Fast Westfall-Young Permutation Procedure for Combinatorial Regulation Discovery

Aika Terada*, Koji Tsuda†‡, and Jun Sese*
* *Dept. of Computer Science, Tokyo Institute of Technology, Tokyo, Japan*
† *Computational Biology Research Center (CBRC),*
National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
‡ *Minato Discrete Structure Manipulation System Project, ERATO,*
Japan Science and Technology Agency, Hokkaido, Japan
Email: terada@ss.cs.titech.ac.jp, koji.tsuda@aist.go.jp, sesejun@cs.titech.ac.jp

***Abstract*—Three or more transcription factors (TFs) often work together, and the combinatorial regulations are essential in cellular machinery. However, it is impossible to discover statistically significant sets of TF binding motifs due to the necessity of the multiple testing procedure. To improve the sensitivity of widely used Bonferroni correction or its modified methods, such as Holm procedure, Westfall-Young permutation procedure (WY-procedure) has often been applied. However, few studies have used WY-procedure for the discoveries of the combinatorial effects of the motifs because of the extremely large computational time. In this paper, we propose an efficient branch-and-bound algorithm to perform WY-procedure to enumerate statistically significant motif combinations. When we use WY-procedure for the combinatorial regulation discovery, finding the minimum P-value from each permuted dataset consumes an enormous amount of time. We show that a combination that has the possibility to achieve the minimum P-value appears with high frequency over the threshold in dataset. This property enables a frequent itemset mining algorithm to efficiently select the candidates to achieve the minimum P-value. Our demonstrations using yeast and human transcriptome datasets show that the proposed algorithm is orders-of-magnitude faster than WY-procedure, and can practically list statistically significant motif combinations even when any combinations are considered.**

***Keywords*-Multiple Test, Combinatorial Regulation, Westfall-Young Permutation Procedure, Motif Combination, Gene Expression**

I. Introduction

Combinatorial activities of different transcription factors (TFs) are essential to respond to a wide spectrum of environmental and developmental signals [1]. However, computational scanning of TF binding motifs is often limited to single motifs or pairs due to the necessity of a multiple testing procedure. To detect statistically significant motif combinations associated with a gene expression profile, statistical tests for all combinations are performed. Such multiple test results in a high false discovery ratio. For example, when we set the significance level α to 0.05 and perform 100 tests, the probability that at least one false discovery happens is $1 - 0.95^{100} = 0.994$.

Multiple testing correction procedures have been proposed [2]–[4] to avoid false discoveries. A simple and widely used theoretical approach is Bonferroni correction [2]. Given M tests, Bonferroni correction controls the probability of occurrence of at least one false discovery, called family-wise error rate (FWER), to be under α by calibrating the adjusted significance level to $\delta = \alpha/M$. When we check all possible motif combinations, M increases exponentially to the number of motifs, and δ becomes a very small value. Hence, the discovery of statistically significant motif combinations is extremely unlikely. Even when we use methods improving detection power of Bonferroni correction, such as Holm procedure [3], the same problem hides the statistical significance of many combinations.

Another strategies, such as Westfall-Young permutation procedure (WY-procedure) [4], generate a null distribution from thousands of randomly permuted datasets, and determines δ based on the distribution. It is known that WY-procedure has higher detection power than Bonferroni correction, and improved methods of it [5], [6] have been widely used in practice [7], [8]. While the high sensitivity may allow us to discover statistically significant combinations, a large amount of computing time is required in permutation tests, and this prohibits the application of WY-procedure to combination discovery. When WY-procedure is used to find combinatorial regulations on 1,000 permutation tests for 100 motifs, statistical tests are performed $1000 \cdot (2^{100} - 1) \approx 10^{33}$ times.

In this paper, we propose an efficient algorithm to perform WY-procedure to enumerate statistically significant sets of motifs. As a test statistic, we use Fisher's exact test due to its popularity and accuracy. However, this algorithm can be extended to use Chi-square test and Mann-Whitney U test instead of using Fisher's exact test. To describe the algorithm, we first introduce the calculation of the lower bound of the P-value of a motif set. With the lower bound, we identify motif sets that never achieve the minimum P-value. These sets do not affect the multiple testing result. We next associate the lower bound of a motif set with the number of genes that are targeted by the set. This property enables us to list motif sets whose lower bounds are less than the threshold by using the motif set frequency. Such motif sets can be efficiently listed with a frequent itemset mining (FIM) algorithm. It is known that FIM algorithm requires a large computing time when the threshold for the frequency of motif sets is small. To avoid the small threshold, we gradually decrease the threshold to find the minimum P-value.

We apply the algorithm to gene expression profiles in yeast and human transcriptome datasets. We demonstrate that our method achieves orders of magnitude acceleration over WY-procedure, and can successfully calibrate the significance level even when all motif sets are considered. Our method finds statistically significant motif sets of up to nine motifs from real human dataset.

II. Westfall-Young Permutation Procedure

In this section, we introduce WY-procedure to control FWER under the given significance level α.

To simplify the problem, we assume here that each of the N genes has a gene expression level. If a motif sequence exists in the promoter region of a gene, the motif is regarded to target the gene. When we focus on a set of multiple motifs, we assume that the targeted genes of the set have all of the motifs in the set in their promoter regions. Here, we describe a set of motifs as a set of items, called *itemset*. For a given itemset I, we classify N genes into two ways: targeted or untargeted, and highly expressed or not. We define $x(I)$ as the number of targeted genes by I. n and $a(I)$ indicate the number of highly expressed genes and the number of genes that are targeted by I and are highly expressed, respectively.

The P-value of I is calculated as the following equation with one-sided Fisher's exact test.

$$P(I) = \sum_{a_i = a(I)}^{a_{\max}} \frac{\begin{pmatrix} n \\ a_i \end{pmatrix} \begin{pmatrix} N-n \\ x(I)-a_i \end{pmatrix}}{\begin{pmatrix} N \\ x(I) \end{pmatrix}}, \tag{1}$$

where $a_{\max} = \min\{x(I), n\}$.

WY-procedure uses an empirical distribution computed from randomly permuted datasets to determine δ [4]. When FWER is controlled to α, the α percentage of all permuted datasets has at least one P-value smaller than δ; that is, the minimum P-value of the α percentage of permuted datasets is less than δ. Therefore, we generate the empirical distribution of the minimum P-values from each randomly permuted dataset, and then the α percentile point in the distribution becomes δ in WY-procedure. After the calibration, when $P(I) \leq \delta$, I is regarded as the significant set of motifs. We here call such I a *significant itemset*.

The algorithm of WY-procedure is shown as follows. N genes and a set of itemsets $\mathcal{I}$ to be tested are given. WY-procedure generates a permuted dataset by randomly shuffles of the relationships between genes and the expression levels. The associations between a gene and TFs that target the gene are held. The minimum P-value among all tests in each permuted dataset is computed. Gathering K minimum P-values provides the simulated null distribution by repetition of this procedure. δ is calculated as the α percentile point in the distribution.

The running time of WY-procedure increases linearly with the number of itemsets to be tested. When all possible itemsets are tested, the time increases exponentially with the number of motifs. Even for a small number of motifs, an intractable amount of time might be required.

III. Properties to Find the Minimum P-value Efficiently

We will propose an efficient algorithm to find the minimum P-value among all of the itemsets in each permuted dataset in Section IV. Since this task is NP-hard [9], in this section, we introduce a pruning strategy for itemsets to solve the problem in practice. Because minimum P-value calculation of thousands of permuted datasets is required to generate the empirical distribution of minimum P-values, the strategy has a large impact on acceleration of the entire computing time.

Fig. 1 illustrates the important properties of our algorithm. In Fig. 1(a), a light red bar shows the P-value of an itemset. A red point represents the lower bound of the P-value of the itemset, which will be shown in Property 1. Assume that we have a threshold t for the lower bound (the red solid line). Itemsets whose lower bounds are at most t are defined as *candidate itemsets* (itemsets in the red arrow region). The other itemsets are defined as *non-candidate itemsets*. When a candidate itemset has the P-value smaller than t, like itemset $\{1, 2, 3\}$, all of the non-candidate itemsets such as $\{1, 2, 3, 4\}$ never achieve the minimum P-value because their lower bounds for the P-value are larger than t. Hence, we can avoid P-value calculation of non-candidate itemsets to find the minimum P-value.

However, there are two problems to this approach. One is how to efficiently enumerate the candidate itemsets with t. The other is how to determine t. To solve the first problem, we show that the candidate itemsets with t are identical to the itemsets targeting λ or more genes, where λ corresponds to t. The lower bound of P-value has one-to-one correspondence to the number of genes that contain all of the motifs in the itemset (Fig. 1(b)). This relation enables us to convert t to λ. With λ, we can efficiently list itemsets contained in λ or more genes by using FIM algorithm. Since the frequent itemsets correspond to the candidate itemsets, we can accelerate the findings of the candidate itemsets. For the second problem, we use λ instead of t. To find the optimal threshold, we propose an algorithm in which λ decreases gradually in Section IV.

A. Lower Bound of Fisher's Exact Test

Suppose that an itemset I targets $x = x(I)$ genes. The lower bound of $P(I)$ can be described independently of the choice of I and only depends on x from Terada *et al.* [10].

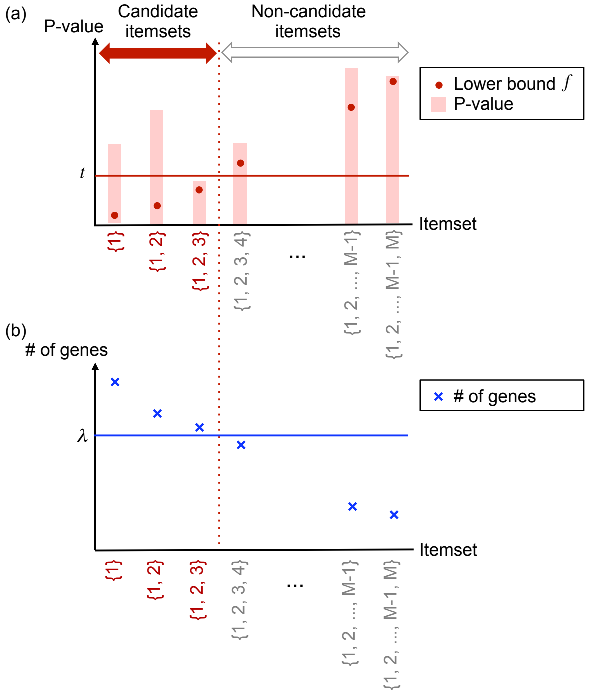

Figure 1. Illustration of the properties of candidate itemsets. (a) Candidate and non-candidate itemsets. Each point and bar indicate the lower bound $f(x)$ and its P-value, respectively. The candidate and non-candidate itemsets are represented as the red and white arrows, respectively. Upon adding a motif to an itemset, $f(x)$ increases while the P-value does not always increase. (b) Relationship between an itemset and the number of genes that have all of the motifs in I. On adding a motif to an itemset, the number of genes that contain I decreases. A candidate itemset contains TF motifs targeting λ or more genes. Our algorithm selects the candidate itemsets with the number of genes instead of using the lower bound of the P-value since an FIM algorithm is directly applicable.

Property 1. *For an itemset targeting x genes, the lower bound of the P-value is described as*

$$f(x)=\begin{cases}\binom{n}{x}\Big/\binom{N}{x} & \textit{for } x\le n,\\[2ex] 1\Big/\binom{N}{n} & \textit{otherwise.}\end{cases}\qquad(2)$$

To use two-sided Fisher's exact test, the lower bound is achieved at the smaller value between the two edge points of $\min\{0, n+x-N\}$ and $\min\{n, x\}$.

B. Monotonicity of Lower Bound

$f(x)$ monotonically increases upon adding a motif to I since $x = x(I)$ decreases by adding an item to I and $f(x)$ decreases to x from Terada *et al.* [10].

Property 2. *$f(x)$ decreases to x.*

The lower bound and its monotonicity are held on other types of statistical tests including Mann-Whitney U test for a single ranked series. Therefore, the proposed algorithm, which will be introduced in Section IV, can be extended to these statistical tests.

C. Listing Candidate Itemsets with FIM Algorithm

The following properties enable us to use FIM algorithm to list the candidate itemsets with t.

Property 3. *For itemsets I and $I' \supseteq I$, $f(x(I)) \le f(x(I'))$.*

Proof: Because $x(I) \ge x(I')$, $f(x(I)) \le f(x(I'))$. ■

The monotonicity is useful for listing all itemsets within the candidate itemsets.

Property 4. *Assuming that $f(\lambda) \le t < f(\lambda-1)$, the candidate itemsets with t are identical to the itemsets targeting λ or more genes.*

Proof: When $x(I) \ge \lambda$ for an itemset I, I is the candidate itemset because $f(x(I)) \le f(\lambda)$ from Property 2. Otherwise, I is the non-candidate itemset with t. Hence, Property 4 holds. ■

Properties 3 and 4 allow us to use λ in FIM algorithm to list candidate itemsets. Since FIM algorithm requires a large amount of computing time for a small λ, we start from a large λ and gradually decrease λ to find the minimum P-value of all of the itemsets. When the smallest P-value in the candidate itemsets is larger than $f(\lambda)$, the P-value is the minimum P-value of all of the itemsets.

IV. Algorithm

In this section, we propose an algorithm called FastWY to efficiently find the adjusted significance level δ for keeping FWER $\le \alpha$ using randomly permuted datasets. For each permuted dataset, FastWY discovers the minimum P-value among all of the itemsets by using Properties 1 to 4. Gathering them gives us null distribution of the minimum P-values, in which the α percentile value can be used as the adjusted significance level for multiple testing.

FastWY includes three steps. Step 1: Estimation of a null distribution. Step 2: Calculation of the δ. Step 3: Listing itemsets whose P-values are less than or equal to δ. The overall structure of FastWY is the same as that of WY-procedure in Section II. The main difference is in Step 1. To find the minimum P-value efficiently, we introduce a branch-and-bound method by using Properties 1-4.

In Step 1, for a permuted dataset, we compute the minimum P-value among all of the tests. From Property 1, when $f(\lambda)$ is larger than the minimum P-value, it is impossible for any of the non-candidate itemsets to achieve the minimum P-value. Therefore, given that p is the smallest P-value of the candidate itemsets and $f(\lambda) > p$, p is the minimum P-value of all of the possible itemsets. Since a small λ requires a large amount of computing time for FIM, we start from a large value of λ and gradually decrease λ while $f(\lambda) \le p$. First, λ is assigned $\min\{x_{\max}, n\}$, where

Table I
DATASET

Dataset	# of motifs (M)	# of genes (N)	# of highly exp. genes (n)	Avg. # of motifs
Yeast	102	5,988	639 (10.7 %)	0.509
Human	397	11,610	638 (5.5 %)	6.510

$x_{\max} = \max\{x(I) \text{ for } I = \{i\} \text{ where } i = 1, \ldots, M\}$. FIM algorithm lists candidate itemsets on λ. FastWY computes the smallest P-value p among the candidate itemsets. Then, FastWY compares p with $f(\lambda)$. If $p > f(\lambda)$, a non-candidate itemset can achieve a P-value smaller than p, and hence we replace λ with $\lambda - 1$, and repeat the procedures. A decrease in λ causes an increase in $f(\lambda)$ from Property 2, which brings $f(\lambda)$ closer to p. If $p \leq f(\lambda)$, then p is the minimum P-value.

Step 1 is performed on K different permuted datasets, and generates K minimum P-values. Based on the P-values, adjusted significance level δ is found from Step 2. In Step 3, the significant itemsets for δ are listed. The step is similar to the minimum P-value findings in Step 1 except using δ instead of p. After listing the candidate itemsets on δ and calculating P-values, the itemsets whose P-values are less than δ are statistically significant itemsets after multiple testing correction.

V. PERFORMANCE EVALUATION

We check the computation efficiency of FastWY using the real dataset. All programs are written in Python 2.7 except for FIM. For the FIM program, we use LCM [11] due to its fast calculation. All experiments are tested on a machine with 2 AMD Opteron processors at 2.3 GHz with 32 GB of RAM running RedHat Linux.

A. Datasets and Experimental Settings

We compare the performance of FastWY and WY-procedure with the two datasets shown in Table I. For the yeast dataset, gene expression levels are obtained from Gasch *et al.* [12]. If the expression level of a gene in the amino acid starvation is two times higher than that in the control environment, the gene is considered as a highly expressed gene. The relationships between motifs and genes are observed in Harbison *et al.* [13]. For the human dataset, we used gene expression level data observed from 84 tissues [14]. When the expression level of a gene in the trachea is two times higher than that of the median expression calculated from 84 tissues, it is regarded as a highly expressed gene. For the relationships between motifs and genes, we used the Molecular Signatures Database [15]. Since there is no standard method to find the optimal threshold to discriminate between genes that are highly expressed and those that are not, we use an arbitrary threshold.

For the statistical measure, a one-sided Fisher's exact test was used. We set α at 0.05. FastWY estimates null distributions from 1,000 permuted datasets. WY-procedure is measured the computation time with the same setting in FastWY. However, because it requires more than a month, we estimate the running time of WY-procedure on $K = 1,000$ from the time on $K = 100$. To check the scalability, we generate datasets of various sizes by random selection from yeast and human datasets. All analyses were repeated 5 times to calculate the means and variances of the values.

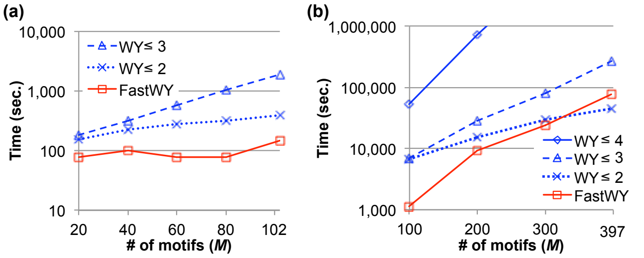

Figure 2. Running time of FastWY and WY-procedure with increasing numbers of motifs. (a) Yeast dataset. (b) Human dataset.

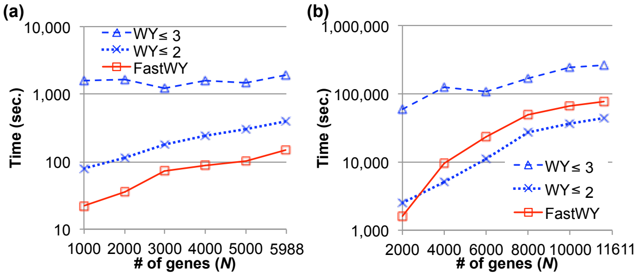

Figure 3. Running time of FastWY and WY-procedure with increasing numbers of genes. (a) Yeast dataset. (b) Human dataset.

B. Comparison between FastWY and WY-procedure

Figs. 2 and 3 indicate that the running time changes with an increasing the number of motifs and genes, respectively. WY ≤ 2 and WY ≤ 3 show WY-procedure for up to two and three motifs, respectively. Note that the y-axis is on a log scale. Since the variance is too small to describe in these figures, we show only the average computation time. In the experiments, FastWY can find the minimum P-values from all permuted datasets even when the sizes of all itemsets are considered.

First, we check the performance upon increasing the number of motifs. Fig. 2(a) shows that FastWY can compute the adjusted significance level δ orders-of-magnitude faster than with WY-procedure in the yeast dataset. The computation time of FastWY increases slightly to M, while that of WY-procedure increases exponentially to M. We performed the same experiments on the human dataset. Fig. 2(b) shows that the computing times for both FastWY and WY-procedure increase to M, and the that computation time of FastWY is faster than that of WY-procedure for up to three motifs (WY ≤ 3).

For an increasing number of motifs, our strategy works more effectively. The entire number of itemsets to be tested increases exponentially to M, while the slope of the computation time of FastWY is much lower than exponential in both Figs. 2(a) and (b). For example in Fig. 2(b), the slope of the computation time of FastWY to M is comparable with that of WY ≤ 3, and is orders of magnitude faster than the slope of WY ≤ 4. Therefore, the slope in Fig. 2(b) implies that our pruning technique works well to reduce the computation time for FastWY. For example, in the human dataset with $M = 397$, FastWY computes P-values for $8.70 \cdot 10^9$ itemsets on average in five repetitions.This value is 83.4 % of WY ≤ 3 since WY-procedure computes the P-value for $1.04 \cdot 10^{10}$ itemsets even when we check combinations of up to three motifs. As a result, FastWY is faster than WY-procedure.

Next, we check the performance with an increasing the number of genes. Fig. 3 shows the running time with varying N. FastWY is faster than WY ≤ 2 in Fig 3(a), and is faster than WY ≤ 3 in Fig. 3(b). Therefore, FastWY succeeds in accelerating WY-procedure.

The computation time of FastWY is increased to N in both figures in Fig. 3 since FastWY tends to finish the computation at a small λ/N for a dataset with large N. For example, in the yeast dataset, FastWY continues its calculation until λ decreases to 1, independent of N. In the human dataset, when N is varied from 2,000 to 11,611, λ gradually increases from 1.0 ± 0.0 to 2.6 ± 0.548 on average in five repetitions. The ratio of λ/N becomes smaller for the larger N in both datasets. Given a small λ/N, FIM algorithm requires a large amount of time for enumerating itemsets whose frequency is larger than λ. Therefore, FastWY requires much time to analyze a dataset with large N. However, FastWY is much faster than WY-procedure, and it enables us to test all of the motif combinations even when any combinations are considered.

FastWY optimizes λ by gradually decreasing the value. FastWY starts from $\lambda = 221$ and finishes at $\lambda = 1$ in yeast dataset, i.e., FIM is performed 220 times. Although all itemsets targeting at least one gene are listed, the computation time is faster than WY-procedure. Hence, the number of repeats does not have huge impact on calculation time. In the human dataset, FIM algorithm is performed 635 times on average and requires about 2,000 seconds, and FastWY finishes when λ is 2.6 on average in five repetitions. This result indicates that FastWY can avoid many P-value calculations. If we compute $\lambda = 1$ for each permuted dataset in the human dataset, 19,513,920 P-value calculations are required, which is 2.26 times more than that for FastWY. It will take a large amount of computation time. For similar reasons, other searching algorithms, such as a binary search, are difficult to apply in finding the optimal λ.

C. Detected Motif Sets

We check significant itemsets in both the yeast and human datasets. We used $\alpha = 0.05$ and $K = 1,000$ for the calculation.

In the yeast dataset, the adjusted significance level δ is 0.000580, and 12 itemsets are detected as significant itemsets. The largest itemset contains four motifs, *CBF1*, *MET4*, *MET31*, and *MET32* whose P-value is 0.000129. This four-motif combination is confirmed that they work in sulfur amino acid metabolism with interactions by biological experiments [16]. Bonferroni correction [2], which is often used theoretically in multiple testing procedure, cannot show the significance due to its too conservative correction. When we test all of the itemsets up to four motifs with Bonferroni correction, the P-value for a significant itemset requires to be less than $1.13 \cdot 10^{-11}$.

In the human dataset, the adjusted significance level δ is 1.781 $\cdot 10^{-7}$. Seven itemsets are detected as significant itemsets, and the largest itemset contains eight motifs, *MYOD*, *E4F1*, *FOXO4*, *ATF*, *CREB*, *CREBP1*, *E12*, and *ATF3*. The P-value is $2.693 \cdot 10^{-8}$. We cannot find this combination as long as we focus on a singleton because any single motif does not have significance. Furthermore, none of the itemsets that contains up to four motifs become a significant itemset. Because no biological investigation has been performed on such large combinations, the findings of statistically significant combinations can motivate biologists to confirm whether these combinations function in cells.

VI. Related Work

To find the combinatorial effects of motifs, screening of all pairs of motifs [17], [18] have been provided. Although these procedures might possibly be improved to screen combinations of three or more motifs, a crucial problem is how to assess the statistical significance of discovered motifs.

As multiple testing procedures for controlling FWER, Bonferroni correction and its variants have been proposed [2], [3]. Their conservativeness results in low detection power especially when there are a large number of tests. These methods could overlook combinatorial regulations because we test a large number of motif combinations.

The large computation time of WY-procedure [4] prohibits its application in biological problems. To overcome this problem, Zhang *et al.* proposed a fast computing method for WY-procedure to find statistically significant pairs of single nucleotide polymorphisms (SNPs) [19]. Extending this method to detect three or more itemsets is not trivial.

VII. Conclusions and Future Work

We proposed an efficient algorithm to accelerate WY-procedure to list statistically significant motif combinations. Our algorithm uses the property that distribution of FWER only depends on the minimum P-values of permuted dataset, and we developed the method to accurately find the minimum value. We showed that our algorithm can successfully calibrate the significance level for combinatorial regulation discovery even when all possible combinations are considered. The computation time is faster than that of WY-procedure when up to three motif combinations are computed, showing that the algorithm achieved orders-of-magnitude acceleration to WY-procedure. The results contain a combination of up to nine motifs from a real human dataset. Since the statistical significance of the combination is guaranteed, this result motivates the biologists or the medical scientists to verify these results experimentally.

This problem has many applications. For examples, the drug combination detection problem requires investigation of combinations of 10,000 drugs, and the discovery of SNPs combinations in medical science requires the consideration of combinations of more than three million positions. However, the number of elements, such as drugs and SNPs, in some of these problems is larger than that we solved. Therefore, improvement of the computational speed of our algorithm would be useful. One solution would be the development of FIM algorithm suitable for our problem. We ran an existing FIM algorithm on every λ, but some overlaps were found between them. Reducing duplication may accelerate the running time of our algorithm. Another solution is fast computation of the FWER distribution. Instead of using the exact minimum value from each permuted dataset, estimation of the value would be enough to adjust the significance level in practice.

Acknowledgment

A.T. is supported by JSPS Research Fellowships for Young Scientists. This work was partially supported by JSPS KAKENHI Grants 24240044, 24651227, 24680032, 25128704 and 25125709 to J.S. K.T. is supported by the FIRST Program, JST CREST and JSPS Kakenhi 25106005. The super computing resource was provided by NIG (ROIS) and HGC (Univ. of Tokyo).

References

[1] T. Ravasi *et al.*, "An atlas of combinatorial transcriptional regulation in mouse and man," *Cell*, vol. 140, no. 5, pp. 744–752, 2010.

[2] C. E. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilità," *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, vol. 8, pp. 3–62, 1936.

[3] S. Holm, "A simple sequentially rejective multiple test procedure," *Scand. J. Stat.*, vol. 6, no. 2, pp. 65–70, 1979.

[4] P. H. Westfall and S. S. Young, *Resampling-based multiple testing: Examples and methods for p-value adjustment*. New York: Wiley, 1993.

[5] S. Dudoit, J. P. Shaffer, and J. C. Boldrick, "Multiple hypothesis testing in microarray experiments," *Statist. Sci.*, vol. 18, no. 1, pp. 71–103, 2003.

[6] N. Meinshausen, M. H. Maathuis, and P. Bühlmann, "Asymptotic optimality of the WestfallYoung permutation procedure for multiple testing under dependence," *The Annals of Statistics*, vol. 39, no. 6, pp. 3369–3391, 2011.

[7] V. G. Cheung *et al.*, "Mapping determinants of human gene expression by regional and genome-wide association." *Nature*, vol. 437, no. 7063, pp. 1365–9, 2005.

[8] J. Winkelmann *et al.*, "Genome-wide association study of restless legs syndrome identifies common variants in three genomic regions." *Nature genetics*, vol. 39, no. 8, pp. 1000–6, 2007.

[9] S. Morishita and J. Sese, "Traversing Itemset Lattices with Statistical Metric Pruning," in *PODS '00*, 2000, pp. 226–236.

[10] A. Terada, M. Okada-Hatakeyama, K. Tsuda, and J. Sese, "Statistical significance of combinatorial regulations." *Proc. Natl. Acad. Sci. USA*, vol. 110, no. 32, pp. 12 996–13 001, 2013.

[11] T. Uno, T. Asai, Y. Uchida, and H. Arimura, "LCM : An efficient algorithm for enumerating frequent closed item sets," in *FIMI'03*, 2003.

[12] A. P. Gasch *et al.*, "Genomic expression programs in the response of yeast cells to environmental changes," *Mol. Biol. Cell*, vol. 11, no. 12, pp. 4241–4257, 2000.

[13] C. T. Harbison *et al.*, "Transcriptional regulatory code of a eukaryotic genome," *Nature*, vol. 431, no. 7004, pp. 99–104, 2004.

[14] A. I. Su *et al.*, "A gene atlas of the mouse and human protein-encoding transcriptomes." *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 16, pp. 6062–6067, 2004.

[15] X. Xie *et al.*, "Systematic discovery of regulatory motifs in human promoters and 3' UTRs by comparison of several mammals." *Nature*, vol. 434, no. 7031, pp. 338–345, 2005.

[16] P. L. Blaiseau, A. D. Isnard, Y. Surdin-Kerjan, and D. Thomas, "Met31p and Met32p, two related zinc finger proteins, are involved in transcriptional regulation of yeast sulfur amino acid metabolism." *Molecular and cellular biology*, vol. 17, no. 7, pp. 3640–3648, 1997.

[17] Y. Pilpel, P. Sudarsanam, and G. M. Church, "Identifying regulatory networks by combinatorial analysis of promoter elements." *Nat. Genet.*, vol. 29, no. 2, pp. 153–159, 2001.

[18] Z. Zhu, J. Shendure, and G. M. Church, "Discovering functional transcription-factor combinations in the human cell cycle." *Genome Res.*, vol. 15, no. 6, pp. 848–855, 2005.

[19] X. Zhang, F. Zou, and W. Wang, "FastANOVA: an Efficient Algorithm for Genome-Wide Association Study," in *KDD'08*, 2008.