A NOVEL WEIGHTED RANK AGGREGATION ALGORITHM WITH APPLICATIONS IN GENE PRIORITIZATION

BY

FARDAD RAISALI

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Professor Olgica Milenkovic

ABSTRACT

We propose a new family of algorithms for bounding/approximating the optimal solution of rank aggregation problems based on weighted Kendall distances. The algorithms represent linear programming relaxations of integer programs that involve variables reflecting partial orders of *three or more candidates*. Our simulation results indicate that the linear programs give near-optimal performance for a number of important voting parameters, and outperform methods based on PageRank and Weighted Bipartite Matching. Finally, we illustrate the performance of the aggregation method on a set of test genes pertaining to the Bardet-Biedl syndrome, schizophrenia, and HIV and show that the combinatorial method matches or outperforms state-of-the art algorithms such as ToppGene.

To my parents, for their kindness and devotion and their endless support.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Prof. Olgica Milenkovic, for her excellent guidance, caring, sacrifices and helping me to get admitted at the University of Illinois. This work would not have been possible without her help and support. I would never have been able to finish my thesis without the guidance of my committee members, help from friends and support from my family. I would like to thank Prof. Vaidya, Prof. Varshney, Prof. Hajek and Prof. Kiyavash for guiding and supporting me while I was studying at the University of Illinois. Many thanks to my CSL friends, Farzad, Philip, Navid, Minji and Hossein. We had a great time being together. I would also like to thank my parents and sister for always supporting me and encouraging me with their best wishes.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 A NEW EQUIVALENT FORMULATION FOR ACYCLIC POLYTOPES 3
2.1 Background 3
2.2 Problem Reformulation 5

CHAPTER 3 WEIGHTED KENDALL RANK AGGREGATION: LINEAR PROGRAMMING APPROACH 9
3.1 Weighted Rank Aggregation 9
3.2 Quadratic Weight Functions 12
3.3 Simulations 14

CHAPTER 4 GENE PRIORITIZATION VIA WEIGHTED KENDALL RANK AGGREGATION 17
4.1 Aggregation Algorithm for Weak Orders 19
4.2 Former Prioritization Methods 21
4.3 Results for Disease Related Gene Identification 22

CHAPTER 5 CONCLUSION 26

REFERENCES 27

CHAPTER 1

INTRODUCTION

The problem of rank aggregation may be simply stated as follows: a set of voters or agents is presented with a list of candidates that have to be ranked according to some criteria. The aggregate ranking is chosen to best reflect the ordering provided by the voters. Due to the fact that large volume datasets in social science, search engines, and biology are ordinal data, frequently obtained from multiple sources and using different ranking functions, rank aggregation has found many applications in web metasearch engines, social sciences, spam control and other applications [9, 6].

One of the best known methods for rank aggregation is distance based aggregation, where the problem is cast as the computation of the median of a set of full rankings (permutations). The distance measure used for computing the median is the Kendall distance, which has also found many applications outside of social choice theory and computer science – for example, in rank modulation coding for flash memories [2]. The Kendall distance counts the number of pairwise disagreements between two permutations ([17], [16]), and can be computed efficiently. On the other hand, computing the aggregate ranking under the Kendall distance is known to be NP hard [3]. To overcome this computational bottleneck, a number of algorithms for approximate aggregation were put forward, including PageRank (PR), Weighted Bipartite Graph Matching (WBGM), and relaxed Integer Programming (IP) (in particular, linear programing (LP) methods) [9, 6].

PR methods for rank aggregation mimic the principles used for ranking webpages by Google, and they reduce to computing equilibrium probabilities of Markov chains. WBGM algorithms utilize the fact that the Kendall distance may be approximated up to a multiplicative constant by the ℓ_1 norm of permutations. The close connection between transitive tournaments and rankings was the basis for developing IP aggregation methods [21].

It is well known that the Kendall distance is not suitable for many practical

applications in which human subjects are involved, since the Kendall distance does not account for the fact that one inevitably pays more attention to the top of a list than to the remainder of the list. To overcome this problem, in our recent work we introduced the notion of a *weighted Kendall distance*, where higher weights are assigned to adjacent swaps at the top of a list. This ensures that in an aggregate, strong showings of candidates are emphasized compared to their weaker showings. In a companion paper [11], we presented extensions of the PR and WBGM methods for weighted Kendall distances. In what follows, we present a novel combinatorial optimization framework for computing the weighted Kendall aggregate with near-optimal performance. The algorithm is based on a new representation of permutations using partial orderings of three or more candidates as constraints. The method is of especially simple form when the weights are monotonically decreasing functions, and we therefore focus our attention to this case. Decreasing weights are suitable for capturing the importance of the top of a list, as they ensure that changes at the top are costlier than changes at the bottom.

The thesis is organized as follows. In Chapter 2, we present an alternative formulation for acyclic polytopes. In Chapter 3, we derive a closed form expression for linearly decreasing weighted Kendall distances, describe a corresponding IP aggregation method, and its LP relaxation. We also describe how this approach may be viewed as a special scoring procedure on rankings. Chapter 3 also contains extensions of the aforementioned results to the case of polynomially decreasing weight functions. In Chapter 4, we show the applications of the aforementioned aggregation algorithm for identifying relevant genes that might cause particular diseases. Conclusions based on the results of the thesis are discussed in Chapter 5.

CHAPTER 2

A NEW EQUIVALENT FORMULATION FOR ACYCLIC POLYTOPES

In this chapter, we present an alternative formulation for acyclic polytopes in n-dimensional space which was primarily formulated based on the relative relation between every pair of the objects. In this chapter, we propose a new polytope by applying new variables. The new variables store the relations between every set of triplets of the objects in the rankings. Later, we propose a new mapping of the aforementioned polytope to acyclic polytope.

2.1 Background

We consider the problem of rank aggregation involving n candidates and m voters. For simplicity, the set of candidates is chosen as $\{1, \ldots, n\}$, and denoted by $[n]$. A vote is a ranking of the candidates with no ties, and hence a permutation in $\mathbb{S}_n$, the symmetric group of order $n!$. We write each permutation $\sigma \in \mathbb{S}_n$ as $\sigma(1) \cdots \sigma(n)$, where $\sigma(i)$ represents the candidate with rank i. Note that $\sigma^{-1}(i)$ is the rank of candidate i, where σ^{-1} denotes the inverse of σ.

Suppose that the voters are numbered from 1 to m. Voters are allowed to cast the same vote, and the multiset of the voters' permutations (rankings) is denoted by Σ.

In *distance-based* rank aggregation, the goal is to find a ranking, called the *aggregate ranking*, that is as "close" as possible to all the votes simultaneously. Closeness is measured via a chosen distance function over $\mathbb{S}_n$. For a given distance d, the aggregate ranking π is formally evaluated according to

$$\pi^* = \arg\min_{\pi \in \mathbb{S}_n} \sum_{\sigma \in \Sigma} d(\pi, \sigma). \tag{2.1}$$

The most commonly used distance for the purpose of rank aggregation is

the Kendall distance, although other distances, such as the Cayley distance, Spearman's footrule, and Spearman's rank correlation have found relevant applications [8]. The Kendall distance between two permutations π and σ, denoted by $\mathrm{d}_K(\pi, \sigma)$, is the number of disagreements between π and σ, i.e., the number of ordered pairs (i, j) such that π ranks i higher than j, and σ ranks j higher than i. Formally, the distance may be defined as

$$\mathrm{d}_K(\pi, \sigma) = \left|\{(i, j) : \pi^{-1}(i) < \pi^{-1}(j), \sigma^{-1}(j) < \sigma^{-1}(i)\}\right|.$$

The solution of (2) for the Kendall distance is known as the *Kemeny aggregate*.

For $\sigma \in \mathbb{S}_n$, and $i, j \in [n]$, let

$$\sigma_{ij} = \begin{cases} 1, & \text{if } \sigma^{-1}(i) < \sigma^{-1}(j), \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Let P be the set of points $x = (x_{ij})$ satisfying

$$x_{ij} + x_{ji} = 1, \qquad \text{for distinct } i, j \in [n], \tag{2.3}$$

$$x_{ij} + x_{jk} + x_{ki} \leq 2, \qquad \text{for distinct } i, j, k \in [n], \tag{2.4}$$

$$x_{ij} \in \{0, 1\}, \qquad \text{for distinct } i, j \in [n], \tag{2.5}$$

$$x_{ii} = 0, \qquad \text{for } i \in [n]. \tag{2.6}$$

Note that there is a one-to-one correspondence between points $x \in P$ and permutations $\pi \in \mathbb{S}_n$, since $\pi^{-1}(i) < \pi^{-1}(j)$ if and only if $x_{ij} = 1$.

Using (2.2) and the definition of the Kendall distance, for each $x \in P$, one can write

$$\sum_{\sigma \in \Sigma} \mathrm{d}_K(x, \sigma) = \sum_{\sigma \in \Sigma} \sum_{i,j} x_{ij} \sigma_{ji} = \sum_{i,j} c_{ij} x_{ij}, \tag{2.7}$$

where $c_{ij} = \sum_{\sigma \in \Sigma} \sigma_{ji}$.

From (2.7) and the fact that the constraints (2.3)-(2.5) define a permutation, we find that a Kemeny aggregate is a solution of the integer programing

(IP) problem

$$\min_{x} \sum_{\sigma \in \Sigma} \sum_{i,j} c_{ij} x_{ij}$$

$$\text{subject to } x_{ij} \in P.$$

This formulation was independently proposed in [6], while relaxations of the IP method were shown to provide good approximations to the exact solution in [18].

In what follows, we describe how to generalize this simple idea for a broad class of *weighted* Kendall distance measures. Weighted Kendall distances were introduced by the authors in [11], and may be defined as follows. An adjacent transposition in a permutation is a swap of two elements ranked consecutively. Endow the set of adjacent transpositions A with a weight function $\rho : A \to \mathbb{R}^+$, i.e., assign to each adjacent transposition $(i\ i+1)$ a non-negative weight ρ_i.

The weighted Kendall distance under ρ, applied to two permutations π and σ, equals the smallest cost of any sequence of adjacent transpositions needed to transform π into σ. For example, let $\rho_1 = 2$ and $\rho_2 = 1$. The weighted Kendall distance between 132 and 213 equals $\rho_2 + \rho_1 = 3$, since one may first swap candidates 2 and 3 with weight ρ_2, and then swap candidates 2 and 1 with weight ρ_1.

In many applications, the top of a ranking is more important than the bottom, and thus it is reasonable to require that changes to the top of a ranking induce a larger distance than similar changes applied to the bottom of a ranking. Unfortunately, the classical Kendall distance does not take into account positional significance of candidates in a ranking, as any adjacent transposition contributes one point to the total distance. Weighted distances can overcome this problem, since they do not require uniform weights for adjacent swaps.

2.2 Problem Reformulation

In what follows, we describe an alternative formulation for P that will be useful in our subsequent analysis.

Let $\mathcal{T}_{a,b,c} = \{(abc), (acb), (bac), (bca), (cba), (cab)\}$. In addition, let Q be the set of points (x, w), with $x = (x_{ij})$, $i, j \in [n]$, and $w = (w_{ijk})$, with

$i, j, k \in [n]$, satisfying

$$\sum\nolimits_{(rst)\in\mathcal{T}_{i,j,k}} w_{rst} = 1, \qquad \text{for distinct } i, j, k \in [n], \tag{2.8}$$

$$w_{ijk} + w_{ikj} + w_{kij} = x_{ij}, \qquad \text{for distinct } i, j, k \in [n], \tag{2.9}$$

$$x_{ij},\, w_{ijk} \in \{0, 1\}, \qquad \text{for distinct } i, j, k \in [n], \tag{2.10}$$

$$w_{ijk} = 0, \qquad \text{for } i, j, k \text{ not distinct.} \tag{2.11}$$

Note that there is a one-to-one correspondence between points $(x, w) \in Q$ and permutations $\pi \in \mathbb{S}_n$, where $x_{ij} = 1$ if and only if $\pi^{-1}(i) < \pi^{-1}(j)$, and $w_{ijk} = 1$ if and only if $\pi^{-1}(i) < \pi^{-1}(j) < \pi^{-1}(k)$.

Define $\bar{Q}$ similarly to Q, by replacing the integrality condition (2.10) with $0 \le w_{ijk} \le 1$. In other words, let $\bar{Q}$ be the convex hull of Q. Clearly, $\bar{Q}$ is a polytope. Also, define $\bar{P}$ by replacing (2.5) with $0 \le x_{ij} \le 1$ in the definition of P. Finally, let $Q_p = \{x : (x, w) \in Q\}$ and $\bar{Q}_p = \{x : (x, w) \in \bar{Q}\}$.

Theorem 1 *The sets P and Q_p are identical.*

Proof: We first show that $x \in Q_p$ implies $x \in P$. For $x \in Q_p$ and distinct $i, j, k \in [n]$, one has

$$x_{ij} + x_{ji} = \sum\nolimits_{(rst)\in\mathcal{T}_{i,j,k}} w_{rst} = 1,$$

where the first equality follows from (2.9) and the second equality follows from (2.8). This proves (2.3).

To prove (2.4), for distinct $i, j, k \in [n]$, one may write

$$\begin{aligned} x_{ij} + x_{jk} + x_{ki} &= w_{ijk} + w_{ikj} + w_{kij} + w_{jki} + w_{jik} \\ &\quad + w_{ijk} + w_{kij} + w_{kji} + w_{jki} \\ &= 1 + w_{kji} + w_{kij} + w_{jki} \le 2, \end{aligned}$$

where the first equality follows from (2.9), while the other two equalities follow from (2.8).

From (2.8) and (2.9), one has $x_{ij} \le 1$, and from (2.9) and (2.10), it follows that x_{ij} is a non-negative integer. Hence, x_{ij} is either 0 or 1, proving (2.5). To complete the proof of the claim that $Q_p \subset P$, observe that (2.6) follows from (2.9) and (2.11).

Suppose next that $x \in P$. For $i, j, k \in [n]$, let $w_{ijk} = x_{ij}x_{jk}$. We show that $x \in Q_p$ by proving that $(x, w) \in Q$. It is clear that (2.10) is satisfied.

When $i = j$ or $j = k$, the proof of (2.11) follows from (2.6). If $i = k \neq j$, then (2.11) follows from (2.3).

To see that (2.9) holds, note that, for distinct $i, j, k \in [n]$,

$$x_{ij} = x_{ij}x_{jk} + x_{ik}x_{kj} + x_{ki}x_{ij} = w_{ijk} + w_{ikj} + w_{kij}.$$

The first equality can be verified by considering all possible choices for (x_{ij}, x_{jk}, x_{ki}), i.e. by observing that

$$(x_{ij}, x_{jk}, x_{ki}) \quad \in \quad \big\{(0,0,1), (0,1,0), (1,0,0), (0,1,1), (1,1,0), (1,0,1)\big\},$$

since $(x_{ij}, x_{jk}, x_{ki}) = (0,0,0)$ and $(x_{ij}, x_{jk}, x_{ki}) = (1,1,1)$ are excluded by (2.4). As a result, (2.8) follows from (2.3) and (2.9).

Theorem 2 *The sets $\bar{P}$ and $\bar{Q}_p$ are identical.*

Proof (Sketch): Each triple (r, s, t) appears in the definition of $\bar{Q}_p$ as part of the following constraints:

$$\begin{aligned} \textstyle\sum_{(ijl) \in \mathcal{T}_{r,s,t}} w_{ijl} &= 1, \\ w_{ijl} + w_{ilj} + w_{lij} &= x_{ij}, \quad \forall (ijl) \in \mathcal{T}_{r,s,t}. \end{aligned} \tag{2.12}$$

Similarly, each triple (r, s, t) appears in the definition of $\bar{P}_p$ as part of the following constraints:

$$\begin{aligned} x_{rs} + x_{st} + x_{tr} &\leq 2, \\ x_{sr} + x_{ts} + x_{rt} &\leq 2, \\ x_{ij} + x_{ji} = 1 \quad \forall (ijl) &\in \mathcal{T}_{r,s,t}. \end{aligned} \tag{2.13}$$

Consider the tuples $(x_{rs}, x_{st}, x_{tr}, x_{sr}, x_{ts}, x_{rt})$ as restricted by 2.12 and 2.13, and denote them by $\bar{P}^{rst}$ and $\bar{Q}_p^{rst}$, respectively.

We first show that $\hat{x} \in \bar{Q}_p^{rst}$ implies $\hat{x} \in \bar{P}^{rst}$. Note that $\bar{Q}_p^{rst}$ is the convex hull of the points

$$(1,1,0,0,0,1), (0,1,1,1,0,0), (1,0,1,0,1,0),$$

$$(0,0,1,1,1,0),(1,0,0,0,1,1),(0,1,0,1,0,1).$$

It is easy to check that these points belong to $\bar{P}^{rst}$ as well, which completes the claim.

Next, we show that $\hat{x} \in \bar{P}^{rst}$ implies $\hat{x} \in \bar{Q}_p^{rst}$. Assume that there exists a $\hat{x} \in \bar{P}^{rst}$ such that $\hat{x} \notin \bar{Q}_p^{rst}$. Since $\hat{x} \notin \bar{Q}_p^{rst}$, there exists a facet of $\bar{Q}_p^{rst}$ which serves as a separating hyperplane between $\hat{x}$ and the interior of the polytope. Moreover, this facet is also a separating hyperplane for at least one vertex of the unit cube which does not belong to the convex hull [14]. Note that the vertices of the unit cube that do not belong to the convex hull are

$$(1,1,1,*,*,*),(*,*,*,1,1,1),(1,*,*,1,*,*),(0,*,*,0,*,*),$$

$$(*,1,*,*,1,*),(*,0,*,*,0,*),(*,*,1,*,*,1),(*,*,0,*,*,0);$$

the symbol "$*$" stands for either 1 or 0.

The facet $x_{rs}+x_{st}+x_{tr}=2$ is a separating hyperplane for $(1,1,1,*,*,*)$. The three vertices of the facet are $(1,1,0,0,0,1)$, $(0,1,1,1,0,0)$ and $(1,0,1,0,1,0)$, and for all points in the polytope not incident with the facet we have $x_{rs}+x_{st}+x_{tr}<2$. Since $\hat{x}$ is assumed not to belong to $\bar{Q}_p^{rst}$, it must hold that $\hat{x}_{rs}+\hat{x}_{st}+\hat{x}_{tr}>2$. But this contradicts the assumption that $\hat{x} \in \bar{P}^{rst}$.

The proof follows by considering all other vertices of the unit cube.

CHAPTER 3

WEIGHTED KENDALL RANK AGGREGATION: LINEAR PROGRAMMING APPROACH

In this chapter, we present a novel aggregation algorithm for calculating the median of rankings using weighted Kendall distance as a metric. The algorithms is based on Integer Programming in which the relaxed version is converted to Linear Programming. The algorithms have been given for two forms of weights: the first one for monotonic linear weights and the second one for monotonic second-degree polynomial case. At the end, the performance of the algorithm is compared with other methods.

3.1 Weighted Rank Aggregation

3.1.1 Linear Weighted Distances

While an efficient algorithm for computing the weighted Kendall distance with an arbitrary weight function ρ is not known, a polynomial-time algorithm exists if the weight function is decreasing, i.e., if $\rho_i \geq \rho_{i+1}$.

Consider the following linear weight function:

$$\rho_i = 1 + \frac{\epsilon}{n-2}(n-1-i), \tag{3.1}$$

where $\epsilon \geq 0$. This function assigns weight $1+\epsilon$ to a swap involving the first and the second candidate, and weight 1 to a swap involving the last and the next to last candidate. The weights decrease linearly between these two points. Note that with this choice, swapping candidates at the top induces a larger distance between permutations. We subsequently make use of the following weight functions as well,

$$\rho_i = 1 + \frac{\epsilon}{n-2}(n-1-i)^k, \tag{3.2}$$

where k is a positive integer, and $\epsilon > 0$.

Let $I(\pi, \sigma)$ denote the set of ordered pairs (a, b) for which $\pi^{-1}(a) < \pi^{-1}(b)$ and $\sigma^{-1}(b) < \sigma^{-1}(a)$.

Lemma 1 *For permutations $\pi, \sigma \in \mathbb{S}_n$, and the weight function ρ of (4.1), we have*

$$\mathrm{d}_\rho(\pi, \sigma) = \sum_{i,j} \pi_{ij}\sigma_{ji}\Big(1 + \frac{\epsilon}{n-2}\sum_k \pi_{ik}\sigma_{jk}\Big). \tag{3.3}$$

Proof: It was shown in [11] that the minimum weight sequence of adjacent transpositions that converts π to σ is obtained as follows: for $\ell = 1, \ldots, n$, find $\sigma(\ell)$ in π and move it to position ℓ in π using adjacent transpositions. It then follows that the transposition that swaps $(i, j) \in I(\pi, \sigma)$ has weight ρ_s, where

$$s = \pi^{-1}(i) + \left|\left\{k : \sigma^{-1}(k) < \sigma^{-1}(j), \pi^{-1}(i) < \pi^{-1}(k)\right\}\right|.$$

It is not hard to show that s can also be written as

$$s = n - 1 - \left|\left\{k : \pi^{-1}(i) < \pi^{-1}(k), \sigma^{-1}(j) < \sigma^{-1}(k)\right\}\right|.$$

Using (2.2), we have $s = n - 1 - \sum_k \pi_{ik}\sigma_{jk}$. The lemma follows from (4.1).

The objective function of the rank aggregation problem (2.1), with weights given by (4.1), equals

$$\begin{aligned}\sum_{\sigma\in\Sigma}\mathrm{d}_\rho(x, \sigma) &= \sum_{\sigma\in\Sigma}\sum_{i,j} x_{ij}\sigma_{ji}\Big(1 + \frac{\epsilon}{n-2}\sum_k x_{ik}\sigma_{jk}\Big) \\ &= \sum_{i,j} x_{ij}\sum_{\sigma\in\Sigma}\sigma_{ji} + \frac{\epsilon}{n-2}\sum_{i,j,k} x_{ij}x_{ik}\sum_{\sigma\in\Sigma}\sigma_{ji}\sigma_{jk}.\end{aligned} \tag{3.4}$$

Let d_{ijk} denote the number of voters who prefer i to j, and j to k. Note that $\sum_{\sigma\in\Sigma}\sigma_{ji}\sigma_{jk} = d_{jik} + d_{jki}$. Hence, for $x \in P$,

$$\sum_{\sigma\in\Sigma}\mathrm{d}_\rho(x, \sigma) = \sum_{i,j} c_{ij}x_{ij} + \frac{\epsilon}{n-2}\sum_{i,j,k}\left(d_{jik} + d_{jki}\right)x_{ij}x_{ik}.$$

The objective function consequently reduces to

$$\min_{x\in P}\sum_{i,j} c_{ij}x_{ij} + \frac{\epsilon}{n-2}\sum_{i,j,k}\left(d_{jik} + d_{jki}\right)x_{ij}x_{ik}. \tag{3.5}$$

Theorem 1 implies that $x \in P$ if and only if $x \in Q_p$. Hence, one can replace $x \in P$ in (3.5) with $(x, w) \in Q$. For every $(x, w) \in Q$ and $i, j, k \in [n]$, it is straightforward to see that $x_{ij}x_{ik} = w_{ijk} + w_{ikj}$. Hence, we may rewrite (3.5) as

$$\min_{(x,w)\in Q} \sum_{i,j} c_{ij}x_{ij} + \frac{\epsilon}{n-2} \sum_{i,j,k} (d_{jik} + d_{jki}) (w_{ijk} + w_{ikj}) . \tag{3.6}$$

Since $c_{ij} = d_{jik} + d_{jki} + d_{kji}$, and

$$x_{ij} = \frac{1}{n-2} \sum_{k} (w_{ijk} + w_{ikj} + w_{kij}) ,$$

it is apparent that (3.6) is equivalent to

$$\min_{w\in W} \frac{1}{n-2} \sum_{i,j,k} \alpha_{ijk} w_{ijk}, \tag{3.7}$$

where $W = \{w : (x, w) \in Q\}$ and

$$\begin{aligned} \alpha_{ijk} = d_{ikj} + (1+\epsilon)d_{jik} + (2+\epsilon)d_{kij} \\ + (2+\epsilon)d_{jki} + (3+\epsilon)d_{kji}. \end{aligned}$$

The coefficients on the right side of the above equation have an interesting interpretation. For each permutation (rst) of $\{i, j, k\}$, the coefficient of d_{rst} equals the weighted Kendall distance between the permutations (rst) and (ijk), based on the weight function (4.1) and for $n = 3$. In other words,

$$\alpha_{ijk} = \sum_{(rst)\in \mathcal{T}_{i,j,k}} \mathrm{d}_\rho(rst, ijk)\, d_{rst},$$

which for $\epsilon = 1$ reduces to

$$\alpha_{ijk} = d_{ikj} + 2d_{jik} + 3d_{kij} + 3d_{jki} + 4d_{kji}.$$

3.1.2 The Dual Problem

The dual of the problem (4.1) can be written as

$$\begin{aligned}
&\max_{\lambda} \sum_{i<j<k} \lambda_{\{i,j,k\}} \\
&\text{s.t. for all distinct } i,j,k \in [n]: \\
&\quad \lambda_{\{i,j,k\}} + \nu_{ijk} + \nu_{ikj} + \nu_{jki} \\
&\quad - \nu_{ijh_{ij}(k)} - \nu_{ikh_{ik}(j)} - \nu_{jkh_{jk}(i)} \leq \alpha_{ijk},
\end{aligned}$$

The brackets in the subscript of λ indicate that $\lambda_{\{i,j,k\}} = \lambda_{\{i,k,j\}} = \cdots$, i.e., that the order of i, j, and k does not matter. Here, $h_{ij}(k)$ is the element that (circularly) precedes k in the vector $(1, \cdots, i-1, i+1, \cdots, j-1, j+1, \cdots, n)$. For example, $h_{25}(4) = 3$ and $h_{12}(3) = n$.

There does not seem to be a clear interpretation of the dual problem. However, if we let the ν variables equal to zero, we obtain the following problem:

$$\begin{aligned}
&\max \sum_{i<j<k} \lambda_{\{i,j,k\}} \qquad (3.8) \\
&\text{s.t. } \lambda_{\{i,j,k\}} \leq \min\{\alpha_{rst} : (rst) \in \mathcal{T}_{i,j,k}\}, \quad \forall\ i<j<k.
\end{aligned}$$

The optimal value of the latter problem has a clear interpretation as a lower bound: for each set of distinct values $\{i, j, k\}$ at least one of the w's is one, and thus at least a value of $\min\{\alpha_{ijk}, \alpha_{ikj}, \alpha_{kij}, \alpha_{jik}, \alpha_{jki}, \alpha_{kji}\}$ is contributed to the total sum.

3.2 Quadratic Weight Functions

In Section 3.1, we derived a linear programming relaxation of the rank aggregation problem with the linear weight function

$$\rho_i = 1 + \frac{\epsilon}{n-2}(n-1-i).$$

A similar approach can be used for weight functions of the more general form of (3.2), with k a positive integer. For simplicity, we illustrate the general

problem on the quadratic weight function

$$\rho_i = 1 + \frac{\epsilon}{n-2}(n-1-i)^2. \tag{3.9}$$

For the quadratic weight function ρ, the distance between rankings π and σ is

$$\mathrm{d}_\rho(\pi,\sigma) = \sum_{i,j} \pi_{ij}\sigma_{ji}\Big(1 + \frac{\epsilon}{n-2}\big(\sum_k \pi_{ik}\sigma_{jk}\big)^2\Big). \tag{3.10}$$

Hence, for $x \in P$,

$$\begin{aligned}\sum_{\sigma\in\Sigma}\mathrm{d}_\rho(x,\sigma) &= \sum_{\sigma\in\Sigma}\sum_{i,j} x_{ij}\sigma_{ji}\Big(1 + \frac{\epsilon}{n-2}\big(\sum_k x_{ik}\sigma_{jk}\big)^2\Big)\\ &= \sum_{\sigma\in\Sigma}\sum_{i,j} x_{ij}\sigma_{ji} + \sum_{\sigma\in\Sigma}\sum_{i,j,k}\frac{\epsilon}{n-2}x_{ik}x_{ij}\sigma_{jk}\sigma_{ji}\\ &\quad + \sum_{\sigma\in\Sigma}\sum_{i,j,k}\sum_{l\neq k}\frac{\epsilon}{n-2}x_{ik}x_{il}x_{ij}\sigma_{jk}\sigma_{jl}\sigma_{ji}.\end{aligned}$$

Let R be the set of points (x,w), with $x = (x_{ij})$ and $w = (w_{ijkl})$, satisfying

$$\begin{aligned}\sum\nolimits_{(rstu)\in\mathcal{T}_{i,j,k,l}} w_{rstu} &= 1, && \text{for distinct } i,j,k,l \in [n],\\ \sum\nolimits_{(rstu)\in\mathcal{T}^{i>j}_{i,j,k,l}} w_{rstu} &= x_{ij}, && \text{for distinct } i,j,k \in [n],\\ w_{ijkl} &\in \{0,1\}, && \text{for distinct } i,j,k,l \in [n],\\ w_{ijkl} &= 0, && \text{for } i,j,k,l \text{ not distinct},\end{aligned}$$

where $\mathcal{T}_{i,j,k,l}$ denotes the set of permutations of $\{i,j,k,l\}$ and $\mathcal{T}^{i>j}_{i,j,k,l}$ denotes the set of permutations of $\{i,j,k,l\}$ in which i appears before j. Note that there is a one-to-one correspondence between points $(x,w) \in R$ and permutations $\pi \in \mathbb{S}_n$, where $x_{ij} = 1$ if and only if $\pi^{-1}(i) < \pi^{-1}(j)$ and $w_{ijkl} = 1$ if and only if $\pi^{-1}(i) < \pi^{-1}(j) < \pi^{-1}(k) < \pi^{-1}(l)$.

Similar to Theorem 1, one can show that $P = \{x : (x,w) \in R\}$. Furthermore, it is straightforward to show that $x_{ik}x_{ij}$ and $x_{ik}x_{ij}x_{il}$ are linear in $w_{rstu}, r,s,t,u \in [n]$. Let e_{ijkl} be the number of permutations $\sigma \in \Sigma$ with $\sigma^{-1}(i) < \sigma^{-1}(j) < \sigma^{-1}(k) < \sigma^{-1}(l)$. The rank aggregation problem with quadratic weight function is equivalent to

$$\arg\min_{(x,w)\in R} \sum_{i,j,k,l} \beta_{ijkl} w_{ijkl}, \tag{3.11}$$

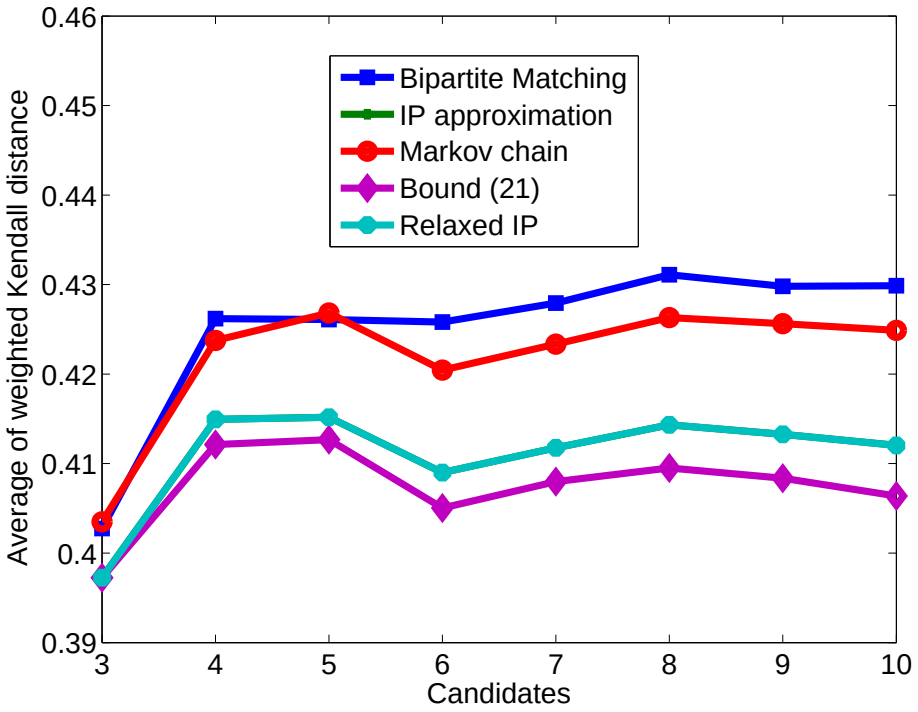

Figure 3.1: m=10.

where β_{ijkl}, for $i, j, k, l \in [n]$, are linear combinations of $e_{rstu}, r, s, t, u \in [n]$. Note that the objective function of (3.11) is linear. Furthermore, if we replace the integrality condition $w_{ijkl} \in \{0, 1\}, \quad$ for $i, j, k, l \in [n]$, with $0 \leq w_{ijkl} \leq 1, \quad$ for $i, j, k, l \in [n]$, we obtain a linear programming relaxation for the rank aggregation problem with a quadratic weight function.

3.3 Simulations

We evaluate the performance of the bound (3.8), the IP approximation (4.1) and relaxed IP bound (when condition (2.10) is replaced with $0 \leq w_{ijk} \leq 1$). Moreover, we considered the WBM and Markov chain (PR) methods, adapted for the weighted Kendall distance measures in [10]. We compared the averages of the objective function based on the weighted Kendall distance given in section 3.1 (here $\epsilon = 1$). The average value refers to

$$\frac{1}{m} \sum_{\sigma \in \Sigma} d_\rho(\hat{\pi}, \sigma),$$

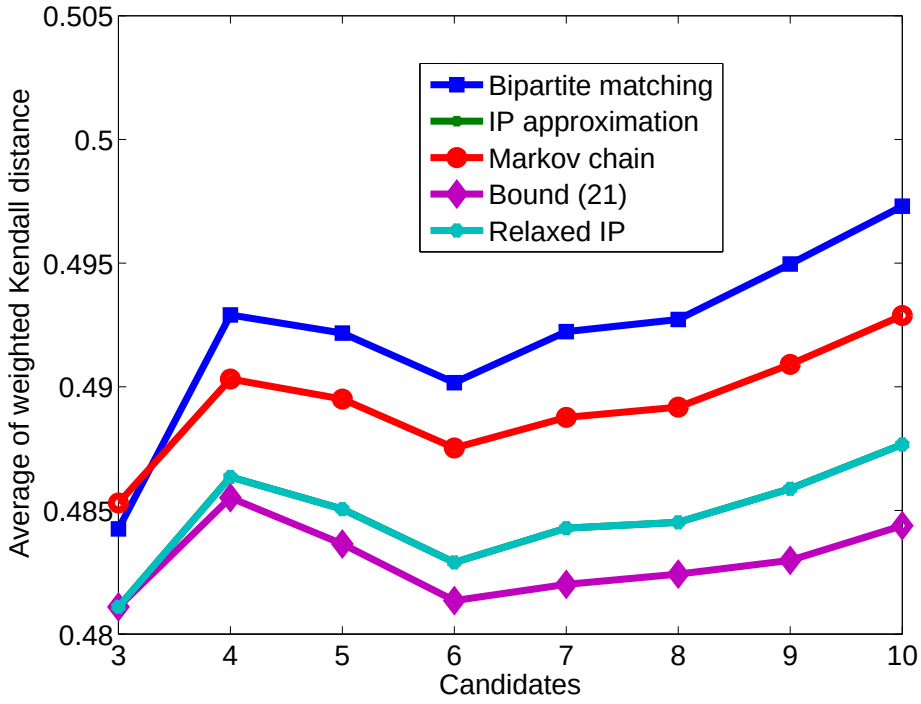

Figure 3.2: m=50.

where $\hat{\pi}$ represents a solution found by a particular algorithm. The minimum of the average value is attained by the optimal solution. Note that in relaxed IP and for the bound (3.8), the solutions do not necessarily represent permutations. In these cases, we use a lower bound on the average value of the optimal solution based on the weighted Kendall distance.

We generated different sets of votes with varying number of candidates. The votes were chosen in an iid manner and generated uniformly. The number of candidates varies from $n = 3$ to $n = 10$. For $m = 10, 50$ the results obtained by the aforementioned algorithms are depicted in Figures 3.1 and 3.2. More precisely, Figures 3.1 and 3.2 illustrate the average value of solutions obtained by IP approximation, bipartite matching, and the Markov chain method. They also illustrate lower bounds on the optimal average value obtained from (3.8), and from the relaxed integer programming approach. For each data point, we created 500 samples of votes.

To find the solution for the IP approximation, we used a branch and bound method. Notice that the curves for the integer programming approximation and the relaxed integer program match very well. This means that integer programming approximations are quite successful in finding the correct opti-

mal solution based on the weighted Kendall distance. Integer programming approximations outperform bipartite matching and Markov chain techniques. The bound (3.8) remains below the other curves, as expected. Surprisingly, it does not exhibit large deviations from the optimal average value. This is interesting, since the bound (3.8) is attained with much smaller computational cost.

CHAPTER 4

GENE PRIORITIZATION VIA WEIGHTED KENDALL RANK AGGREGATION

It is known that humans have roughly $25,000$ genes, some of which – when mutated – may lead to a host of diseases, conditions and abnormal phenotypes. Despite decades of intense research focus, the underlying gene aberrations that lead to even the most frequently encountered diseases are not completely known. Usually, the main impediment to identifying disease genes is the time-consuming and costly process of testing a working hypothesis, further exacerbated by alternative splicing and by the fact that typically, multiple genes have to be jointly mutated to trigger the onset of a disease. Even for experiments involving only up to three genes, one would have to test as many as 4×10^{12} combinations of genes in order to check if they are linked to a given disease. This is clearly an infeasible experimental endeavor which will remain difficult to accomplish for decades to come.

One approach to mitigate the problem is to preprocess available biological side-information about genes and then reduce the set of test genes accordingly. The problem of identifying a small subset of genes likely to be causally linked with a disease is known as the *gene prioritization problem*, and the algorithmic solutions for solving the problem are classified as prioritization algorithms. Prioritization algorithms are typically based on using experimentally confirmed disease genes and identifying different qualitative evidence that associates the disease genes with target test genes. For this purpose, linkage analysis, sequence similarity, functional annotation, marker and pedigree analysis are all combined. The evidence obtained establishes the ranking of candidate genes based on the extent of their relationship – or similarity – to the training set of disease genes.

In the past few years, a number of sophisticated computational gene prioritization tools were proposed in [1, 5, 7, 15]. Most of these methods are statistical and *quantitative* in nature. Although offering significant improvements over random search methods, most such methods suffer from the fact

that they tacitly or implicitly rely on the assumptions that a) a test gene has to be close to the training genes *under all similarity criterion*; in other words, the top-ranked genes have to be highly ranked in all individual lists reflecting different criteria for comparison; and b) no distinction is to be made about the accuracy of ranking genes in any part of the list; in other words, the aggregate ranking has to be *uniformly accurate* at the top, middle and bottom of the list. Clearly, neither of the two aforementioned assumptions is justified in the gene prioritization process: there are many instances where genes similar only under a few criteria (such as sequence similarity or linkage distance) are involved in the same disease pathways. Given that the goal of prioritization is to produce a list of genes to be experimentally tested in a wet lab, only highly relevant candidate genes are to be considered, and consequently, such genes have to be ranked with higher accuracy than other genes on the list. Furthermore, aggregation of rankings based on statistical methods is often highly sensitive to outliers and ranking errors.

To overcome the above issues of classical prioritization approaches, we employ a combinatorial median approach to ordinal data fusion using the weighted Kendall τ distance, first introduced by the authors in [12]. The aggregation approach is henceforth referred to as the *generalized Kemeny approach*. The ranking obtained using the weighted Kendall τ distance is more influenced by top positions in the rankings obtained from different criteria so it is robust to negative outliers – i.e., a small number of low rankings of some candidate gene. These properties are useful for gene prioritization, as weighted Kendall τ distance does not penalize genes for not being similar to training genes under *every* possible similarity criteria, and it allows for fusing weak orders in which several candidate genes may be ranked the same, which helps in resolving frequent scoring ambiguities. Although fundamental results from social choice theory and political sciences have shown that there exists no "optimal" rank aggregation method that is consistent, fair, and impossible-to-manipulate [22], the Kemeny method is one of the few aggregation solutions that provably offers a large number of performance guarantees. The properties of the generalized Kemeny method were investigated in our companion papers [12, 19].

We apply the generalized Kemeny approach to lists of rankings generated by Endeavour and ToppGene [1, 5], using criteria such as sequence similarity, CisReg modules, expression profiles, transcription factor binding sites,

annotation in different databases, pathways, etc. Our sets of test genes pertain to the Bardet-Biedl syndrome (a genetic condition affecting cellular cilia and causing obesity, retinal failure and sometimes mental retardation), schizophrenia, and HIV (Human Immunodeficiency Virus) infections. Despite the fact that generalized Kemeny aggregation is purely combinatorial in nature and hence discards all quantitative information in data, i.e., it does not make use of the p-values but only the underlying *rankings* of genes, it usually outperforms Endeavour [1] and matches/outperforms ToppGene [5]. In many instances, it produces ties in the rankings, potentially indicative of insufficient evidence to accurately discern the most similar genes (note that ToppGene and Endeavour always produce complete linear orders).

4.1 Aggregation Algorithm for Weak Orders

In the same way as Chapter 3, assume that one is given a set of n genes, ranked according to N different similarity criteria. For simplicity, one may assume that the genes are indexed by the positive integers $[n] = \{1, 2, \ldots, n\}$. Each ranking without ties may be viewed as a permutation over $[n]$, i.e., an element of the symmetric group $\mathbb{S}_n$. Similarly, a ranking with ties may be viewed as an ordered set partition, i.e., an ordered partition of the set $[n]$ into classes, where all genes in the same class are considered to have the same rank. As an example, for $n = 6$, $\sigma = (1, 5, 4, 3, 2, 6)$ is a ranking without ties, while $\sigma = (\{2, 3\}, \{1\}, \{4, 5, 6\}) = (2 - 3, 1, 4 - 5 - 6)$ is a ranking with ties. In the latter case, genes indexed by 2 and 3 share the first position, i.e. they are the top ranked genes. Usually, we represent ranking with ties through their *median scores*, defined as the average position of an element within a part. For the previous example, 2 and 3 have a median score of 1.5, given that they occupy the 1st and 2nd position, and $(1 + 2)/2 = 1.5$.

For a linearly decreasing weight function of the form

$$\rho_i = 1 + \frac{\epsilon}{n-2}(n - 1 - i),$$

with $\epsilon \geq 0$, it can be shown that the LP relaxation of the corresponding

aggregation problem equals

$$\min_{w \in W} \frac{1}{n-2} \sum_{i,j,k} \alpha_{ijk} w_{ijk}, \tag{4.1}$$

where W represents the set of points $w = (w_{ijk})$, with $i, j, k \in [n]$, satisfying

$$\begin{aligned}
\textstyle\sum_{(r,s,t) \in \mathcal{T}_{i,j,k}} w_{rst} &= 1, && \text{for distinct } i, j, k \in [n], \\
w_{ijk} + w_{ikj} + w_{kij} &= x_{ij}, && \text{for distinct } i, j, k \in [n], \\
x_{ij},\, w_{ijk} &\in [0,1], && \text{for distinct } i, j, k \in [n], \\
w_{ijk} &= 0, && \text{for } i, j, k \text{ not distinct.}
\end{aligned}$$

Here, the variables x_{ij} have the same interpretation as in the classical Kemeny aggregation framework, $\mathcal{T}_{r,s,t} \equiv \mathbb{S}_3 = \{(r,s,t), (r,t,s), (s,r,t), (s,t,r), (t,r,s), (t,s,r)\}$, and

$$\alpha_{ijk} = \sum_{(r,s,t) \in \mathcal{T}_{i,j,k}} \mathrm{d}_\rho((r,s,t),(i,j,k))\, d_{rst},$$

where d_{rst} denotes the number of $\sigma \in \Sigma$ that rank r higher than s higher than t. Note that for the given linear choice of the weight ρ, it suffices to use $\mathcal{T}_{r,s,t}$ on triples of variables only. Furthermore, this definition easily extends to rankings with ties, by replacing $\mathcal{T}_{r,s,t}$ with

$$\begin{aligned}
\mathcal{T}^{(*)}_{r,s,t} = {} & \{(r,s,t), (r,t,s), (s,r,t), (s,t,r), (t,r,s), (t,s,r)\} \\
\cup\ & \{(r,s-t), (s,r-t), (t,r-s)\} \\
\cup\ & \{(r-s,t), (r-t,s), (s-t,r), (r-s-t)\},
\end{aligned}$$

and defining $\mathrm{d}_\rho(\pi_1, \pi_2)$, for $\pi_1, \pi_2 \in \mathcal{T}^{(*)}_{r,s,t}$, as the shortest path between π_1 and π_2 in the graph shown in Figure 4.1.

As a final remark, we observe that the LP program for weighted aggregation with ties of lists of n genes involves $O(n^3)$ constraints and $O(n^2)$ variables. Still, the constraints are sparse, which allows for efficient computational savings.

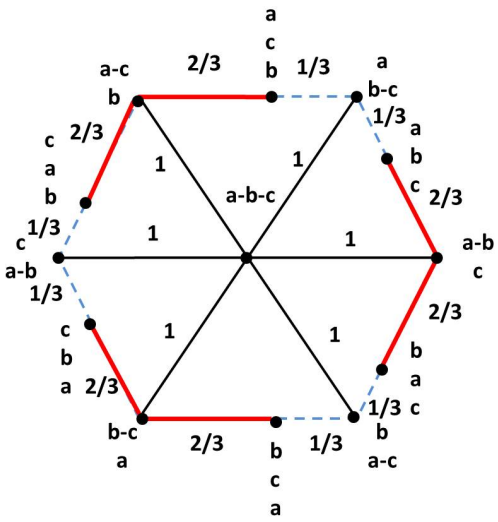

Figure 4.1: A graph for the weighted Kendall distance between rankings with ties, involving three elements. The weights of the edges have to satisfy certain symmetry constraints, as described in [9,13]. The weights in our example are chosen to illustrate this symmetry property. To avoid confusion between the numerical values of the weight and the identity of candidates, we used the set $\{a, b, c\}$ to represent the candidates.

4.2 Former Prioritization Methods

One of the earliest gene prioritization software package is Endeavour [1]. For different criteria, Endeavour ranks the candidate test genes based on their similarity to a set of known training genes. For each similarity criteria, Endeavour first calculates the average p-value with respect to the training genes, i.e., the probability of obtaining a test statistic as extreme as the one observed, under suitably chosen null hypotheses (the method which Endeavour uses to calculate the p-values is beyond the scope of this thesis). It subsequently ranks the test genes from lowest to highest p-values. The rankings are aggregated via the Q-statistic, calculated from all rank ratios r_i, $i = 1, \ldots, N$, using the joint cumulative distribution of an N – dimensional order statistic,

$$Q(r_1, r_2, ..., r_N) = N! \int_0^{r_1} \int_{s_1}^{r_2} \cdots \int_{s_{N-1}}^{r_N} ds_N ds_{N-1} \ldots ds_1.$$

Here, the indices i refer to data sources, where N is the total number of data sources. Also, $r_0 = 0$.

ToppGene, a more recent software described in [5], also ranks candidate genes according to average p-values for different criteria, but the choice of criteria and the aggregation method differ from those proposed in Endeavour. The main difference is that ToppGene employs Human and Mouse phenotypes as one of the criterion, because direct comparison of human and mouse phenotypes provides vital information for identifying disease genes [4]. ToppGene aggregates rankings via Fisher's inverse chi-square method, which aggregates the p-values of different criteria, p_i, $i = 1, \ldots, N$, into $-2\sum_{i=1}^{N} \log p_i$. Assuming that the p-values $p_i, i = 1, \ldots, N$, come from independent tests and that the null hypotheses are all true, one has $-2\sum_{i=1}^{N} \log p_i \to \chi^2(2n)$, where $\chi^2(2n)$ denotes a χ^2 distribution with $2n$ degrees for freedom. Despite the fact that the p-values of gene prioritization criteria may not be independent, ToppGene currently appears to be the state-of-the-art prioritization method in terms of accuracy.

One of the most recently developed prioritization methods, NetworkPrioritizer [15], uses distances between genes in regulatory networks as additional criteria, and performs combinatorial aggregation based on Weighted Borda Fuse (WBF), Weighted AddScore Fuse (WASF), and MaxRank Fuse. However, these methods have the same drawbacks as the classical aggregation methods and differ substantially from the generalized Kemeny approach pursued in this thesis.

4.3 Results for Disease Related Gene Identification

We tested the generalized Kemeny method on three diseases, and compared the overall rankings with those of ToppGene and Endeavour. For each disease, we obtained a list of phenotype genes on OMIM (Online Mendelian Inheritance in Man) [13], some of which are labeled as "training genes" and some as "test genes". For example, OMIM lists 14 genes known to be involved in the Bardet-Biedl syndrome, 11 of which are listed as "training genes" in Table 4.1, and 3 genes, colored in red – TTC8, CEP290, MKS1– are part of the 12 "test genes". These phenotype test genes are expected to be ranked high in the overall aggregate, since there is strong evidence that they are similar to the training genes. The rest of the test genes are selected from GeneCards (www.genecards.org) [20] such that they are not known to

be related to the disease. Although the sets of training and test genes are identical for Endeavour and ToppGene, the criteria used by Endeavour and ToppGene are different. For fairness of comparison, we took the *intersection* of Endeavour and ToppGene criteria. From the ToppGene suite, we used GO: Molecular Function, GO: Biological Process, GO: Cellular Component, Domain, Pathway, Pubmed, Interaction, Transcription Factor Binding Site, Gene Family. From the Endeavour suite, we used GeneOntology, Interpro, Kegg, Motif, and Text.

We performed generalized Kemeny aggregation with ties via the LP method of Chapter 2; the results are shown in Tables 4.1-4.3. The first two columns label the gene symbols with numbers, and those "Gene numbers" are used throughout columns 4-6. Note that column 3 simply indexes the ranking from 1 to 12, and the numbers are *not* gene numbers. Columns 4-6 contain rankings of genes according to ToppGene, generalized Kemeny, and Endeavour, respectively. In the case of the Bardet-Biedl syndrome, the generalized Kemeny method matches the performance of ToppGene, as it ranked the three phenotype genes at the top, and it outperforms Endeavour. A similar result is true for schizophrenia. The HIV results are interesting in that both ToppGene and Endeavour placed the three phenotype genes between the 2nd and 6th position, whereas the generalized Kemeny approach ranked all three phenotype genes at the top, tied along with 3 other non-phenotype genes.

Table 4.1: Results for training genes CCDC28B, BBS5, ARL6, BBS7, BBS12, TMEM67, TRIM32, BBS1, BBS10, BBS4, BBS2, implicated with the **Bardet-Biedl syndrome**.

Gene #	**HGNC Symbol**	**Rank #**	**ToppGene**	**Generalized Kemeny**	**Endeavour**
1	TTC8	1	1	1	1
2	CEP290	2	2	3	2
3	MKS1	3	3	2	9
4	APP	4	4	5	3
5	ASPM	5	5	4	7
6	IL10	6	6	10 - 11	8
7	MYOD1	7	7		5
8	BDNF	8	8	7	11
9	SRY	9	9	9	12
10	CD4	10	10	12	10
11	SDHD	11	11	8	4
12	ZBTB7A	12	12	6	6

Table 4.2: Results for training genes MTHFR, CHI3L1, DISC1, SYN2, DRD3, DTNBP1, HTR2A, RTN4R, APOL4, implicated with **schizophrenia**.

Gene #	**HGNC Symbol**	**Rank #**	**ToppGene**	**Generalized Kemeny**	**Endeavour**
1	AKT1	1	1	1	1
2	HCN4	2	2	2	4
3	DAO	3	3	3	6
4	ADCY3	4	4	4	5
5	EPO	5	5	5 - 6	12
6	SOX3	6	6		7
7	LRAT	7	7	7	3
8	FGG	8	8	8	9
9	FGD3	9	9	9	2
10	NNT	10	10	10	8
11	ACLY	11	11	11	11
12	ICOS	12	12	12	10

Table 4.3: Results for training genes CX3CR1, TLR3, HLA-C, CXCL12, IFNG, IL4R, CCL2, implicated with **HIV**.

Gene #	**HGNC Symbol**	**Rank #**	**ToppGene**	**Generalized Kemeny**	**Endeavour**
1	CXCR4	1	1	1 - 2 - 3 - 4 - 5 - 6	1
2	IL10	2	2		3
3	OSM	3	3		2
4	CRH	4	4		5
5	CD209	5	5		6
6	KIR3DL1	6	6		7
7	HFE	7	7	7	9
8	APC	8	8	10	8
9	RHO	9	9	8 - 9	11
10	SLC18A2	10	10		4
11	ABO	11	11	11	10
12	MCM6	12	12	12	12

CHAPTER 5

CONCLUSION

In this thesis, we presented novel algorithms for aggregation of rankings using Integer Programming. While older methods store the orders of the pairs as variables, in this work we considered the order of the triplets of the objects to be stored in variables. The method that we used calculates the median of rankings which employs weighted Kendall distance as the metric. The algorithm is based on LP and we observed that the relaxed IP method shows close approximation to the actual integer programming. The applications of the aggregation method for gene prioritization showed better performance for identifying genes related to certain diseases in comparison with other ranking methods which were used in TopGene and Endeavor.

In this work, we presented algorithms for two types of weight functions and this might be generalized to newer forms of weight functions in the future. Furthermore, the data formats that we considered in this work were permutations. In the future, the results of this thesis can potentially be extended to more general ranking formats like weak orders and partial orders.

REFERENCES

[1] S. Aerts *et al.*, "Gene prioritization through genomic data fusion," *Nat. Biotechnol.*, vol. 24, pp. 537–544, 2006.

[2] A. Barg and A. Mazumdar, "Codes in permutations and error correction for rank modulation," *IEEE Trans. Information Theory*, vol. 56, pp. 3158–3165, July 2010.

[3] J. Bartholdi, C. Tovey, and M. Trick, "The computational difficulty of manipulating an election," *Social Choice and Welfare*, vol. 6, no. 3, pp. 227–241, 1989.

[4] J. Chen *et al.*, "Improved human disease candidate gene prioritization using mouse phenotype," *BMC Bioinformatics*, vol. 8, pp. 398, 2007.

[5] J. Chen *et al.*, "ToppGene Suite for gene list enrichment analysis and candidate gene prioritization," *Nucleic Acids Res.*, vol. 37, pp. W305–W311, 2009.

[6] V. Conitzer, A. Davenport, and J. Kalagnanam, "Improved bounds for computing Kemeny rankings," in *Proc. of the 21st National Conf. on Artificial Intelligence*, Boston, Massachusetts, 2006.

[7] T. De Bie *et al.*, "Kernel-based data fusion for gene prioritization," *Bioinformatics*, vol. 23, pp. i125–i132, 2007.

[8] P. Diaconis, "Group representations in probability and statistics," *Lecture Notes-Monograph Series*, vol. 11, 1988.

[9] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation revisited," unpublished manuscript, 2001.

[10] F. Farnoud, B. Touri, and O. Milenkovic, "Nonuniform vote aggregation algorithms," in *Int. Conf. Signal Processing and Communications (SPCOM)*, Bangalore, India, July 2012.

[11] F. Farnoud, B. Touri, and O. Milenkovic, "Novel distance measures for vote aggregation," *arXiv preprint arXiv:1203.6371*, 2012.

[12] F. Farnoud, O. Milenkovic, and B. Touri, "A novel distance-based approach to constrained rank aggregation," CoRR abs/1212.1471, 2012.

[13] A. Hamosh *et el.*, "Online Mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders," *Nucleic Acids Research*, vol. 33, pp. D514-D517, 2005.

[14] R. G. Jeroslow, "On defining sets of vertices of the hypercube by linear inequalities," *Discrete Mathematics*, vol. 11, pp. 119–124, 1975.

[15] T. Kacprowski *et al.*, "NetworkPrioritizer: a versatile tool for network-based prioritization of candidate disease genes or other molecules," *Bioinformatics*, vol. 29, pp. 1471–1473, 2013.

[16] M. Kendall, *Rank Correlation Methods.* London: Griffin, fourth ed., 1970.

[17] J. G. Kemeny, "Mathematics without numbers," *Daedalus*, vol. 88, no. 4, pp. 577–591, 1959.

[18] K. Pedings, A. Langville, and Y. Yamamoto, "A minimum violations ranking method," *Optimization and Engineering*, vol. 13, no. 2, pp. 349–370, 2012.

[19] F. Raisali, F. Farnoud, and O. Milenkovic, "Weighted rank aggregation via relaxed integer programming," *Proceedings of the ISIT*, 2013.

[20] M. Rebhan *et el.*, "GeneCards: a novel functional genomics compendium with automated data mining and query reformulation support," *Bioinformatics*, vol. 14, pp.656–664, 1998.

[21] G. Reinelt, *The Linear Ordering Problem: Algorithms and Applications,* Research and Exposition in Mathematics 8, Berlin: Heldermann, 1985.

[22] D. Saari and R. Merlin, "A geometric examination of Kemeny's rule," *Social Choice and Welfare*, pp. 403-438, 2002.