold $s$ by applying Modified FP-growth. This produces $F$.
3. For each motif $m \in F$, compute its $z$-value using its $f$-counter and $b$-counter.
4. For each motif $m \in F$, add its information into MotifRecords.
5. For $j = 1, 2, \ldots, L$ ($j$ is the permutation index and $L$ is the number of permutations),
    (a) Generate independent data sets by randomly permuting the class labels.
    (b) Search and count all motifs whose support is above a threshold $s$ using Modified FP-growth. This produces a set of frequent motifs $G_j$.
    (c) For each motif $m \in G_j$, look for $m$ in MotifRecords. If it is already there, update its $b$-counter through MotifRecords; Otherwise, update its $b$-counter through traversing $N$ set, and then add its information into MotifRecords.
    (d) For each motif $m \in G_j$, compute its $z$-value using its $f$-counter and $b$-counter.
    (e) Add $z$-value of each motif $m \in G_j$ to $Z$.
6. For each motif $m \in F$, get the number of $z$-values which are larger than the $z$-value of $m$ as $T$, then the permutation $p$-value, $P^*(m)$, is given by $P^*(m) = T/|Z|$.
7. Obtain the set of significant motifs $G$ as $\{m \in F$: $P^*(m) < \alpha\}$, where $\alpha$ is the significance threshold for the $p$-values.
```

So as the permutation test goes on, the number of motifs in MotifRecords becomes larger and larger so that more and more motifs can update their $b$-counter through MotifRecords rather than traversing $N$ set. Obviously, this will save a lot of time. In a test using data PhAtY, we perform 1,000 permutations. DSP takes 98 minutes while ISP takes only 9 minutes. It shows that the ISP method can drastically improve the efficiency of a standard permutation test.

The algorithm description is shown in Algorithm 2.

## 2.6 Adaptive Marginal Effect Permutation (AMEP)

The Adaptive Marginal Effect Permutation method is initially proposed for detecting epistasis in disease association studies [8]. With minor changes, it can be used for assessing phosphorylation motifs as well. This kind of permutation differs from standard permutation in two important ways.

Firstly, in standard permutation method $p$-values for the motifs of different size are tested together, which means that they use the same permutation null. Since the number of possible motifs increases combinatorially with their size and motifs of larger size are prone to have small $p$-values, the motifs of smaller size may be overwhelmed if we use the same permutation null. This problem will become more and more serious as the size of motifs under investigation increases. To overcome this limitation, AMEP tests motifs of different size separately with different permutation nulls.

Secondly, since we conduct a separate permutation test for motifs of different size, the permutation null for longer motifs should take into account the effects detected among the shorter ones. For example, if an attribute combination (R1, R5) has been determined to be significant, then many 3-motifs that contain this attribute combination may be significant as well. More generally, if the effect of an attribute combination can be explained by one or more of its sub-combinations, then we should try to recover such sub-combinations rather than declare the longer one to be significant. Thus, the significant motifs of size up to $n$-1 should be considered in constructing the null hypotheses for motifs of size $n$.

The basic idea of AMEP is to test 1-motifs first followed by the 2-motifs using the effects detected in the 1-motifs as the null, and then test the 3-motifs using the effects detected in the 1-motifs and 2-motifs as the null, so forth and so on until for testing $n$-motifs there are no frequent motifs with size $n$ found by Modified FP-growth.

```
0-DGYDRRYGDRYSPGGRSPGFE   (P)
1-DGNEVVEPVDYGKSKADDEFE   (P)

2-AEKKKTKKPSYPSSSMKSKVY   (N)
3-MTKDELTEEEYLSGKDYLDPP   (N)
4-RHKDSLAAAEYPDGMKVSNSH   (N)
5-GGTAVGKDLLYDGDSVKSSTD   (N)
```

*Figure 1. Original data set $P$ and $N$.*

```
0-DGYDRRYGDRYSPGGRSPGFE   (N)
1-DGNEVVEPVDYGKSKADDEFE   (N)

2-AEKKKTKKPSYPSSSMKSKVY   (N)
3-MTKDELTEEEYLSGKDYLDPP   (P)
4-RHKDSLAAAEYPDGMKVSNSH   (P)
5-GGTAVGKDLLYDGDSVKSSTD   (N)
```

*Figure 2. Standard permutation results.*

```
0-MTYDRRYGDRYSPGGRSPGPP   (N)
1-AENEVVEPVDYGKSKADDEVY   (N)

2-RHKKKTKKPSYPSSSMKSKSH   (N)
3-DGKDELTEEEYLSGKDYLDFE   (P)
4-DGKDSLAAAEYPDGMKVSNFE   (P)
5-GGTAVGKDLLYDGDSVKSSTD   (N)
```

*Figure 3. Adaptive marginal effect permutation results.*

Suppose we have completed our testing for motifs of size up to $n-1$, and have arrived at a set, $S_{(n-1)}$, of significant attribute combinations up to size $n-1$. Here we declare that an attribute combination is significant as long as one of its corresponding motifs is significant. To test the $n$-motifs, we first divide them into $C$ groups in such a fashion that the motifs within each group share exactly the same set of significant attribute combinations. The idea is to construct a separate permutation null for each of the $C$ groups. It should be noted that $C$ depends on both $n$ and $S_{(n-1)}$. For example, suppose we have tested 2-motifs, and the significant attribute combinations $S_2 = \{(\mathrm{R1}), (\mathrm{R2}), (\mathrm{R2}, \mathrm{R3})\}$. When testing 3-motifs, we divide them into at most $C = 6$ groups:

- $G_1$: those that contain (R1), but do not contain either (R2) or (R2, R3).
- $G_2$: those contain (R2), but do not contain either (R1) or (R2, R3).
- $G_3$: those that contain (R2, R3), but do not contain (R1).
- $G_4$: those that contain (R1) and (R2), but do not contain (R2, R3).
- $G_5$: those that contain (R1), (R2), and (R2, R3).
- $G_6$: those that contain none of (R1), (R2), (R2, R3).

Here "at most" means that in practice when partitioning 3-motifs, we may get no more than 6 groups. For instance, if there are no 3-motifs that contain (R1), (R2) and (R2, R3), then $G_5$ is empty, so it is not necessary to generate $G_5$ in our test and the number of groups we get is smaller than 6.

The $C$ permutation nulls, one for each of the $C$ groups, can be constructed simultaneously by permuting the class label together with all the attribute combinations in $S_{(n-1)}$. Here we give a concrete example to explain details of the procedure and show how it differs from the standard permutation method. As shown in Fig. 1, suppose the first two peptides belong to the set of phosphorylated peptides $P$, and the remaining 4 are present in the set of unphosphorylated peptides $N$. We have tested 1-motifs, and the set of attribute combinations $S_1$ we get contains four elements (R0), (R1), (R19), (R20). Suppose we generate a random permutation of sample id: 3, 4, 1, 0, 2, 5. That means in the new permutation data, $P$ contains 3, 4 and $N$ contains 1, 0, 2, 5. Then according to this random permutation, we should do the following assignment: 0 to 3, 1 to 4, 2 to 1, 3 to 0, 4 to 2 and 5 to 5. In the standard permutation method, this just means in the new permutation data, $P$ contains sample 3, 4 and $N$ contains sample 1, 0, 2, 5. There are no data swap but just labels. For instance, 0 to 3 just means the label of sample 0 is given to sample 3. The final results are given in Fig. 2. However, in AMEP, we should permute the class label together with all the attribute combinations in $S_1$, which means that we should not only assign the sample label, but also assign the residues associated with $S_1$. Since $S_1$ contains (R0), (R1), (R19) and (R20), the residues in position of 0, 1, 19 and 20 are supposed to be permuted together with the label at the same time. The results are shown in Fig. 3, in which the blue residues are assigned based on $S_1$.

For each permutation, we apply Modified FP-growth just as we did for the original data. By pooling the $z$-values of the motifs belonging to each of the $C$ groups from all the permutations, we obtain a sample of $z$-values from the permutation null for each of the $C$ groups. Then the $p$-value for a motif can be computed as the proportion of permutations that generated a more significant $z$-value in the corresponding group. Those motifs whose $p$-values pass a significance threshold, for example, 5%, are declared significant, and their corresponding attribute combinations are joined with $S_{(n-1)}$ to form $S_n$. The algorithm description is shown in Algorithm 3, adapted from [8].

## 2.7 Modified adaptive marginal effect permutation

We propose the Modified Adaptive Marginal Effect Permutation because of the special characteristics of our data. The motivation behind this new method and the corresponding changes are follows.

The first one is that the case data set and control data set are of the similar size in [8] when performing a permutation test. However, our case data (the set of phosphory-

**Algorithm 3** Adaptive Marginal Effect Permutation (AMEP)

```
1. Initialization: Let $F$ be the set of frequent motifs found from the original data set. Let $S$ be the set of significant attribute combinations, and $S_0 = \emptyset$.
2. Search all motifs whose support is above a threshold $s$ by applying Modified FP-growth. This produces $F$.
3. For each motif $m \in F$, compute its $z$-value using its $f$-counter and $b$-counter.
4. For $n = 1, 2, \ldots, max.length$,
    (a) Take all frequent motifs with size $n$ from $F$ to construct $G$.
    (b) Partition frequent motifs that $G$ contains into groups $G_1, G_2, \ldots, G_C$ according to the significant attribute combinations in $S_{(n-1)}$ they contain. Let $Z_1, Z_2, \ldots, Z_C$ be the corresponding collections of $z$-values.
    (c) For $j = 1, 2, \ldots, L$ ($j$ is the permutation index and $L$ is the number of permutations),
        i. Permute the response label together with the attributes involved in $S_{(n-1)}$.
        ii. Search and count all motifs whose support is above a threshold $s$ using Modified FP-growth. This produces a set of frequent motifs $G^{(j)}$.
        iii. For each motif $m \in G^{(j)}$, compute its $z$-value using its $f$-counter and $b$-counter.
        iv. Classify frequent motifs that $G^{(j)}$ contains into groups $G_1^{(j)}, G_2^{(j)}, \ldots, G_C^{(j)}$ according to the significant attribute combinations in $S_{(n-1)}$ they contain. Let $Z_1^{(j)}, Z_2^{(j)}, \ldots, Z_C^{(j)}$ be the corresponding collections of $z$-values.
    (d) For each motif $m \in G_i$, $i = 1, 2, \ldots, C$, the permutation $p$-value, $P^*(m)$, is given by $P^*(m) = \#\{j\colon \max Z_i^{(j)} > Z(m)\}/L$.
    (e) Set $S_n = S_{(n-1)} \bigcup$ {the corresponding attribute combinations of $m \in G : P^*(m) < \alpha$}, where $\alpha$ is the significance threshold for the $p$-values.
```

lated peptides $P$) and control data (the set of unphosphorylated peptides $N$) are very different in size. For example, in PhAtY data set [4], the size of $P$ is 80 while the size of $N$ is 304344. Such extreme unbalance may cause a potential problem when performing a permutation directly. More precisely, the permuted sample ids are totally random, so almost all peptides in the new set $P$ come from the original set $N$. From the original $P$ set, we can find a lot of frequent motifs, especially some long motifs, In contrast, we may only find a very small number of frequent motifs from the new set $P$ since most peptides it contains are from the original set $N$. As a result, in the final null distribution, there are very few motifs whose $z$-values are larger than that of frequent motifs of a larger size found from the original $P$ set. Hence, it is very hard to prune those frequent motifs even permuting the class label together with lower-order attribute combinations they contain. In order to alleviate the influence of unbalanced data distribution, we make some changes in the procedure of permuting response labels. Here we use a new sample id permutation method in which only 50 percent of phosphorylated peptides change their response labels. We name this modification as the fixed percentage constraint.

We define $|P|$ as the size of $P$ and $|N|$ as the size of $N$. Suppose $I$ is the set of ids from 0 to $|P| + |N| - 1$. First we permute $I$ to get random sample ids for generating $I_0$. Second, we take $|P|/2$ ids whose values are smaller than $|P|$ from the beginning of $I_0$ to construct $|I_1|$, analogously, we take $|P|/2$ ids whose values are not smaller than $|P|$ from the beginning of $I_0$ to construct $I_2$, and the remaining ids of $I_0$ are used to construct $I_3$. Third, we combine $I_1$ and $I_2$ to construct $I_4$. Fourth, we randomly permute the ids in $I_4$ to construct $I_p$ as well as the ids in $I_3$ to construct $I_n$. Finally, we use $I_p$ and $I_n$ as the set of random sample ids to finish the permutation. For example, suppose there are 4 peptides in $P$ and 6 peptides in $N$, the procedure for permuting sample ids is shown in Fig. 4.

The second one is that in detecting epistasis in disease association studies [8], for one genetic pattern, the number of potential markers is very large compared to the number of true significant markers. However, in phosphorylation motif discovery, our data set consists of only 20 attributes. In this case, if we fix all the significant attributes in $S_{(n-1)}$ at the same time, the number of free attributes will be reduced greatly. To be more precise, as the size of tested motifs increases, more and more residues of peptides in the original data set and the new data set will be the same. As a result, the distribution of $z$-values of these long frequent motifs found from the original data set and the new data set are almost the same. Obviously this is not reasonable since permutation test for long frequent motifs finally becomes unnecessary. Our solution to this problem is to fix only one significant attribute combination in $S_{(n-1)}$ in each permutation. We call this modification as the fixed attribute constraint. For example, suppose we have completed the testing of 2-motifs, and get $S_2 = \{(\text{R1}), (\text{R2}), (\text{R2, R3})\}$. To test the 3-motifs, we use following permutations:

- In the first permutation, we permute the response label together with R1.
- In the second permutation, we permute the response label together with R2.
- In the third permutation, we permute the response label together with R2 and R3.
- In the fourth permutation, we permute the response label together with R1, so forth and so on until all the permutations are finished.

We permute the response label together with only one attribute combination involved in $S_2$ circularly until all the permutations are finished.

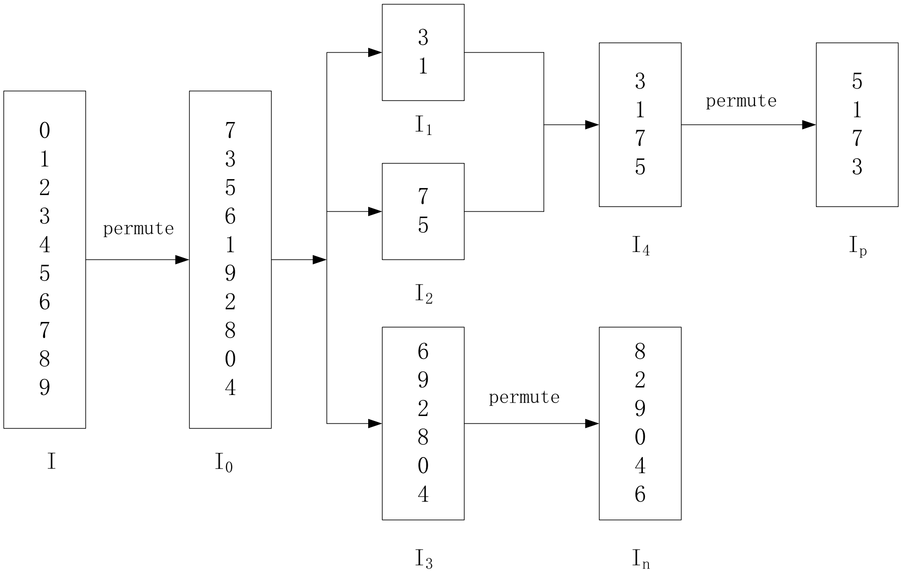

*Figure 4. The procedure of permuting sample ids for MAMEP. In this method, only 50 percent of phosphorylated peptides change their response labels in the new permutation data.*

Since there are special characteristics of data, we make two major improvements as described above. Overall, the null hypothesis underlying MAMEP is that the distributions of motifs in foreground data and background data are the same under the fixed percentage constraint and fixed attribute constraint. A formal algorithm-style description of MAMEP is provided in Algorithm 4.

# 3. EXPERIMENTAL STUDIES ON REAL DATA

## 3.1 Data

We use the PhosPhAt database 3.0 of Arabidopsis phosphorylation sites [6, 7] to construct the set of phosphorylated peptides $P$. Only the unambiguous site identifications are utilized in the constructing process. The length of each extracted peptide is 21 with a measured phosphorylated residue in the 11th position. To generate the background data set $N$, we first extract all 21-mers with a phosphorylated residue in the center position from the TAIR7 protein database. Then, we remove all peptides already in $P$. The remaining peptides form $N$. Overall, we generate three groups of data for serine (denoted by PhAtS), threonine (denoted by PhAtT) and tyrosine (denoted by PhAtY), respectively. Their characteristics are the following:

- PhAtS: It contains 2,734 foreground sequences ($P$ set) and 982,050 background sequences ($N$ set).
- PhAtT: It contains 415 foreground sequences ($P$ set) and 550,574 background sequences ($N$ set).
- PhAtY: It contains 80 foreground sequences ($P$ set) and 304,344 background sequences ($N$ set).

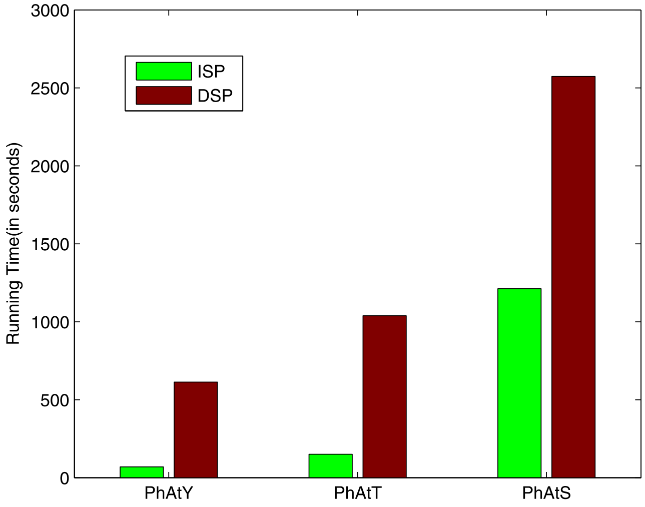

*Figure 5. Time comparison between DSP and ISP. Here we perform 100 permutations in both methods. The algorithms are implemented with Java and tested on a Lenovo notebook computer with 2.10GHz CPU and 2GB RAM.*

## 3.2 Time comparison between DSP and ISP

ISP needs only some additional space for storing MotifRecords, which consists of several hundred frequent motifs. That means DSP and ISP has the same space complexity. Hence, we only conduct a time comparison between ISP and DSP in the experimental section. We perform 100 permutations on PhAtY, PhAtT and PhAtS, respectively. The results are plotted in Fig. 5. It shows that ISP can drastically improve the efficiency. As ISP is much more efficient

**Algorithm 4** Modified Adaptive Marginal Effect Permutation (MAMEP)

```
1. Initialization: Let $F$ be the set of frequent motifs found from the original data set. Let $S$ be the set of significant attribute combinations, and $S_0 = \emptyset$. Let $|S|$ be the size of $S$.
2. Search all motifs whose support is above a threshold $s$ by applying Modified FP-growth. This produces $F$.
3. For each motif $m \in F$, compute its $z$-value using its $f$-counter and $b$-counter.
4. For $n = 1, 2, \ldots, max.length$,
   (a) Take all frequent motifs with size $n$ from $F$ to construct $G$.
   (b) Classify frequent motifs that $G$ contains into groups $G_1, G_2, \ldots, G_C$ according to the significant attribute combinations in $S_{(n-1)}$ they contain. Let $Z_1, Z_2, \ldots, Z_C$ be the corresponding collections of $z$-values.
   (c) For $j = 1, 2, \ldots, L$ ($j$ is the permutation index and $L$ is the number of permutations),
       i. Permute the sample ids using the new sample id permutation method.
       ii. Let $r$ be the remainder obtained through $j$ divided by $|S_{(n-1)}|$. Let attribute combination $a$ be the element of $S_{(n-1)}$ with the index $r$.
       iii. Permute the response label using the new sample ids together with the attributes involved in $a$.
       iv. Search all motifs whose support is above a threshold $s$ with Modified FP-growth. This produces a set of frequent motifs $G^{(j)}$.
       v. For each motif $m \in G^{(j)}$, compute its $z$-value using its $f$-counter and $b$-counter.
       vi. Classify frequent motifs that $G^{(j)}$ contains into groups $G_1^{(j)}, G_2^{(j)}, \ldots, G_C^{(j)}$ according to the significant attribute combinations in $S_{(n-1)}$ they contain. Let $Z_1^{(j)}, Z_2^{(j)}, \ldots, Z_C^{(j)}$ be the corresponding collections of $z$-values.
   (d) For each motif $m \in G_i$, $i = 1, 2, \ldots, C$, the permutation $p$-value, $P^*(m)$, is given by $P^*(m) = \#\{j\colon \max Z_i^{(j)} > Z(m)\}/L$.
   (e) Set $S_n = S_{(n-1)} \bigcup$ {the corresponding attribute combinations of $m \in G : P^*(m) < \alpha$}, where $\alpha$ is the significance threshold for the $p$-values.
```

than DSP, we take ISP as the practical method of SP to conduct the following experiments.

## 3.3 Number of reported motifs

In Motif-All, we perform its software using its default configurations: the support threshold is 0.05 and the significance threshold is $10^{-6}$. In SP, AMEP and MAMEP, we set the support threshold to 0.05 and the $p$-value threshold (significance level) to 0.05. We perform 1,000 permutations for each method and the detailed information on the number of reported motifs of different size is shown in Tab. 2. The comparison on the total number of reported motifs is shown in Fig. 6.

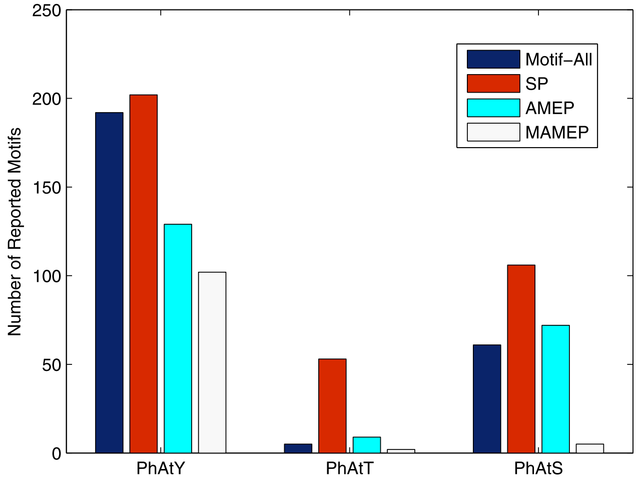

*Figure 6. The comparison on the total number of reported motifs on the real data.*

Firstly, we compare our methods according to the motifs that have been detected by at least three algorithms in previous studies. From [4], we know that motifs found simultaneously by Motif-X, MoDL and Motif-All from these three data sets are (.......R..$\underline{S}$..........), (..........$\underline{S}$P.........), (..........$\underline{T}$P.........) and (..........$\underline{Y}$...R......). SP and AMEP can find all the four motifs while MAMEP eliminates (.......R..$\underline{S}$..........) and can find the other three motifs. We then check the $z$-value of (.......R..$\underline{S}$..........) and find that it is only 7.35, which is not very large compared with other significant motifs (e.g., the $z$-value of (..........$\underline{S}$P.........) is 51.32).

Secondly, we compare our methods with Motif-All. Applying Modified FP-growth and Apriori can find the same set of frequent motifs, so the main difference between our methods and Motif-All is the significance testing method. Motif-All calculates the $p$-value of each motif without a multiple testing correction. Our methods use three kinds of permutation approaches to estimate the significance of motifs. From the results shown in Tab. 2 and Fig. 6, we can find that the distribution of motifs reported by AMEP is similar to that of Motif-All. On PhAtY, AMEP filters more 2-motifs and 3-motifs than Motif-All while Motif-All filters more 1-motifs than AMEP on PhAtT and PhAtS. Across all three data sets, we can see that MAEMP usually tends to eliminate more motifs than Motif-All.

Thirdly, we compare the three permutation methods. As shown in Tab. 2 and Fig. 6, SP retains too many motifs, especially short motifs. This indicates that using standard permutation directly is not effective enough in filtering motifs. AMEP retains less motifs than SP since it removes more

*Table 2. The number of reported motifs of different size. Here the size of motifs ranges from 1 to 7. In PhAtT and PhAtS, no significant motifs with a size larger than 2 could be detected. Therefore, the corresponding number is 0*

| *Data* | *Algorithms* | *The size of motif* | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | *1* | *2* | *3* | *4* | *5* | *6* | *7* |
| PhAtY | *Motif-All* | 1 | 74 | 50 | 38 | 21 | 7 | 1 |
| | *SP* | 3 | 82 | 50 | 38 | 21 | 7 | 1 |
| | *AMEP* | 1 | 16 | 45 | 38 | 21 | 7 | 1 |
| | *MAMEP* | 1 | 6 | 29 | 37 | 21 | 7 | 1 |
| PhAtT | *Motif-All* | 3 | 2 | 0 | 0 | 0 | 0 | 0 |
| | *SP* | 51 | 2 | 0 | 0 | 0 | 0 | 0 |
| | *AMEP* | 7 | 2 | 0 | 0 | 0 | 0 | 0 |
| | *MAMEP* | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| PhAtS | *Motif-All* | 57 | 4 | 0 | 0 | 0 | 0 | 0 |
| | *SP* | 102 | 4 | 0 | 0 | 0 | 0 | 0 |
| | *AMEP* | 68 | 4 | 0 | 0 | 0 | 0 | 0 |
| | *MAMEP* | 1 | 4 | 0 | 0 | 0 | 0 | 0 |

nonsignificant motifs by taking into account the effect of significant low-order motifs. MAMEP prunes more motifs than AMEP and retains the least motifs. If we regard the motifs found simultaneously by Motif-X, MoDL and Motif-All as the true phosphorylation motifs, we can conclude that AMEP has a lower FDR than Motif-All and SP. MAMEP has a lower power but its FDR is obviously much lower than the other three methods.

Since the true phosphorylation motifs are not known for the real data, we can only make inference in such a manner. In order to quantitatively demonstrate the effectiveness of our methods, we conduct simulation studies in the next section.

## 4. SIMULATION STUDIES

In order to further demonstrate the effectiveness of our permutation methods, we perform simulation studies. Here we conduct two simulation experiments: the first one is used for demonstrating AMEP's advantage of eliminating false positives against SP; the second one is conducted for comparing the performance of Motif-All, SP, AMEP and MAMEP on phosphorylation motif discovery.

### 4.1 Simulation study for demonstrating AMEP's advantage over SP

In section 2.5, we have described that AMEP differs from SP in two important ways: the first one is that AMEP tests motifs of different size separately with different permutation nulls; the second one is that, AMEP takes the effects detected among shorter motifs into the construction of permutation null for longer motifs. When the effect of an attribute combination can be explained by one or more of its sub-combinations, AMEP is able to recover such sub-combinations rather than declare the longer one to be significant. In order to demonstrate this important capability, we generate four groups of data in the first simulation study as follows.

#### 4.1.1 Data

We first construct a synthetic group of data consisting of 20 instances as the foreground data, where each instance has two planted motifs, (..........Y..PE......) and (.......ND.Y..........). For each instance, the non-conserved positions are chosen uniformly according to the background distribution (here we use set $N$ of PhAtY as the background distribution). We then add 20, 40 and 60 peptides chosen randomly from the background data to the initial foreground data, yielding four groups of foreground data consisting of 0%, 33.3%, 50% and 60% background peptides respectively. For each foreground data set $P$, we eliminate the peptides already in $P$ from the background data of PhAtY and the remaining peptides form the new background data set $N$. Their characteristics are the following:

- PhAtY_1: It contains 40 foreground sequences ($P$ set) and 304,343 background sequences ($N$ set).
- PhAtY_2: It contains 60 foreground sequences ($P$ set) and 304,323 background sequences ($N$ set).
- PhAtY_3: It contains 80 foreground sequences ($P$ set) and 304,303 background sequences ($N$ set).
- PhAtY_4: It contains 100 foreground sequences ($P$ set) and 304,283 background sequences ($N$ set).

#### 4.1.2 Results

Similar to experiments on the real data, we take $p$-value threshold as 0.05, which means that we keep the type I error or significance level at 0.05. The support threshold s for the four data sets is set to be 0.15, 0.1, 0.075 and 0.06, respectively. Here this parameter is manually tuned so as to report as less false positives as possible while retaining all true positives. As the synthetic motifs we plant are (..........Y..PE......) and (.......ND.Y..........), we treat (.......N..Y..........), (........D.Y..........), (..........Y..P.......), (..........Y...E......), (.......ND.Y..........) and (..........Y..PE......) as true phosphorylation motifs. We perform 1,000 permutations for each

Table 3. The reported motifs of different size on PhAtY_1 and PhAtY_2. Here the support threshold $s$ is 0.15 and 0.1 respectively and the size of motifs ranges from 1 to 3. On PhAtY_1, AMEP returns no false positives containing true sub-motifs while SP returns 4 such motifs; on PhAtY_2, AMEP returns 0 such motifs while SP returns 5

| | *PhAtY_1* | | *PhAtY_2* | |
|---|---|---|---|---|
| | *SP* | *AMEP* | *SP* | *AMEP* |
| *1-motifs* | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y..........<br>.........KY.......... | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y.......... | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y..........<br>.........KY..........<br>..........Y.....K.... | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y.......... |
| *2-motifs* | ..........Y..PE......<br>.......ND.Y..........<br>.L.....N..Y..........<br>.L......D.Y..........<br>...S......Y..P....... | ..........Y..PE......<br>.......ND.Y.......... | ..........Y..PE......<br>.......ND.Y..........<br>.L.....N..Y..........<br>.L......D.Y..........<br>...S......Y..P.......<br>.L...S....Y..........<br>.........KY..P....... | ..........Y..PE......<br>.......ND.Y..........<br>.L...S....Y.......... |
| *3-motifs* | .L.....ND.Y.......... | | .L.....ND.Y.......... | |

Table 4. The reported motifs of different size on PhAtY_3 and PhAtY_4. Here the support threshold $s$ is 0.075 and 0.06 respectively and the size of motifs ranges from 1 to 3. On PhAtY_3, AMEP returns 1 false positive containing true sub-motifs while SP returns 9 such motifs; on PhAtY_4, AMEP returns 2 such motifs while SP returns 10

| | *PhAtY_3* | | *PhAtY_4* | |
|---|---|---|---|---|
| | *SP* | *AMEP* | *SP* | *AMEP* |
| *1-motifs* | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y..........<br>.........KY..........<br>..........Y.....K....<br>..........Y......S...<br>.....S....Y..........<br>..........Y.D........<br>....F.....Y.......... | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y.......... | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y..........<br>.........KY..........<br>..........Y.....K....<br>..........Y......S...<br>.....S....Y..........<br>......H...Y..........<br>....F.....Y..........<br>...S......Y.......... | ..........Y...E......<br>..........Y..P.......<br>........D.Y..........<br>.......N..Y.......... |
| *2-motifs* | ..........Y..PE......<br>.......ND.Y..........<br>.L.....N..Y..........<br>.L......D.Y..........<br>...S......Y..P.......<br>.L...S....Y..........<br>.........KY..P.......<br>..G....N..Y..........<br>..G.....D.Y..........<br>........D.Y...L......<br>..........Y......S.L. | ..........Y..PE......<br>.......ND.Y..........<br>.L...S....Y..........<br>..........Y......S.L.<br>........D.Y...L...... | ..........Y..PE......<br>.......ND.Y..........<br>.L.....N..Y..........<br>.L......D.Y..........<br>...S......Y..P.......<br>.L...S....Y..........<br>.........KY..P.......<br>..G....N..Y..........<br>..G.....D.Y..........<br>........D.Y...L......<br>..........Y......S.L.<br>........S.Y..P....... | ..........Y..PE......<br>.......ND.Y..........<br>........S.Y..P.......<br>........D.Y...L...... |
| *3-motifs* | .L.....ND.Y..........<br>..G....ND.Y.......... | | .L.....ND.Y..........<br>..G....ND.Y.......... | |

method on each dataset and the results are described in Tab. 3, Tab. 4 and Fig. 7.

From the reported motifs in Tab. 3 and Tab. 4, we can see that AMEP is much more powerful on eliminating false positives than SP, especially on false positives which contain a true positive as its sub-motif. We summarize the information in Fig. 7, from which we can infer that when the effect of motifs can be explained by one or more of its sub-motifs, AMEP can effectively recover such sub-motifs rather than declare the longer ones to be significant.

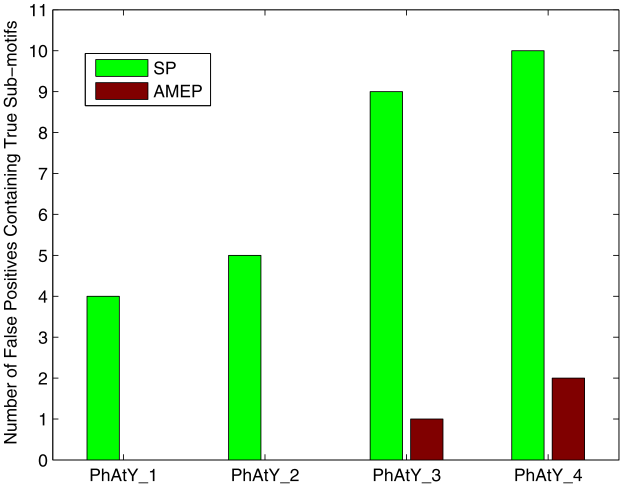

*Figure 7. The comparison on the number of false positives containing true sub-motifs between SP and AMEP. Here we perform 1,000 permutations in both methods on PhAtY_1, PhAtY_2, PhAtY_3 and PhAtY_4.*

## 4.2 Simulation study for comparing the performance

### 4.2.1 Data

In order to demonstrate the effectiveness of our methods in phosphorylation motif discovery, we should generate simulated data whose characteristics are as close as possible to the real data. To achieve this objective, we construct the simulated data according to the method shown in Algorithm 5.

Using Algorithm 5, we obtain 10 simulated datasets. Low support or low odds ratio motifs are rarely reported by the existing methods or our permutation methods on the real data, so such motifs planted in our simulated data should not be regarded as true phosphorylation motifs. In order to alleviate this problem, we sort those 6 planted 1-motifs in descending order with respect to their odds ratio and keep the first 4 motifs as the true phosphorylation motifs. We perform the same operation for 6 planted 2-motifs as well. As a result, 8 motifs are treated as true phosphorylation motifs.

### 4.2.2 Results

In Motif-All, we use its default configurations: the support threshold = 0.05 and the significance threshold = $10^{-6}$. In SP, AMEP and MAMEP, we set the support threshold to 0.05 and the $p$-value threshold (significance level) to 0.05. We perform 1,000 permutations for each permutation method. The comparison result in terms of the power and false discovery rate[1] on all ten simulated data sets are given in Tab. 5. To describe the results in Tab. 5 in a more vivid manner, we further summarize the comparison results using boxplot graph in Fig. 8 and Fig. 9, respectively.

**Algorithm 5** Simulation Data Generation Process

```
1. Initialize: Let $A$ be the set of amino acids whose frequency value in PhAtY is above a threshold, $A$={"G", "A", "S","P", "V", "T", "L", "I", "N", "D", "Q", "K", "E", "M", "H", "F", "Y", "R"}.
2. Randomly choose a subset of $k$ peptides from $N$ set of PhAtY as the original $P$ set (here we set $k = 100$).
3. Plant a set of 1-motifs and a set of 2-motifs. These two sets of planted motifs have the same size $L$ (here we use $L$=6). The support of injected 1-motifs ranges from 0.12 to 0.23 and we enforce that the sum of their support equals 1. The support of planted 2-motifs ranges from 0.05 to 0.1. Both the 1-motifs and 2-motifs are planted in an iterative manner. For $j = 1, 2, \ldots, L$, let $\hat{P}$ be the subset of peptide sequences from $P$ that have not been planted into any motifs. Firstly, we randomly choose two amino acids $S_{j1}$ and $S_{j2}$ from $A$ and two successive positions $Pos_{j1}$ and $Pos_{j2}$. The two positions should be different from those that have been selected and none of them should be the position of phosphorylated residue. Then, we choose the first $C_{j1}$ peptides from $\hat{P}$ and set their residues at $Pos_{j1}$ to be $S_{j1}$. Finally, we choose next $C_{j2}$ peptides from $\hat{P}$ and set their residues at $Pos_{j1}$ and $Pos_{j2}$ to be $S_{j1}$ and $S_{j2}$. This process plants one 1-motif with support $(C_{j1}+C_{j2})/k$ and another 2-motif with support $C_{j2}/k$.
4. Eliminate all peptides that appear in our simulated $P$ set from $N$ set of PhAtY, and the remaining peptides form simulated $N$ set.
```

From Tab. 5, Fig. 8 and Fig. 9, we can find that both Motif-All and SP can discover all true phosphorylation motifs since their power value is always 1 throughout all 10 data sets. However, their false discovery rates are higher. In average, 72.6% of motifs reported by Motif-All and 79% of motifs found by SP are false positives. Since it is very expensive and time-consuming to perform biological validation on reported phosphorylation motifs, too many false positives will lead to an unnecessary cost. From this perspective, the performances of Motif-All and SP are not so satisfactory. In contrast, AMEP has lower FDR compared with Motif-All and SP and its power almost equals 1. And MAEMP achieves a power as high as 0.863 at the FDR of 0.174. Furthermore, we have observed that true motifs MAMEP leaves out mostly have lower significance. In other words, MAMEP

[1] Let $M$ be the set of reported phosphorylation motifs, $T$ be the true ones and $F$ be the false ones so that $M = T \cup F$. Let $\hat{T}$ be the set of true phosphorylation motifs planted in the simulated data. Then power is computed as $|T|/|\hat{T}|$ and false discovery rate is computed as $|F|/|M|$.

Table 5. Performance comparison of different methods in terms of both power and false discovery rate. Here we perform 1,000 permutations for each permutation method on ten simulated datasets

| *Data* | *Power* | | | | *False Discovery Rate* | | | |
|---|---|---|---|---|---|---|---|---|
| | *Motif-All* | *SP* | *AMEP* | *MAMEP* | *Motif-All* | *SP* | *AMEP* | *MAMEP* |
| *Data_1* | 1 | 1 | 1 | 0.875 | 0.556 | 0.652 | 0.5 | 0 |
| *Data_2* | 1 | 1 | 0.875 | 0.875 | 0.724 | 0.795 | 0.5 | 0.222 |
| *Data_3* | 1 | 1 | 1 | 1 | 0.742 | 0.805 | 0.619 | 0.333 |
| *Data_4* | 1 | 1 | 1 | 0.75 | 0.771 | 0.814 | 0.652 | 0 |
| *Data_5* | 1 | 1 | 1 | 1 | 0.724 | 0.795 | 0.5 | 0 |
| *Data_6* | 1 | 1 | 1 | 0.875 | 0.714 | 0.789 | 0.619 | 0.125 |
| *Data_7* | 1 | 1 | 1 | 0.75 | 0.742 | 0.771 | 0.619 | 0.143 |
| *Data_8* | 1 | 1 | 1 | 0.625 | 0.758 | 0.833 | 0.579 | 0.286 |
| *Data_9* | 1 | 1 | 1 | 1 | 0.742 | 0.814 | 0.667 | 0.333 |
| *Data_10* | 1 | 1 | 1 | 0.875 | 0.784 | 0.83 | 0.5 | 0.3 |
| *Ave* | 1 | 1 | 0.988 | 0.863 | 0.726 | 0.79 | 0.576 | 0.174 |

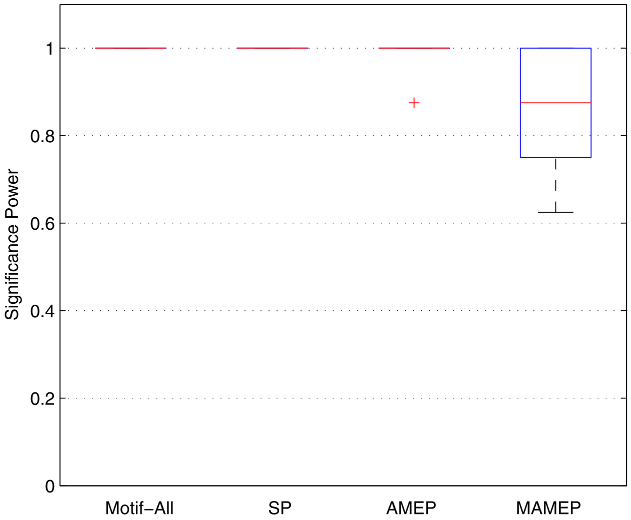

Figure 8. Power of Motif-All, SP, AMEP and MAMEP on the simulated data.

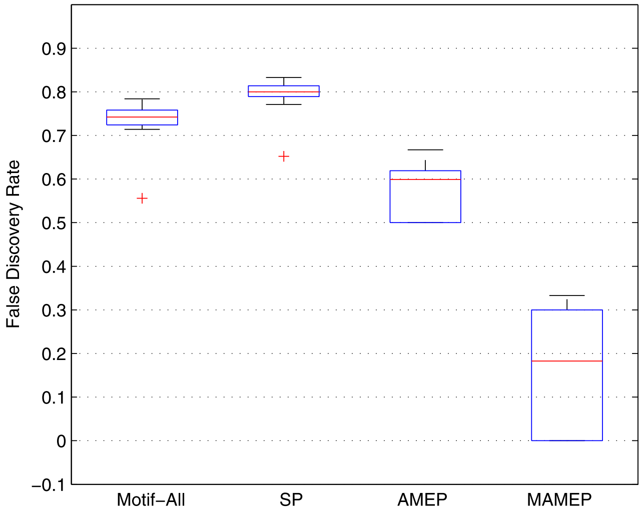

Figure 9. False discovery rate of Motif-All, SP, AMEP and MAMEP on the simulated data.

retains true motifs which are the most statistically significant.

According to different goals, AMEP and MAMEP can both be very useful: if we don't want to leave out true motifs and don't care so much about the cost, AMEP is a better choice; however, if we aim at discovering most of the significant motifs at a lower validation cost, MAMEP is strongly recommended.

## 5. CONCLUSION

We introduce three permutation methods, namely, SP, AMEP and MAMEP for significance testing of phosphorylation motifs. Both the experimental results on real data and simulated data show that our methods are powerful in separating true phosphorylation motifs from false ones.

*Received 29 March 2011*

## REFERENCES

[1] Amanchy, R., Periaswamy, B., Mathivanan, S., Reddy, R., Tattikota, S. G., and Pandey, A. (2007). A curated compendium of phosphorylation motifs. *Nature Biotechnology* **25** 285–286.

[2] Schwartz, D. and Gygi, S. P. (2005). An iterative statistical approach to the identification of protein phosphorylation motifs from large-scale data sets. *Nature Biotechnology* **23** 1391–1398.

[3] Ritz, A., Shakhnarovich, G., Salomon, A. R., and Raphael, B. J. (2009). Discovery of phosphorylation motif mixtures in phosphoproteomics data. *Bioinformatics* **25** 14–21.

[4] He, Z., Yang, C., Guo, G., Li, N., and Yu, W. (2011). Motif-all: Discovering all phosphorylation motifs. *BMC Bioinformatics* **12** **S22**.

[5] Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. *Proc. of VLDB'94* **20** 487–499.

[6] Durek, P., Schmidt, R., Heazlewood, J. L., Jones, A., MacLean, D., Nagel, A., Kersten, B., and Schulze, W. X. (2010). PhosPhAt: The *Arabidopsis thaliana* phosphorylation site database. An update. *Nucleic Acids Research* **38** D828–D834.

[7] Heazlewood, J. L., Durek, P., Hummel, J., Selbig, J., Weckwerth, W., Walther, D., and Schulze, W. X. (2007). PhosPhAt: A database of phosphorylation sites in Arabidopsis thaliana and a plant-specific phosphorylation site predictor. *Nucleic Acids Research* **36** D1015–D1021.

[8] Ma, L., Assimes, T. L., Asadi, N. B., Iribarren, C., Quertermous, T., and Wong, W. H. (2010). An almost exhaustive search-based sequential permutation method for detecting epistasis in disease association studies. *Genetic Epidemiology* **34** 434–443.

[9] Han, J. and Kamber, M. (2006). *DataMining: Concepts and Techniques*, Second Edition, Elsevier Inc.

[10] Hesterberg, T., Monaghan, S., Moore, D. S., Clipson, A., and Epstein, R. (2005). *Introduction to the Practice of Statistics*, W.H.Freeman & Co.

[11] Han, J., Pei, J., Yin, Y., and Mao, U. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery* **8** 53–87. MR2037831

Haipeng Gong
School of Software
Dalian University of Technology
China
E-mail address: haipengxf@gmail.com

Zengyou He
School of Software
Dalian University of Technology
China
E-mail address: zyhe@dlut.edu.cn