On functional annotation with gene co-expression networks

1st Vladimír Kunc
Department of Computer Science
Czech Technical University in Prague, FEE
Prague, Czech Republic
kuncvlad@fel.cvut.cz

2nd Jiří Kléma
Department of Computer Science
Czech Technical University in Prague, FEE
Prague, Czech Republic
klema@fel.cvut.cz

***Abstract*—Gene co-expression networks have frequently been used for functional annotation. In these networks, an unknown gene is annotated with terms that have already been associated with genes whose expression profiles tend to correlate with the expression profile of the unknown gene. Despite the biological plausibility of this principle referred to as guilt-by-association, its applicability has not been thoroughly experimentally verified yet. In our paper, we formulate several statistical hypotheses concerning the principle and test them on a representative expression dataset. We demonstrate that gene annotation carried out with co-expression networks clearly outperforms random annotation and improves with increasing sample size and the knowledge of gene co-location. Eventually, we discuss the practical significance of this way of functional annotation.**

***Index Terms*—gene expression, co-expression network, functional annotation, guilt-by-association principle**

I. Introduction

Correlation networks [1], [2] have proven to be useful for analysis of large and high-dimensional biological data sets [3], [4]. In a correlation network, each node represents a variable (entity), and links represent correlations between the variables. In general, the networks are used to address a couple of issues. Most often, they can help to identify clusters of densely interconnected nodes. These clusters often correspond to important functional units of the system represented by the network [5].

Further, other structural network characteristics such as highly connected hub nodes could be found. These hubs may serve as treatment targets, for example [6]. Last but not least, the networks can serve to answer predictive tasks such as node annotation, or link prediction [7]–[9].

Our paper focuses on a special kind of biological correlation network called co-expression networks. In these networks, the individual variables correspond to expression profiles. Co-expression networks represent a major application of correlation network methodology [3]. Gene co-expression networks have frequently been used to explore the system-level functionality of genes [10]. Later, co-expression has also been applied to interpret non-coding RNA (ncRNA) expression data [11]. When building a co-expression network, one has to make a few key decisions. A measure of association or correlation between two expression profiles has to be selected [12]. The network could either be binary or weighted [13]. In the first case, there is a correlation threshold, a pair of variables is linked if and only if their absolute correlation exceeds this threshold, and all the existing links have unit weight [14], [15]. In the second case, the network graph is full, and the links are most often weighted by the magnitude of absolute correlation [3], [10].

One of the most important goals in co-expression network analysis is annotating previously unknown genes, and ncRNAs [4], [13], [16]–[18]. The main principle that drives this annotation is the guilt-by-association (GBA) principle [19]. An unknown gene or ncRNA is annotated with terms that have already been associated with protein-coding mRNAs and other ncRNAs whose expression profiles tend to correlate with the profile under examination [20], [21]. In this way, expensive and time-demanding experimental functional verification carried out in vitro can be supplemented by cheaper and faster computational predictive annotation performed in silico. An example of a simple co-expression network demonstrating the principle is in Figure 1.

Currently, there are several publicly available tools for gene co-expression network analysis and function annotation. These tools could be considered as a methodological standard in

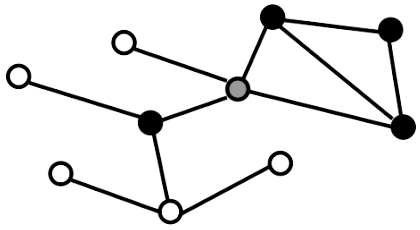


Fig. 1: A binary co-expression network. The individual nodes correspond to genes; edges connect correlated genes. The genes annotated with a term of interest are in black; the genes in white are known to be without this annotation. A gene whose annotation is unknown is in grey. The goal is to decide whether the gene should be annotated with the term of interest. According to the GBA principle, the gene will likely be linked with the term.

This study was supported by the AZV CR (the grant NU20-03-00412) and the Research Center for Informatics (CZ.02.1.01/0.0/0.0/16_019/0000765).
978-1-6654-6819-0/22/$31.00 ©2022 IEEE

annotation with co-expression networks. WGCNA is a popular R package for gene co-expression network construction and module identification [3]. In WGCNA, functional annotation can be done through the following steps: 1) identification of clusters/modules of highly correlated genes, 2) enrichment analysis of the constructed modules, and 3) annotation of the nodes of interest with respect to how close they are to the identified modules. ncFANs is a platform for the functional annotation of non-coding RNAs [22]. The functions of ncRNAs are proposed using hub- and module-based methods described by Liao et al. [23]. The hub-based methods select the directly connected nodes of the hub ncRNAs and assign the enriched functions of these nodes to ncRNAs. The module-based methods work on the same principle as the above-described annotation in WGCNA. GCEN is a cross-platform command-line toolkit to easily build a gene co-expression network and predict gene function recently proposed in [24]. It implements the module-based methods as well as the annotation through a random walk with restart (RWR) proposed in [25]. RWR propagates annotations through the network from seed nodes with the known gene function to nearby nodes until convergence. However, a number of questions arise when compiling the co-expression network and implementing the general GBA principle. It has been pointed out that (gene) function is not systemically encoded in networks but dependent on specific and critical interactions [26]. It has also been observed that the fraction of genes in a module that relate to its main biological function is often $<$20%, and module-trait correlations can be relatively low (correlation < 0.5) even when statistically significant [13]. It has been shown that there is an influence of the physical distance between genes and the degree of their co-expression in breast cancer [27]. On the other hand, in healthy breast tissue, gene co-expression was independent of the chromosome location of genes [27].

The notion of the relevant neighborhood of the annotated entity is therefore not trivial at all. It is crucial to find a suitable size for the neighborhood and incorporate additional knowledge in its construction. For this reason, we decided to carry out a comprehensive experimental study that verifies the applicability of the GBA principle in gene co-expression networks and suggests a suitable method for its practical application. We implement and test several different ways of gene annotation in a large and representative co-expression network that stems from a large gene expression dataset. The subsequent statistical evaluation helps to assess whether the gene annotation with the aid of co-expression networks has both statistical and practical significance. To the best of our knowledge, co-expression networks have frequently been used for gene and ncRNA annotation; however, their applicability has not been thoroughly verified yet. The existing reviews on co-expression networks [4], [28] rather focus on methodological comparison of the individual approaches than their experimental verification.

The paper is structured as follows. The next section presents our study's overall design, including the formulation of fundamental statistical hypotheses. Further, the annotation algorithm based on over-representation analysis is described. The evaluation section gives the quality measures used in our ablation study, including the comparison of our algorithm with a random benchmark. Eventually, the results are evaluated and summarized in a conclusion.

II. The overall design and goals of the study

This paper will experimentally study how far we can use co-expression to predict gene function. In particular, we will deal with a human genome and pathway annotation terms available in Kyoto Encyclopedia of Genes and Genomes (KEGG) [29], [30].

Co-expression will be calculated from a large expression set from the Affymetrix microarray platform curated by the Broad Institute. The dataset was used originally for learning gene expression model using an artificial neural network called D–GEX [31], and the networks with transformative adaptive activation functions [32]. The main advantage of this expression dataset is its representative size. It consists of 111,009 biological samples; each sample comprises 22,268 probes. Another advantage is the uniform measurement platform used for all the samples. The gene location information was obtained from the Ensemble database [33].

We will construct a co-expression network for all the genes whose profiles are available in the expression dataset and have at least one KEGG annotation resulting in 3,305 genes with available annotations (denoted as $\mathcal{G}$). We will also propose and test a family of prediction methods that will give a ranking of KEGG annotations for a gene. Then, the methods will be applied for every single gene (symbol $\in \mathcal{G}$), and they will construct a ranking of all the KEGG annotations ($\mathcal{A}$=KEGG annotations) for each of the genes. The more related a term is to the gene, the higher it appears in the ranking. Next, the predicted rankings will be compared with the actual gene annotation; the higher the true gene annotations appear in the rankings, the better the prediction. Eventually, the methods will be statistically evaluated, and the best ranking method will be reported and discussed further.

The goal of this paper is to illustrate the performance of the discussed method using known annotations. Therefore, it is assumed that we know the annotations for each gene except for the target, whose annotations are inferred, in order to evaluate the performance of a method.

In particular, we will test several statistical hypotheses (only alternative hypotheses are mentioned):

1) The GBA principle can be used for gene annotation with KEGG pathway terms. The ranking of KEGG terms delivered by our algorithm that implements this principle is closer to the true annotation than a random ranking.
2) Increased sample size makes the estimation of correlation more robust and improves the gene annotations.
3) The knowledge of gene co-location improves the gene annotation (co-located genes are considered related in the co-expression network independently of correlation in their expression profiles).

4) The treatment of correlation thresholds in the algorithm matters. Multiple thresholds make the annotation prediction more precise.

III. Annotation algorithm

The prediction will be based on over-representation analysis [34]. The importance of every single KEGG term will be evaluated through its over-representation in the annotations within the set of genes that most correlate with the target gene. This over-representation will be statistically evaluated with the aid of Fisher's exact test; the smaller the p-value of the test, the better the term ranking.

The proposed method creates a ranking of candidate terms for each gene. This ranking is based on the correlations of the gene with other genes for which the terms are known. We have used two ranking approaches; one is based mainly on a single Fisher's test for each term, while the other uses multiple Fisher's tests for each term.

In the first approach, the correlation of the target gene with each of the genes with known terms is calculated, and we then use the absolute value of the correlations. Since the gene expression might have non-linear relationships between each other, a Kendall correlation coefficient τ variant b [35] has been used to control for monotonic transformation. Then a threshold for splitting the gene set into correlated and uncorrelated genes is determined as a function of the calculated correlations – e.g., a median. These two sets of correlated and uncorrelated genes are then used to create a contingency table for the given term, see Table I.

	Correlated with g	**Uncorrelated with g**	**Total**
Annotated with t	p_{ca}	$p_{\overline{c}a}$	p_a
Not related with t	$p_{c\overline{a}}$	$p_{\overline{c}\overline{a}}$	$p_{\overline{a}}$
Total	p_c	$p_{\overline{c}}$	p

TABLE I: A contingency table that quantifies the relationship between a gene g and a KEGG term t. Each entry contains the number of genes that meet the conditions. The total number of genes is p.

This contingency table summarizing the numbers of correlated and uncorrelated genes with and without the term is then tested for statistically significant over- or under-representation using Fisher's exact test. The p-values of the tests for each term for a particular gene are then used to construct the ranking of the terms. The procedure for obtaining the single p-value for a given threshold is depicted in Alg. 1. The ranking method of the first approach is depicted in Alg. 2 and is denoted as the *corr ranker*.

The second method is similar, but it does not select a single threshold over correlations but evaluates multiple thresholds to increase the robustness of the method. For a particular gene and for each term that might be associated with the gene, we threshold the absolute correlations using 200 individual thresholds spaced uniformly over the interval of (0, 0.5). We get 200 contingency tables and then 200 p-values from Fisher's exact tests. We aggregate the p-values for a given term and gene using a function (e.g., a median or mean) to produce a single value that is then used to create rankings of terms for a given gene. The ranking method of the second approach is depicted in Alg. 3. It is denoted as the *p-value ranker*. Several aggregation functions used for either selecting thresholds over correlations or aggregating p-values over multiple thresholds as described above were tested.

For very small samples, it might be beneficial to utilize the gene position information as closely co-located genes are more likely to have a common function [36], [37]. For the purposes of the proposed method, a gene is considered to be co-located with another gene if their mutual distance on a chromosome is less than *n* bps. The co-located genes are then considered to be correlated for the purposes of the methods described above, i.e., they are always added to the set of positively correlated genes used for the calculation of Fisher's exact test.

Algorithm 1 Valuer providing the value used for ranking the symbols.

procedure VALUER(symbol, annotation, thresh) ▷ Provides the value of the given annotation for the symbol for ranking
    correlated_symbols $\leftarrow$ (correlation_matrix[symbol] $\geq$ threshold) $-$ {symbol} $+$ co $-$ located_genes
    return fisher(# uncorrelated with annotation,
                # uncorrelated without annotation,
                # correlated with annotation,
                #correlated without annotation).pvalue ▷ p-value of the Fisher's exact test
end procedure

Algorithm 2 Ranking using a single Fisher's exact test for a single symbol.

procedure RANKER_CORR(symbol, quantile_f)
    values $\leftarrow$ []
    for $a \in \mathcal{A}$ **do**
        $t \leftarrow$ quantile_f(correlation_matrix[a] ▷ threshold as a quantile of the correlation coefficients
        values[a] $\leftarrow$ valuer(symbol, a, t)
    end for
    return rank(values)
end procedure

A. Evaluation

Since the goal of the method is to produce a ranking of candidate annotations for each gene, we evaluate it using the average difference of average ranks of the real annotations of the gene (the known annotations for the gene) and the rest of the candidate annotations. The advantage of this evaluation measure is that it can be used even though individual genes might have different numbers of correct annotations. Furthermore, such a method is beneficial as it closely resembles the

Algorithm 3 Ranking using multiple Fisher's exact tests for a single symbol.

procedure RANKER_PVALS(symbol, quantile_f)
  values ← []
  for $a \in \mathcal{A}$ **do**
    values[a] ← quantile_f([valuer(symbol, a, t) for t $\in \{0.005, 0.01, \ldots, 0.5\}$]
  end for
  return rank(values)
end procedure

work of biologists when considering individual annotations by a prioritized list of possible candidates without the need to a priori set a threshold for selecting the candidates for further research.

Specifically, for each gene g and ranking method m, we define difference of mean ranks $DMR^m(g)$:

$$DMR^m(g) = \frac{1}{|\mathcal{A}_g|} \sum_{a \in \mathcal{A}_g} r_g^m(a) - \frac{1}{|\mathcal{A} \setminus \mathcal{A}_g|} \sum_{a \in \mathcal{A} \setminus \mathcal{A}_g} r_g^m(a) \tag{1}$$

where $\mathcal{A}$ is the set of all available annotations, $\mathcal{A}_g$ are the real annotations of the gene g and $r_g^m(a)$ is the rank of annotation a for the gene produced by method m — all annotations $a \in \mathcal{A}$ are ordered for each gene g and then ranked, i.e., the highest possible rank is $|\mathcal{A}|$. Since we allow for ties in the ranking, the average rank of the tie group is used for individual tied annotation.

The performance of a method is measured as the mean $DMR^m(g)$ over all genes $g \in G$ where G is the target set of genes. Such measure is denoted as $MDMR^m$:

$$MDMR(m) = \frac{1}{|G|} \sum_{g \in G} DMR^m(g) \tag{2}$$

For direct comparison of two candidate methods m_1 and m_2, we use a pairwise measure — mean difference of DMRs denoted as MDDMR:

$$MDDMR(m_1, m_2) = \frac{1}{|G|} \sum_{g \in G} (DMR^{m_1}(g) - DMR^{m_2}(g)) \tag{3}$$

Since subsampled data from a larger set were used, ten samplings for each sample size were obtained to reduce performance differences of individual methods due to the individual samples that were used for correlation calculation. To avoid further complicating the notation, this is not reflected in the formulas above; all presented values were obtained as the mean over the relevant ten subsampling runs.

Also, a notion of a *minimum rank* (MR) was used for evaluation; an MR is the rank of the first correct annotation of a gene:

$$MR^m(g) = \min\{r_g^m(a) | a \in \mathcal{A}_g\}. \tag{4}$$

B. Statistical evaluation

The impact of individual parameters relative to a baseline using $MDDMR$ is tested using Wilcoxon signed-rank test over all available variants with the respective parameters.

IV. Implementation

The whole workflow is summarized in Figure 2. It was implemented in Python 3 using the libraries *SciPy* [38], *pandas* [39]. However, since the repeated computation of the Fisher's exact test is costly, we have used the publicly available crate *fishers_exact* that implements the test in low-level, compiled language Rust which we interfaced from Python as an installable library using *maturin* package. We have also created a small Python module using the stack described above for fast computation of the correlation matrix using Kendall's τ correlation coefficient.

V. Results

Several factors influence the performance of the class of the evaluated methods — sample size used for correlation matrix calculation, usage of the co-location information (and the co-location distance threshold), and the used thresholder — that are analyzed in this paper. We show the influence of each parameter separately by comparing the $MDMRs$ for different parameters and also by showing the relative difference $MDDMR$ of two variants that share the exact same parameters with the exception of the examined parameter.

A. Sample size

Since biological data are notoriously difficult to obtain, much research has to do with only a few tens of samples. At the same time, robustness of correlation estimates strongly depends on the sample size. To show the dependence, we have created a few subsamples of the original dataset — for each sample size in $\{20, 50, 100, 1000, 10000\}$, ten subsamples were created.

As shown in Figure 3a, the performance improves significantly with increasing sample size with slowly diminishing returns. While Figure 3a shows the overall performance,

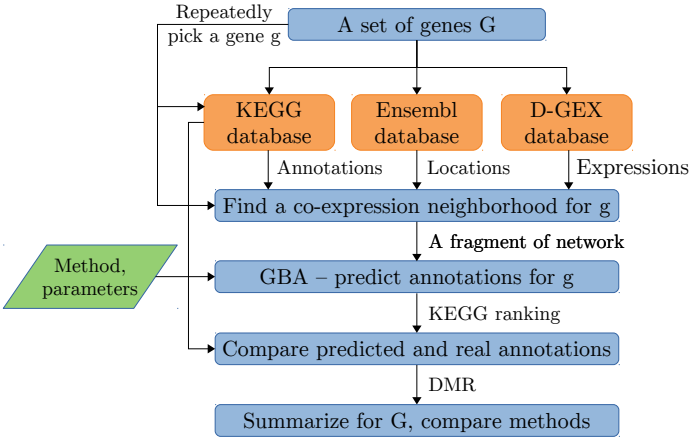
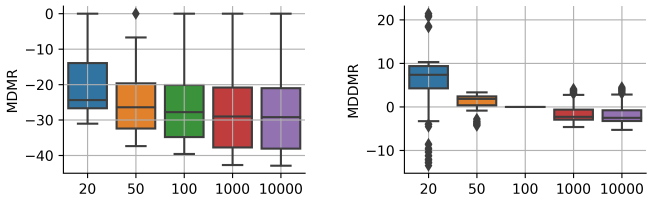


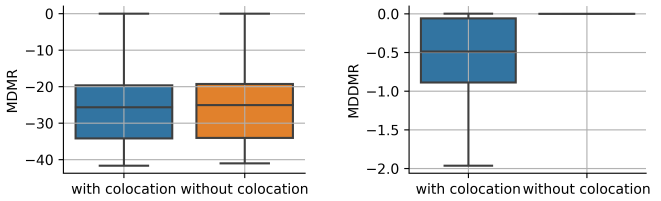
Fig. 2: The flowchart of the proposed framework.



(a) The absolute performance in terms of $MDMR$.

(b) The relative performance of variants that have the same parameters except for the sample size shown relative to the sample size 100.

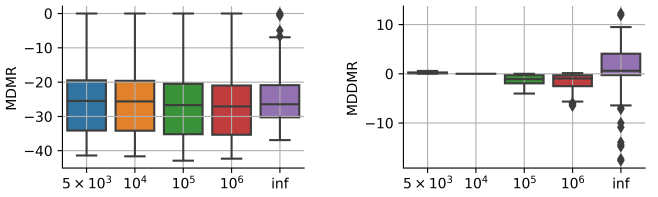
Fig. 3: Performance of all runs broken by the sample size used for correlation matrix calculations.



(a) The absolute performance in terms of $MDMR$.

(b) The relative performance of co-location usage relative to the variant without co-location.

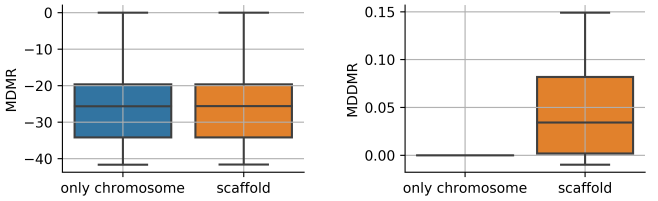
Fig. 4: Co-location usage.



(a) The absolute performance in terms of $MDMR$.

(b) The relative performance to the co-location distance threshold 10,000.

Fig. 5: Co-location distance.



(a) The absolute performance in terms of $MDMR$.

(b) The relative performance of using the most specific chromosome location compared to using the information about a chromosome only.

Fig. 6: Co-location determination type.

Figure 3b compares identical variants directly and thus isolates the effect of the sample size on performance. The difference in sample size was statistically significant (p-value < 0.001) when compared to the baseline sample size of 100 samples, which aligns with the hypothesis that sample size matters.

B. Co-location

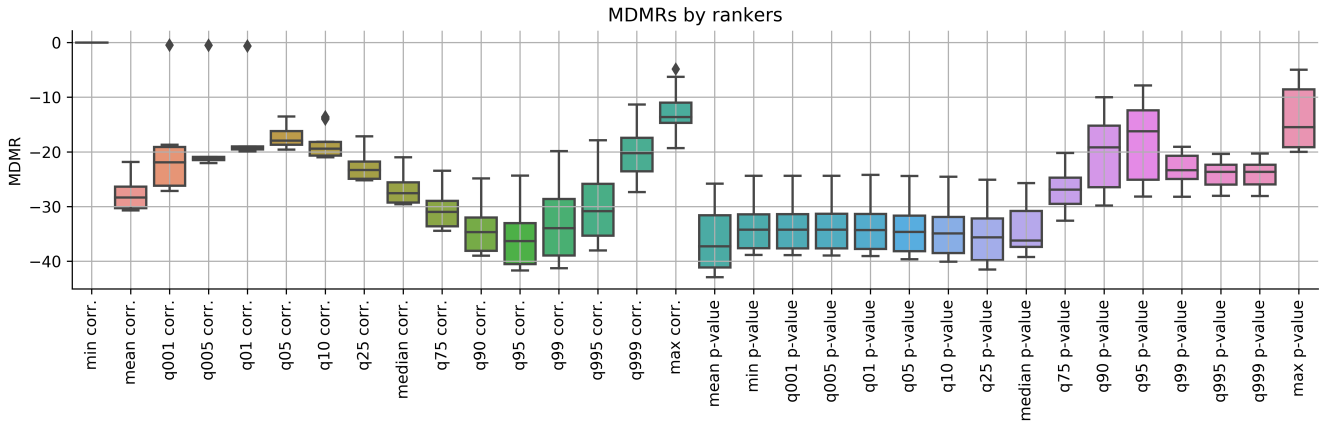
Since closer genes are more likely to have a common function than more distant genes, the co-location of two genes might improve the method's performance. First, we establish the advantages of the co-location usage in Figure 4, where variants with a co-location threshold of 10,000 bps are used, and only the chromosome number is used - i.e., two genes are considered to be co-located if they are on the same chromosome and their position is less than 10,000 bps apart. The effect of the co-location is small but significant (p-value < 0.001).

The distance threshold up to which two genes should be considered co-located is not a priori obvious — a co-location thresholds of 5,000 bps, 10,000 bps, 100,000 bps, and 1,000,000 bps were tested together with the unlimited variant (denoted as the *inf* threshold; genes are considered colocated if they are on the same chromosome) as shown in Figure 5. The methods' performance was the best for the co-location distance threshold of 1,000,000 and this difference was statistically significant when compared to the baseline with a co-location distance threshold of 10,000 (p-value < 0.001) established above. This is in line with our hypothesis that knowledge of gene co-location improves the gene annotation. We explain the high location threshold value by saying that completeness prevails over accuracy (despite decreasing relevance, it is better to consider even distant genes correlated).

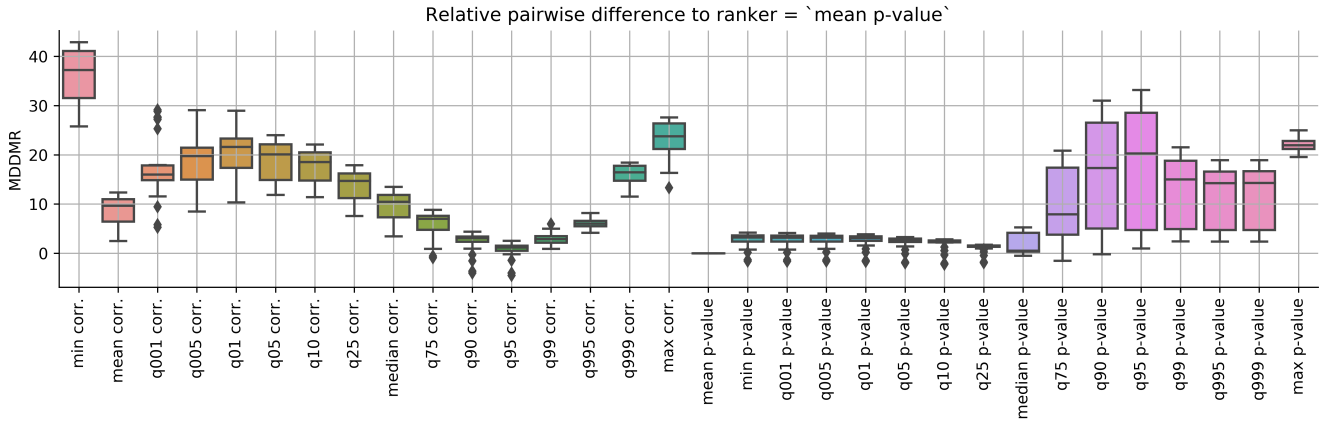
Last but not least, we examine whether there is any advantage to using the scaffold information for location specification if it is available — only genes that share the same scaffold and are within the threshold distance apart are considered to be co-located. Such co-location definition was found to perform marginally worse compared to the default variant as shown in Figure 6 except for the *inf* threshold where the scaffold information effectively limits the threshold and thus, it has a less negative impact on the performance and thus seemingly improves the performance.

C. Rankers

The most important parameter of the presented method is the *ranker* and the aggregation function that produces the final ranking of candidates. For both the *corr* and p-value rankers, several aggregation functions were tested — the mean, minimum, maximum and several percentiles: $\{0.1\%, 0.5\%, 1\%,$ 5%, 10%, 25%, 50%, 75%, 95%, 99%, 99.5%, 99.9%$\}$. Note that the *corr ranker* with a minimum for the aggregation function leads to degenerate results when all function candidates receive the same rank and is only kept for completeness. The most suitable aggregation functions for the *corr* ranker are higher percentiles, with the optimum being at 95%. However,



(a) The absolute performance in terms of $MDMR$.



(b) The relative performance w.r.t the *mean p-value* ranker.

Fig. 7: Ranker and aggregation function variants.

the *p-value* ranker leads to slightly better performance on average with the mean aggregation function as shown in Figure 7b; this performance gain compared to other ranker variants was statistically significant using Wilcoxon signed-rank test with p-value < 0.001 for all comparison which confirms that the treatment of correlation thresholds in the algorithm matters. Also, the *p-value* rankers were generally performing better than *corr* rankers for the same sample sizes and co-location usage as determined by a paired Wilcoxon signed-rank test over the pairs where the best performing variant over available quantile functions for each of the rankers was selected (p-value < 0.001).

Overall ranking

The list of the top ten configurations is shown in Table II. The primary performance measure was the MDMR; the mean minimum rank (MMR) was used as the secondary performance measure. The best overall configuration is the *mean p-value* ranker with the usage of co-location information. The method's performance is less sensitive to the exact value of the co-location distance threshold than the ranker selection; nevertheless, the final recommendation is to use the threshold set to 100,000 bps.

However, this recommendation is not universal and depends on the sample size used for correlation matrix calculation; higher quantile aggregation functions and the *p-value* ranker are more robust for smaller sample sizes, as shown in Figure 8. Even though the *q75 p-value* and *median p-value* rankers led to slightly better results for sample size 20 and 50, respectively, the *mean p-value* ranker is still recommended as the gains of the other two are rather small, and the *mean p-value* ranker behaves well for all the sample sizes unlike the named two that start to behave much worse for bigger samples.

D. Practical significance

We have already shown that the proposed methods are statistically significantly better than random. Their MDDMRs (when compared with the random baseline) are strictly negative. However, the absolute size of MDDMRs is far from its ideal value, which approaches half of the mean number of KEGG pathways considered. This value is 186 in our case. In order to assess the practical significance of the general application of the GBA principle, we offer also additional

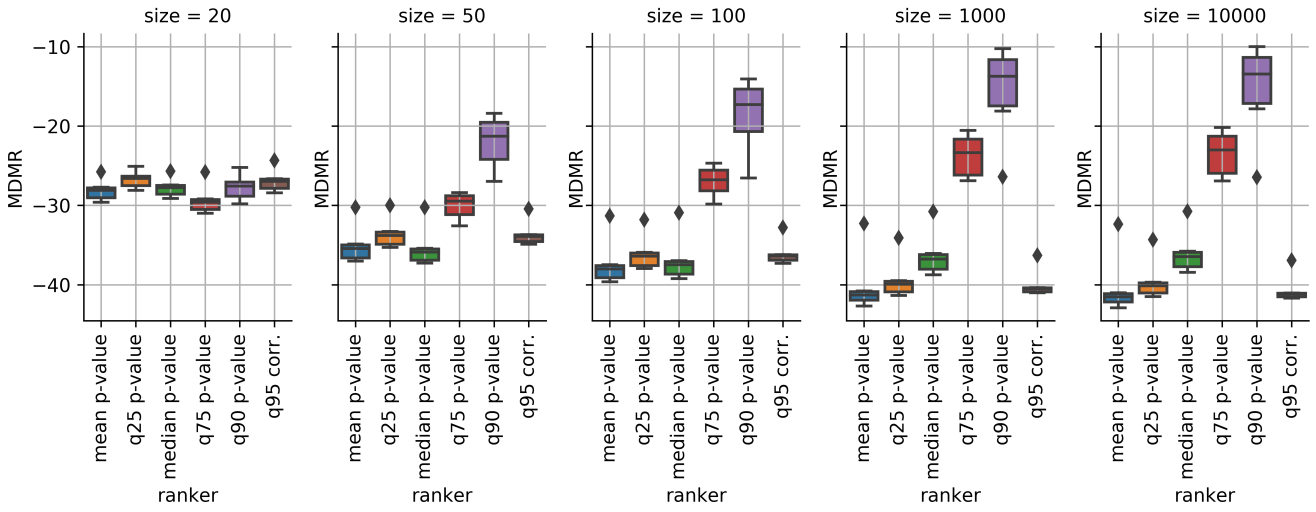


Fig. 8: The performance of a selected subset of rankers changing with increasing sample size.

	$MDMR$	MMR	ranker	co-location	dist. threshold	sample size
0	-42.90	38.47	mean p-value	yes	100,000	10000
1	-42.69	38.40	mean p-value	yes	100,000	1000
2	-42.33	38.94	mean p-value	yes	1,000,000	10000
3	-42.13	38.93	mean p-value	yes	1,000,000	1000
4	-41.67	39.62	q95 corr.	yes	100,000	10000
5	-41.65	39.35	mean p-value	yes	10,000	10000
6	-41.63	39.67	q95 corr.	yes	1,000,000	10000
7	-41.48	40.01	q25 p-value	yes	100,000	10000
8	-41.41	39.32	mean p-value	yes	10,000	1000
9	-41.40	39.60	mean p-value	yes	5,000	10000
-	0.01	69.08	random	—	—	—

TABLE II: The top ten over all evaluated parameter combinations and sample sizes.

quality measures: 1) the percentage of genes whose first annotation is right, 2) the percentage of genes that contain a true KEGG annotation in top 10 annotations assigned.

The percentage of genes whose first annotation is right for the best method from Table II averaged over individual runs is 17.65% compared to 1.42% of the random method. When considering the first ten annotations, the best method leads to at least o acccne correct annotation in 45.48% genes, while the random method only in 12.42% genes (again averaged over all the runs). Figure 9 illustrates the difference in minimum rank distribution for the best and the random methods.

The supplementary measures suggest that the methods demonstrate reasonable general practical applicability. It is expected that this applicability will grow in studies dealing with biologically homogeneous sets of samples and targeting specific terms (e.g., diseases related to the samples under consideration). This is in line with the previous observations that co-expression patterns are tissue-dependent [27].

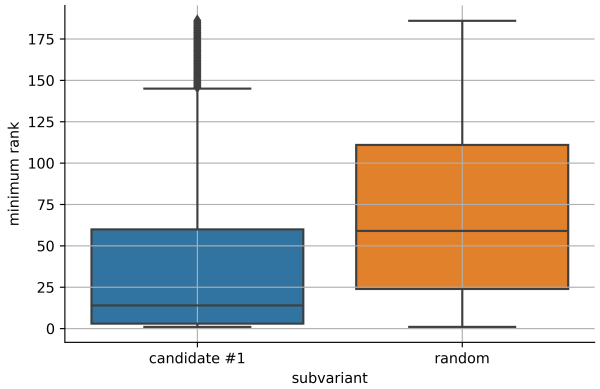


Fig. 9: The distribution of minimum ranks of correct annotations across genes for the best and random method.

VI. Conclusion

The functional annotation of genes is very important for biological and medical applications; however, functional annotations are sometimes lacking. The functional annotation for genes can be derived using the co-expression networks from related genes. The described method produces rankings of available annotation candidates for further biological evaluation. The KEGG annotations of genes were used to evaluate the performance of discussed methods. It was shown that the guilt-by-association principle can be used for gene

annotation as the rankings produced using such approach were statistically significantly better than random rankings. Furthermore, it was shown that larger sample sizes and co-location information improves the performance of the gene annotation. The multiple threshold variant denoted as *p-value ranker* was also shown to perform statistically significantly better compared to a single correlation threshold. From the evaluated configurations of the proposed method, the variant with *mean p-value* ranker with co-location information is recommended as the default configuration. In terms of practical significance, this method yields a correct annotation within much fewer candidates than a random ranking would produce. Still, the annotations proposed through correlation networks should be considered as candidate terms rather than plausible annotation terms.

REFERENCES

[1] D. Steinhauser, L. Krall, C. Müssig *et al.*, "Correlation networks," *Analysis of Biological Networks*, vol. 305, p. 333, 2008.

[2] D. Yu, M. Kim, G. Xiao, and T. H. Hwang, "Review of biological network data and its applications," *Genomics & Informatics*, vol. 11, no. 4, pp. 200–210, 2013.

[3] P. Langfelder and S. Horvath, "WGCNA: an R package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, no. 1, pp. 1–13, 2008.

[4] E. A. Serin, H. Nijveen, H. W. Hilhorst, and W. Ligterink, "Learning from co-expression networks: possibilities and challenges," *Frontiers in Plant Science*, vol. 7, no. 444, 2016.

[5] J. Ruan and W. Zhang, "Identification and evaluation of functional modules in gene co-expression networks," in *Systems Biology and Computational Proteomics*. Springer, 2006, pp. 57–76.

[6] Y. Liu, H.-Y. Gu, J. Zhu *et al.*, "Identification of hub genes and key pathways associated with bipolar disorder based on weighted gene co-expression network analysis," *Frontiers in Physiology*, p. 1081, 2019.

[7] X. Guo, L. Gao, Q. Liao *et al.*, "Long non-coding RNAs function annotation: a global prediction method based on bi-colored networks," *Nucleic Acids Research*, vol. 41, no. 2, pp. e35–e35, 2013.

[8] Y. Hao, W. Wu, F. Shi *et al.*, "Prediction of long noncoding RNA functions with co-expression network in esophageal squamous cell carcinoma," *BMC Cancer*, vol. 15, no. 1, pp. 1–10, 2015.

[9] A. Emamjomeh, E. Saboori Robat, J. Zahiri *et al.*, "Gene co-expression network reconstruction: a review on computational methods for inferring functional information from plant-based expression data," *Plant Biotechnology Reports*, vol. 11, no. 2, pp. 71–86, 2017.

[10] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, 2005.

[11] M. Giulietti, A. Righetti, G. Principato, and F. Piva, "LncRNA co-expression network analysis reveals novel biomarkers for pancreatic cancer," *Carcinogenesis*, vol. 39, no. 8, pp. 1016–1025, 2018.

[12] L. Song, P. Langfelder, and S. Horvath, "Comparison of co-expression measures: mutual information, correlation, and model based indices," *BMC Bioinformatics*, vol. 13, no. 1, pp. 1–21, 2012.

[13] S. Van Dam, U. Vosa, A. van der Graaf *et al.*, "Gene co-expression analysis for functional classification and gene–disease predictions," *Briefings in Bioinformatics*, vol. 19, no. 4, pp. 575–592, 2018.

[14] S. L. Carter, C. M. Brechbühler, M. Griffin, and A. T. Bond, "Gene co-expression network topology provides a framework for molecular characterization of cellular state," *Bioinformatics*, vol. 20, no. 14, pp. 2242–2250, 2004.

[15] J. J. Burns, B. T. Shealy, M. S. Greer *et al.*, "Addressing noise in co-expression network construction," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab495, 2022.

[16] W. Chen, X. Zhang, J. Li *et al.*, "Comprehensive analysis of coding-lncRNA gene co-expression network uncovers conserved functional lncRNAs in zebrafish," *BMC Genomics*, vol. 19, no. 2, pp. 73–85, 2018.

[17] W. Liu, L. Li, X. Long *et al.*, "Construction and analysis of gene co-expression networks in escherichia coli," *Cells*, vol. 7, no. 3, p. 19, 2018.

[18] G. de Anda-Jáuregui, S. A. Alcalá-Corona, J. Espinal-Enríquez, and E. Hernández-Lemus, "Functional and transcriptional connectivity of communities in breast cancer co-expression networks," *Applied Network Science*, vol. 4, no. 1, pp. 1–13, 2019.

[19] S. Oliver, "Guilt-by-association goes global," *Nature*, vol. 403, no. 6770, pp. 601–602, 2000.

[20] S. Lefever, J. Anckaert, P.-J. Volders *et al.*, "decodeRNA—predicting non-coding RNA functions using guilt-by-association," *Database*, vol. 2017, 2017.

[21] P. Mestdagh, S. Lefever, F. Pattyn *et al.*, "The microRNA body map: dissecting microRNA function through integrative genomics," *Nucleic Acids Research*, vol. 39, no. 20, p. e136, 2011.

[22] Y. Zhang, D. Bu, P. Huo *et al.*, "ncFANs v2.0: an integrative platform for functional annotation of non-coding RNAs," *Nucleic Acids Research*, vol. 49, no. W1, pp. W459–W468, 2021.

[23] Q. Liao, C. Liu, X. Yuan *et al.*, "Large-scale prediction of long non-coding RNA functions in a coding–non-coding gene co-expression network," *Nucleic Acids Research*, vol. 39, no. 9, pp. 3864–3878, 2011.

[24] W. Chen, J. Li, S. Huang *et al.*, "GCEN: An easy-to-use toolkit for gene co-expression network analysis and lncRNAs annotation," *Current Issues in Molecular Biology*, vol. 44, no. 4, pp. 1479–1487, 2022.

[25] L. Cowen, T. Ideker, B. J. Raphael, and R. Sharan, "Network propagation: a universal amplifier of genetic associations," *Nature Reviews Genetics*, vol. 18, no. 9, pp. 551–562, 2017.

[26] J. Gillis and P. Pavlidis, ""Guilt by association" is the exception rather than the rule in gene networks," *PLoS Computational Biology*, vol. 8, no. 3, p. e1002444, 2012.

[27] D. García-Cortés, G. de Anda-Jáuregui, C. Fresno *et al.*, "Gene co-expression is distance-dependent in breast cancer," *Frontiers in Oncology*, p. 1232, 2020.

[28] L. G. Leal, C. Lopez, and L. Lopez-Kleine, "Construction and comparison of gene co-expression networks shows complex plant immune responses," *PeerJ*, vol. 2, p. e610, 2014.

[29] M. Kanehisa, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 28, no. 1, pp. 27–30, Jan. 2000.

[30] M. Kanehisa, M. Furumichi, Y. Sato *et al.*, "KEGG: integrating viruses and cellular organisms," *Nucleic Acids Research*, vol. 49, no. D1, pp. D545–D551, Oct. 2020.

[31] Y. Chen *et al.*, "Gene expression inference with deep learning," *Bioinformatics*, vol. 32, no. 12, pp. 1832–1839, 2 2016.

[32] V. Kunc and J. Kléma, "On transformative adaptive activation functions in neural networks for gene expression inference," *PLoS ONE*, vol. 16, no. 1, p. e0243915, 2021.

[33] F. Cunningham, J. E. Allen, J. Allen *et al.*, "Ensembl 2022," *Nucleic Acids Research*, vol. 50, no. D1, pp. D988–D995, Nov. 2021.

[34] The Gene Ontology Consortium, "The Gene Ontology resource: enriching a GOld mine," *Nucleic Acids Research*, vol. 49, no. D1, pp. D325–D334, 2021.

[35] M. G. Kendall, "The treatment of ties in ranking problems," *Biometrika*, vol. 33, no. 3, pp. 239–251, 1945.

[36] R. Overbeek, M. Fonstein, M. D'souza *et al.*, "The use of gene clusters to infer functional coupling," *Proceedings of the National Academy of Sciences*, vol. 96, no. 6, pp. 2896–2901, 1999.

[37] O. Q. Zinani, K. Keseroğlu, and E. M. Özbudak, "Regulatory mechanisms ensuring coordinated expression of functionally related genes," *Trends in Genetics*, vol. 38, pp. 73–81, 2022.

[38] P. Virtanen, R. Gommers, T. E. Oliphant *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[39] W. McKinney, "Data structures for statistical computing in Python," in *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman, Eds., 2010, pp. 56–61.