

# Extended Spearman and Kendall Coefficients for Gene Annotation List Correlation

Davide Chicco<sup>1,2</sup>, Eleonora Ciceri<sup>1</sup>, and Marco Masseroli<sup>1</sup>

<sup>1</sup> Dipartimento di Elettronica Informazione e Bioingegneria,  
Politecnico di Milano, Milan, Italy

<sup>2</sup> Princess Margaret Cancer Centre,  
University of Toronto, Toronto, Canada

davide.chicco@gmail.com,  
eleonora.ciceri@polimi.it,  
masseroli@elet.polimi.it

**Abstract.** Gene annotations are a key concept in bioinformatics and computational methods able to predict them are a fundamental contribution to the field. Several machine learning algorithms are available in this domain; they include relevant parameters that might influence the output list of predicted gene annotations. The amount that the variation of these key parameters affect the output gene annotation lists remains an open aspect to be evaluated. Here, we provide support for such evaluation by introducing two list correlation measures; they are based on and extend the Spearman  $\rho$  correlation coefficient and Kendall  $\tau$  distance, respectively. The application of these measures to some gene annotation lists, predicted from Gene Ontology annotation datasets of different organisms' genes, showed interesting patterns between the predicted lists. Additionally, they allowed expressing some useful considerations about the prediction parameters and algorithms used.

**Keywords:** Biomolecular annotations, Spearman coefficient, Kendall distance, top-K queries.

## 1 Introduction

In molecular biology and bioinformatics, a *controlled biomolecular annotation* is an association of a biomolecular entity (mainly a gene, or gene product) with a concept, described by a term of a controlled vocabulary (in this case part of an ontology), which represents a biomedical feature. This association states that the biomolecular entity has such feature. For instance, the association  $\langle \text{Entrez Gene ID 1080, GO:0055085} \rangle$  is a typical annotation of the human *CFTR* gene (*Cystic fibrosis transmembrane conductance regulator (ATP-binding cassette sub-family C, member 7)*), which has Entrez Gene ID 1080, with the concept represented by the *transmembrane transport* term of the Gene Ontology, which has ID GO:0055085. Thus, such annotation states that the human *CFTR* gene is involved in the *transmembrane transport*.

Despite their biological significance, there are some issues concerning available biomolecular annotations [1]. In particular, they are incomplete: only a subset of the biomolecular entities of the sequenced organisms is known, and among those entities only a small part has been annotated by researchers so far. In addition, they may be erroneously annotated and not yet revisited, prior to their being stored into online data banks. Within this context, computational methods and software tools able to produce lists of available or new predicted annotations ranked based on their likelihood of being correct are an excellent contribution to the field [2].

For this reason, starting from a state-of-the-art algorithm [3] based on truncated Singular Value Decomposition (SVD) [4], we developed some enhanced variants that take advantage of available Gene Ontology (GO) [5] annotation data to predict new gene annotations of different organisms, including *Homo sapiens*. Specifically, we designed an automated algorithm that chooses the best the truncation level [6] for the truncated SVD method and developed some alternatives to the SVD, based on gene clustering [7] and Resnik’s [8] term-term similarity metrics [9]. Similar to Khatri and colleagues papers [10] [11], we additionally implemented another version of this method with the enhancement of frequency and probability weights [12]. To this end, we also applied some *topic modeling* algorithms, such as Probabilistic Latent Semantic Analysis (pLSA) by Hofmann et al. [13], and Latent Dirichlet Allocation (LDA) by Blei et al. [14], obtaining relevant results, respectively, in [15] and [16]. Additionally, one of the authors recently took advantage of a *deep learning* algorithm, built on a multi-layer autoencoder neural network, that lead to interesting prediction results in reference [17].

All these methods involve key parameters that strongly influence their output. To understand how the resulting annotation lists vary when these key parameters change, a similarity measure that compares different output annotation lists is required. Currently, several metrics are available to compare ranked lists of elements. A good example is the Goodman-Kruskal’s  $\gamma$  [18], which measures the difference between rank-concordant or rank-discordant pairs of objects in two lists. However, Fagin and colleagues [19] state that the most useful and consistent measures are the Spearman  $\rho$  correlation coefficient [20] and Kendall  $\tau$  distance [21]. In recent years, many variants were proposed to meet new needs that came with some state-of-the-art applications, e.g., top- $K$  queries [22]. For example, the work proposed in [23] by Kumar and colleagues adapts the original formulation to measure weighted correlations, by placing more emphasis on items with high rankings. Applications have been shown in music signal prediction [24], recommendation systems [25] and computer vision [26].

In this work, we depart from a recent work by Ciceri et al. [27] to develop new weighted correlation metrics able to better compare biomolecular annotation lists. We adapt the weighted Kendall  $\tau$  distance (proposed in [23]) and the Spearman  $\rho$  rank correlation coefficient (proposed in [20]) to the case in which multiple lists (initially containing the same items in different orders) are

truncated up to a level  $K$ , thus resulting in sub-lists whose sets of contained items may not coincide.

The remainder of this chapter is organized as follows. Section 2 illustrates aspects related to the prediction of gene annotations. Section 3 introduces the Spearman  $\rho$  correlation coefficient and Kendall  $\tau$  distance variants for the comparison of annotation lists. Section 4 describes some significant test results of the proposed measure variants. Finally, we conclude in Section 5.

## 2 Prediction of Gene Ontology Annotations

Let  $\mathbf{A} = [a_{ij}]$  be an  $m \times n$  matrix, where each row corresponds to a gene and each column corresponds to a Gene Ontology feature term ( $a_{ij} = 1$  if gene  $i$  is annotated to feature term  $j$ ,  $a_{ij} = 0$  otherwise). Moreover, let  $\theta$  be a fixed threshold value. The prediction algorithm elaborates the matrix  $\mathbf{A}$  to produce an output matrix  $\tilde{\mathbf{A}}$ , with the same dimensions of  $\mathbf{A}$ , where each likelihood value  $\tilde{a}_{ij}$  is used to categorize an annotation:  $\langle \mathbf{gene}_i, \mathbf{feature}_j, \tilde{a}_{ij} \rangle$ . A high  $\tilde{a}_{ij}$  value indicates that the probability for  $\mathbf{gene}_i$  to be associated with the feature  $\mathbf{feature}_j$  is high. Each annotation  $\langle \mathbf{gene}_i, \mathbf{feature}_j, \tilde{a}_{ij} \rangle$  can be classified in four categories:

- *Annotation Predicted (AP)*:  $a_{ij} = 0 \wedge \tilde{a}_{ij} > \theta$  (similar to False Positive);
- *Annotation Confirmed (AC)*:  $a_{ij} = 1 \wedge \tilde{a}_{ij} > \theta$  (similar to True Positive);
- *Non-Annotation Confirmed (NAC)*:  $a_{ij} = 0 \wedge \tilde{a}_{ij} \leq \theta$  (similar to True Negative);
- *Annotation to be Reviewed (AR)*:  $a_{ij} = 1 \wedge \tilde{a}_{ij} \leq \theta$  (similar to False Negative).

Since APs and ARs can be considered as *presumed errors* with respect to the available annotations, we chose the value of  $\theta$  as the one that minimizes their sum (APs + ARs), as Khatri et al. did in [3]. After a *ten fold cross validation* phase, in which the 10% of annotations are randomly set to zero (as explained in reference [9]), the software compares each input annotation with its corresponding output prediction. Based on the value of the threshold  $\theta$ , a list of annotations with  $\tilde{a}_{ij} > \theta$  is created; it is subdivided in two sections: an **APlist**, i.e., *Annotation Predicted* list, and a **NAClist**, i.e., *Non-Annotation Confirmed* list, respectively containing the annotations from the original list that were classified as belonging to the AP or NAC category. Moreover, the defined categories are used to create a Receiver Operating Characteristic (ROC) curve, a graphical plot depicting the performance of a binary classifier system for different discrimination threshold values [28]. Similar to its original definition, which uses **TPrate** and **FPrate**, our ROC curve depicts the trade-off between the **ACrate** and the **APrate**, where:

$$\mathbf{ACrate} = \frac{\mathbf{AC}}{\mathbf{AC} + \mathbf{AR}} \quad \mathbf{APrate} = \frac{\mathbf{AP}}{\mathbf{AP} + \mathbf{NAC}} \quad (1)$$

for all possible values of  $\theta$ . Notice that, in statistical terms, **ACrate** = *Sensitivity* and **APrate** =  $1 - \textit{Specificity}$ . The output ROC space is thus defined by **ACrate** and **APrate** as  $x$  and  $y$  axes, respectively. In all of our test results, reported

in Section 4, we consider only the **APrate** in the normalized interval  $[0, 1]\%$ , in order to evaluate the best predicted annotations (APs) having the highest likelihood score. Furthermore, only if the obtained ROC Area Under the Curve (AUC) percentage is greater than a fixed threshold  $\theta_1 = 2/3 = 66.67\%$ , we consider to have good reliability of reconstruction. We use the annotation category labels we just introduced to define our measures in the following sections.

### 3 Annotation List Correlation Measures

Each annotation prediction algorithm has some key parameters; changing their values usually leads to different output predicted annotation lists, i.e. **APlists**. For instance, the truncated SVD (tSVD) algorithm may produce quite different results when its truncation level  $k$  varies. In the two variants of tSVD that we enhanced with gene clustering and term-term similarity metrics (named Semantically IMproved tSVD variants, i.e., SIM1 and SIM2), different results may be obtained when varying the number of gene clusters  $C$  [9]. In Probabilistic Latent Semantic Analysis [13], a topic modeling method, a key role is played by the number of topics  $T$  selected before performing the evaluation [15]. To understand the amount by which the selected parameter values are able to influence the output results, it is important to define similarity metric that compares two output **APlists** resulting from different algorithm parameterizations. To this end, we present novel variants to two well-known similarity metrics:

- the *Spearman rank correlation coefficient* ( $\rho$ ) [20]
- the *Kendall rank distance* ( $\tau$ ) [21]

which are respectively described in Subsection 3.1, and in Subsection 3.2.

#### 3.1 Spearman Rank Correlation Coefficient

The *Spearman rank correlation coefficient* ( $\rho$ , sometimes also called *foot-rule*) [20] measures the statistical dependence between two variables  $X$  and  $Y$ . The measure expresses either *positive* correlation, i.e.,  $Y$  increases when  $X$  increases, or *negative* correlation, i.e.,  $Y$  decreases when  $X$  increases. A similar definition can be applied to pair of ranked lists. Let  $l_a$  and  $l_b$  be two ranked lists of biomolecular annotations. A maximum positive correlation  $\rho = +1$  is returned when  $l_a$  and  $l_b$  are identical (i.e. having the same elements in the same order), while the maximum negative correlation  $\rho = -1$  is returned when  $l_a$  and  $l_b$  contain the same elements, but in reverse order. The minimum correlation  $\rho = 0$  (i.e., maximum diversity) is instead detected when the element order in  $l_a$  and  $l_b$  strongly diverges. Suppose  $l_a$  and  $l_b$  have the same length  $n$ . Given an element  $i$ , let  $x_i$  denote its position in  $l_a$ ,  $y_i$  its position in  $l_b$  and  $d_i = |x_i - y_i|$ . The final normalized Spearman  $\rho$  value is then computed as:

$$\rho = 1 - \frac{6 \cdot \sum d_i^2}{n(n^2 - 1)} \quad (2)$$

**Table 1.** Example of the application of the Spearman rank correlation coefficient;  $l_a$  and  $l_b$  are, respectively, the first and second compared lists;  $x_i$  is the position of the  $i^{th}$  element of  $l_a$  in  $l_a$ , and  $y_i$  is the position of the  $i^{th}$  element of  $l_a$  in  $l_b$ ;  $d_i$  is the difference between positions.

$l_a$	$x_i$	$l_b$	$y_i$	$d_i$	$d_i^2$
a	1	c	4	3	9
b	2	b	2	0	0
c	3	e	1	2	4
d	4	a	5	1	1
e	5	d	3	2	4

For example, the Spearman rank coefficient computed for the  $l_a$  and  $l_b$  in Table 1 is:  $\rho = 1 - \frac{6 \cdot 18}{5 \cdot 24} = 1 - \frac{18}{20} = 1 - 0.9 = 0.1$ . The low value of  $\rho$  indicates that  $l_a$  and  $l_b$  have a low correlation, as shown in Table 1.

**Weighted Spearman Rank Coefficient.** As mentioned earlier, two lists  $l_a$  and  $l_b$  may contain different elements or have different length; in this case, the lists would not be properly handled by the classical Spearman rank correlation coefficient. In order to additionally consider this case, based on the work by Ciceri et al. [27], we introduce a new *Weighted Spearman rank coefficient* featuring penalty distance weights  $w_i$  for each element  $i$  absent from one list.

Let  $q = |l_a \cup l_b|$ . Thus, the penalty weight  $w_{si}$  for an object  $i$  in the lists  $l_a$  or  $l_b$  is computed as follows:

$$w_{si} = \begin{cases} 1 - \frac{1}{|x_i - y_i| + 1}, & i \in l_a \wedge i \in l_b \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

The Weighted Spearman rank coefficient value is then computed as:

$$\rho_w = \frac{\sum_{i=1}^q w_{si}}{q} \quad (4)$$

High correlation is found when  $\rho_w \simeq 0$  (i.e., very few penalties are assigned), while low correlation is found when  $\rho_w \simeq 1$  (i.e., many penalties are assigned). If the two lists have no common elements, i.e.,  $q = |l_a| + |l_b|$ ,  $\rho_w = 1$ .

**Extended Spearman Rank Coefficient.** The Weighted Spearman rank coefficient shows a flaw in our biomolecular annotation prediction context: all elements  $\{i : i \notin l_a \vee i \notin l_b\}$  are weighted equally.

As an example, let  $l_a$  ( $l_b$ ) be a ranked list containing an **APlist**  $l_a^{\text{AP}}$  ( $l_b^{\text{AP}}$ ) and a **NAClist**  $l_a^{\text{NAC}}$  ( $l_b^{\text{NAC}}$ ), and let  $a'$  and  $a''$  be two biomolecular annotations. If an annotation is not present in  $l_a^{\text{AP}}$  ( $l_b^{\text{AP}}$ ), then it is likely present in the related **NAClist**  $l_a^{\text{NAC}}$  ( $l_b^{\text{NAC}}$ ) (see Section 2 for details). However, the  $\rho_w$  coefficient would

assign both lists the annotations  $\{a' : a' \notin l_a^{\text{AP}}, a' \in l_a^{\text{NAC}}\}$  and  $\{a'' : a'' \notin l_a^{\text{AP}}, a'' \notin l_a^{\text{NAC}}\}$  a maximum (equal) penalty (i.e.,  $w_{a'} = w_{a''} = 1.0$ ). The same holds for  $l_b$ . Thus, we designed a new coefficient more well-suited to our domain, where  $a'$  gets a lower penalty than  $a''$ . To do so, we first modified the position weight of each element  $i$  in the **NAClist**:

$$\hat{z}_i = z_i + 2 \cdot m \quad (5)$$

where  $z_i$  is the position of  $i$  in the **NAClist**  $l_b^{\text{NAC}}$ ,  $m$  is the length of the associated **APlist**  $l_b^{\text{AP}}$  and 2 is a penalty factor for  $i$  not to be in  $l_b^{\text{AP}}$  and being in  $l_b^{\text{NAC}}$  (the value 2 keeps the penalty proportional to the position of  $i$  in the list). Then, we expressed the new penalty weight as follows:

$$v_{si} = \begin{cases} 1, & i \in l_a^{\text{AP}} \notin l_b^{\text{AP}} \notin l_b^{\text{NAC}} \\ 1 - \frac{1}{|x_i - \hat{z}_i| + 1}, & i \in l_a^{\text{AP}} \notin l_b^{\text{AP}} \in l_b^{\text{NAC}} \\ 1 - \frac{1}{|x_i - y_i| + 1}, & i \in l_a^{\text{AP}} \in l_b^{\text{AP}} \end{cases} \quad (6)$$

where  $x_i$  is the position of the  $i$  element in  $l_a^{\text{AP}}$ ,  $y_i$  is its position in  $l_b^{\text{AP}}$  and  $\hat{z}_i$  is its position in  $l_b^{\text{NAC}}$  ( $l_b^{\text{NAC}}$ ).

We selected this function to reduce the penalties of those elements found in both the first AP lists ( $l_a^{\text{AP}}$ ) and the second NAC lists ( $l_b^{\text{NAC}}$ ), with respect to those only found in the first AP list ( $l_a^{\text{AP}}$ ) but absent from the second one.

The Extended Spearman rank coefficient is thus computed as:

$$\rho_e = \frac{\sum_{i=1}^q v_{si}}{q} \quad (7)$$

As for the Weighted Spearman rank coefficient, high  $\rho_e$  values lead to low correlation, while  $\rho_e \simeq 0$  suggests high correlation.

### 3.2 Kendall Rank Distance

The *Kendall rank distance* ( $\tau$ ) [21] counts the normalized number of pairwise disagreements between two ranked lists  $l_a$  and  $l_b$ , i.e., the number of bubble-sort swaps needed to sort  $l_a$  in the same order of  $l_b$ . Obviously, when the two lists are identical,  $\tau = 0$ . Conversely, if  $l_a$  is obtained by reversing the order of  $l_b$ , then  $\tau = 1$ . Let  $l_a$  and  $l_b$  be two lists of length  $n$  containing the same elements; given an element  $i$ ,  $x_i$  is its position in  $l_a$ , while  $y_i$  is its position in  $l_b$ . Thus, the set  $\mathcal{K}$  of required swaps between elements in lists  $l_a$  and  $l_b$  is computed as follows:

$$\mathcal{K}(l_a, l_b) = \{(i, j) : (x_i < y_i \wedge x_j > y_j) \vee (x_i > y_i \wedge x_j < y_j)\} \quad (8)$$

The normalized Kendall rank distance is given by:

$$\tau = \frac{|\mathcal{K}(l_a, l_b)|}{n(n-1)/2} \quad (9)$$

Notice that the Kendall rank distance does not express negative correlation between lists. Moreover, while the Spearman rank coefficient is focused on the

**Table 2.** Example of the application of the Kendall rank correlation metrics. For each pair of elements in the set (1<sup>st</sup> column), the ranks in  $l_a$  and  $l_b$  are provided (2<sup>nd</sup> and 3<sup>rd</sup> columns), along with the necessity of performing a bubble-sort swap (4<sup>th</sup> column).

Pair	$l_a$ ranks	$l_b$ ranks	Bubble-sort swap	Pair	$l_a$ ranks	$l_b$ ranks	Bubble-sort swap
(a, b)	1 < 2	4 > 2	✓	(b, d)	2 < 4	2 < 5	
(a, c)	1 < 3	4 > 1	✓	(b, e)	2 < 5	2 < 3	
(a, d)	1 < 4	4 < 5		(c, d)	3 < 4	1 < 5	
(a, e)	1 < 5	4 > 3	✓	(c, e)	3 < 5	1 < 3	
(b, c)	2 < 3	2 > 1	✓	(d, e)	4 < 5	5 > 3	✓

distance between the ranks of each element in the lists, the Kendall rank distance considers only the number of swaps in the element rank.

For example, consider the lists  $l_a$  and  $l_b$  in Table 1, whose rankings are summarized in Table 2. The number of bubble-sort swaps needed to give  $l_a$  and  $l_b$  the same ranking is  $|\mathcal{K}(l_a, l_b)| = 5$ . Thus, the Kendall rank distance between  $l_a$  and  $l_b$  is:  $\tau = \frac{5}{(5 \cdot 4)/2} = \frac{5}{10} = 0.5$ , i.e., the lists have medium correlation. This is discordant with the result obtained by applying the Spearman coefficient (see Section 3.1), which states that the lists have low correlation. Thus, the proposed example highlights the different nature of the two metrics.

**Weighted Kendall Rank Distance.** Like the Spearman rank coefficient (Subsection 3.1), a flaw of the classical normalized Kendall rank distance is that it works properly only when the two lists have the same size and contain the same elements. Analogous to what we did for the Spearman rank coefficient, we introduce weights to penalize elements that are present in one list, but absent from the other [27]. In this case, penalties are added so as to give more weight to bubble-sort swaps concerning elements in earlier positions and less weight to the those in later positions. Moreover, elements that appear only in one list are penalized further .

Let  $x_i$  and  $y_i$  be the positions of the element  $i$  in the lists  $l_a$  and  $l_b$ , respectively. In case  $i \notin l_a$ ,  $x_i = |l_a| + 1$ ; else if  $i \notin l_b$ , then  $y_i = |l_b| + 1$ . Then, the penalty for  $i$  is computed as:

$$w_{ki} = \begin{cases} \frac{1}{\log(x_i+2)} - \frac{1}{\log(x_i+3)} & i \in l_a \wedge i \in l_b \\ 0.5 & \text{otherwise} \end{cases} \quad (10)$$

We chose this weight function to make the weight larger if the element is in the earliest positions of the list and lower if it is in the later ones; we consider bubble-sort swaps in the first positions more important. The Weighted Kendall rank distance is then computed as:

$$\tau_w = \frac{\sum_{(i,j) \in \mathcal{K}(l_a, l_b)} w_{ki} w_{kj}}{|\mathcal{K}(l_a, l_b)|} \quad (11)$$

Thus, low correlations correspond to high distance values. On the other hand, if the two lists are identical,  $\tau_w = 0$  (since no swaps occur).

**Extended Kendall Rank Distance.** Similar to what we did with the Extended Spearman rank coefficient (in Subsection 3.1), we introduce the Extended Kendall rank distance to further handle cases where an element  $i$  may be absent from an **APlist**, but present in its **NAClist**. If such an element  $i$  has a likelihood value  $h$  (that is,  $\tilde{a}_{ij}$  is in the output matrix and in the **APlist**), we set its weight to  $v_{ki} = 0.5 - h$ . Notice that annotations in the **APlist** have likelihood values ranked from the maximum to the minimum, in the interval  $[1, \theta)$ , while annotations in the **NAClist** are in the interval  $[\theta, 0]$ , i.e., the prediction likelihood real value  $h$  decreases along the rank. Accordingly, it is used to define the weight for the element  $i$  as follows:

$$v_{ki} = \begin{cases} 0.5 & i \in l_a^{\text{AP}} \notin l_b^{\text{AP}} \notin l_b^{\text{NAC}} \\ 0.5 - h & i \in l_a^{\text{AP}} \notin l_b^{\text{AP}} \in l_b^{\text{NAC}} \\ \frac{1}{\log(x_i+2)} - \frac{1}{\log(x_i+3)} & i \in l_a^{\text{AP}} \in l_b^{\text{AP}} \end{cases} \quad (12)$$

where  $x_i$  is the position of the  $i$  element in  $l_a^{\text{AP}}$ . In particular, the penalty reduction  $h$  can be defined as:  $h = 0.5 - z_i/(m \cdot 2)$ , where  $z_i$  is the position of  $i$  in  $l_a^{\text{NAC}}$  ( $l_b^{\text{NAC}}$ ) and  $m$  is the length of  $l_a^{\text{NAC}}$  ( $l_b^{\text{NAC}}$ ).

We selected the weight function in Equation 12 such that it decreases when the element gets a lower rank. Consequently:

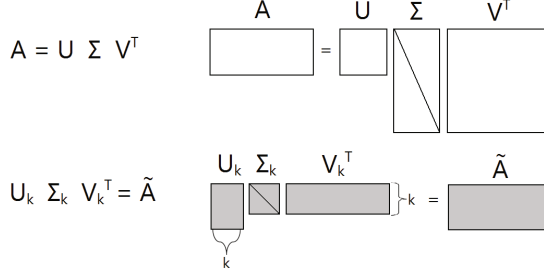
$$\tau_e = \sum_{(i,j) \in \mathcal{K}(l_a, l_b)} v_{ki} v_{kj} \quad (13)$$

As for the Weighted Kendall rank coefficient (in Subsection 3.2),  $\tau_e$  is high when  $l_a$  and  $l_b$  are very different and  $\tau_e \simeq 0$  when the two lists are very similar.

## 4 Results and Discussion

In this Section we evaluate our proposed Extended Spearman rank coefficient and Extended Kendall rank distance measures on lists of annotations produced via the common truncated SVD method, explained in Figure 1. Specifically, we measure the ir prediction performance, while varying the SVD truncation level  $k$ . The measures are tested on a small dataset of GO Cellular Component (CC) and Molecular Function (MF) annotations of *Homo sapiens*, *Gallus gallus* (red junglefowl) and *Bos taurus* (cattle) genes, which were obtained from the Genomic and Proteomic Data Warehouse (GPDW) [29] (*H. sapiens* CC: number of genes: 7,868; number of CC feature terms: 684; number of annotations: 14,381. *G. gallus* MF: number of genes: 309; number of MF feature terms: 225; number of annotations: 509. *B. taurus* MF: number of genes: 543; number of MF feature terms: 422; number of annotations: 934).

Table 3a shows the quantitative amounts of **AP** and **NAC** annotations for different truncation levels  $k$ , provided by our best truncation level algorithm (described in [6]). In the case of *Homo sapiens* CC, the best chosen value is  $k = 378$ ;



**Fig. 1.** An illustration of the Singular Value Decomposition (upper white image) and the Truncated SVD reconstruction (lower gray image) of the  $A$  matrix. In the classical SVD decomposition,  $A \in \{0, 1\}^{m \times n}$ ,  $U \in \mathbb{R}^{m \times m}$ ,  $\Sigma \in \mathbb{R}^{m \times n}$ ,  $V^T \in \mathbb{R}^{n \times n}$ . In the Truncated decomposition, where  $k \in \mathbb{N}$  is the truncation level,  $U_k \in \mathbb{R}^{m \times k}$ ,  $\Sigma_k \in \mathbb{R}^{k \times k}$ ,  $V_k^T \in \mathbb{R}^{k \times n}$ , and the output matrix  $\tilde{A} \in \mathbb{R}^{m \times n}$ . The choice of  $k$  strongly influences the content of the output  $\tilde{A}$  matrix, during reconstruction, even if it does not change the matrix dimensions

the application of the tSVD algorithm with this truncation level produces the *List0*. The values in Table 3a show that a fairly small change in the truncation level  $k$  may lead to different numbers of APs and NACs.

Table 4 shows the values of the Extended Kendall rank distances and the Extended Spearman rank coefficients between the nine lists in Table 3a. By analyzing the Extended Spearman rank coefficients (gray cells), we do not notice any interesting trends or patterns, since the values seem to vary stochastically. Conversely, the Extended Kendall rank distance values increase as the list difference increases monotonically, getting high values for almost all the list pairs, except for the pairs:  $\langle \text{List0}, \text{List1} \rangle$ ,  $\langle \text{List0}, \text{List2} \rangle$  and  $\langle \text{List1}, \text{List2} \rangle$ .

The analysis of the Spearman  $\rho$  metrics, whose computation is based on the difference between list element ranks, suggests that the ROC AUC size does not directly influence the rank dissimilarity in the list. On the contrary, the Kendall  $\tau$  values, whose computation is based on the number of bubble-sort swaps needed to make two lists identical, increase as the AUC difference increases; their analysis suggests the existence of a relationship between the ROC AUC values and the rankings of the APs in the lists from the tSVD generating high AUC percentages and very low AP numbers.

#### 4.1 SVD Truncation Patterns

Another interesting result is revealed by sorting the lists on the basis of the truncation level  $k$ , as done in Table 5. While the Extended Spearman rank coefficients give no additional clues on specific trends, the Extended Kendall rank distances show that the higher the SVD truncation level difference between two lists is, and the less similar the lists are. Apart from the comparison between  $\langle \text{List0}, \text{List1} \rangle$ , all other lists show value trends that increase when the distance between truncation levels increase. *List8*, for example, shows near maximum dissimilarity with all the other lists.

**Table 3.** Table 3a shows the number of **AP** and **NAC** annotations for *H. sapiens* CC when varying the SVD truncation level  $k$ , and their corresponding **ROC AUC** percentage. The likelihood threshold is fixed at  $\theta = 0.49$ . Table 3b Numbers of **AP** and **NAC** annotations for *G. gallus* MF and *B. taurus* MF when varying the SVD truncation level  $k$ , and their corresponding **ROC AUC** percentage. The likelihood threshold is fixed at  $\theta = 0.50$

SVD when varying truncation $k$					SVD when varying truncation $k$				
	$k$	AP	NAC	ROC AUC		$k$	AP	NAC	ROC AUC
<i>Homo sapiens</i> CC					<i>Gallus gallus</i> MF				
List0	<b>378</b>	8	4,458,751	83.49%	List0	<b>40</b>	9	39,340	75.38%
List1	<b>402</b>	2	4,458,757	53.64%	List1	<b>53</b>	5	39,344	74.33%
List2	<b>390</b>	7	4,458,752	53.58%	List2	<b>27</b>	10	39,339	73.69%
List3	<b>291</b>	19	4,458,740	53.14%	List3	<b>14</b>	11	39,338	67.25%
List4	<b>349</b>	8	4,458,751	52.97%	List4	<b>1</b>	164	39,185	35.98%
List5	<b>233</b>	48	4,458,711	51.82%	<i>Bos taurus</i> MF				
List6	<b>175</b>	78	4,458,681	48.80%	List0	<b>70</b>	11	120,318	74.74%
List7	<b>117</b>	86	4,458,673	45.11%	List1	<b>93</b>	8	120,321	74.59%
List8	<b>59</b>	95	4,458,664	38.82%	List2	<b>47</b>	11	120,318	73.57%
					List3	<b>24</b>	32	120,297	68.59%
					List4	<b>1</b>	369	119,960	35.21%

**Table 4.** Extended Spearman rank coefficient values (gray cells) and Extended Kendall rank distance values (white cells) from the comparison of the nine annotation lists in Table 3a, generated by the truncated SVD method applied to the *Homo sapiens* CC dataset with varying truncation level  $k$ . Intervals: **MaxCorrelation** = 0; **MinCorrelation** = 1. All lists are ordered from the one corresponding to the maximum **ROC AUC percentage** (*List0*, **AUC** = 83.49%) to the one corresponding to the minimum **AUC** (*List8*, **AUC** = 38.82%).

Extended Spearman values										
		List0	List1	List2	List3	List4	List5	List6	List7	List8
E.	List0		0.129	0.07	0.301	0.224	0.593	0.841	0.443	0.501
	List1	0.621		0.01	0.535	0.512	0.708	0.838	0.585	0.539
K.	List2	0.491	0.558		0.324	0.245	0.656	0.872	0.499	0.509
	List3	0.984	0.962	0.978		0.57	0.158	0.362	0.249	0.772
e.	List4	0.868	0.894	0.913	0.935		0.416	0.302	0.306	0.885
	List5	0.995	0.995	0.995	0.952	0.984		0.283	0.069	0.805
n.	List6	0.999	0.999	0.999	0.988	0.998	0.936		0.427	0.628
	List7	0.996	0.998	0.997	0.997	0.998	0.990	0.946		0.522
d.	List8	0.998	0.998	0.999	0.999	0.999	0.996	0.984	0.982	

**Table 5.** Extended Spearman rank coefficient values (gray cells) and Extended Kendall rank distance values (white cells) from the comparison of the nine annotation lists in Table 3a, generated by the truncated SVD method applied to the *Homo sapiens* CC dataset with varying truncation level  $k$ . Intervals: **MaxCorrelation** = 0; **MinCorrelation** = 1. All lists are ordered from the one generated with the greatest **SVD truncation level** (*List1*,  $k = 402$ ) to the one generated with the lowest level (*List8*,  $k = 59$ ).

		Extended Spearman values								
		List1	List0	List2	List4	List3	List5	List6	List7	List8
E. K e n d a l	List1		0.129	0.01	0.512	0.535	0.708	0.838	0.585	0.539
	List0	0.621		0.07	0.224	0.301	0.593	0.841	0.443	0.501
	List2	0.558	0.491		0.245	0.324	0.656	0.872	0.499	0.509
	List4	0.894	0.868	0.913		0.935	0.416	0.302	0.306	0.885
	List3	0.962	0.984	0.978	0.57		0.158	0.362	0.249	0.772
	List5	0.995	0.995	0.995	0.984	0.952		0.283	0.069	0.805
	List6	0.999	0.999	0.999	0.998	0.988	0.936		0.427	0.628
	List7	0.998	0.996	0.997	0.998	0.997	0.99	0.946		0.522
	List8	0.998	0.998	0.999	0.999	0.999	0.996	0.984	0.982	

## 4.2 ROC AUC Patterns

Beyond its importance in providing new knowledge about the variation of the annotation lists, varying the SVD truncation level  $k$  has no utility for our validation process. Conversely, interesting trends can be found by comparing the ROC AUC percentages, the Extended Kendall rank distance and the Extended Spearman rank coefficient. As general examples, we show the cases of the *Bos taurus* MF and *G. gallus* MF datasets, in Table 3b and Table 6. As one may notice, these lists corresponding to similarly AUCs (*List0*, *List1*, *List2*), have similar low Extended Spearman rank coefficients. This means that the elements present in these lists have similar rankings.

Since there is a correlation between ROC AUC and list similarity, this may be helpful in finding the best predicted annotations. In fact, our prediction methods are able to produce ROC with maximum AUC, and the AUCs have an oscillatory trend. Since these oscillations produce low changes to the ROC AUC percentages, these will slightly influence the final prediction results. Thus, our methods, based on the optimization of the ROC AUCs, are quite robust. There is no need to find the overall best SVD truncation since it is sufficient to find truncation values that are close to the best ones; the algorithm can then select the best predicted annotations accordingly.

**Table 6.** Extended Spearman rank coefficient values (gray cells) and Extended Kendall rank distance values (white cells) from the comparison of the annotation lists in Table 3b, generated by the truncated SVD method applied to the *G. gallus* MF dataset with varying truncation level  $k$ . Intervals: **MaxCorrelation** = 0; **MinCorrelation** = 1. All lists are ordered from the one generated with the greatest SVD truncation level  $k$  to the one generated with the lowest level. We report in **bold** the cases showing low Spearman or Kendall values for lists with similar AUCs.

		Extended Spearman values					Extended Spearman values				
		List0	List1	List2	List3	List4	List0	List1	List2	List3	List4
		<i>Gallus gallus</i> MF					<i>Bos taurus</i> MF				
E x t. K.	List0		<b>0.370</b>	<b>0.200</b>	0.620	0.85		<b>0.192</b>	<b>0.166</b>	0.364	0.794
	List1	0.790		<b>0.330</b>	0.780	0.94	0.687		<b>0.279</b>	0.425	0.888
	List2	0.640	0.790		0.180	0.74	0.812	0.822		0.128	0.772
	List3	0.980	0.960	0.950		0.78	0.980	0.980	0.950		0.773
	List4	1.000	1.000	1.000	1.000		0.999	0.999	0.999	1.000	

## 5 Conclusions

Both our Extended Spearman rank coefficient and Extended Kendall rank distance measures resulted effective and useful to compute the level of “similarity” between two gene annotation lists, by focusing on either the list element position difference (Spearman) or on the number of list elements having different rankings (Kendall). Using them, we discovered a negative correlation between two generated annotation lists: the more the truncation level difference increases, the more dissimilar (in terms of bubble-sort swaps needed to make them identical) the annotation lists are. We also observed a positive correlation between the ROC AUCs and the similarity between two lists: the closer two AUC percentages are, the more similar the related predicted annotation lists are. In this case the similarity is expressed through the Extended Kendall rank distance, which measures the difference between ranked position of an element present in both analyzed lists.

In general, we can state that the Spearman coefficient is the most useful one when the user wants to take advantage of the global order of the items in the two compared lists; on the contrary, the Kendall distance is the best choice when the user wants to highlight the relative raking among items in the lists.

In the future, we plan to apply our metrics to annotation lists produced through different algorithms and study their variation when changing algorithm’s key parameter values. For example, we will explore the similarity between annotation lists from the topic modeling algorithms (pLSA [15] and LDA [16]) when the number of topics changes.

**Acknowledgments.** This work is partially funded by the CUBRIK project ([www.CubrikProject.eu](http://www.CubrikProject.eu)).

## References

- [1] Karp, P.D.: What we do not know about sequence analysis and sequence databases. *Bioinformatics* 14(9), 753–754 (1998)
- [2] Pandey, G., Kumar, V., Steinbach, M.: Computational approaches for protein function prediction: A survey. Twin Cities: Department of Computer Science and Engineering, University of Minnesota (2006)
- [3] Khatri, P., Done, B., Rao, A., Done, A., Draghici, S.: A semantic analysis of the annotations of the human genome. *Bioinformatics* 21(16), 3416–3421 (2005)
- [4] Golub, G.H., Reinsch, C.: Singular value decomposition and least squares solutions. *Numerische Mathematik* 14(5), 403–420 (1970)
- [5] Consortium, G.O., et al.: Creating the gene ontology resource: design and implementation. *Genome Research* 11(8), 1425–1433 (2001)
- [6] Chicco, D., Masseroli, M.: A discrete optimization approach for svd best truncation choice based on roc curves. In: 2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 1–4. IEEE (2013)
- [7] Drineas, P., Frieze, A., Kannan, R., Vempala, S., Vinay, V.: Clustering large graphs via the singular value decomposition. *Machine Learning* 56(1-3), 9–33 (2004)
- [8] Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007* (1995)
- [9] Chicco, D., Tagliasacchi, M., Masseroli, M.: Genomic annotation prediction based on integrated information. In: Biganzoli, E., Vellido, A., Ambrogi, F., Tagliaferri, R. (eds.) CIBB 2011. LNCS, vol. 7548, pp. 238–252. Springer, Heidelberg (2012)
- [10] Done, B., Khatri, P., Done, A., Draghici, S.: Semantic analysis of genome annotations using weighting schemes. In: IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology, CIBCB 2007, pp. 212–218. IET (2007)
- [11] Done, B., Khatri, P., Done, A., Draghici, S.: Predicting novel human gene ontology annotations using semantic analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 7(1), 91–99 (2010)
- [12] Pinoli, P., Chicco, D., Masseroli, M.: Enhanced probabilistic latent semantic analysis with weighting schemes to predict genomic annotations. In: 2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 1–4. IEEE (2013)
- [13] Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM (1999)
- [14] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. *the Journal of Machine Learning Research* 3, 993–1022 (2003)
- [15] Masseroli, M., Chicco, D., Pinoli, P.: Probabilistic latent semantic analysis for prediction of gene ontology annotations. In: The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2012)
- [16] Pinoli, P., Chicco, D., Masseroli, M.: Latent dirichlet allocation based on gibbs sampling for gene function prediction. In: 2014 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, pp. 1–8. IEEE (2014)
- [17] Chicco, D., Sadowski, P., Baldi, P.: Deep autoencoder neural networks for gene ontology annotation predictions. In: Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics, pp. 533–540. ACM (2014)

- [18] Goodman, L.A., Kruskal, W.H.: Measures of association for cross classifications\*. *Journal of the American Statistical Association* 49(268), 732–764 (1954)
- [19] Fagin, R., Kumar, R., Sivakumar, D.: Comparing top k lists. *SIAM Journal on Discrete Mathematics* 17(1), 134–160 (2003)
- [20] Spearman, C.: The proof and measurement of association between two things. *The American Journal of Psychology* 15(1), 72–101 (1904)
- [21] Kendall, M.G.: A new measure of rank correlation. *Biometrika*, 81–93 (1938)
- [22] Ilyas, I.F., Beskales, G., Soliman, M.A.: A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys (CSUR)* 40(4), 11 (2008)
- [23] Kumar, R., Vassilvitskii, S.: Generalized distances between rankings. In: *Proceedings of the 19th International Conference on World Wide Web*, pp. 571–580. ACM (2010)
- [24] Bertin-Mahieux, T., Eck, D., Maillet, F., Lamere, P.: Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research* 37(2), 115–135 (2008)
- [25] Chen, Q., Aickelin, U.: Movie recommendation systems using an artificial immune system. *arXiv preprint arXiv:0801.4287* (2008)
- [26] Payne, J.S., Stonham, T.J.: Can texture and image content retrieval methods match human perception?. In: *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 154–157. IEEE (2001)
- [27] Ciceri, E., Fraternali, P., Martinenghi, D., Tagliasacchi, M.: Crowdsourcing for Top-K Query Processing over Uncertain Data. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 1–14 (preprint, 2015)
- [28] Fawcett, T.: Roc graphs: Notes and practical considerations for researchers. *Machine Learning* 31, 1–38 (2004)
- [29] Canakoglu, A., Masseroli, M., Ceri, S., Tettamanti, L., Ghisalberti, G., Campi, A.: Integrative warehousing of biomolecular information to support complex multi-topic queries for biomedical knowledge discovery. In: *2013 IEEE 13th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 1–4. IEEE (2013)

Computational Intelligence Methods for Bioinformatics  
and Biostatistics

11th International Meeting, CIBB 2014, Cambridge, UK,

June 26-28, 2014, Revised Selected Papers

di Serio, C.; Liò, P.; Nonis, A.; Tagliaferri, R. (Eds.)

2015, XIII, 314 p. 90 illus. in color., Softcover

ISBN: 978-3-319-24461-7