Faster in Time and Better in Randomness Algorithms for Matching Subjects with Multiple Controls

Hung-Jui Chang
Department of Applied Mathematics
Chung Yuan Christian University
Taoyung, Taiwan
Email: hjc@cycu.edu.tw

Yu-Hsuan Hsu
Google
Taipei, Taiwan
Email: poloo5582@gmail.com

Chih-Wen Hsueh
Department of Computer Science
and Information Engineering
National Taiwan University
Taipei, Taiwan
Email: cwhsueh@csie.ntu.edu.tw

Mei-Lien Pan, Hsiao-Mei Tsao, Da-Wei Wang, Tsan-sheng Hsu*
Institute of Information Science, Academia Sinica
Taipei, Taiwan
Email: {mlpan66, hmtsao, wdw, tshsu}@iis.sinica.edu.tw
*Corresponding Author

Abstract—**In the era of learning healthcare systems and big data, observational studies play a vital role in discovering hidden (causal) associations within a dataset. To reduce bias in these observational studies, a matching step usually is adopted to randomly match each case subject with one or more control candidates. A high-quality matching algorithm, RandFlow, is proposed and compared with the commonly used – Simple Match, Matchit and Optmatch algorithms. The execution time, the memory usage, the successful matching rate, the statistical variation of relative risk, and the randomness computed employing the different algorithms are compared. The execution time of RandFlow was at least 30 times faster than commonly used methods, with at least a 66% reduction in memory usage. The variation of relative risk computed by RandFlow usually was smaller than by Simple Match. Simple Match had varying relative entropy, ranging from 0.2 to 0.95, while RandFlow almost uniformly had relative entropy close to 1. RanfFlow could find a matching so long as the maximum matching ratio was not reached. For obtaining more reliable study results, a two-phase matching is proposed. The first phase is to identify the maximum matching ratio, then is followed by matching multiple times and taking an average.**

Keywords–matching; observational study; relative entropy.

I. Introduction

Observational studies are often used for investigating causal relationships [1]. Given two events α and β, researchers can analyze whether the occurrence probability of event β is affected by a previous event α. In the medical field, an event can be a diagnosis, a prescription or a treatment. To reduce bias, several approaches have been applied, one of them being matching [2]. Hence, an observational study process begins by identifying the study group G_α (those with α), matching to the control candidates $G_{\neq\alpha}$ (those without α), and then performing statistical analysis to draw a conclusion. For example, Relative Risk (RR) is used to estimate the relative risk of having β with and without the previous occurrence of α. For example, in Table I, there are $a+b$ individuals with the event α, and a of them also with the event β. The conditional probability, R_1, which denotes the probability of having β under the condition of with the event α is therefore $a/(a+b)$. Also, there are $c+d$ individuals without the event α, and c of them with the event β. The conditional probability, R_2, which denotes the probability of having β under the condition of without α is therefore $c/(c+d)$. The RR value is defined as $RR = R_1/R_2$. RR values greater than, less than, or equal to 1, respectively, indicate positive, negative or no relationships, respectively. Other statistics, such as Odds Ratio (OR), may be used instead of RR depending on the study design.

TABLE I. EXAMPLE OF STUDY GROUP AND ITS MATCHED CONTROL GROUP

	α	$\neg\alpha$
β	a	c
$\neg\beta$	b	d
Sum	$a+b$	$c+d$

Matching is a critical step in the analysis of observational study. Generally, a matching algorithm randomly permutes the order of the input of the study case s, with the control candidate c, and then checks whether the input s-c pair can be matched, and finally matches s with K-fold eligible controls, one by one. The constant K is called the matching ratio. Some matching methods assign a propensity score to each pair [3] and return a matching with the best total score. However, if the distribution of cases is skewed, the study case may not be matchable to the required munber of controls, and so would be dropped to avoid incurring further bias. Therefore, the output matching needs to satisfy some quality criteria such as randomness and successful matching rate. In a good quality matching algorithm, a control candidate has roughly an equal chance of being matched with any of the matchable study cases. It is desirable as well to retain as many successful matchings as possible.

There are some commonly used matching methods: for example, Simple Match [4], Matchit [5], [6], and Optmatch [7]. The first is based on a simple greedy approach using SAS and there is no proof in [4] of it being able to deliver a matching in reasonable time, and the latter two are variations of the well-known max flow algorithm [8] having a performance guarantee, but with no consideration of randomness. If the matching is only performed once along with a small matching ratio, the result may not be stable in the sense that there is the possibility that different matchings may yield fluctuating statistics such as RR or OR. To obtain a reliable result, it is better to match multiple times and take an average of all the outcomes. However, it is not practical to do repeated matching due to the heavy time consumption. Moreover in practice, the determining matching ratio is also a cloudy issue. During the past few decades, the suggested method in case-control study has been to match each subject with four or five controls [9]. It was reported that ”beyond a ratio of about 4/1, little power improvement results from increasing the number of controls” [10]. However, a matching ratio of 10 or 15 has also appeared in some studies [11], [12]. In Hennessy's study, it was indicated that a higher matching ratio may be needed when the disease prevalence is low [13], which implies that the matching ratio should be data dependent [14]. To date, there has been little study investigating the issue of finding a good matching ratio.

Previous researches have focused on the impact of the matching ratio [14], and whether to use a matching or not [15]. But how to determine the matching ratio is less discussed. To resolve the above problems, we proposed a high-quality matching algorithm called *RandFlow*, which adopts the idea from maximum flow in graph theory. In RandFlow, we have added some vital functions towards raising the randomness and matching efficiency. Furthermore, we leveraged the high efficiency of RandFlow to determine the optimal matching ratio. By using RandFlow, the maximum matching ratio of each data set is calculated, and the range of the suitable matching ratio is also determined. The researcher can choose a preferred matching ratio according to the suggested range.

The remains of this paper are organized as follows: In Section II, we describe our matching algorithm, the data source used in this study, and the factor compared between different matching methods. In Section III, we show the experiment results of RandFlow and comparison between RandFlow and the original method. In Section IV, we discuss the comparison results. Finally, in Section V, we conclude this paper.

II. Methods

The approach of our method is to formulate our problem in the well-known framework of flows in networks [8]. Hence our methods come with performance and correctness guarantees. In our study, we used Taiwan's National Health Insurance Research Database (NHIRD) [16] as the data source and examined the validity of RandFlow by three causal relations reported in the published papers [17], [18]. We then compared RandFlow with the above matching methods with regard to execution time and memory, successful matching rates, RR values and quality of randomness.

A. RandFlow Algorithm

We illustrate the idea of the original matching problem in Figure 1(a). The study cases are listed on the left-hand side, and the control candidates are listed on the right-hand side. The dashed line indicates the potential matching between a study case and a control candidate. We transform the matching problem in Figure 1(a) to the well-known max flow problem [8] in Figure 1(b) by adding one source node, one sink node, outgoing edges from the source to all study nodes, and incoming edges from the control nodes to the sink. There is a capacity constraint set on each edges where the outgoing edges of the source is K, and the rest of the edges are 1. Each edge (x, y) in E is associated a non-negative number called a weight $w(x, y)$ and, for each pair of *intermediate nodes*, namely not sources or sinks, x and y, the equality $w(x, y) = w(y, x)$ always holds. Thus w is function on E.

In a max flow problem, we assign maximum integer weights w, not exceeding the pre-assigned capacity, to the edges so that for each vertex other than the source and sink, the sum of weights on its incoming edges equals the sum of weights on its outgoing edges.

We consider a subset f of w, which we shall call a *flow*. We shall require three things for this flow to be a *legal flow*: (1) the weight of each edge is not exceeding the pre-assigned capacity, (2) for each vertex other than the source and sink, the sum of weights on its incoming edges equals the sum of weights on its outgoing edges, (3) the sum of weights on the outgoing edges of the source equals to the sum of weights on the incoming edges of the sink. A legal flow f in the max flow problem is a possible matching in the original matching problem. Those control candidates in the chosen f, whose incoming edges have nonzero weight, are the matched control cases in the original matching problem. Therefore, we can calculate the corresponding RR values of a legal flow f.

A study case S_i is matched with those control candidates C_j so that the weight of the edge from S_i to C_j is 1. The outcome is called a *max flow*. We further require that each study case has the same sum of incoming edge weights, which is called the *maximum matching ratio*, denoted by r. Thus each study case is matched with exactly r candidates, and each candidate is matched at most once. Since a max flow is to be found, r is should be as large as possible. Note that the value of r is data dependent. Each data set has its own maximum matching ratio. In addition to whether a matching of a specified size can be found efficiently or not, we also are concerned whether the resulting matching is random or not, i.e., whether each candidate has an equal chance of being selected by any case subject. Without considering constraints incurred from competitions between case subjects, we use the well-known *entropy* [19] E of the ideal distribution among all possible candidates that can be matched to a case subject. Then we measure the entropy E' of the actual distribution of candidates being found by applying the matching repeatedly say 1000 times. To quantify the quality of randomness in the matching obtained, we define the *relative entropy* to be $\frac{E'}{E}$. Some of our results infra show that finding matchings with ratio exactly r provides good randomness properties.

There are known algorithms for finding such a max flow in $O(|E||F|)$ time, where E is the set of edges and F, called *flow*, is the set of edges with weight 1 between the study cases and candidates. The value of $|F|$ is the number of edges inside. In the general case, there are certain known algorithms, which we shall call special augmentation algorithms, for augmenting given legal flows step by step, eventually leading to legal flows taking on the maximal flow value for such a flow.

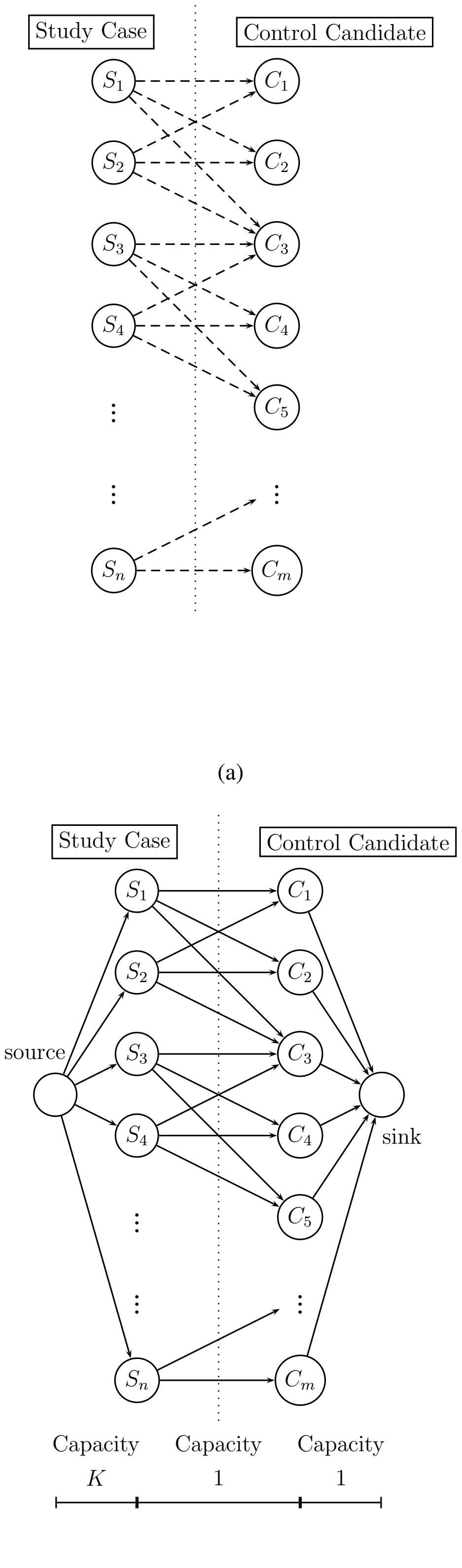
To each special augmentation algorithm corresponds an integer n. When one applies such an algorithm to a given legal flow f, which is not a maximum flow, the algorithm identifies an edge (s, c) between intermediate nodes and a set X of n edges (x, y) in the domain of f such that the nodes x are all distinct and such that each edge (a, x) is in the domain of f. Let D denote the union of three sets: the domain of f, the set $\{(s.c), (c, t)\}$ and the set of all (y, x) such that x is in X. The special augmentation algorithm acts in such a way that the restriction f' of w to D is again a legal flow. It follows from the weight property $w(x, y) = w(y, x)$ that the flow value of f' exceeds the flow value of f by the weight $w(s, c)$.

For our application here we set all weights equal to 1, and so each application of a special augmentation algorithm increases the flow value by 1.

We extend the original algorithm by finding a random augmenting flow, instead of a fixed one using a randomized version of Depth First Search (RDFS). In addition, we use a merging technique so that given two candidates C_i and C_j, they are merged if they have incoming edges from the same set of study cases. Furthermore, we randomly shuffle the ordering of study cases from the input to obtain better randomness quality. Our revised algorithm runs faster and uses less memory in practice than the original one. The technical details can be found in our technique report [20].

B. Data source

The NHIRD is a nationwide database extracted from the claim data of the National Health Insurance (NHI) program in Taiwan for research purposes. In recent years, NHIRD has been widely used to identify potential causal relationships. Our study also used NHIRD as the data source and was reviewed by the Institutional Review Board of Academia Sinica, Taiwan (approval number: AS-IRB-BM-16043). As a benchmark, we randomly generated data sets according to NHIRD data distribution. We denote P_α as the probability that an α event happens, and $P_{\beta|\neq\alpha} = P_\beta$ as the probability that a β event happens without an α event happens. And the probability that a β event happens after an α event happens is denoted as $P_{\beta|\alpha} = RR \times P_\beta$. In the following, P_α and P_β is selected from {0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50}, and RR is selected from {1/2, 2/3, 4/5, 1, 5/4, 3/2, 2}. There are 10 possible values of P_α, 10 possible values of P_β and 7 possible values of RR. The total number of test cases is 700. For each test case, we randomly draw 1,000 person from NHIRD. According to P_α and P_β, we randomly pick α and β. We executed the RandFlow Algorithm 1,000 times and calculate the average and the standard deviation of R_1, R_2, and RR, respectively. Additionally, we selected three distinct causal relations from two published papers [17], [18]. One paper investigated the bidirectional relationship between obstructive sleep apnea (OSA) and depression [17]. The study showed the positive relationship that patients with OSA have increased risk of occurring depression, and vice versa. The other paper examined whether previous statin use in patients with stroke affects the subsequent risk of dementia [18]. The study found



(a)

(b)

Figure 1. An example of transforming the matching problem into a flow problem.

a negative relationship in such a way that statin use in patients with stroke decreases the risk of dementia. In our study, we define an event pair to be one for which the former event affects the occurrence of the following event. Hence, the relationship between depression and subsequent OSA is denoted as Event Pair I, and the reverse is Event Pair II. The relationship between statin use in patients with stroke and subsequent dementia is Event Pair III.

C. Comparisons between matching methods

We identified the study cases of the selected event pairs according to the case criteria of [17], [18] and matched them to the control candidates by the following matching algorithms: Simple Match [4] in SAS, Matchit [5], [6] in R, Optmatch [7] in R, and RandFlow in C. Because Simple Match is the most popular matching program used in epidemiology, we implemented Simple Match using C, which we denote by Simple (C) in the sequel, in order to compare with RandFlow from a common basis. Because of software limitations and language nature, programs and packages in SAS and R run much slower and use more memory than those in C. As for Simple (C) and RandFlow, the time complexity of the former is $O(nm)$ and that of the latter is $O(n^2m)$, where n is the number of nodes and m is the number of edges in the graph. Therefore, it is expected that Simple (C) will run faster than RandFlow.

In the original studies, Event Pairs I and II were performed by the exact match method. Among these two event pairs, each study case was matched with five controls. Regarding Event Pair III, each study case was instead matched with one control by propensity score match [21]. In our study, all experiments were done by exact match. We used the ratio of control candidates to study cases on order to conjecture the maximum matching ratio. The number of total edges provided an estimate of computing time and memory consumption.

We then compared the matching methods with regard to execution time and memory usage, successful matching rates, RR values and quality of randomness. The *successful matching rate* is defined to be the percentage of matched study cases that are not dropped. We assessed the average execution time, the corresponding successful matching rates and the RR values with matching ratios from 1 to 30 (to 90 in the case of Event Pair II). To further understand the variation of RR values, we also examined the standard deviation of RR for R_1 and R_2 where R_1 and R_2, respectively, represent the risks of β occurring in the study group (G_α) and control group ($G_{\neq\alpha}$), respectively. The ratio of R_1/R_2 is RR. For quality of randomness, we calculated the relative entropy of the matched control candidates using three different matching ratios: 70%, 100% and 110% of the maximum matching ratio. The RR values and relative entropies were run 100 times, after which the average was taken. In our study, we used only Event Pair I as the benchmark to evaluate the execution time and memory usage. Because the programs implemented in C are more efficient and memory sparing, we just compared C implementations in terms of successful matching rates, RR value and quality of randomness. All the experiments were performed on a Ubuntu 14.04 system with an Intel(R) Core(TM) i7-3770 CPU 3.40 GHz, and 16 Gbytes RAM.

III. RESULTS

We present our experiment results in the follows.

TABLE II. THE STATISTIC RESULTS OF REAL RR VALUE AND THE ESTIMATED RR VALUE OF RANDFLOW.

Real RR	Estimated RR	Δ	Variance	STD
0.50 (1/2)	0.462	0.038	0.021	0.144
0.66 (2/3)	0.658	0.002	0.033	0.182
0.80 (4/5)	0.854	0.054	0.041	0.202
1.00 (1/1)	1.016	0.016	0.029	0.169
1.25 (5/4)	1.259	0.009	0.049	0.209
1.50 (3/2)	1.542	0.042	0.056	0.236
2.00 (2/1)	2.049	0.049	0.105	0.325

A. General result of the randomly sampled data

Figure 2 shows the result of the randomly generated data. The x-axis denotes the real RR value, and the y-axis denotes the estimated RR value, which is calculated by RandFlow. Each point in Figure 2 represents one data set. For each real RR value, there are 100 test data sets. The results show RandFlow can get an estimated RR value very close to the real RR value. The statistic results are summarized in Table II. The first and second column denotes the real RR value and the estimated RR value. The third to fifth column denotes the absolute error between the real and the estimated RR value, the variance of the estimated RR value and the standard deviation of the estimated RR value. The experiment results show the absolute error RandFlow Algorithm is less than 0.06 and the variance and standard deviation is only 0.10 and 0.33, respectively.

B. General information of the selected event pairs

Table III shows the general information of the selected event pairs from the original papers together with our results, including the number of controls/control candidates, the ratio of control candidates to study cases, and the maximum matching ratio.

Among these event pairs, the greatest number of study cases was found in Event Pair I. With such a larde number of study cases, there were a total of more than 149 million edges generated while matching using RandFlow. We speculated that the maximum matching ratio would be different among the event pairs since it turned out to be the ratios 11, 51 and zero, respectively, for Event Pair I, II and III, respectively. In addition, these event pairs covered both positive and negative relationships. As a result, we believed that they can serve as representatives for testing matching quality - Event Pair I especially for execution time and memory usage comparison.

C. Execution time and memory usage

The benchmark experiments used the Event-Pair I dataset to measure the performance. Table IV shows the comparison of average execution times and memory usage among the selected matching methods. We measure the amount of memory usage at a matching ratio equals 1. And we only shows two digits after decimal point for execution time. Because of the different time complexity, Simple (C) ran faster than RandFlow. Regarding memory usage, these two methods are comparable. However, the average execution times of Matchit increased roughly linearly with the increase of the required matching ratio (Figure 3).

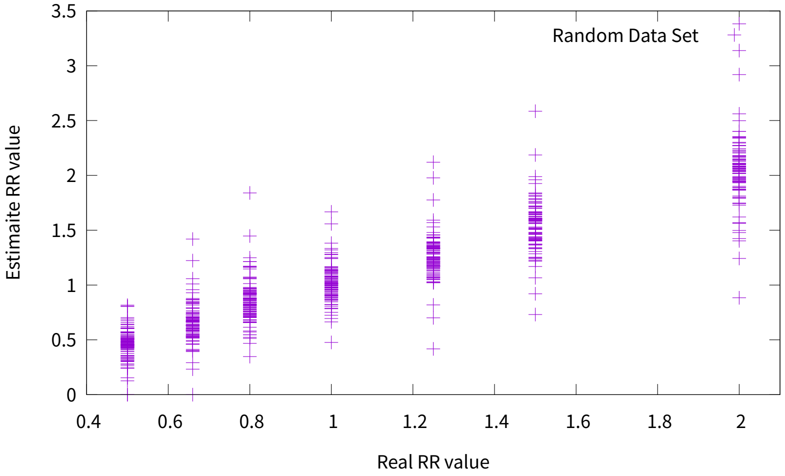


Figure 2. The distribution of real RR value and the estimated RR value of RandFlow.

TABLE III. GENERAL INFORMATION OF THE SELECTED EVENT PAIRS.

	Event Pair I	Event Pair II	Event Pair III
Original results			
No. study cases	27,073	6,427	5,527
No. control cases	135,365	32,135	5,527
Matching ratio	5	5	1
Our results			
No. control candidates	562,707	619,904	9,102
Control candidates/Study cases	≈21	≈97	≈2
Maximum matching ratio	11	51	0
Total edge	149,676,628	38,629,676	404,835

TABLE IV. AVERAGE EXECUTION TIMES (IN SECONDS[B]) AND MEMORY USAGE (IN GB) OF EACH METHOD.

Method	Lang.	Average execution times at different matching ratio(sec)							Memory usage (Gb)
		1	5	10	15	20	25	30	
Simple Match	SAS	1221.00	959.00	894.00	859.00	2350.00	1228.00	1078.00	1.039
Optmatch	R	8473.28	9132.34	9676.67	9407.05	9605.70	9037.06	8399.11	0.277
MatchIt	R	182.81	928.57	1956.58	2890.76	3891.28	4821.20	5924.35	0.232
Simple (C)	C	0.51	0.51	0.51	0.51	0.51	0.51	0.51	0.068
RandFlow	C	7.06	7.03	7.04	5.97	5.56	5.19	4.71	0.070

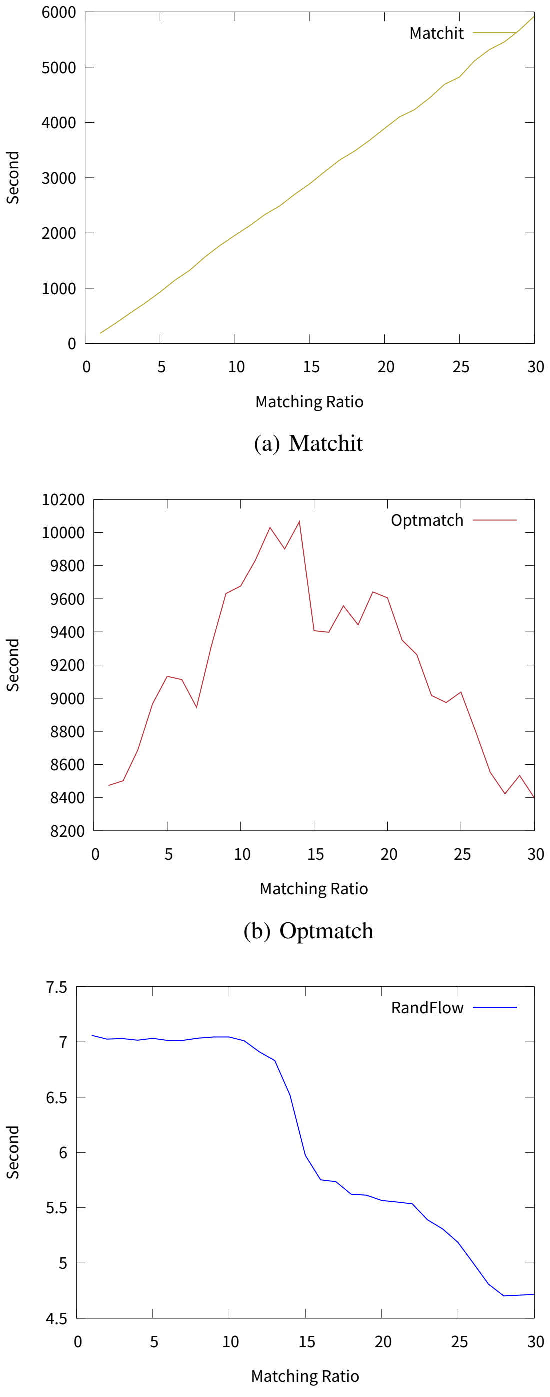
D. RR values and Successful matching rates

Flow-based matching methods in nature continue matching until they use up all the matchable control candidates. They are expected to have the same traits in terms of RR value variation and successful matching rate. Hence, in Section III, we only show the comparisons between Simple (C) and RandFlow.

Figure 4 shows the comparison results between RandFlow and Simple Match. Each column denotes the experiment results of different event pair, and each row denotes one specific comparison. The first row (Figure 4(a)-4(c)) shows the comparison of the RR value under different matching ratio. The second row (Figure 4(d)-4(f)) shows the comparison of the standard deviation of the RR value under different matching ratio. And the last row (Figure 4(g)-4(i)) shows the standard deviation of R1 and R2, which are performed by RandFlow under different matching ratio.

Overall, the average RR values of Simple (C) were greater than the values of RandFlow. In both methods, the average RR values were fairly stable while the matching ratio was small, and then gradually decreased when the matching ratio exceeded a certain value. In RandFlow, the decline occurred at the maximum matching ratio. By contrast, the decline of Simple (C) occurred earlier than that (Figure 4(a) and 4(b)). In the case of a negative relationship in Event Pair III, the average RR values increased rather than decreased (Figure 4(c)).

Generally speaking, the variation of RR values of Simple (C) was more unstable than that of RandFlow. In both methods, the variation of RR values steadily decreased and then turned up after a certain matching ratio. The least variation of the RR values of RandFlow occurred right at the maximum matching ratio. That of Simple (C) occurred before the maximum matching ratio (Figure 4(d)-4(f)).



(a) Matchit

(b) Optmatch

(c) RandFlow

Figure 3. Average execution times of the flow-based matching methods.

Since RR is calculated as R_1 divided by R_2, we examined the variation of R_1 and R_2 in RandFlow to survey further where the RR variation originates. When the matching ratio was less than the maximum matching ratio, no study cases were dropped; thus, the standard deviation of R_1 remained zero. On the other hand, the standard deviation of R_2 decreased with matching ratio, until it reached its maximum. As the size of the control group increased up to a certain number, the standard deviation of R_2 continued relatively small and steady. Beyond the maximum, the standard deviation of R_1 surged, resulting from the dropping of study cases (Figure 4(g)-4(i)).

Figure 5 shows the comparison of successful matching rates between Simple (C) and RandFlow. Because Simple (C) is based on a simple greed algorithm, its matching results may vary. We used both the minimal (Simple_min) and the maximal (Simple_max) results from the 100 trials for comparison. Whether or not we used the minimal or the maximal result from Simple (C), the successful matching rates dropped before reaching the maximum matching ratio, whereas those of RandFlow remained at 100%. At any fixed matching ratio, RandFlow had the highest successful matching rates. Although Simple (C) ran faster than RandFlow, when the execution time was fixed it failed to achieve the successful matching rate of RandFlow.

E. Quality of randomness

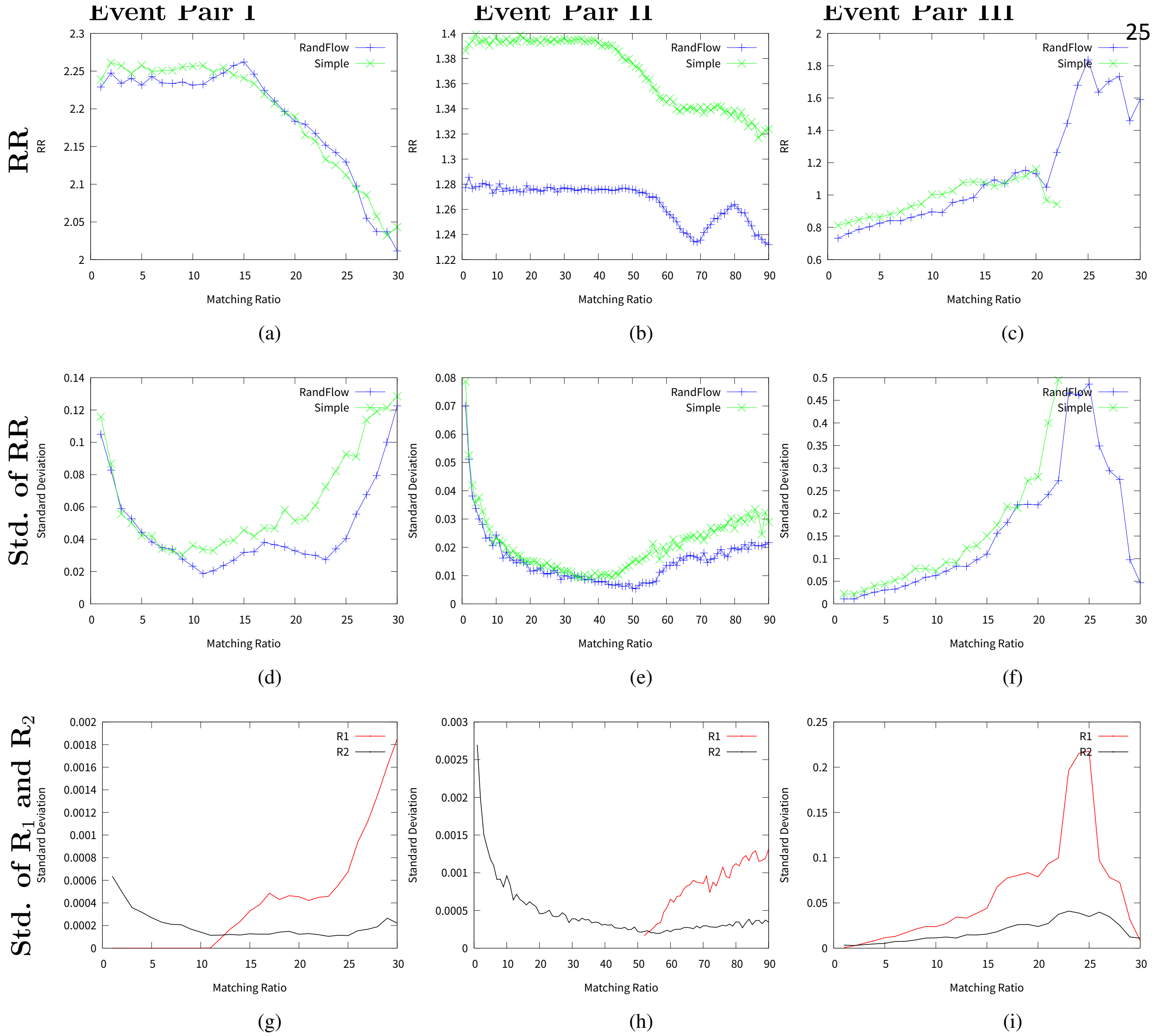
Optmatch and Matchit are both flow-based matching methods and do not randomly shuffle the input graph. In other words, their matched results remain unchangeable with no randomness at all. By contrast, we implemented RandFlow with inputting random graph and RDFS to enhance the quality of randomness. In this section, we present a comparison of quality of randomness between RandFlow and Simple (C).

Figure 6 shows the relative entropy of the chosen control candidates of the study cases in Event Pair I and Event pair II. The first column shows the result of Event Pair I, and the second column shows the result of Event Pair II. Each row denotes different matching ratio, the first row, the second row, and the last row shows the experiment result with matching ratio equals to 70%, 100%, and 110% of the maximum matching ratio, respectively.

Figure 6 shows that Randflow had better quality of randomness than Simple (C). The estimated relative entropy of RandFlow was about 1 and generally higher than that of Simple (C). In addition, RandFlow had consistently stable entropy for all matching ratio and study cases. Even when the ratio was set at 110% of the maximum matching ratio, the relative entropy of Randflow only decreased slightly. For those study cases having small sets of control candidates, the relative entropy of Randflow remained high. By contrast, the relative entropy of Simple (C) fluctuated widely as the matching ratio increased. For those study cases having less matchable control candidates, that of Simple (C) plunged.

IV. Discussion

In this study, we adopted maximum flow theory to develop a highly efficient and good-quality matching method, *RandFlow*, for matching subjects with multiple controls. This method can accomplish difficult matching tasks, such as matching 20 thousand study cases to each to 30 controls within a few seconds. Compared with the most popular matching method,



(a) (b) (c)

(d) (e) (f)

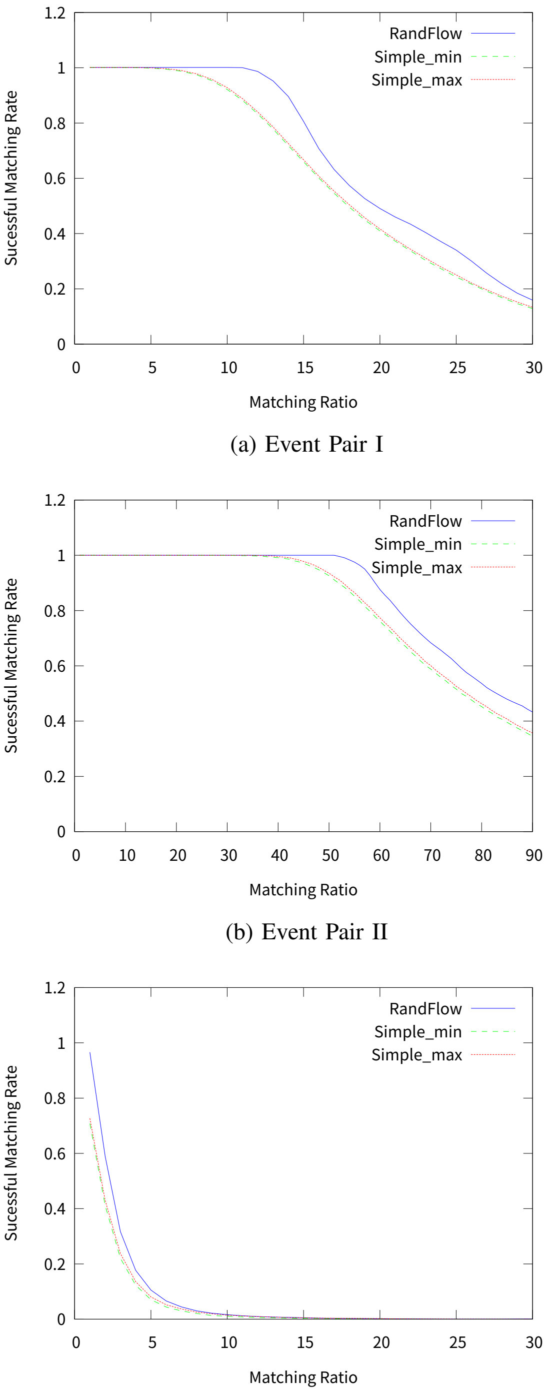
(g) (h) (i)

Figure 4. RR values and standard deviation of RR, R_1 and R_2 of Simple (C) and RandFlow.

Simple Match (on a SAS platform), it consumed merely 0.5-2.4% of execution time and 7% of memory usage. Among the flow-based matching methods, Optmatch and RandFlow are much alike in terms of execution time versus matching ratio. In both of these methods, the average execution times remained about the same until the required matching ratio exceeded the maximum matching ratio, after which they decreased because more and more case subjects were subsequently dropped.

Most importantly, RandFlow exhibited a good quality of randomness and rather than dropping study cases found a matching whenever such a matching existed. Matching is used to cause study cases and controls to have similar distributions across confounding variables. During the matching process, the controls are expected to be randomly selected from all the control candidates. Anything that may affect the sampling design, such as the dropping of cases, should be avoided. Our study used relative entropy to quantify randomness, and then verified that RandFlow had a good quality of randomness. The randomness of RandFlow does not vary with the chosen matching ratios since it is no more than the maximum ratio. With regards to successful matching rate, RandFlow outperformed simple greedy algorithms due to the nature of those algorithms. Overall, RandFlow surpassed those commonly used matching methods. It is not only a highly efficient matching method but also includes processes for avoiding undesirable biases during matching.

The matching ratio is data dependent and should be set differentially at the maximum matching ratio to obtain consistent results. During the past few decades, the suggested matching method of case-control study was to match each case subject with four or five controls. Previous studies had indicated that a higher matching ratio might be desired [10], [13], [14]. Beyond the previous studies, we tested three distinct data sets and performed matching multiple times at a range



(a) Event Pair I

(b) Event Pair II

(c) Event Pair III

Figure 5. Successful matching rate of Simple (C) and RandFlow. Simple_min and Simple_max represent the minimal and the maximal matching rate from the 100 trials run by Simple (C).

of matching ratios. In our experiments, we found that the maximum matching ratio varied with the input data set and the least variation of RR values always occurred when we set the matching ratio to be the maximum. This can be explained from the perspective of graph theory. If the matching ratio h requested is no more than the maximum matching ratio w, then we have many possible different matchings. From the law of large numbers, the RR value calculated from many instances must be stable and close to the real average case. If h is more than w, then we do not have many choices in selecting the pairings. The deviation of the computed RR tends to be higher than in the former case. Therefore, rather than using an empirical fixed matching ratio for any given study, we suggest matching at the maximum matching ratio multiple times and then taking an average for consistent results.

RandFlow being an exact matching has an inherent limitation: that of being unable to match some study cases with the required number of controls when the distribution of the confounding variables is skewed. In the extreme case, even a 1:1 match cannot be reached; thus, the RR values will be unstable at any matching ratio. In this circumstance, in order to obtains reliable results, other matching methods should be considered.

V. Conclusions

In this study, we developed a highly efficient matching method and demonstrated that it provides a good quality of randomness. From our experiments, we further concluded that the matching ratio is data dependent and should be set differentially at the maximum matching ratio. For future study, we suggest that matching should be done in two phases. The first phase is to identify the maximum matching ratio. Then, the second phase is to carry out matching using the maximum matching ratio several times and taking an average statistics. Using this two-phase matching, researchers can obtain stable results and accordingly draw unbiased study conclusions.

Acknowledgment

We thank Dr. Kelly McKennon for English editing and proofreading. This research was partially funded by MOST, Taiwan grants number 104-2221-E-001-021-MY3, and 107-2221-E-001-005, and Multidisciplinary Health Cloud Research Program: Technology Development and Application of Big Health Data, Academia Sinica, Taipei, Taiwan. It is based in part on data from the National Health Insurance Research Database provided by the Bureau of National Health Insurance, Department of Health and managed by National Health Research Institutes. The interpretation and conclusions contained herein do not represent those of the National Health Insurance Administration, Department of Health or National Health Research Institutes.

References

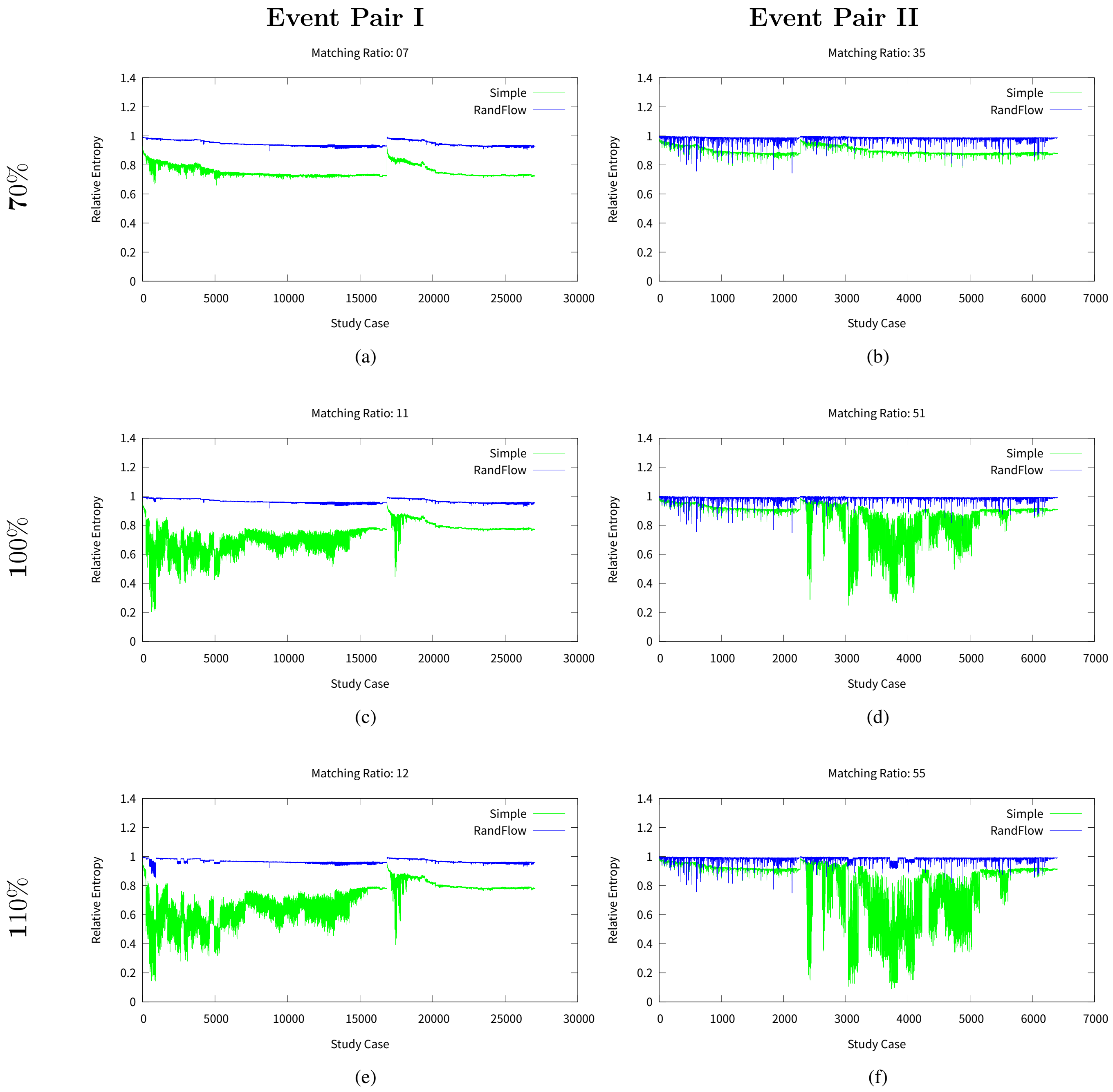
[1] H.-J. Chang, Y.-H. Hsu, C.-W. Hsueh, and T.-s. Hsu, “Efficient qualitative method for matching subjects with multiple controls,” in Proceedings of the Fifth International Conference on Big Data, Small Data, Linked Data and Open Data, ALLDATA 2019.

[2] E. A. Stuart, “Matching methods for causal inference: A review and a look forward,” Statistical science: a review journal of the Institute of Mathematical Statistics, vol. 25, no. 1, 2010, p. 1.

[3] P. R. Rosenbaum and D. B. Rubin, “Constructing a control group using multivariate matched sampling methods that incorporate the propensity score,” The American Statistician, vol. 39, no. 1, 1985, pp. 33–38.

[4] H. Kawabata, M. Tran, and P. Hines, “Using SAS® to match cases for case control studies,” in Proceeding of the Twenty-Ninth Annual SAS® Users Group International Conference, vol. 29, 2004, pp. 173–29.

[5] D. E. Ho, K. Imai, G. King, and E. A. Stuart, “Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference,” Political analysis, vol. 15, no. 3, 2007, pp. 199–236.



(a) (b)

(c) (d)

(e) (f)

Figure 6. Relative entropy of Simple (C) and RandFlow in Event Pair I, II. Relative entropy of Event Pair I and II was tested at 70%, 100% and 110% of the maximum matching ratio in 100 trials.

[6] ——, “Matchit: Nonparametric preprocessing for parametric causal inference,” Journal of Statistical Software, 2007.

[7] B. B. Hansen, “Full matching in an observational study of coaching for the SAT,” Journal of the American Statistical Association, vol. 99, no. 467, 2004, pp. 609–618.

[8] L. R. F. Jr and D. R. Fulkerson, “Maximal flow through a network,” Canadian journal of Mathematics, vol. 8, no. 3, 1956, pp. 399–404.

[9] S. Wacholder, D. T. Silverman, J. K. McLaughlin, and J. S. Mandel, “Selection of controls in case-control studies: Iii. design options,” American journal of epidemiology, vol. 135, no. 9, 1992, pp. 1042–1050.

[10] D. A. Grimes and K. F. Schulz, “Compared to what? finding controls for case-control studies,” The Lancet, vol. 365, no. 9468, 2005, pp. 1429–1433.

[11] M.-L. Pan, L.-R. Chen, H.-M. Tsao, and K.-H. Chen, “Relationship between polycystic ovarian syndrome and subsequent gestational diabetes mellitus: a nationwide population-based study,” PloS one, vol. 10, no. 10, 2015, p. e0140544.

[12] K.-J. Tien, C.-W. Chou, S.-Y. Lee, N.-C. Yeh, C.-Y. Yang, F.-C. Yen, J.-J. Wang, and S.-F. Weng, “Obstructive sleep apnea and the risk of atopic dermatitis: A population-based case control study,” PloS one, vol. 9, no. 2, 2014, p. e89656.

[13] S. Hennessy, W. B. Bilker, J. A. Berlin, and B. L. Strom, “Factors

influencing the optimal control-to-case ratio in matched case-control studies," American Journal of Epidemiology, vol. 149, no. 2, 1999, pp. 195–197.

[14] K. J. Rothman, S. Greenland, and T. L. Lash, Modern epidemiology. Lippincott Williams & Wilkins, 2008.

[15] T. Faresjö and Å. Faresjö, "To match or not to match in epidemiological studiesxsame outcome but less power," International journal of environmental research and public health, vol. 7, no. 1, 2010, pp. 325–332.

[16] M. o. H. National Health Insurance Administration and R. Welfare, Taiwan, "National health insurance research database, Taiwan." retrieved: December, 2019. [Online]. Available: http://nhird.nhri.org.tw/en/index.htm

[17] M.-L. Pan, H.-M. Tsao, C.-C. Hsu, K.-M. Wu, T.-s. Hsu, Y.-T. Wu, and G.-C. Hu, "Bidirectional association between obstructive sleep apnea and depression: A population-based longitudinal study," Medicine, vol. 95, no. 37, 2016, p. e4833.

[18] M.-L. Pan, C.-C. Hsu, Y.-M. Chen, H.-K. Yu, and G.-C. Hu, "Statin use and the risk of dementia in patients with stroke: A nationwide population-based cohort study," Journal of Stroke and Cerebrovascular Diseases, 2018.

[19] C. E. Shannon, "A mathematical theory of communication," ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, no. 1, 2001, pp. 3–55.

[20] H.-J. Chang, Y.-H. Hsu, C.-W. Hsueh, and T.-s. Hsu, "Efficient randomized algorithms for large-scaled exact matching with multiple controls: Implementation and applications," Institution of Information Science, Academia Sinica, Taiwan, Tech. Rep. TR-IIS-17-005, 2017.

[21] L. S. Parsons, "Reducing bias in a propensity score matched-pair sample using greedy matching techniques," The Twenty-Sixth Annual SAS Users Group International Conference, vol. 21426, 01 2001.