High-speed Westfall-Young permutation procedure for genome-wide association studies

Aika Terada*
Research Fellow of JSPS
Dept. of Comp. Biol., The Univ. of Tokyo
BRD, AIST
terada@cbms.k.u-tokyo.ac.jp

Hanyoung Kim *
Dept. of Comp. Sci.,
Tokyo Inst. of Technol.
hanyoung@ss.cs.titech.ac.jp

Jun Sese
BRD, AIST
sese.jun@aist.go.jp

ABSTRACT

Genome-wide association studies (GWASs) are widely used to investigate statistically significant associations between diseases and single nucleotide polymorphisms (SNPs) to identify causal factors of diseases. In GWAS, statistical significance of more than one million SNPs have been recently assessed, but in many case, no associations are found because of the application of conservative multiple testing corrections, such as Bonferroni correction. While more sensitive methods, such as Westfall-Young permutation procedure (WY), would relate more SNPs with diseases, its extremely long computational time has prohibited from the application of WY to GWAS. We introduce an algorithm to accelerate WY, named High-speed Westfall-Young permutation procedure (HWY). HWY utilizes three techniques to make WY computationally practical. First, P-value calculations for SNPs that cannot affect the adjusted significance level are pruned. Second, a lookup table of P-values is used to avoid frequent duplicate calculations. Finally, computations are parallelized using a GPGPU. HWY was 619 times faster than WY and more than 122 times faster than PLINK, a widely used GWAS software, and analyzed a dataset contained one million SNPs and one thousand individuals in approximately two hours. Re-analysis of existing GWAS datasets with HWY may uncover additional hidden SNP-trait associations.

Categories and Subject Descriptors

J.3 [**Life and Medical Science**]: Biology and genetics; G.3 [**Probability and Statistics**]: Statistical computing; H.2.8 [**Database Applications**]: Data mining

*AT and HK contributed equally to this work.

BCB'15, September 09–12, 2015, Atlanta, GA, USA.

http://dx.doi.org/10.1145/2808719.2808721.

General Terms

Algorithms, Performance

Keywords

Multiple testing procedure, Westfall-Young permutation procedure, Genome-wide association study

1. INTRODUCTION

More than a thousand associations between single nucleotide polymorphisms (SNPs) and traits have been uncovered by genome-wide association studies (GWASs) [19], which enumerate SNPs that are statistically significantly associated with a target trait. GWASs are not limited to identifying causal genes for humans [7, 10] but have also been used to determine chromosomal regions related to environmental adaptation in agriculture [1, 9].

Current analyses, however, have two problems. The first is that some statistically significant SNP-trait associations may be overlooked. Because many statistical tests are performed in a GWAS, a multiple testing procedure needs to be applied to avoid false discoveries. While a Bonferroni-like correction [4, 8, 17] has been used in most studies, this causes very conservative adjusted significance levels which may, therefore, overlook substantial associations. The problem arises from the assumption in the Bonferroni inequation that all tests are independent; however, SNPs in each haplotype block are strongly associated with each other. The Westfall-Young permutation procedure [20] (WY) is a multiple testing procedure that can asymptotically compute an optimal adjusted significance level without the assumption of independence [11] and would yield more associations than Bonferroni-like corrections. However, applying WY to GWAS has been limited [5, 21] because of the extensive computational time required to obtain a reliable adjusted significance level.

The second problem involves substituting an asymptotic test, such as the Chi-square test, for an exact test. Fisher's exact test is recommended for GWAS analyses because the approximation of Chi-square test is incorrect when the number of individuals is small [13]. However, this also causes the problem of requiring a large amount of computational time, making it prohibitively difficult for the application to GWAS.

We propose an algorithm called High-speed Westfall-Young permutation procedure (HWY). HWY accelerates WY and

conducts the statistical assessment using not only the Chi-square test but also Fisher's exact test. HWY involves three techniques. The first involves pruning P-value calculations. If a P-value calculation would not affect the estimation of the null distribution, we can immediately stop the calculation without any change in the result. The second uses a lookup table of P-values to eliminate redundant calculations. We found that many calculations of WY are duplicated; therefore, using the lookup table avoids repetitive calculations. The third technique involves the use of parallel computing. Recent computer servers have multi-core CPUs and a GPGPU. Using these can substantially decrease calculation times. Our demonstrations have shown that HWY is 619 times faster than WY and 122 times faster than PLINK [12]. As a result, HWY successfully calculated the adjusted significance level for a dataset containing one million SNPs and one thousand individuals in two hours. When applied to a HapMap dataset [16] and an Alzheimer GWAS dataset [18], the adjusted significance levels by HWY were at least three times higher than those by Bonferroni-like corrections. Because the estimated null distribution is mathematically identical to WY, HWY also provides an asymptotically optimal adjusted significance level.

2. RELATED WORK

Bonferroni correction is the most widely used multiple testing procedure for GWAS and theoretically controls family-wise error rate (FWER), which is the probability of at least one false discovery, below a given significance level α. However, because the theoretical FWER bound is too loose in practice, this correction often calculates a conservative significance level. To solve this problem, several methods have been proposed [8, 14, 17]. However, none of them markedly improve the level because most of them assume that tests are independent.

WY approximates the null distribution of FWER using random permutations and estimates an adjusted significance level. While the adjusted significance level is asymptotically optimal when the number of tests is infinity [11], the intractable number of calculations has prevented its wide application in GWAS. Ge *et al.* proposed an algorithm for accelerating WY for functional enrichment analyses [6]. However, the calculation speed is still not sufficient for practical use in GWAS analyses.

Some research has been tackled to accelerate WY to enumerate significant SNP pairs associated with a target trait [22, 23]. FastChi prunes SNP pairs using the upper bound of the Chi-square test [23]. PBOOST implemented parallelization of the permutation test on GPU [22]. However, both methods use Chi-square test, whereas Fisher's exact test is recommended to assess the significance of categorical data [13]. Terada *et al.* developed FastWY to accelerate WY to find significant combinations of features [15]. While FastWY can perform Fisher's exact test, the calculation is too slow to apply to GWAS.

The false discovery rate (FDR) [2, 3] is another measure to control false discoveries caused by multiple tests. When applied to GWAS, we assume that some SNPs are statistically significantly associated with a trait. However, because this assumption is wrong in some studies, FDR is not always appropriate for GWAS. Therefore, we are only concerned with controlling FWER rather than FDR.

Table 1: A contingency table for a SNP S_i.

	Case	Control	Total
Minor allele	n_i	$x_i - n_i$	x_i
Major allele	$N_{\text{case}} - n_i$	$N - N_{\text{case}} - x_i + n_i$	$N - x_i$
Total	N_{case}	$N - N_{\text{case}}$	N

3. PRELIMINARY

In this section, we formally state the statistical problem in GWAS and introduce WY [20] to control the FWER below a given significance level α.

3.1 Statistical test in GWAS

Given a GWAS dataset, including M SNPs and a trait of N individuals, $\mathcal{S}$ and S_i denote the set of the M SNPs and the i-th SNP in $\mathcal{S}$, respectively. The individuals consist of N_{case} cases and $N - N_{\text{case}}$ controls. The GWAS problem is to enumerate $\mathcal{S}' \subseteq \mathcal{S}$ whose members are statistically significantly associated with the trait. To test if these SNPs are significantly associated, we use Fisher's exact test, because of its generality. However, our algorithm can be applied to other statistical tests, such as the Chi-square test.

When we focus on a SNP S_i, the statistical significance can be assessed with the contingency table in Table 1. When the target species is diploid, such as humans, each SNP site has two alleles, one from the father and one from the mother. Among all individuals, a nucleotide having the largest frequency on S_i is called the *major allele* and the others are called *minor alleles*. x_i and n_i in the table are the total number of individuals having a minor allele at S_i and the number of case individuals having a minor allele at S_i, respectively. $P(S_i)$ denotes the P-value at S_i using the one-sided Fisher's exact test. $P(S_i)$ can be calculated with the following equation.

$$P(S_i) = \sum_{k=n_i}^{\min\{x_i, N_{\text{case}}\}} f(x_i, k),$$

$$\text{where} \quad f(x_i, k) = \frac{\binom{N_{\text{case}}}{k}\binom{N - N_{\text{case}}}{x_i - k}}{\binom{N}{x_i}}. \quad (1)$$

Note that this calculation requires a high computational cost due to numerous multiplications and additions.

3.2 Westfall-Young permutation procedure

When conducting multiple tests, the significance level must be adjusted to control false discoveries. Given the significance level α, WY determines the adjusted significance level δ to control the FWER below α. If no SNPs in $\mathcal{S}$ cause a particular trait, the following equation allows computation of the FWER on δ.

$$\begin{aligned} \text{FWER} &= \Pr(P(S_i) \leq \delta \text{ for at least one } S_i \in \mathcal{S}) \\ &= \Pr\left(\min_{S_i \in \mathcal{S}}\{P(S_i)\} \leq \delta\right). \end{aligned}$$

Therefore, with the probability distribution of the minimum P-value of $\mathcal{S}$, the α percentile of the distribution may be used as δ. However, the distribution is not known in advance; hence, WY calculates δ based on the distribution estimated from thousands of randomly permuted datasets [20]. After calibration, SNPs S_i whose $P(S_i) \leq \delta$ are regarded as

Procedure 1 Westfall-Young permutation procedure (WY)

Input: GWAS dataset G: N individuals including N_{case} cases and a set of SNPs to be tested $\mathcal{S}$.
α: FWER upper limit. K: the number of permutations.
Output: Enumerate significant SNPs.
// Estimate a null distribution
1: $Q \leftarrow \{\}$ // An array of the minimum P-values
2: **repeat**
3: Generate a dataset G' whose associations of the individuals with the trait in G are randomly permuted.
4: $q \leftarrow \min\{P(S) \mid S \in \mathcal{S}\}$ on G' // The minimum P-value
5: Add q to Q
6: **until** K times repeated
// Calibrate the threshold δ
7: Sort Q in ascending order.
8: $\delta \leftarrow Q[\alpha K]$
// Enumerate all SNPs whose P-values are less than δ
9: Output a set of SNPs S where $P(S) \leq \delta$ for $S \in \mathcal{S}$ on G.

statistically significantly associated for that trait. The WY algorithm is described in Procedure 1.

4. ACCELERATING WY

We develop three techniques to accelerate WY. The first is focused on accelerating calculations for each permuted dataset. In a permuted dataset, only the minimum P-value among all tests is required; hence, we develop a method to prune the calculation of tests that have no possibility of achieving the minimum P-value. The second involves elimination of repeated calculations over multiple permuted datasets. Since N and N_{case} are identical values independent of permutations, many duplicate calculations may be performed on each SNP. By preparing a lookup table, we avoid repetitive calculation and increase calculation speed. The third involves the use of parallel computing. Recent computers have multi-core CPUs and/or a GPGPU. Because the calculations between permuted datasets are almost independent, fundamentally, the calculations can be parallelized. However, the pruning method and the lookup table may affect parallelization efficiency. Therefore, we will verify that parallel computing successfully accelerates the calculations in Section 5.4.

4.1 Pruning strategy

We prune P-value calculations for SNPs that do not affect the null distribution in WY. In each permuted dataset, WY calculates the minimum P-value among all tests, which generates the null distribution. In other words, we do not need to calculate precise P-values of SNPs unless they could become the minimum P-value.

When we calculate the P-value for a SNP S_i, the sum of the positive terms in Equation (1) is computed. Therefore, $P(S_i) \geq f(x_i, k)$, where $k \in \{n_i, \ldots, \min\{x_i, N_{\text{case}}\}\}$ holds. From this inequation, because we can calculate the lower bound of the P-value of S_i that is independent from n_i, the lower bound can be used for overall random permutation results.

PROPERTY 1. **Lower bound of P-value.** *For a SNP S_i, the lower bound of the P-value $L(S_i)$ for the one-sided*

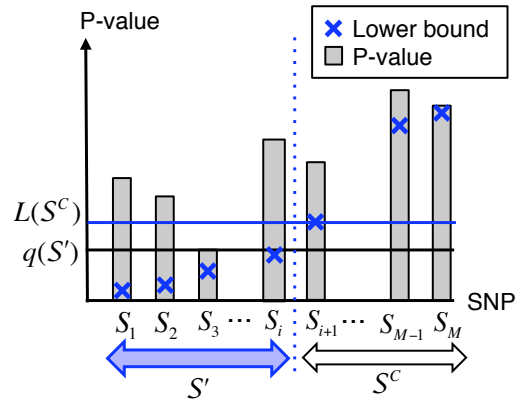

Figure 1: Minimum P-value and lower bound. Each cross mark and bar indicate the lower bound and its P-value, respectively. SNPs are sorted by their lower bounds. $q(\mathcal{S}')$ indicates the minimum P-value in $\mathcal{S}'$. When the lower bounds of all members in $\mathcal{S}^C$ ($L(\mathcal{S}^C)$) are larger than $q(\mathcal{S}')$, SNPs in $\mathcal{S}^C$ must have P-values larger than $q(\mathcal{S}')$ and none achieve the minimum P-value.

Fisher's exact test is expressed by the following equation.

$$L(S_i) = \begin{cases} f(x_i, x_i) & \text{for } x \leq N_{\text{case}}, \\ f(x_i, N_{\text{case}}) & \text{otherwise.} \end{cases} \quad (2)$$

Note that $L(S_i)$ only depends on x_i and is independent from n_i.

For the two-sided Fisher's exact test, the lower bound is smaller value of the two edge points of $\max\{0, N_{\text{case}}+x_i-N\}$ and $\min\{N_{\text{case}}, x_i\}$. The lower bound can be computed for other types of statistical tests as well, including the Chi-square test or the Mann-Whitney U test for a single ranked series. Therefore, our algorithm can be extended to utilize these statistical tests as well.

We next define the minimum P-value q that decreases with the calculations of P-values of SNPs.

PROPERTY 2. **q monotonically decreases to P-value calculation.** *Let $\mathcal{S}$ be a given SNP set. We define $q(\mathcal{S}')$ where $\mathcal{S}' \subseteq \mathcal{S}$ as $\min\{P(S) \mid S \in \mathcal{S}'\}$. For any SNP $S \in \mathcal{S} \backslash \mathcal{S}'$, $q(\mathcal{S}') \geq q(\mathcal{S}' \cup \{S\})$ holds.*

By combining Properties 1 and 2, we can calculate the minimum P-value as the following procedure on a permuted dataset.

1. Let q be 1.0 as an initial value.
2. For each $S \in \mathcal{S}$:
   (a) If $L(S) > q$, we move to the next S without calculating $P(S)$ because S never provides a P-value less than or equal to q.
   (b) Otherwise, calculate $P(S)$ and $q \leftarrow \min\{q, P(S)\}$.
3. After iteration, q is the minimum P-value in the permuted dataset.

Sorting the SNPs to $L(S)$ in advance yields a more efficient algorithm using the following property.

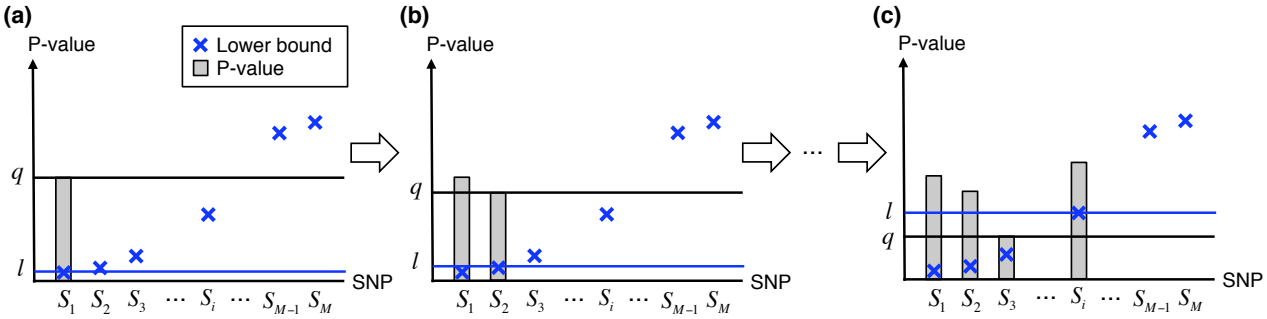

Figure 2: Pruning algorithm to efficiently find the minimum P-value from $\mathcal{S}$. SNPs are sorted by their lower bounds in advance. (a) l is assigned to $L(S_1)$. $P(S_1)$ is computed and $q = P(S_1)$ (black line). (b) Because $q > l$, $S_i (i \geq 2)$ can achieve a P-value smaller than q. Hence, we set l to $L(S_2)$, and q is updated if needed. (c) If $q \leq l$, all remaining S_i have P-values larger than q and therefore do not need to computed.

PROPERTY 3. **Property for an efficient pruning algorithm.** *Let $\mathcal{S}'$ be a subset of $\mathcal{S}$. We define $L(\mathcal{S}')$ to be $\min\{L(S) \mid S \in \mathcal{S}'\}$. Let $\mathcal{S}^C$ be $\mathcal{S} \backslash \mathcal{S}'$. If $q(\mathcal{S}') < L(\mathcal{S}^C)$ holds, no SNP in $\mathcal{S}^C$ results in the minimum P-value.*

Property 3 is illustrated in Fig. 1. In this figure, SNPs are sorted by $L(S)$. $\mathcal{S}'$ and $\mathcal{S}^C$ are $\{S_1, \ldots, S_i\}$ and $\{S_{i+1}, \ldots, S_M\}$, respectively. $q(\mathcal{S}')$ and $L(\mathcal{S}^C)$ are $P(S_3)$ and $L(S_{i+1})$, respectively. Because $q(\mathcal{S}') < L(\mathcal{S}^C)$, for any SNP $S \in \mathcal{S}^C$, $q(\mathcal{S}') < P(S)$. Therefore, we do not need to calculate the precise P-values of SNPs in $\mathcal{S}^C$.

With this property, we propose the following procedure to calculate the minimum P-value on a permuted dataset. Fig. 2 illustrates the procedure.

1. Let $S_1, \ldots, S_M$ be a list of SNPs in $\mathcal{S}$ sorted in ascending order of $L(S)$.
2. Let q be 1.0 as an initial value.
3. For $i = 1$ to M:
   (a) If $L(S_i) > q$, we immediately end this loop because $S_j (j \geq i)$ never provides a P-value less than or equal to q.
   (b) Otherwise, calculate $P(S_i)$ and $q \leftarrow \min\{q, P(S_i)\}$.
4. After iteration, q is the minimum P-value in the permuted dataset.

There are two differences between this and the previous procedure: SNPs are first sorted at number 1 and the calculation is stopped at number 3(a). We can avoid the comparison of $L(S_j)$ with $q(\mathcal{S}')$ while providing an identical result by using Property 3.

Importantly, the sorting process is only performed once and is not performed again even after the dataset is permuted. $L(S_i)$ does not depend on n_i, and the other variables x_i, N_{case} and N are constant values on the permutation. Therefore, the sorted list of SNPs is independent from the permuted result.

4.2 Lookup table of P-values

Statistical tests require large computing costs because they include many multiplication and addition steps. Faster calculation of these would reduce the overall computational time for WY. When we focus on the P-values of SNP S_i over multiple permuted datasets, while x_i and n_i in Table 1 are variable, N and N_{case} are constant. Furthermore, both N and N_{case} are independent from the selection of the SNP. Therefore, P-values only depend on x_i and n_i. Based on these observations, we can generate a single lookup table of the calculated P-values whose keys are x_i and n_i, and if these key pairs have already been computed in a previous calculation, we use the value in the lookup table without recalculating the P-value.

However, x_i and n_i range from 1 to N_{case} and from 0 to x_i, respectively; therefore, the lookup table may be required to keep a substantial number of values for a large GWAS dataset. This situation arises if each pair of keys is only used

Procedure 2 Proposed: High-speed Westfall-Young (HWY)

Input: GWAS dataset G: N individuals including N_{case} and a set of M SNPs $\mathcal{S}$.
α: FWER upper limit. K: The number of permutations.
Output: Enumerate statistically significant SNPs

```
1: S_1, ..., S_M is a list of SNPs in ascending order of L(S).
2: T ← {} // A lookup table of the P-values.
3: Q ← a priority queue to keep the minimum P-values
4: for k = 1 to K do
5:    Generate a permuted dataset G'
6:    q ← 1.0
7:    q ← αK-th value in Q if |Q| ≥ αK.
8:    for i = 1 to M do
9:       break if L(S_i) ≥ q
10:      if T has a key of (x_i, n_i) then
11:         P(S_i) ← T(x_i, n_i)
12:      else
13:         Compute P(S_i) and store the value in T(x_i, n_i)
14:      end if
15:      q ← min{P(S_i), q}
16:   end for
17:   Add q to Q
18: end for
    // Calibrate the threshold δ
19: δ ← Q[αK]
    // Enumerate all SNPs whose P-values are less than δ
20: Output a set of SNPs S where P(S) ≤ δ for S ∈ S on G.
```

once, which causes no efficiency improvement. In Section 5.2, we will investigate whether the lookup table is effective for real GWAS datasets.

4.3 Algorithmic details of HWY

Our proposed algorithm HWY efficiently finds the adjusted significance level δ to keep the FWER $\leq \alpha$ using randomly permuted datasets. The pseudo code is shown in Procedure 2. This algorithm is basically identical to that discussed in Section 4.1, with the addition of two parts. One is the use of the lookup table introduced in Section 4.2 (lines 10-14), and the other is the change in the initial q on each permuted dataset (line 7). The reason we can change the initial q is that we only need the αK-th value in K permuted datasets, and if the minimum P-value on a permuted dataset is larger than the αK-th value, we do not need to know it precisely. Furthermore, a smaller q can result in higher pruning efficiency. Therefore, we start q as the αK-th value in Q.

The adjusted significance level is computed at line 19. After calculating real P-values on the original GWAS dataset, SNPs with P-values less than δ are regarded as statistically significantly associated with the target trait.

4.4 Parallelization of HWY

Because recent computer servers have multi-core CPUs and GPGPUs, acceleration with hardware is useful. Fortunately, the calculations in lines 5 to 17 in Procedure 2 are not dependent on k except when updating q; hence, these can be readily parallelized.

We implemented parallelization using a GPGPU called HWY-GPU. This involves two differences from HWY. First, Q is an array instead of a priority queue, because many updates on Q may occur once, and many threads may have to wait on this update to avoid simultaneous updates of Q. By using an array for Q, this problem is solved. However, this change causes a problem with the initial value of q at line 7. Therefore, as the second change, HWY-GPU does not use line 7 in Procedure 2.

When lines 5 to 17 are parallelized, we generate random SNP-trait associations on a CPU at line 5, then the real permuted dataset is generated in local memory on each core in a GPGPU.

5. PERFORMANCE EVALUATION

We applied HWY to two GWAS datasets, one from an Alzheimer study and one from the HapMap project, to demonstrate its efficiency and utility. First, we examined the computational efficiency of HWY with variously sized datasets. In addition to HWY, we implemented the original WY (WY) and WY using a lookup table (WY-LT) to individually examine the efficiency of pruning in Section 4.1 and the lookup table for maintaining the calculated P-values in Section 4.2. HWY was also compared to the multiple testing procedures proposed by Ge *et al.* (Ge) [6] and to the performance of PLINK with maxT and the adaptive permutation procedure (PLINK (maxT) and PLINK (adaptive)) [12].

5.1 Experimental datasets

Table 2 summarizes the two datasets used in our experiments. The Alzheimer dataset [18] originated from the GWAS of 380,157 SNPs with 176 Alzheimer patients and 188 controls. The HapMap dataset [16] originated from 1,311,113 SNPs of 82 Han Chinese in Beijing, China (CHB) individuals and 1,272,736 SNPs of 82 Japanese in Tokyo, Japan (JPT) individuals in HapMap phase 3. We analyzed a total of 1,340,770 SNPs observed in CHB or JPT individuals.

Table 2: Statistics of two real GWAS datasets.

	Alzheimer	HapMap
# of SNPs (M)	380,157	1,340,770
# of alleles (N)	728	328
# of cases (N_{case})	352	164
Avg. of x_i	163.6	72.7

We treated individual alleles identically to Fisher's exact test of PLINK [12], which is the most widely used software for GWAS; that is, paternal and maternal alleles were independently considered. Therefore, for each SNP, we observed N alleles from $N/2$ individuals and performed a statistical test with N observations on each SNP.

The variously sized datasets were generated from Alzheimer and HapMap datasets by random selection. All analyses were repeated five times to calculate the means and variances of the calculation times. To evaluate the time performances of the algorithms, we generated a null distribution from $K = 1{,}000$ permuted datasets. The one-sided Fisher's exact test was used for the statistical measure, and FWER was controlled to be under $\alpha = 0.05$. The experiments for the algorithms were run on a machine with two Intel Xeon E5-2680v2 processors at 2.8 GHz with 64 GB of RAM and an nVidia Tesla K20 GPGPU with 2,496 cores running Red Hat Enterprise Linux 6. All programs were written in C. Intel C Compiler version 14.0.0 was used for all experiments, and CUDA version 5.5 was used for the experiments on a GPGPU.

5.2 Increasing number of permutations

We evaluated the performance with an increasing number of permutations K (Figs. 3(a) and (b)). Fig. 3(a) shows the running times for the Alzheimer dataset using only a single core of a CPU. Comparison of the times between WY and WY-LT confirmed that the lookup table for storing P-values dramatically reduced the running time. For example, WY-LT was approximately 52 times faster than WY at $K = 100$. While the running times of both algorithms grew linearly with the number of permutations, the increasing speed for WY-LT was much slower than that for WY. Moreover, WY-LT reduced the running time by 58.1% from Ge. These results indicate the lookup table works efficiently and accelerates the minimum P-value calculation in each permutation. A similar tendency was observed in the HapMap dataset (Fig. 3(b)), where WY-LT was orders of magnitude faster than WY.

You might think if the lookup table works efficiently, the calculation time in each permutation will gradually become faster with the increase in permutations because the values in the lookup table are used more frequently than in earlier permutations. However, our results show that the times increase nearly linearly with the number of permutations. This is because most of the values in the lookup table were calculated in the first permutation. Figs. 3(c) and (d) show the magnified results regarding the running times in first 5 permutations. The figures also include the frequencies of P-value calculations (the frequencies that the lookup table

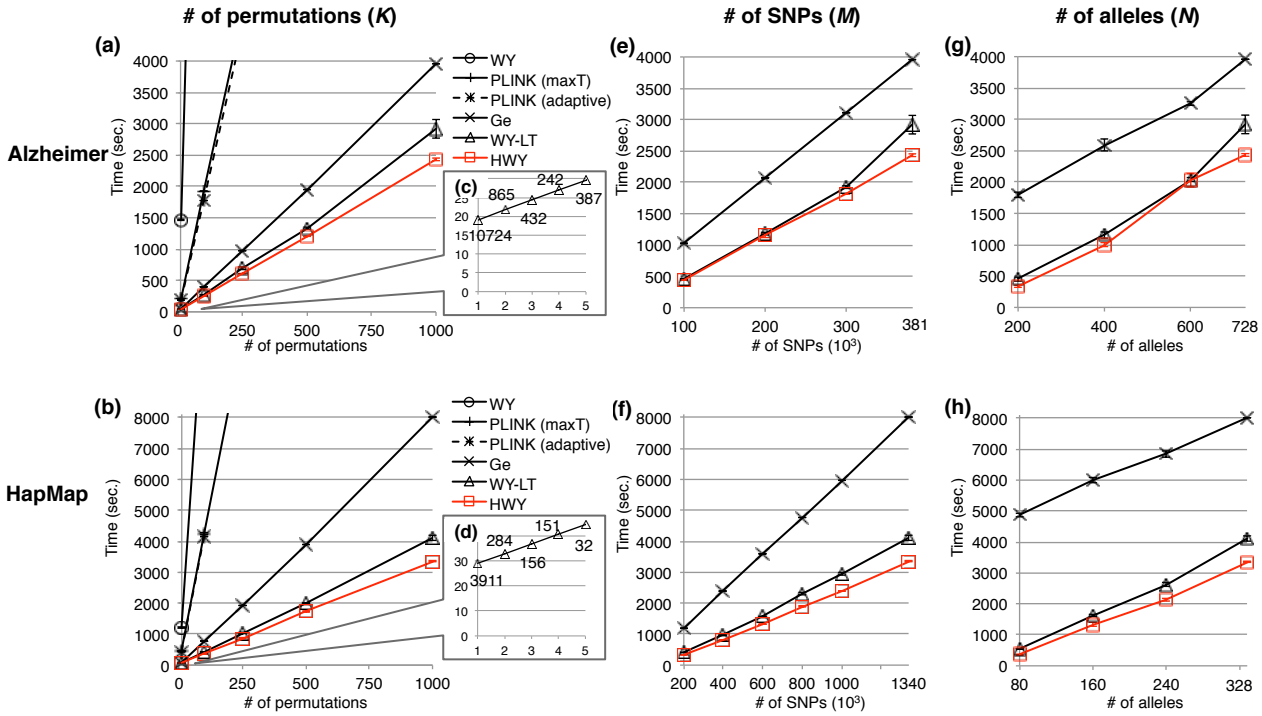

Figure 3: Calculation times for the Alzheimer and HapMap datasets on a CPU. Each plot reports average running time and standard error. (a) (b) Running time with increasing number of permutations. (c) (d) Enlarged figure of (a) and (b) in the first five permutations. Each line indicates running time of WY-LT, and values close to markers are numbers of SNPs whose P-values were calculated because they were not maintained in the lookup table on each permutation. (e) (f) Running time with increasing number of SNPs. The running times of WY and PLINKs were omitted from both figures because they required at least ten hours (36,000 seconds) performed on $M = 100 \times 10^3$ and 200×10^3 SNPs. (g) (h) Running time with increasing number of alleles.

was not used). When $K = 1$ in Fig. 3(c), P-values were calculated 10,724 times among 380,157 SNPs, and the other P-values had already been calculated previously. This suggests that even in the first permutation, the lookup table works very effectively. When $K = 2$, the frequency of P-value calculations was 865, which is substantially decreased from the first permutation. These observations indicate that almost all P-values were no longer calculated but were found in the lookup table. When using the lookup table, the running time after the second permutation is occupied deriving the value from the lookup table. Therefore, the slope of running time to the number of permutations did not gradually decrease, but grew linearly. Fig. 3(d) shows the amount of calculated P-values in the HapMap dataset. Compared with the Alzheimer results, the number of the calculations was smaller, which may come from the fact that both x_i and n_i were smaller than the Alzheimer values, and hence the unique pairs of x_i and n_i were limited.

The number of pairs of keys of the lookup table is $x_i \cdot n_i/2$. In Alzheimer dataset, $364 \cdot 176/2 = 32{,}032$ patterns were possible keys, while Fig. 3(c) suggests that a small number of keys appeared frequently. Fig. 4 shows the frequencies of the keys used in the calculation. We can see the single blue line in this figure, which means that a limited numbers of the keys were extremely frequently used. This figure, therefore, indicates that the lookup table worked extremely well to accelerate WY.

Figs. 3(a) and (b) also show that the pruning technique introduced in Section 4.1 was effective because the HWY was faster than WY-LT. In Alzheimer and HapMap datasets, HWY reduced the running time by at most 16.6% and 18.5% from WY-LT, respectively. While the lookup table was quite useful in reducing the running time, WY-LT required P-values for all SNPs in each permutation. Because the pruning technique reduced the number of the P-value calculations, HWY successfully further accelerated the computation.

5.3 Increasing number of SNPs and alleles

We investigated the performance when increasing the number of SNPs (Figs. 3(e) and (f)). In these experiments, we excluded WY, PLINK (maxT) and PLINK (adaptive) because they required more than ten hours when $M = 100 \times 10^3$ or $M = 200 \times 10^3$, while HWY, WY-LT and Ge finished in three hours even when using whole datasets.

Figs. 3(e) and (f) show that HWY was the fastest among Ge, WY-LT and HWY over any M. The computation times linearly increased with the number of SNPs M, because the number of P-values to be investigated increases linearly according to the number of SNPs. The difference between HWY and WY-LT indicates that the pruning of HWY was effective when the number of SNPs varied. HWY was faster

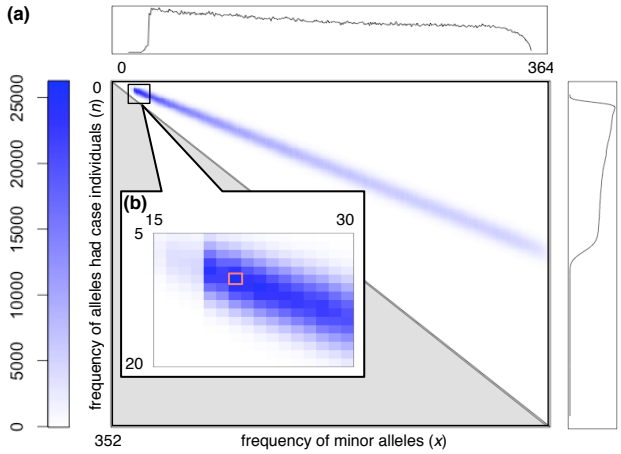

Figure 4: Heatmap illustrating frequencies of key pairs (x_i and n_i) in the lookup table for analyzing the Alzheimer dataset with $K = 100$. (a) In the middle figure, x- and y-axis show x_i and n_i, respectively, and the color illustrates frequency. The gray region (the bottom half of the table) is not used because $x_i \geq n_i$. The histograms on upper and right sides show the frequencies of x_i and n_i, respectively. (b) Enlarged view around the most frequent pair ($x_i = 21$ and $n_i = 10$), indicated with the red square. This pair was derived 26,223 times from the lookup table.

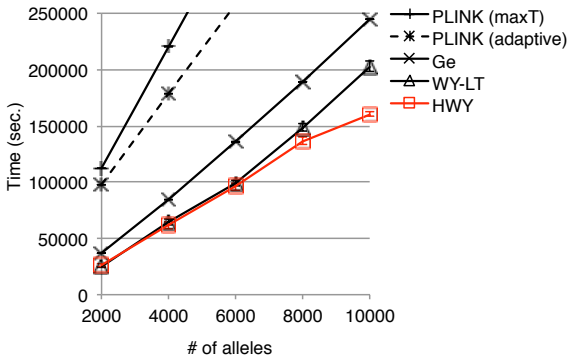

Figure 5: Running time on simulated datasets containing 10^6 SNPs and more than 2000 alleles. The running time of WY was omitted because it required at least five days performed on $N = 2000$ alleles.

than Ge in both datasets. Especially, with the HapMap dataset, HWY was 2.4-fold faster than Ge.

Figs. 3(g) and (h) indicate that HWY was the fastest among Ge, WY-LT and HWY over any N. The calculation time increases were not linear but faster than quadratic to the number of alleles. We computed Equation (1) to perform Fisher's exact test, which requires computational time that increases exponentially in the worst case according to the number of alleles. However, because the lookup table avoided the heavy calculations, the increase of calculation time was almost linear.

To compare performance for larger number of alleles, we generated simulated datasets that consist of up to $N =$ 10,000 using the SNP simulation routine of PLINK. Half of the alleles come from case samples and the others from control samples. All datasets contains 10^6 SNPs. We conducted six methods on $K =$ 10,000 and show the result in Fig. 5. The result of WY is not shown because it required at least five days for $N = 2000$ alleles. Similar to Figs. 3(g) and (h), HWY was the fastest among Ge, WY-LT and HWY over any N, with almost linear increases in time required. Therefore, the pruning technique and the lookup table in HWY efficiently reduce the running time when the dataset contains a large number of alleles.

5.4 Acceleration by the use of a GPGPU

Figs. 6(a)-(c) compare the running time among HWY, WY with a lookup table using a GPGPU (WY-LT-GPU), and HWY using a GPGPU (HWY-GPU) on three datasets. Because the CPUs on the computer have 20 cores in total, we reported the time of HWY as the running time on a CPU divided by 20. We also implemented WY on a GPGPU, but as the calculation time was 100-fold longer than that of HWY-GPU, we did not put the computation time on the figures.

The figures indicate that both the pruning and the lookup table were efficient on a GPGPU. WY-LT-GPU was much faster than WY on a GPGPU for all datasets but required almost double the time of HWY-GPU. As the result, when the dataset consisted of 10^6 SNPs and 1,000 individuals, HWY-GPU could compute the adjusted significance level in approximate two hours, whereas HWY-LT-GPU required more than four hours (Fig. 6(c)).

Comparison between HWY-GPU and HWY indicates that the GPGPU is useful for the acceleration of HWY. Figs. 6 show that HWY is slower than HWY-GPU even when using 20 CPU cores. Because we could run hundreds to thousands of almost independent permutation calculations at a time with the GPGPU, the GPGPU parallelization worked effectively.

6. STATISTICAL RELEVANCE

The huge calculation time of WY has severely limited its application to GWAS; therefore it is unknown whether WY-like methods can find a reliable significance threshold on GWAS and new statistically significantly SNP-trait associations can be found by replacing Bonferroni-like corrections with WY or HWY. In this section, we evaluate the statistical relevance and the biological usefulness of HWY.

We analyzed both the Alzheimer and HapMap datasets with HWY for $\alpha = 0.05$. The one-sided Fisher's exact test was used to detect SNPs associated with case individuals for the Alzheimer dataset, while the two-sided Fisher's exact test was used to identify the SNPs associated with the difference between Japanese and Chinese for the HapMap dataset.

6.1 Stability of adjusted significance level

The important point is to determine K to achieve a stable adjusted significance level δ. We repeated HWY ten times with varying K from 100 to 10,000. The results are presented in Fig. 7, where the x- and y-axes indicate K and δ, respectively. The gray lines represent the results of each of the 10 repetitions. The mean and standard errors are plotted along the red line. For the Alzheimer dataset, δ was

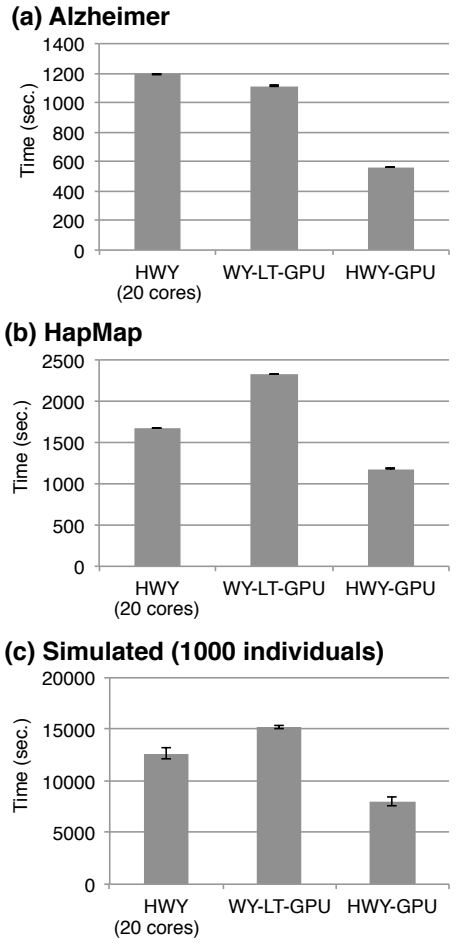

Figure 6: Performance comparison among HWY (20 cores), WY-LT-GPU and HWY-GPU on K = 10,000. We omitted the times of WY with a GPGPU because it required running times of more than one day.

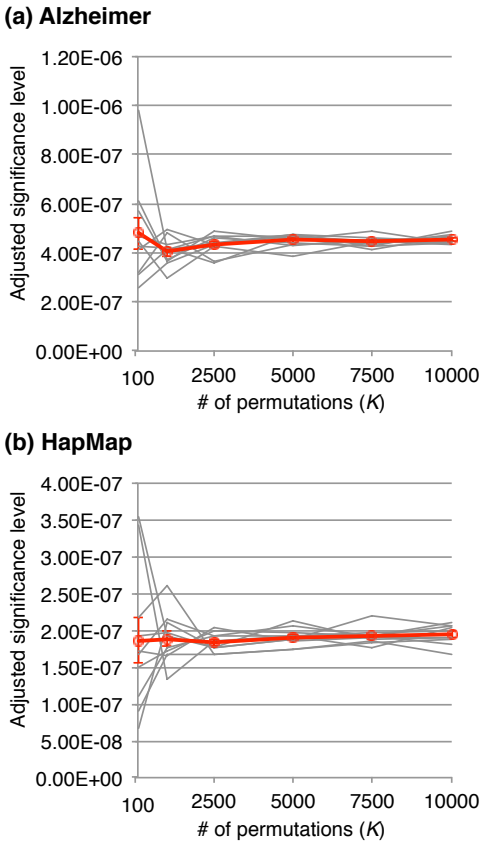

Figure 7: Convergence of δ. The gray and red lines represent the results of each repetition and the means of the ten repetitions, respectively. (a) Alzheimer dataset. The result shows that K = 10,000 would be enough because the standard error of δ is at most 5.82 $\times 10^{-9}$, and the value is two orders smaller than δ. (b) HapMap dataset. Similar to (a), δ was enough converged at K = 10,000 because the standard error of δ is at most 4.23 $\times 10^{-9}$.

almost identical when $K \geq$ 2,500, and the standard errors gradually decreased to 5.82×10^{-9} for K = 10,000, which is two orders of magnitude smaller than δ. Therefore, we concluded that the adjusted significance level converged enough for K = 10,000 to calculate a reliable significance level using HWY. In the following experiments, we used K = 10,000 in the Alzheimer dataset.

We also investigated the convergence of the adjusted significance level for the HapMap dataset. Fig. 7(b) shows that δ converged when $K \geq$ 2,500, and the average was more than 100 times larger than the standard errors for $K \geq$ 7,500. Because the adjusted significance level converged sufficiently, we also used K = 10,000 in the HapMap experiments.

6.2 Sensitivity

We report on the adjusted significance levels and the numbers of statistically significant SNPs in Table 3. We compared HWY with two other multiple testing procedures to control FWER, Bonferroni correction [4] and Holm procedure [8].

Table 3 shows that the δ computed by HWY was more than three times larger than those by Bonferroni and Holm for the Alzheimer dataset. While five statistically significant SNPs were identified by HWY, two of them were overlooked by the other procedures, suggesting that HWY can find SNPs overlooked by Bonferroni-like corrections.

In the HapMap dataset, the δ computed by HWY was approximately 4.5-fold higher than both Bonferroni and Holm, and HWY detected 429 SNPs more than both of the other procedures. While Bonferroni-like corrections often compute a conservative significance level, especially when SNPs are highly correlated, HWY can adjust δ with consideration of the correlations. With these experiments, the replacement of Bonferroni-like correction with HWY may detect more larger number of statistically associated SNPs under keeping FWER below α.

7. CONCLUSIONS

We propose an efficient algorithm called HWY to accelerate WY for GWAS. While providing results identical to

Table 3: Adjusted significance levels of HWY, Bonferroni and Holm.

	Alzheimer		HapMap	
	δ	# of SNPs detected	δ	# of SNPs detected
HWY	4.36×10^{-7}	5	1.67×10^{-7}	8,731
Bonferroni	1.32×10^{-7}	3	3.73×10^{-8}	8,302
Holm	1.32×10^{-7}	3	3.75×10^{-8}	8,302

In the Alzheimer dataset, the adjusted significance level δ computed by HWY was more than three times higher than those of Bonferroni and Holm corrections. In the HapMap dataset, 429 SNPs were overlooked by Bonferroni and Holm corrections, whereas HWY detected significance of them.

WY, HWY achieved at most 1,000 times faster than WY using three techniques: pruning the search space with the lower bound of P-values, introducing a lookup table to avoid duplicate P-value calculations, and GPGPU based parallel computing. Finally, HWY enumerated statistically significant SNPs associated with a trait within two hours for a dataset consisting of more than one million SNPs and one thousand individuals.

Replacing Bonferroni-like corrections with HWY increased the sensitivity of SNP detection from GWAS datasets. HWY achieved an adjusted significance level more than three times higher than Bonferroni and Holm corrections while all procedures controlled the FWER below α. With the use of HWY, many more new associations between SNPs and traits may be uncovered, not only from unpublished GWAS datasets but also published datasets, resulting in a better understanding of underlying biological mechanisms.

8. ACKNOWLEDGEMENTS

We thank Prof. Yutaka Akiyama, Assoc. Prof. Takashi Ishida, and Dr. Shuji Suzuki for insightful discussions and suggestions.

Supercomputing resources were provided by NIG (ROIS). AT is supported by JSPS Research Fellowships for Young Scientists. This work was partially supported by JSPS KAKENHI Grants 24240044 and 15H01717 to JS.

9. REFERENCES

[1] S. Atwell, Y. S. Huang, B. J. Vilhjálmsson, et al. Genome-wide association study of 107 phenotypes in Arabidopsis thaliana inbred lines. *Nature*, 465(7298):627–631, 2010.

[2] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Series B*, 57(1):289–300, 1995.

[3] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Ann Stat.*, 29(4):1165–1188, 2001.

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

[5] V. G. Cheung, R. S. Spielman, K. G. Ewens, et al. Mapping determinants of human gene expression by regional and genome-wide association. *Nature*, 437(7063):1365–1369, 2005.

[6] Y. Ge, S. Dudoit, and T. P. Speed. Resampling-based multiple testing for microarray data analysis. *Test*, 12(1):1–77, 2003.

[7] L. A. Hindorff, P. Sethupathy, H. A. Junkins, et al. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. *PNAS*, 106(23):9362–9367, 2009.

[8] S. Holm. A simple sequentially rejective multiple test procedure. *Scand J Stat.*, 6(2):65–70, 1979.

[9] X. Huang and B. Han. Natural variations and genome-wide association studies in crop plants. *Annu Rev Plant Biol*, 65:531–551, 2014.

[10] M. I. McCarthy, G. R. Abecasis, and L. R. o. Cardon. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. *Nature Reviews Genetics*, 9(5):356–369, 2008.

[11] N. Meinshausen, M. H. Maathuis, and P. Bühlmann. Asymptotic optimality of the WestfallâĂŞYoung permutation procedure for multiple testing under dependence. *Ann Stat.*, 39(6):3369–3391, 2011.

[12] S. Purcell, B. Neale, K. Todd-Brown, et al. PLINK: a tool set for whole-genome association and population-based linkage analyses. *American journal of human genetics*, 81(3):559–75, 2007.

[13] G. D. Ruxton and M. Neuhäuser. Good practice in testing for an association in contingency tables. *Behavioral Ecology and Sociobiology*, 64(9):1505–1513, 2010.

[14] A. Terada, M. Okada-Hatakeyama, K. Tsuda, et al. Statistical significance of combinatorial regulations. *Proc Natl Acad Sci USA.*, 110(32):12996–13001, 2013.

[15] A. Terada, K. Tsuda, and J. Sese. Fast Westfall-Young Permutation Procedure for Combinatorial Regulation Discovery. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2013.

[16] The International HapMap Consortium. A haplotype map of the human genome. *Nature*, 437(7063):1299–320, 2005.

[17] Z. Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *J Am Stat Assoc.*, 62(318):626–633, 1967.

[18] J. A. Webster, J. R. Gibbs, J. Clarke, et al. Genetic control of human brain transcript expression in Alzheimer disease. *American journal of human genetics*, 84(4):445–58, 2009.

[19] D. Welter, J. MacArthur, J. Morales, et al. The NHGRI GWAS Catalog, a curated resource of SNP-trait associations. *Nucleic acids research*, 42(Database issue):D1001–6, 2014.

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

[21] J. Winkelmann, B. Schormair, P. Lichtner, et al. Genome-wide association study of restless legs syndrome identifies common variants in three genomic regions. *Nat Genet.*, 39(8):1000–1006, 2007.

[22] G. Yang, W. Jiang, Q. Yang, et al. PBOOST: A GPU based tool for parallel permutation tests in genome-wide association studies. *Bioinformatics*, 2014.

[23] X. Zhang, F. Zou, and W. Wang. FastChi: an efficient algorithm for analyzing gene-gene interactions. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 528–39, 2009.