BMC Bioinformatics

RESEARCH

Open Access

circGPA: circRNA functional annotation based on probability-generating functions

Petr Ryšavý[1*], Jiří Kléma[1] and Michaela Dostálová Merkerová[2]

*Correspondence: petr.rysavy@fel.cvut.cz

[1] Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic
[2] Department of Genomics, Institute of Hematology and Blood Transfusion, Prague, Czech Republic

Abstract

Recent research has already shown that circular RNAs (circRNAs) are functional in gene expression regulation and potentially related to diseases. Due to their stability, circRNAs can also be used as biomarkers for diagnosis. However, the function of most circRNAs remains unknown, and it is expensive and time-consuming to discover it through biological experiments. In this paper, we predict circRNA annotations from the knowledge of their interaction with miRNAs and subsequent miRNA–mRNA interactions. First, we construct an interaction network for a target circRNA and secondly spread the information from the network nodes with the known function to the root circRNA node. This idea itself is not new; our main contribution lies in proposing an efficient and exact deterministic procedure based on the principle of probability-generating functions to calculate the *p*-value of association test between a circRNA and an annotation term. We show that our publicly available algorithm is both more effective and efficient than the commonly used Monte-Carlo sampling approach that may suffer from difficult quantification of sampling convergence and subsequent sampling inefficiency. We experimentally demonstrate that the new approach is two orders of magnitude faster than the Monte-Carlo sampling, which makes summary annotation of large circRNA files feasible; this includes their reannotation after periodical interaction network updates, for example. We provide a summary annotation of a current circRNA database as one of our outputs. The proposed algorithm could be generalized towards other types of RNA in way that is straightforward.

Keywords: Circular RNA, Annotation term, Interaction network

Introduction

Recent research has found circular RNAs (circRNAs) to be broadly expressed in eukaryotes in tissue- and species-specific manner [1]. Some circRNAs have already been shown to regulate gene expression and potentially relate to diseases [2–4]. Due to their stability, circRNAs can also be used as biomarkers for diagnosis [2, 5, 6]. However, the function of most circRNAs remains unknown.

Functional annotation aims to attach biological information to genomic elements. The traditional functional annotation stems from sequential similarity with annotated sequences. Statistically significant similarity often reflects common ancestry and then also a common function [7]. This reasoning applies mostly to genes and proteins.

CircRNAs are known to regulate gene expression by influencing the transcription, the mRNA turnover, and translation by sponging RNA-binding proteins and microRNAs [8]. Their annotation databases often include basic information on their tissue-specificity, disease associations, and miRNA interactions [9]. Furthermore, advanced circRNA annotations could be obtained from the knowledge of their interaction with miRNAs and all other interactions of these miRNAs [10]. At the same time, the circRNA annotations may stem from the known annotations of their host genes [11].

In this paper, we propose an algorithm that allows for annotating circRNAs with annotation terms as, for example, gene-ontology terms, phenotype terms, or diseases. The algorithm implements an integrative approach which firstly constructs an interaction network for a target circRNA, and secondly, spreads the information from the network nodes with the known function to the root circRNA node. The algorithm employs the simple principle that a circRNA should be annotated with a term that is over-represented in the set of its interacting nodes. The most common solution is to quantify this over-representation in terms of p-value through stochastic algorithms [6, 12–15]. An analogical principle has also been implemented in protein function annotation [16]. The main disadvantage of these algorithms is their randomness and low sampling efficiency. They can be computationally intensive, especially when estimating low p-values in multiple testing settings frequent in genetics [17]. The low p-values need to be estimated accurately in order to identify and possibly sort the most interesting terms for the circRNA under examination; moreover, the p-values must be estimated with high accuracy to control the family-wise error rate. There have been numerous past efforts to reduce the computational burden of stochastic algorithms. Those which are universally applicable [18, 19] rely on stopping early when the p-value is obviously large. The efficiency of these approaches depends on the frequency of low p-values (truly interesting annotation terms in our case), acceleration in the order of tens is typically reported. More efficient algorithms are often dedicated to specific tasks such as mean comparison in two-sample tests where the individual runs could be partitioned with p-values showing a predictable trend across the partitions [17]. For example, these methods can be used for the identification of differentially expressed genes. To conclude, bulk annotation of multiple circRNAs remains computationally challenging for stochastic algorithms.

Our main contribution lies in the proposal of an efficient and exact procedure that is based on the principle of probability-generating functions [20]. For this reason, we named our algorithm circGPA (circRNA generating-polynomial annotator). The algorithm has four steps. First, an interaction network for a circRNA is constructed. Second, a statistic that quantifies the size of the neighborhood of the circRNA that is annotated with a term of certain cardinality is introduced. Third, the probability mass function of the statistic, which is a discrete random variable, is represented as a power series (the generating function). Fourth, the coefficients of the power series serve to calculate the p-value for the pair of circRNA and annotation term efficiently and exactly.

We show that circGPA is both more effective and efficient than the commonly used stochastic sampling approach; in particular, circGPA calculates all the p-values exactly, and it is at least two orders of magnitude faster than stochastic sampling. This feature enables the summary annotation of large circRNA files, including their reannotation after periodical interaction network updates, for example. In the supplement, we provide a summary annotation of a current circRNA database as one of our outputs. The annotation contains around 3000 circRNAs available in the CircInteractome database and about 10,000 annotation terms taken from the C5 category of MSigDB database (ontology gene sets consisting of the gene-ontology terms and the human phenotype ontology). To illustrate the practical significance of the new algorithm, the summary annotation was processed in 20 months of CPU time[1], with the stochastic approach this would be technically unreachable. circGPA, together with the code used to generate the figures in this paper, is publicly available on https://github.com/petrrysavy/circgpa-paper. The outputs of our code can be downloaded from https://ida.fel.cvut.cz/~rysavy/circgpa/. The proposed algorithm could be generalized in a straightforward manner towards different types of RNA that could be characterized with interaction graphs of similar properties that we show in this paper.

The rest of our paper is organized as follows. The "Problem statement" section defines the problem to be solved as root node classification in tripartite circRNA interaction graph. The "Proposed statistic" section introduces the key interaction statistic for a paired circRNA and annotation term. The statistic is easy to compute; however, its p-value estimation is more time-consuming. For this reason, the "p-value calculation" section introduces an exact p-value calculation algorithm based on generating functions. The "Implementation" section gives the algorithm pseudocode as well as further details on interaction network construction. The basic concepts are summarized in an all-in-one example in the "All in one example" section. The "Results" section experimentally evaluates the new algorithm and compares it with the Monte Carlo algorithm. The "Conclusion" section concludes the paper and outlines future work.

Materials and methods

Problem statement

This paper aims to devise an algorithm to annotate circRNA molecules with annotation terms. To do so, we exploit the interaction graph between circRNA–miRNA and miRNA–mRNA molecules. As we assume that the annotations of circRNAs are independent, we can process the individual circRNAs sequentially and restrict ourselves to a single circRNA molecule in our description. Assume a fixed ordering on miRNA and mRNA molecules. Assume that the count of mRNAs (miRNAs) is $|m|$ ($|\mu|$).

Formally, we can define the interaction graph between the selected circRNA and miRNAs using a vector $\mathbf{a}^{\mu,c} \in \{0,1\}^{|\mu|}$ where each field represents whether a particular miRNA interacts with the circRNA. Interactions between miRNAs and mRNAs are represented by an adjacency matrix $\mathbf{A}^{m,\mu} \in \{0,1\}^{|m|\times|\mu|}$ where each row is a vector indicating which miRNAs interact with a particular mRNA.[2] We assume that the graph edges are directed only from circRNA to miRNA and from miRNA to mRNA, so that a directed path cannot connect two molecules of the same type. A simple network is shown in Fig. 1.

[1] Additional graphical outputs took another 10 months of CPU time.

[2] We define $\mathbf{a}^{\mu,c}$ as a binary vector, and $\mathbf{A}^{m,\mu}$ as a binary matrix. However, the approach can be easily generalized to the situation when $\mathbf{a}^{\mu,c}$ and $\mathbf{A}^{m,\mu}$ contain natural numbers which might capture, for example, the strength of the interactions or alternatively situations when a miRNA has two binding spots on a circRNA. The fields higher than 1 can be represented as parallel edges in the interaction graph which becomes a multigraph in this case.

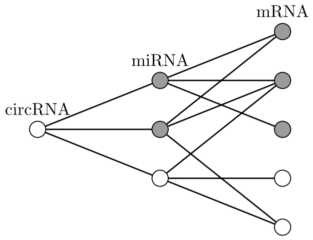

Fig. 1 An example of a network. The grey nodes are part of the annotation term. The circRNA of interest interacts with all three miRNAs, out of which two are annotated with the term of interest. There are five mRNAs, three of them annotated with the term. In the graph, we might find three paths from the circRNA to a miRNA and nine paths from the circRNA to a mRNA. Out of those, 2 + 6 terminate in an annotated mi/mRNA, resulting in $s(c, g) = 8$

In our notation, an annotation term will be defined by a set of mRNAs and miRNAs it annotates. The membership of mRNAs (miRNAs respectively) is formalized using binary vectors $\mathbf{g}^m \in \{0,1\}^{|m|}$ ($\mathbf{g}^\mu \in \{0,1\}^{|\mu|}$ respectively). As a shorthand notation, we will use the symbol g to denote the tuple of the latter, i.e., $g = (\mathbf{g}^m, \mathbf{g}^\mu)$. Having these definitions on hand, we can define the problem to be solved in this paper.

Definition 1 (CircRNA annotation problem) For a circular RNA, let $\mathbf{A}^{m,\mu}$, $\mathbf{a}^{\mu,c}$ be its interaction graph. Decide whether the circRNA should be annotated with a term $g = (\mathbf{g}^m, \mathbf{g}^\mu)$.

Proposed statistic

To solve the problem, we will develop a simple yet powerful statistic to annotate a circRNA. The concept is based on the ”guilt by association” principle [16, 21]. The circRNA should be annotated with a term if and only if this molecule frequently interacts with miRNAs (and through them indirectly with mRNAs) annotated with the term. We will capture this frequency in statistic s. This statistic will quantify the size of the neighborhood of the circRNA that is annotated with the term. As the complete tripartite graph is only a unification of two bipartite graphs and remains fixed for the circRNA, we might calculate this number precisely:

$$s(c, g) = (\mathbf{a}^{\mu,c})^T \mathbf{g}^\mu + (\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c})^T \mathbf{g}^m. \tag{1}$$

The first addend shows how many paths of length one end in a miRNA annotated with the term. The term $\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}$ shows how many paths go from c to each mRNA. The second addend in Formula (1) calculates how many paths of length two terminate in an mRNA that is annotated with the term.

However, the frequency represented by s is hard to explain without knowing the entire neighborhood. Larger neighborhoods, as well as more abundant gene terms, tend to generate a larger frequency. The importance of the term could better be captured by a relative frequency. Assume that we start a random walk in circRNA c. We might calculate the probability that this random walk ends in an RNA annotated with the term g. For a fixed circRNA, the size of the neighborhood is fixed. Therefore, the aforementioned probability is equal to $s(c, g)$ but for a normalization factor.

We will continue to use the number of paths represented in $s(c,g)$, knowing that they are only linearly scaled, preserving the ordering of the above-mentioned probabilities. To increase interpretability, we will further develop a normalized s as well as the p-value for the statistic so that standard statistical reasoning is applicable. In this work, we will avoid random Monte-Carlo sampling (used among others in [6, 12–15]) and simulating the random walk and claim its low efficiency in our setting for the p-value calculation.

Normalization

As the value of statistic s grows by definition with the size of annotation term and the size of the interaction graph (it gives the number of distinct paths to mRNAs and miRNAs annotated with the term), we present the user with a more explainable output. We normalize the statistic (1) by its expected value

$$\mathbb{E}\big(s(c,g)\big) = \frac{\|\mathbf{g}^{\mu}\|_1}{|\mu|}(\mathbf{a}^{\mu,c})^T\mathbf{1} + \frac{\|\mathbf{g}^{m}\|_1}{|m|}(\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c})^T\mathbf{1} \tag{2}$$

where $\mathbf{1} = (1,1,\ldots,1)^T$ is the vector of ones and $\|\cdot\|_1$ denotes the L1-norm. The expected value gives the expected number of random walks that end in an RNA annotated with the term if the annotations were assigned randomly. The user is then presented with the ratio of the statistic and its expected value

$$\frac{s(c,g)}{\mathbb{E}\big(s(c,g)\big)}. \tag{3}$$

This ratio represents the normalized statistic. The value above 1 then stands for terms that tend to interact with the circRNA under observation more than expected as can be seen in Table 1.

Influence of individual RNAs

Once circGPA predicts that a circRNA should be annotated with a term, users might be interested in which miRNAs and mRNAs back up this annotation. In other words, knowledge of which RNAs connect the circRNA to the annotation term is important. Fortunately, it is possible to split $s(c,g)$ among individual molecules. A natural method of explaining how much the RNA adds to the statistic is to remove this RNA with all its incoming and outcoming edges from the graph. On such a modified graph, we recalculate the score and calculate the difference in the score value. We can calculate this difference for all miRNAs and mRNAs at once using linear algebra. We denote the vector of those differences Δ^m (Δ^μ) for mRNAs (miRNAs). For a vector $\mathbf{v}$, let $diag(\mathbf{v})$ denote a diagonal matrix with elements of $\mathbf{v}$ on its diagonal. Then we derive that

$$\Delta^{\mu} = diag(\mathbf{a}^{\mu,c})\Big(\mathbf{g}^{\mu} + (\mathbf{A}^{m,\mu})^T\mathbf{g}^{m}\Big), \tag{4}$$

$$\Delta^{m} = diag(\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c})\mathbf{g}^{m}. \tag{5}$$

One can notice that the L1 norm of Δ^μ is equal to $s(c,g)$. We use values Δ^μ, Δ^m to sort mi/mRNAs in a report that shows the influence of individual RNAs. An example output will be seen in the "Results" section.

p-value calculation

To understand and compare the values of statistic s among different circRNAs and annotation terms, we need to calculate its p-value. The p-value cannot stem solely from s itself as other circular RNAs have a different number of connections to the remaining RNAs. In addition, more frequent annotation terms will reach higher scores. Formally, statistic $s(c, g)$ is an outcome of the statistical test, whose null hypothesis is that *the given* c, g *pair is not related*. In other words, the null hypothesis is that circRNA c has no preference in interactions with miRNAs (or mediated interactions with mRNAs) annotated with term g. The alternative hypothesis states that c *should be annotated with* g as g is overrepresented in the neighborhood of c.

The p-value, in our case, represents the probability that a random annotation term of the same size in the same interaction graph reaches the same statistic s or higher. The literal implementation of the null distribution simulation would thus be empirical random sampling with replacement [22]. In our case, this Monte-Carlo approach would be based on enumerating the random subsets of m/miRNAs of the same size as the evaluated annotation term and calculating the statistic value based on Formula (1).

This paper proposes an exact approach that does not depend on random trials but uses generating polynomials instead to compute the p-value. We should first reformulate the problem so that we can easily describe its mathematical solution. Denote $\|\mathbf{g}^{\mu}\|_1$ the number of miRNA molecules annotated by the term. Formally, $\|\mathbf{g}^{\mu}\|_1$ is the L1-norm of the $\mathbf{g}^{\mu}$ vector. Define $\|\mathbf{g}^{m}\|_1$ similarly. For each miRNA, there is a fixed number denoting its weight in the statistic (1). This weight is 1 if and only if the circRNA of interest is connected to the miRNA, zero otherwise. The weight is stored in the respective field of $\mathbf{a}^{\mu,c}$. Out of all miRNAs, we select $\|\mathbf{g}^{\mu}\|_1$. For mRNA, the weight can be seen in the respective field of $\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}$. Out of all interacting mRNAs, $\|\mathbf{g}^{m}\|_1$ mRNAs are selected.

To calculate the p-value, the molecules of mRNA and miRNA are selected randomly given the weights and the fact that $\|\mathbf{g}^{\mu}\|_1$ and $\|\mathbf{g}^{m}\|_1$ need to be preserved. For now, we consider only miRNAs. Imagine a bag full of balls with numbers written on them. Each number is a field in $\mathbf{a}^{\mu,c}$ (one field equals one ball). Now we randomly select $\|\mathbf{g}^{\mu}\|_1$ balls from the bag and sum the numbers written on them. By repeating this procedure many times, we get the null distribution for the first part of the statistic (1). If we include a second bag with numbers taken from $\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}$, we get the null distribution for the whole statistic.

Having built an informal intuition, we can proceed to introduce the generating polynomials by which we denote a polynomial which is a multiple of the well-established probability-generating functions [20]. Consider an mRNA that is connected by five paths to the circRNA. The weight of this mRNA is 5. In a random annotation term, this mRNA is either included or not. This gives two possibilities. We can formulate the generating polynomial for this mRNA as

$$1 + x^5 y^1. \tag{6}$$

The variable x keeps track of weights, y keeps track of the number of selected mRNAs. Having a simple graph with only one mRNA, we have two options for building a random mRNA set: either we use zero mRNAs, and the sum of weights is zero (the term 1, which equals x^0y^0), or we use one, and the sum is 5 (the term x^5y^1). If we also consider a new mRNA with weight 3, the resulting polynomial that represents the extended graph is

$$(1 + x^5y^1) \cdot (1 + x^3y^1) = 1 + x^3y^1 + x^5y^1 + x^8y^2. \tag{7}$$

We immediately see that if we select no mRNAs, we can only get the sum of weights 0; by selecting one, the weights will be either 3 or 5, and by selecting two, the sum of the weights will be eight. The coefficients by terms with y^1 show a single possibility of getting a weight of three or five. Another helpful view on the formula above might be as on a dynamic programming algorithm in a 2D array where the power of x denotes a row, the power of y denotes a column, and the coefficient is the number at the particular position of the table. Now, we can define the generating polynomial for a weight vector.

Definition 2 (Generating polynomial) Let $\mathbf{w}$ be a vector of weights (of mRNA or miRNA). Then the generating polynomial is

$$\mathrm{genpoly}_{\mathbf{w}}(x, y) = \prod_{w \in \mathbf{w}} (1 + x^w y). \tag{8}$$

Next, we define an operator that restricts the polynomial only on a selected power of one or more variables. We will denote the operator $\mid x^n$ and use it to denote only terms that contain x^n. For example, for the polynomial (7), operator $\mid y^1$ will return $x^3 + x^5$. The following theorem allows us to calculate the null distribution of the statistic (1).

Theorem 1 *Consider statistic s for a fixed circRNA c, interaction graph* $\mathbf{a}^{\mu,c}$, $\mathbf{A}^{m,\mu}$ *and annotation term sizes* $\|\mathbf{g}^{\mu}\|_1, \|\mathbf{g}^{m}\|_1$. *Then coefficients of the polynomial*

$$\left(\mathrm{genpoly}_{\mathbf{a}^{\mu,c}}(x, y) \mid y^{\|\mathbf{g}^{\mu}\|_1}\right)\left(\mathrm{genpoly}_{\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}}(x, y) \mid y^{\|\mathbf{g}^{m}\|_1}\right) \tag{9}$$

are the null distribution of statistic s up to a normalization factor.

Proof From what precedes, it can be seen that the first multiplicand coefficients are the number of ways to reach a particular value of the miRNA part of the statistic (1) by selecting a particular number of miRNAs. The restriction to the $y^{\|\mathbf{g}^{\mu}\|_1}$ ensures that the number of miRNAs in the annotation term is preserved. The same holds for the second multiplicand and mRNAs.

After multiplying the polynomials, the polynomial coefficients will hold the number of unique ways the value of the statistic can be achieved. The normalization to 1 then finishes the calculation of the null distribution. □

Once the null distribution is calculated, the p-value is then obtained by a standard approach in which we sum probabilities of all statistic values greater than $s(c, g)$.

Computational complexity

If we focus on the computational complexity of circGPA, most of the work is done in the `Generating-Polynomial` function. The two inner loops depend on variables *maxx* and *maxy*, where *maxx* is, in the worst case, linearly dependent on the L1-norm of vector $\mathbf{w}$; *maxy* is equal to the number of RNAs annotated with the terms. Their product, therefore, is linearly dependent on the product of $\|\mathbf{w}\|_1$ times the size of the annotation term. The two outer loops in function `Generating-Polynomial` do at most n operations for each unique non-zero entry in the vector of weights of count n. Sum of the fields of weight vector $\mathbf{w}$ is, therefore, the same as the number of evaluations of the two outer loops. The computational complexity of the two outer loops is in the worst case equal to $\|\mathbf{w}\|_1$. We may conclude that one call to the `Generating-Polynomial` function is in $\mathcal{O}(\|\mathbf{w}\|^2 \cdot maxy)$. Other terms in function `AnnotateCircRNA` are asymptotically smaller than the runtime of the `Generating-Polynomial` function. The overall runtime is, therefore, in

$$\mathcal{O}\Big((\|\mathbf{a}^{\mu,c}\|_1)^2\|\mathbf{g}^{\mu}\|_1 + (\|\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}\|_1)^2\|\mathbf{g}^{m}\|_1\Big). \tag{10}$$

Referential approach

A standard approach for p-value calculation would be to enumerate subsets of miRNAs/mRNAs as random annotation terms. The size of the term is preserved, and we count how many times the score is higher than $s(c, g)$. This sampling Monte-Carlo approach then allows estimation of the p-value using the biased estimator $\frac{r+1}{n+1}$, where r is the number of trials with a high enough score and n is the number of all trials [23].

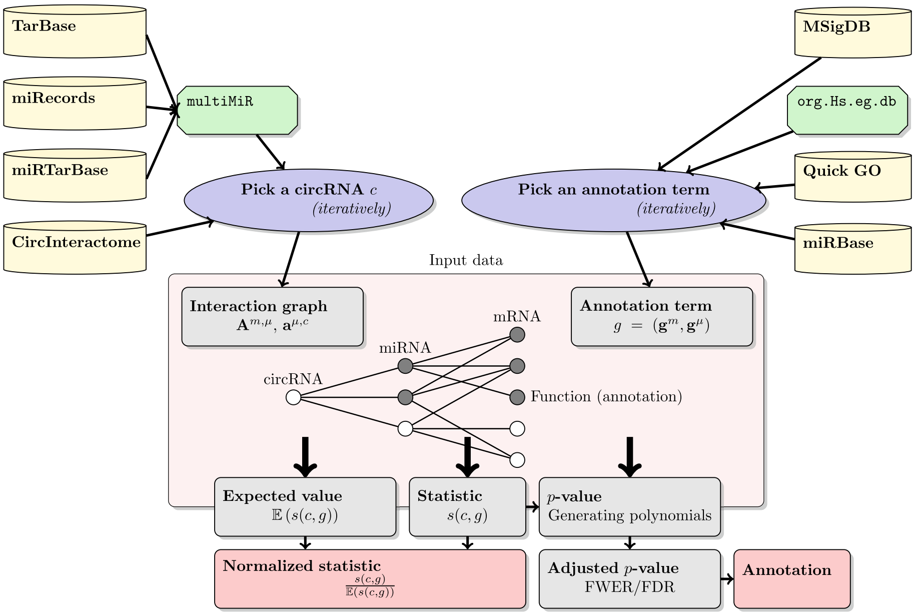
TarBase
miRecords
miRTarBase
CircInteractome
multiMiR
Pick a circRNA c (iteratively)
MSigDB
org.Hs.eg.db
Quick GO
miRBase
Pick an annotation term (iteratively)
Input data
Interaction graph $\mathbf{A}^{m,\mu}$, $\mathbf{a}^{\mu,c}$
Annotation term $g = (\mathbf{g}^m, \mathbf{g}^\mu)$
mRNA
miRNA
circRNA
Function (annotation)
Expected value $\mathbb{E}(s(c,g))$
Statistic $s(c,g)$
p-value Generating polynomials
Normalized statistic $\frac{s(c,g)}{\mathbb{E}(s(c,g))}$
Adjusted p-value FWER/FDR
Annotation

Fig. 2 An illustration of the whole pipeline. The input graph is used for multiple annotation terms to calculate the statistic and its *p*-value. Later, the *p*-values are adjusted and used for annotation

Implementation

Using a naive implementation, we need at most $2^{|m|} \cdot 2^{|\mu|}$ multiplications to evaluate polynomial (9). This number is the theoretically possible maximum number of terms in polynomial (7). The real number is, however, much smaller. As polynomial exponents repeat, the bound can be tightened. The power of x goes from 0 to $\|\mathbf{a}^{\mu,c}\|_1$ in the case of miRNAs, and from 0 to $\|\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}\|_1$ in the case of mRNAs. The y variable goes from 0 to $|\mu|$ (0 to $|m|$ in case of mRNA); however, relevant fields are only up to $\|\mathbf{g}^{\mu}\|_1$ ($\|\mathbf{g}^{m}\|_1$). The x variable can be trimmed similarly using the fact that only $\|\mathbf{g}^{\mu}\|_1$ (or $\|\mathbf{g}^{m}\|_1$) highest terms of $\mathbf{a}^{\mu,c}$ (or $\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}$) may be used.

If the weight of a mi/mRNA occurs more than once, we can exploit the binomial expansion instead of term-by-term multiplication in the Equation (8). The multiplication could be implemented using the dynamic programming approach and pointers. A similar approach was used for the fast p-value calculation of the unweighted GSEA [24]. Details can be seen in Algorithm 1. The whole pipeline is illustrated in Fig. 2.

Algorithm 1 The circGPA algorithm.

```
function AnnotateCircRNA($c$, $\mathbf{a}^{\mu,c}$, $\mathbf{A}^{m,\mu}$, $\mathbf{g}^{\mu}$, $\mathbf{g}^{m}$)
    $s(c,g) \leftarrow (\mathbf{a}^{\mu,c})^T\mathbf{g}^{\mu} + (\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c})^T\mathbf{g}^{m}$    ▷ Calculate the statistic
    $\mathbb{E}(s(c,g)) = \frac{\|\mathbf{g}^{\mu}\|_1}{|\mu|}(\mathbf{a}^{\mu,c})^T\mathbf{1} + \frac{\|\mathbf{g}^{m}\|_1}{|m|}(\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c})^T\mathbf{1}$
        ▷ The expected value of the statistic

    $miRNAPoly \leftarrow$ Generating-Polynomial($\mathbf{a}^{\mu,c}$, $\|\mathbf{g}^{\mu}\|_1$)
    $mRNAPoly \leftarrow$ Generating-Polynomial($\mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c}$, $\|\mathbf{g}^{m}\|_1$)
    $null \leftarrow$ Coefficients($miRNAPoly \cdot mRNAPoly$)
    $pval \leftarrow$ Sum($null[s(c,g):]$)/Sum($null$)
        ▷ $p$-value is the probability of reaching a higher value than $s(c,g)$ for a random term
    return $\frac{s(c,g)}{\mathbb{E}(s(c,g))}$, $pval$
end function

function Generating-Polynomial($\mathbf{w}$, $maxy$)
        ▷ $maxy$ represents how many weights are selected
    $maxx \leftarrow$ Sum($maxy$ highest terms in $\mathbf{w}$)
        ▷ Since we select only $maxy$ RNAs, higher powers of $x$ will be zero
    $curr, next \leftarrow$ 2D-arrays of 0 indexed from 0 to $maxx$ and from 0 to $maxy$
    $curr[0,0] = next[0,0] = 1$
        ▷ Initialization, 1 way of obtaining sum 0 using 0 molecules

    for all unique $w$ in $\mathbf{w}$ do
        $n \leftarrow$ count of $w$ in $\mathbf{w}$
            ▷ We will use binomial expansion of $(1 + x^w y^1)^n$
        $maxPow \leftarrow$ Min($n$, $maxy$)
            ▷ Only relevant powers of $y$ in expansion of $(1 + x^w y^1)^n$
        for $pow = 1...maxPow$ (incl.) do    ▷ Power of $x^w y^1$
            for $y = pow...maxy$ (incl.) do
                for $x = w \cdot pow...maxx$ (incl.) do
                    $next[x,y] \leftarrow next[x,y] + curr[y - pow, x - w \cdot pow] \cdot \binom{n}{pow}$
                        ▷ Multiply the $curr$ poly. by $\binom{n}{pow}(x^w y^1)^{pow}$; add to $next$
                end for
            end for
        end for
        copy values from $next$ to $curr$    ▷ This includes $\cdot x^0 y^0$
    end for
    return Polynomial in $x$ given by the last row, i.e., $curr[, maxy]$
end function
```

Input data and used libraries

Our implementation combines R code and C++ code for the critical p-value calculation. To connect the C++ code and R code, we use the `Rcpp` package [25].

To construct the graph, we exploit several databases and R packages. The circRNA–miRNA interactions are downloaded from the CircInteractome [26] database that uses the TargetScan [27] interaction prediction algorithm. The miRNA–mRNA interactions are downloaded from the TarBase [28], miRecords [29], and miRTarBase [30] databases via the `multiMiR` package [31]. In the case of miRNA–mRNA interactions, we used verified interactions only.

The GO annotation for the miRNAs is downloaded using the miRBase [32] and ENA Quick GO [33] databases. Annotation of mRNAs is done via the `org.Hs.eg.db` R package. The annotation terms are obtained from the MSigDB database [34] C5 category using the `msigdbr` R package.

Other R packages used include `miRBaseConverter`, `GO.db`, `biomaRt`,`stringr`, `httr`, `openxlsx`, `geometry`, `tictoc`, `ggnet`, `network` and `polynom`.

The graph constructed in the presented way can annotate 3, 009 circRNAs. There are 1,761 miRNAs connected by 81, 391 edges. This means that one circRNA interacts with 29 miRNAs on average. One miRNA interacts with 50 circRNAs on average. The circRNA that interacts most with other molecules is `hsa_circ_0000005` with 307 interactions. The miRNA with the most frequent interactions is `hsa-miR-942`, with 799.

The graph contains 19, 375 mRNAs with 465, 741 known interactions with miRNAs. Therefore, one mRNA interacts with 24 miRNAs on average, while one miRNA interacts with 264 mRNAs on average. The most frequent miRNA is `hsa-miR-1-3p` with 7491 interactions, the most frequent mRNA is `NUFIP2` with 331 interactions.

We work with 10, 189 annotation terms. The average size of those is 82 mRNA or miRNA molecules. If we exclude annotation terms which are too narrow or too broad (see Sect. Results for details), we end up with 7075 annotation terms with 89 molecules on average. One RNA is annotated with 43 terms on average.

All in one example

Consider the interaction network depicted in Fig. 1. In this figure, we have a circRNA of interest connected with 3 miRNAs that connect to 5 mRNAs. The edges in the graph can then be described by a vector and a matrix.

$$\mathbf{a}^{\mu,c} = (1,1,1)^T, \quad \mathbf{A}^{m,\mu} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

The annotation term contains 2 miRNAs and 3 mRNAs. It is formalized as

$$\mathbf{g}^{\mu} = (1,1,0)^T, \quad \mathbf{g}^{m} = (1,1,1,0,0)^T.$$

The weights of the miRNAs and mRNAs are

$$\mathbf{a}^{\mu,c} = (1,1,1)^T, \quad \mathbf{A}^{m,\mu}\mathbf{a}^{\mu,c} = (2,3,1,1,2)^T.$$

Which gives a statistic of

$$s(c,g) = 2 + 6 = 8.$$

The first addend is for miRNAs, the second for mRNAs. The expected value of the statistic is

$$\mathbb{E}\big(s(c,g)\big) = \frac{2}{3}(1+1+1) + \frac{3}{5}(2+3+1+1+2) = 7.4.$$

The normalized statistic is, therefore, $\frac{8}{7.4} \cong 1.08$. The generating polynomial for miRNAs is

$$(1 + x^1 y)^3 = 1 + 3xy + 3x^2y^2 + x^3y^3.$$

As we have two miRNAs that are annotated with the terms, we immediately see that there are three options for achieving two paths from the circRNA that end in an annotated miRNA (see term $3x^2y^2$ – variable y stands for the number of miRNAs, variable x stands for the number of paths). For mRNA, the generating polynomial is

$$(1 + x^1 y)^2(1 + x^2 y)^2(1 + x^3 y) = \cdots + (2x^4 + 3x^5 + 4x^6 + x^7)y^3 + \cdots.$$

The relevant terms contain the variable y to the power of three – the term annotates three mRNAs. And the corresponding terms show that there are two options for selecting mRNAs such that there are 4 paths that go from the circRNA to an annotated mRNA. Those two options are illustrated in Fig. 3. Similarly, we can see that there are four options so that six paths end in an annotated mRNA and so on.

The generating polynomial for the whole statistic is, therefore,

$$3x^2(2x^4 + 3x^5 + 4x^6 + x^7) = 6x^6 + 9x^7 + 12x^8 + 3x^9.$$

From the polynomial, we see that there are 12 ways to obtain a statistic value equal to 8. One of these is the situation depicted in Fig. 1 and solved in this example. The p-value is equal to the ratio of the number of combinations that reach a statistic value greater or equal to 8 to the number of all possible combinations. These can be seen from the polynomial or calculated as $\binom{3}{2}\binom{5}{3} = 30$. Hence, the p-value is $\frac{12+3}{6+9+12+3}$.

We can also see that the expected value is the same if calculated from the null distribution. There are six ways to get statistic equal to six, nine ways to get statistic equal to seven and so forth, i.e., the expected value is $\frac{6\cdot6+9\cdot7+12\cdot8+3\cdot9}{6+9+12+3} = \frac{222}{30} = 7.4$.

Results

We ran circGPA on a graph constructed on the human genome as explained in the "Implementation" section. For presentation purposes, we filtered the annotation terms based on their sizes. We excluded annotation terms which are too broad or too narrow. The narrow terms are difficult to evaluate statistically, the general terms suffer from low interestingness to domain experts. Reimand et al [35] argue that: *...we*

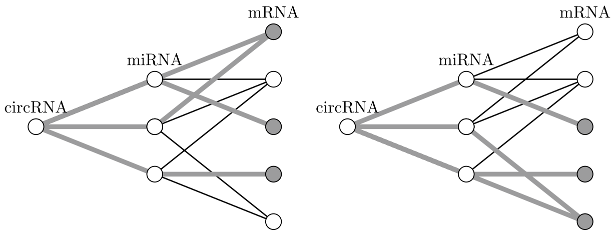

Fig. 3 The generating polynomial of the mRNAs in this graph is equal to $2x^4 + 3x^5 + 4x^6 + x^7$. We see that there are two ways to annotate 3 mRNAs, so that there are 4 paths that end in an annotated mRNA. These two cases are illustrated in the figure, the paths are marked by bold gray

often recommend excluding pathway GO terms with $< 10 - 15$ *genes and* $> 200 - 500$ *genes, although upper bounds of* $200 - 2{,}000$ *genes can be found in the literature.* We decided to stick with the bounds researched by the authors and exclude by default terms smaller than 10 genes and larger than 1, 000 genes.

Outputs of circGPA

For each circular RNA, circGPA generates a table with the statistic (1), the normalized statistic (as described in Sect. Normalization), and the *p*-values. Since we deal with many annotation terms in parallel, we have to adjust *p*-values for multiple comparison. We provide both FWER (Bonferroni [36]) and FDR (Holm [37]) adjusted *p*-values. The runtime of the *p*-value calculation is measured. An example output is shown in Table 1.

For a visual presentation of the provided results, circGPA is able to generate an output in a form that can be processed by the EnrichmentMap plugin [38] of the Cytoscape program [39]. This tool visualizes multiple annotation terms found relevant for a single circRNA in a graph. The vertices correspond to the terms; their size corresponds to the number of genes in the term and the color is calculated from the *p*-value as in a heatmap. The edges are constructed so that their width represents the Jaccard index of the connected annotation terms – a wider line means a bigger overlap between the terms. An example of a produced output is presented in Fig. 4. Therefore, the graph for a circRNA shows predicted annotations in the context of the other annotation terms. As a result, the user is presented with information about the term-term overlap and clustering of the predicted annotation. Figure 5 shows an example of the circGPA report in which miRNAs and mRNAs back up annotation.

"To further test our method of *p*-value calculation, we implemented the usual sampling approach mentioned in the "*p*-value calculation" section. circGPA is deterministic and guarantees the exact *p*-values. The sampling approach is burdened with a random noise caused by the randomness in the selection of the subsets. Therefore, the *p*-values are not the same. At the very beginning, we thus configured the stochastic algorithm to approach the exact *p*-values. We worked with all the annotation terms relevant to `hsa_circ_0000228`. We found out that *p*-values closely match for 10^6 and more random trials. We follow recommendations of [23] and estimated the *p*-values as $\frac{r+1}{n+1}$, where r is the number of positive Monte-Carlo trials out of $n = 10^6$. The Spearman's ρ for the two *p*-value vectors were equal to 0.99991. We also calculated the relative difference with a mean equal to 0.015 (i.e., on average, the *p*-values differ by less than 2 %), standard deviation equal to 0.041, the maximum deviation equal to 0.82, and median equal to 0.006. If

Table 1 A sample circGPA output

Set id	Size	$s(c,g)$	$\frac{s(c,g)}{\mathbb{E}(s(c,g))}$	time (*s*)	*p*-value	Bonferroni	FDR
HP_SOFT_TISSUE_SARCOMA	115	38	3.61	0.04	1.40E-08	1.40E-04	1.39E-04
GOMF_MRNA_BINDING	287	64	2.44	0.16	2.78E-08	2.78E-04	1.39E-04
HP_GENITAL_NEOPLASM	142	41	3.16	0.05	5.78E-08	5.78E-04	1.93E-04
HP_SARCOMA	165	44	2.91	0.06	1.03E-07	1.03E-03	2.57E-04
HP_NEOPLASM_BY_HISTOLOGY	320	66	2.25	0.19	1.60E-07	1.60E-03	3.21E-04
HP_THIN_VERMILION_BORDER	329	66	2.19	0.20	3.48E-07	3.48E-03	5.80E-04
HP_THIN_UPPER_LIP_VERMILION	234	52	2.43	0.11	5.82E-07	5.82E-03	8.32E-04
GOMF_UBIQUITIN_LIKE_PROTEIN_LIGASE_BINDING	309	62	2.19	0.18	7.74E-07	7.75E-03	9.69E-04
HP_URINARY_TRACT_NEOPLASM	133	36	2.96	0.04	1.10E-06	1.10E-02	1.22E-03
GOBP_REGULATION_OF_MRNA_METABOLIC_PROCESS	334	64	2.09	0.22	1.75E-06	1.76E-02	1.63E-03

The table shows ten annotation terms with the smallest *p*-value that were obtained by annotating circRNA `hsa_circ_0000228`

we eliminate the first 20 annotation terms with a p-value $1.6 \cdot 10^{-5}$ and smaller, the mean of the relative p-value difference is 0.014, standard deviation 0.027, maximum deviation 0.372, and median 0.006.

Runtime

Then, we compared the circGPA runtime with the sampling approach using 10^6 trials. The sampling was configured to allow for Monte-Carlo p-values to approach the exact deterministic ones calculated with generating polynomials. The comparison of runtimes is summarized in Fig. 6. In those experiments, we used the sample of circRNAs summarized in Table 2. It is obvious that circGPA overcomes the stochastic algorithm for all the tested annotation terms with speedups that vary from 0.16 to 48, 670. The average speedup per single p-value calculation proved to be 3, 150. If we compare the overall runtime requirements of 59 hours needed to calculate the p-values exactly on the testing circRNAs (see Table 2) and 1967 hours using the sampling approach, we can see that our approach is 33 times faster as a whole. Extrapolating this number means that our approach to calculating the p-values can save 65 years of CPU-time to annotate all circRNAs in our database (compared to approximately two years needed to calculate the p-values using the exact approach). However, the results of circGPA are worse on densely connected circRNAs that are over-represented in our dataset (see Table 2). The real measurements have shown that the circGPA requires 20 months to annotate all circRNAs in the database, including graph construction and disk access.

Further insights into the speedups provided by the algorithm are in Fig. 7, which demonstrates the runtime complexity of Algorithm 1. If we eliminate all terms that are constant for a given circular RNA from Formula (10) and re-evaluate the runtime of the two outer loops, the computational complexity is square of the size of the gene ontology term. As we see from Fig. 7, the runtime measurements are almost a line. The slope of the line is circRNA-dependent. For more connected circular RNAs, the slope is higher.

Figure 9 demonstrates that the decreasing number of trials in the sampling approach leads to a decrease in the accuracy of p-value estimation. It is impossible to

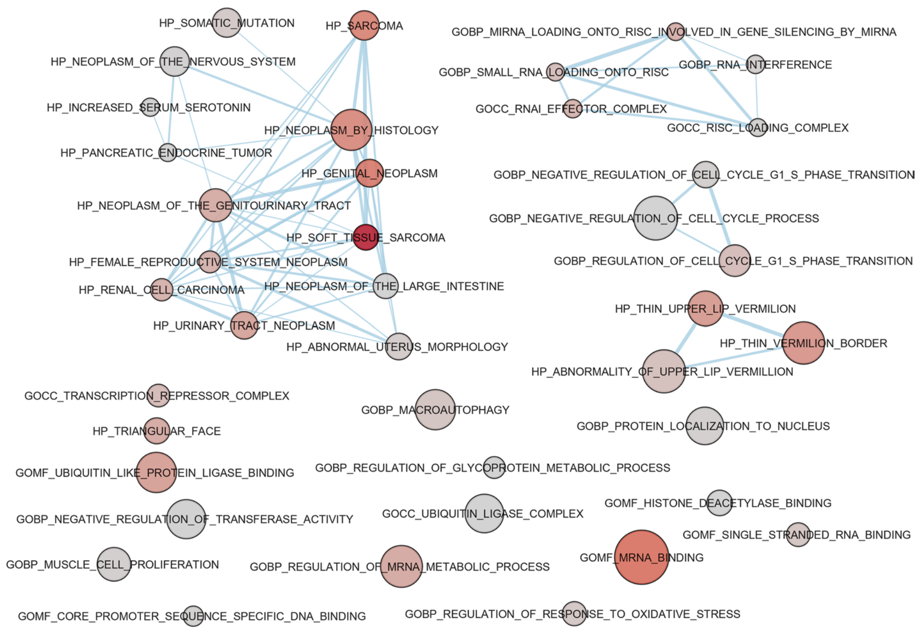
HP_SOMATIC_MUTATION
HP_SARCOMA
GOBP_MIRNA_LOADING_ONTO_RISC_INVOLVED_IN_GENE_SILENCING_BY_MIRNA
HP_NEOPLASM_OF_THE_NERVOUS_SYSTEM
GOBP_SMALL_RNA_LOADING_ONTO_RISC
GOBP_RNA_INTERFERENCE
HP_INCREASED_SERUM_SEROTONIN
GOCC_RNAI_EFFECTOR_COMPLEX
HP_NEOPLASM_BY_HISTOLOGY
GOCC_RISC_LOADING_COMPLEX
HP_PANCREATIC_ENDOCRINE_TUMOR
HP_GENITAL_NEOPLASM
GOBP_NEGATIVE_REGULATION_OF_CELL_CYCLE_G1_S_PHASE_TRANSITION
HP_NEOPLASM_OF_THE_GENITOURINARY_TRACT
GOBP_NEGATIVE_REGULATION_OF_CELL_CYCLE_PROCESS
HP_SOFT_TISSUE_SARCOMA
HP_FEMALE_REPRODUCTIVE_SYSTEM_NEOPLASM
GOBP_REGULATION_OF_CELL_CYCLE_G1_S_PHASE_TRANSITION
HP_RENAL_CELL_CARCINOMA
HP_NEOPLASM_OF_THE_LARGE_INTESTINE
HP_THIN_UPPER_LIP_VERMILION
HP_URINARY_TRACT_NEOPLASM
HP_THIN_VERMILION_BORDER
HP_ABNORMAL_UTERUS_MORPHOLOGY
HP_ABNORMALITY_OF_UPPER_LIP_VERMILLION
GOCC_TRANSCRIPTION_REPRESSOR_COMPLEX
GOBP_MACROAUTOPHAGY
GOBP_PROTEIN_LOCALIZATION_TO_NUCLEUS
HP_TRIANGULAR_FACE
GOMF_UBIQUITIN_LIKE_PROTEIN_LIGASE_BINDING
GOBP_REGULATION_OF_GLYCOPROTEIN_METABOLIC_PROCESS
GOMF_HISTONE_DEACETYLASE_BINDING
GOBP_NEGATIVE_REGULATION_OF_TRANSFERASE_ACTIVITY
GOCC_UBIQUITIN_LIGASE_COMPLEX
GOMF_SINGLE_STRANDED_RNA_BINDING
GOBP_MUSCLE_CELL_PROLIFERATION
GOBP_REGULATION_OF_MRNA_METABOLIC_PROCESS
GOMF_MRNA_BINDING
GOMF_CORE_PROMOTER_SEQUENCE_SPECIFIC_DNA_BINDING
GOBP_REGULATION_OF_RESPONSE_TO_OXIDATIVE_STRESS

Fig. 4 An example of a graph produced by the EnrichmentMap plugin of the Cytoscape program using the 40 most likely annotations of `hsa_circ_0000228`. The labels were moved manually so that they do not overlap. Several red circles that correspond to annotation terms with lower *p*-values can be noticed. Besides that, there are several clusters of terms that share genes. The biggest is located in the top left corner showing a set of terms connected with the reproductive and urinary systems. This indicates that the circular RNA might be connected with those systems. According to circBase [40], the sequence of `hsa_circ_0000228` is located on the `ZEB1` gene. According to the NCBI summary of publication [41], the `ZEB1` gene shows the highest expression in the endometrium (out of 27 tissues presented) and is also highly expressed in the urinary bladder, placenta and prostatic tissues. Also, a recent publication has shown a connection between `hsa_circ_0000228` and cervical cancer [42]. circGPA predicts a link of the circRNA to cancer as well, given the fact that `HP_SOFT_TISSUE_SARCOMA` is the term with the lowest *p*-value

straightforwardly decrease the number of trials and still approach the exact *p*-values. To reach runtimes observed in circGPA, the sampling algorithm would need to work with no more than 350 trials. However, the *p*-values then do not match the exact ones. We consider `hsa_circ_0000228` with 352 sampling trials to illustrate the gap. The Spearman's ρ of the *p*-value vectors that capture those annotation terms that pass the 0.05 *p*-value significance threshold (the most likely to be truly relevant) is equal only to 0.93. The average relative difference between those is equal to 19, with a standard deviation of 178. The maximum relative difference is 2793, and the median 0.33. Excluding the first 20 terms with the lowest *p*-value decreases the maximum deviation to 176, the mean to 2.6, the standard deviation to 11.2, and the median stays 0.32. It needs to be said, however, that we followed the recommendation of [23] to use biased estimates.

The *p*-values distribution

The applicability of calculated *p*-values is demonstrated in Fig. 8. The histograms shown are mostly bimodal. The peak close to 0 *p*-values represents the cases where alternative hypotheses truly apply. The peak close to 1 *p*-value represents the cases where the network size and interactivity are not sufficient. Clearly, this peak is large, especially for the low-interacting circRNAs (see Table 2). We can notice that the most interacting

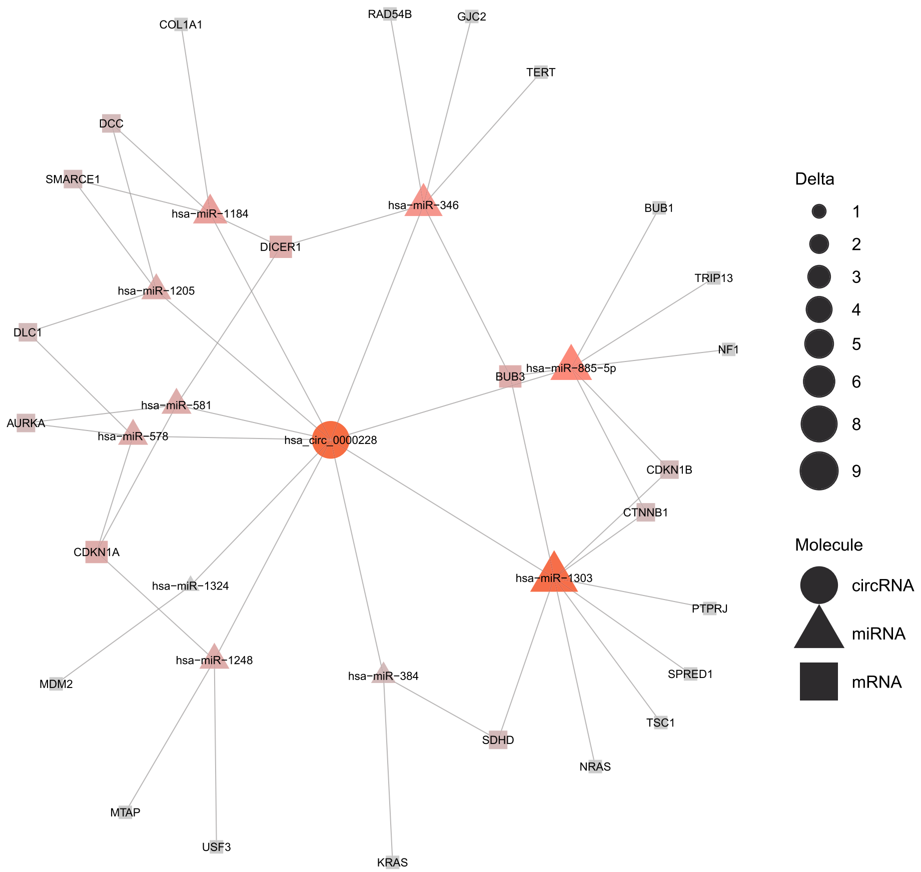

Fig. 5 A network of miRNAs and mRNAs that back up annotation of `hsa_circ_0000228` by the term with the lowest *p*-value - term `HP_SOFT_TISSUE_SARCOMA`. The size and color of each RNA shows how much the statistic (equal to 38) drops if the RNA is excluded from the graph together with all incident edges, i.e., the Δ^{μ} and Δ^{m} values. See the "Influence of individual RNAs" section for details

circRNAs tend to attract more low-*p*-value annotation terms. On the contrary, for circRNAs with few interactions, there are only a few annotation terms with low *p*-values. This is actually the desired behavior as circRNAs that interact a lot may influence many other genes and pathways; however, circRNAs that have only a few interactions influence only a specific part of the cellular machinery. This situation is common in nature and can be explained by the well-known “80-20 rule” [43].

Discussion and related work

The previous section clearly shows that circGPA is an efficient tool for circRNA functional annotation. Let us compare it conceptually with similar existing tools. The closest tool is Cerina [10] which also employs the circRNA–miRNA–mRNA interaction network for circRNA functional annotation, including GO terms. Their stochastic approach is based on permuting the connections between a given circRNA and its interacting miRNAs/mRNAs. To reduce the size of the interaction network, Cerina binds the interaction and expression data and uses the Pareto-front-based algorithm for their integrative analysis. This step increases the efficiency of the stochastic algorithm and gives a

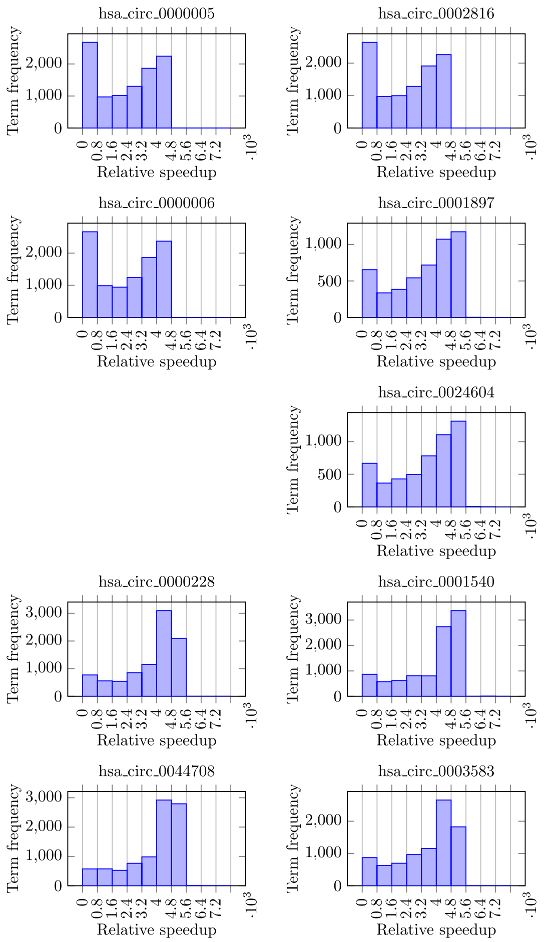

Fig. 6 A histogram of the relative speedup of the *p*-value calculation using the generating polynomials compared to the sampling approach with the sample size 10^6. The speedup is shown only for the case when the *p*-value is not equal to 1 (i.e., when $s(c, g) = 0$). CircRNA `hsa_circ_0004624` is excluded from the plot as all *p*-values are equal to 1, meaning that the circRNA is not connected with any term

chance to work with tissue-specific interactions only. We mention the possibilities of circGPA for integrative analysis in the "Conclusion" section.

The utilization of random-walk algorithms for functional annotation is wide. One of its early applications was the EnrichNet tool [14] for integrative analysis and gene annotation. Another usage is in [15], where the authors use random walks and circRNA similarity to predict circRNA-disease association. A similar approach utilized random walk for drug association prediction [44, 45]. The circRNA-disease association prediction problem was tackled using random walk with restarts [12, 46]. Fang et al. used random walks to predict miRNA-circRNA associations [13]. Close to the random walk with restarts algorithm is the PageRank algorithm [47] that has been developed for internet hyperlinks. If applied to the presented graph with circRNA–miRNA–mRNA interactions, both random walk with restarts and PageRank algorithms would lead to the same results as there is only a single source circRNA.

From the methodological point of view, the circRNA-disease association prediction is very similar to the problem solved by circGPA. The main condition for circGPA

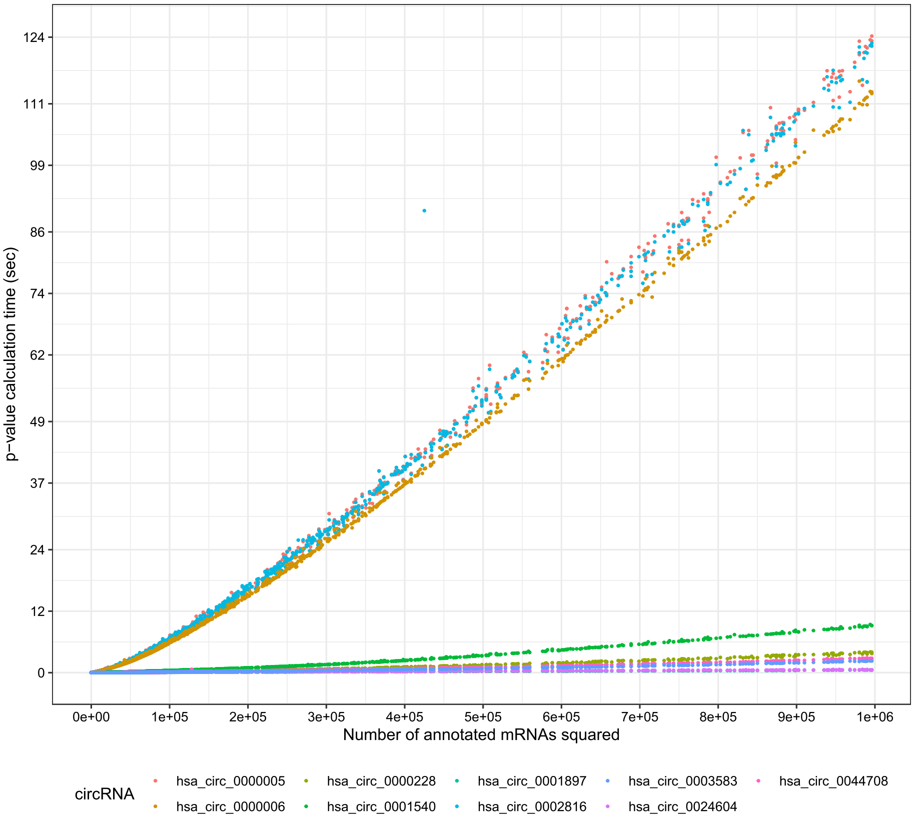

Fig. 7 A plot of the time needed to calculate the *p*-value on $\|\mathbf{g}^m\|_1^2$. The plot was generated under the same conditions as Fig. 6

application in this task is the knowledge of miRNA/mRNA-disease annotations in the circRNA interaction network. These annotation databases exist. The miRNA-disease association databases include miR2Disease [48] with 3, 273 associations, and HMDDv3 [49], which contains 35, 547 manually collected miRNA-disease associations. Those associations were collected from 19, 280 scientific papers. For gene-disease associations, we can mention the DisGeNET database [50]. The accuracy of prediction could be verified against the known circRNA disease annotations. The CircR2Cancer [51] provides a list of 1, 439 manually curated entries. The Circ2Disease [52] is the most extensive database and contains 5, 368 manually curated entries. The CircR2Disease database [53] contains 725 associations.

One of the early papers from the circRNA-disease association prediction field is the Circ2Traits database [54]. The authors worked with two data sources to predict the associations. Firstly, it was a miRNA-circRNA interaction graph together with statistical tests. Secondly, the algorithm incorporates knowledge about single nucleotide polymorphisms. Recent tools for circRNA-disease association prediction usually take advantage of similarities between the circRNA pairs and disease pairs to predict the associations. For example, a tool named PWCDA developed in [55] constructs three graphs. The first graph represents circRNA similarity; the second graph represents disease similarity. Those two graphs are used to form a disease-circRNA association graph, out of which

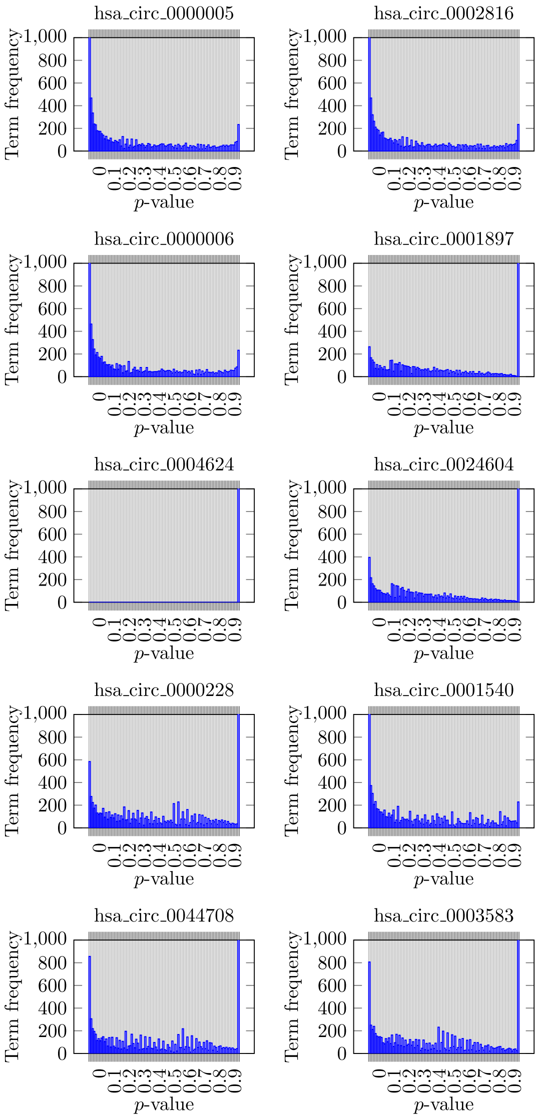

Fig. 8 Histograms of the *p*-values before multiple hypothesis testing correction. The *y* axis is trimmed to 1, 000 annotation terms

association scores are calculated. A similar approach was used in [56], where a Gaussian interacting profile is used to evaluate the similarities. There were frequent machine learning applications to circRNA-disease association prediction too. The work of Lei et al. [57] employs recommender systems to overcome the sparsity of validated annotations. The authors of [58] used convolutionary networks on the *k*-mer representation of the circRNA sequences. The disease similarities are captured using the disease ontology terms. The iGRLCDA tool [59] uses Gaussian interacting profiles, convolutional networks, and graph factorization. The authors of this paper developed a similar tool for drug-disease association prediction named HINGRL [60] too. The tool DWNCPCDA [61] uses DeepWalk. DWNCPCDA is based on the work by a similar set of authors - the NCPCDA tool [62] based on network consistency projection. This brief overview is far from complete. Other tools include, among others, [63–66].

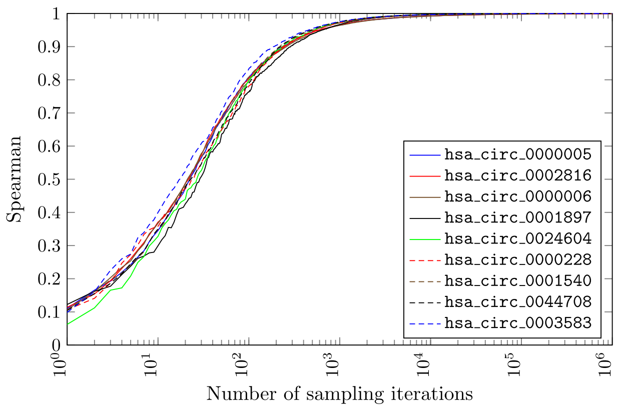

Fig. 9 Dependence of Spearman's correlation between the *p*-values calculated by Algorithm 1 and the sampling approach. The correlation is calculated for the annotation terms with *p*-value smaller than 0.05 (i.e., those that are likely to be checked manually). The plot excludes `hsa_circ_0004624` as all *p*-values for this circRNA were equal to 1

Table 2 CircRNAs used in the experiments, along with the reason we included them

circRNA	Interacting miRNAs	Paths to mRNAs	Reason to include
`hsa_circ_0000005`	307	34043	Top-interacting
`hsa_circ_0002816`	305	34113	Top-interacting
`hsa_circ_0000006`	295	33470	Top-interacting
`hsa_circ_0001897`	2	302	Least-interacting
`hsa_circ_0004624`	2	0	Least-interacting
`hsa_circ_0024604`	2	325	Least-interacting
`hsa_circ_0000228`	26	1771	Used in development
`hsa_circ_0001540`	22	5037	Random choice
`hsa_circ_0044708`	11	1837	Random choice
`hsa_circ_0003583`	12	1234	Random choice

As the *p*-value calculation using the sampling approach takes up to two weeks on a single circRNA, we limited the experiments only to those circRNAs

A clear advantage of circGPA compared to the aforementioned circRNA-disease prediction algorithms is the existence of a *p*-value that allows filtering the predictions in a more advanced manner than selecting top *k* predictions. The runtime requirements of circGPA allow us to bulk annotate all known circRNAs with gene-ontology terms and provide the annotated results on https://ida.fel.cvut.cz/~rysavy/circgpa/.

Conclusion

In this paper, we proposed an annotation algorithm circGPA that identifies prospective links between circRNAs and annotation terms. The algorithm is deterministic and based on generating polynomials. We show that this approach is both more effective and efficient than the alternative stochastic approach frequently applied in a similar context.

Our approach could easily be generalized for related tasks. Besides circRNAs, the long non-coding RNAs can act as miRNA sponges, and their annotation could be predicted too. As a whole, the approach is generalizable on any interactions which can be represented by a directed acyclic graph where leaves are annotated with binary concepts (annotation terms). Our goal is to decide upon the annotation of the roots of the graph (ncRNAs whose annotation is unknown). There are, however, computational limits to our approach. The *p*-value calculation is limited by the fact that we need to fit a potentially large table into the memory. The size of the table is the number of paths from the vertex of interest multiplied by the size of the annotation term.

When we compared circGPA with the sampling approach, we set the number of sampling trials as a constant. The only optimization we did was when the *p*-value was equal to 1. However, for high *p*-values annotation terms, it would be possible to stop the sampling earlier, knowing that the *p*-value will not be smaller than a threshold with a high-enough probability. Such approaches were proposed in [18, 19] and used in `simctest` R package. Similar ideas could apply to circGPA. We might use the generating polynomial to say that the *p*-value will not be smaller than a threshold without evaluating all polynomial coefficients. In the same manner, the weights in the loop of the `Generation-Polynomial` function could be sorted so that a bound on the *p*-value could be provided in the middle of computation.

In future work, we will look at the integrative analysis that deals with additional data modalities. So far, we have only examined interaction graphs. In the future, the annotation should stem from sequential data too in order not to rely on binary interaction records only. Also, tissue-specific expression data can help to minimize the impact of false-positive interactions with negligible expression and focus our analysis. The hierarchy of annotation terms could serve to regularize the eventual annotation records.

Acknowledgements
This work was supported by the Czech Science Foundation project 20-19162S. The authors acknowledge the support of the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

Author Contributions
PR participated in drafting and designing the method, implementation, and text of the paper. JK participated in drafting and designing the method and text of the paper. MDM participated in fundraising, analyses, and the final text of the paper. All authors read and approved the final manuscript.

Funding
This work was supported by the Czech Science Foundation project 20-19162S. The authors acknowledge the support of the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

Availability of data and materials
The source code is publicly available on https://github.com/petrrysavy/circgpa-paper. The outputs of our code can be downloaded from https://ida.fel.cvut.cz/~rysavy/circgpa/.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 20 May 2022 Accepted: 21 September 2022
Published online: 27 September 2022

References

1. Dong R, Ma X-K, Li G-W, Yang L. CIRCpedia v2: an updated database for comprehensive circular RNA annotation and expression comparison. Genom Proteom Bioinform. 2018;16(4):226–33. https://doi.org/10.1016/j.gpb.2018.08.001 (**Bioinformatics Commons (I)**).

2. Verduci L, Tarcitano E, Strano S, Yarden Y, Blandino G. CircRNAs: role in human diseases and potential use as biomarkers. Cell Death Dis. 2021;12(5):468. https://doi.org/10.1038/s41419-021-03743-3.
3. Han B, Chao J, Yao H. Circular RNA and its mechanisms in disease: from the bench to the clinic. Pharmacol Ther. 2018;187:31–44. https://doi.org/10.1016/j.pharmthera.2018.01.010.
4. Wang C-C, Han C-D, Zhao Q, Chen X. Circular RNAs and complex diseases: from experimental results to computational models. Briefings in Bioinformatics (2021). https://doi.org/10.1093/bib/bbab286. bbab286. https://academic.oup.com/bib/advance-article-pdf/doi/10.1093/bib/bbab286/39715891/bbab286.pdf
5. Meng S, Zhou H, Feng Z, Xu Z, Tang Y, Li P, Wu M. CircRNA: functions and properties of a novel potential biomarker for cancer. Mol Cancer. 2017;16(1):94. https://doi.org/10.1186/s12943-017-0663-2.
6. Zhang Z, Yang T, Xiao J. Circular RNAs: promising biomarkers for human diseases. EBioMedicine. 2018;34:267–74. https://doi.org/10.1016/j.ebiom.2018.07.036.
7. Pearson WR. An introduction to sequence similarity ("homology") searching. Curr Protoc Bioinformatics. 2013; Chapter 3: Unit3.1. https://doi.org/10.1002/0471250953.bi0301s42.
8. Panda AC. Circular RNAs act as miRNA sponges. In: Xiao J, editor. Circular RNAs: biogenesis and functions. Singapore: Springer; 2018. p. 67–79. https://doi.org/10.1007/978-981-13-1426-1_6.
9. Vromman M, Vandesompele J, Volders P-J. Closing the circle: current state and perspectives of circular RNA databases. Brief Bioinform. 2020;22(1):288–97. https://doi.org/10.1093/bib/bbz175.
10. Cardenas J, Balaji U, Gu J. Cerina: systematic circRNA functional annotation based on integrative analysis of ceRNA interactions. Sci Rep. 2020;10(1):22165. https://doi.org/10.1038/s41598-020-78469-x.
11. Li S, Chen L, Xu C, Qu X, Qin Z, Gao J, Li J, Liu J. Expression profile and bioinformatics analysis of circular RNAs in acute ischemic stroke in a South Chinese Han population. Sci Rep. 2020;10(1):10138. https://doi.org/10.1038/s41598-020-66990-y.
12. Ding Y, Chen B, Lei X, Liao B, Wu F-X. Predicting novel CircRNA-disease associations based on random walk and logistic regression model. Comput Biol Chem. 2020;87: 107287. https://doi.org/10.1016/j.compbiolchem.2020.107287.
13. Fang Z, Lei X. Prediction of miRNA-circRNA associations based on *k*-nn multi-label with random walk restart on a heterogeneous network. Big Data Min Anal. 2019;2(4):261–72.
14. Glaab E, Baudot A, Krasnogor N, Schneider R, Valencia A. EnrichNet: network-based gene set enrichment analysis. Bioinformatics. 2012;28(18):451–7. https://doi.org/10.1093/bioinformatics/bts389.
15. Lei X, Bian C. Integrating random walk with restart and *k*-nearest neighbor to identify novel circRNA-disease association. Sci Rep. 2020;10(1):1943. https://doi.org/10.1038/s41598-020-59040-0.
16. Oliver S. Guilt-by-association goes global. Nature. 2000;403(6770):601–2. https://doi.org/10.1038/35001165.
17. Segal BD, Braun T, Elliott MR, Jiang H. Fast approximation of small *p*-values in permutation tests by partitioning the permutations. Biometrics. 2018;74(1):196–206. https://doi.org/10.1111/biom.12731.
18. Silva I, Assunção R, Costa M. Power of the sequential monte Carlo test. Seq Anal. 2009;28(2):163–74. https://doi.org/10.1080/07474940902816601.
19. Silva IR, Assunção RM. Optimal generalized truncated sequential monte Carlo test. J Multivar Anal. 2013;121:33–49. https://doi.org/10.1016/j.jmva.2013.06.003.
20. Feller W. Introduction to Probability Theory and Its Applications, (1966)
21. Li Y, Xu J, Shao T, Zhang Y, Chen H, Li X. RNA function prediction. In: Kaufmann M, Klinger C, Savelsbergh A, editors. Functional genomics: methods and protocols. New York, NY: Springer; 2017. p. 17–28. https://doi.org/10.1007/978-1-4939-7231-9_2.
22. Manly B, Navarro Alberto J. Randomization, Bootstrap and Monte Carlo methods in biology. 4th ed. London: Chapman and Hall/CRC; 2020.
23. Phipson B, Smyth GK. Permutation *p*-values should never be zero: calculating exact *p*-values when permutations are randomly drawn. Stat Appl Genet Mol Biol. 2010. https://doi.org/10.2202/1544-6115.1585.
24. Keller A, Backes C, Lenhof H-P. Computation of significance scores of unweighted gene set enrichment analyses. BMC Bioinform. 2007;8(1):290. https://doi.org/10.1186/1471-2105-8-290.
25. Eddelbuettel D, Balamuta JJ. Extending R with C++: a brief introduction to Rcpp. Am Stat. 2018;72(1):28–36. https://doi.org/10.1080/00031305.2017.1375990.
26. Dudekula DB, Panda AC, Grammatikakis I, De S, Abdelmohsen K, Gorospe M. CircInteractome: a web tool for exploring circular RNAs and their interacting proteins and microRNAs. RNA Biol. 2016;13(1):34–42. https://doi.org/10.1080/15476286.2015.1128065.
27. Lewis BP, Burge CB, Bartel DP. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microrna targets. Cell. 2005;120(1):15–20. https://doi.org/10.1016/j.cell.2004.12.035.
28. Karagkouni D, Paraskevopoulou MD, Chatzopoulos S, Vlachos IS, Tastsoglou S, Kanellos I, Papadimitriou D, Kavakiotis I, Maniou S, Skoufos G, Vergoulis T, Dalamagas T, Hatzigeorgiou AG. DIANA-TarBase v8: a decade-long collection of experimentally supported miRNA-gene interactions. Nucleic Acids Res. 2017;46(D1):239–45. https://doi.org/10.1093/nar/gkx1141.
29. Xiao F, Zuo Z, Cai G, Kang S, Gao X, Li T. miRecords: an integrated resource for microRNA-target interactions. Nucleic Acids Res. 2008;37(suppl–1):105–10. https://doi.org/10.1093/nar/gkn851.
30. Hsu S-D, Lin F-M, Wu W-Y, Liang C, Huang W-C, Chan W-L, Tsai W-T, Chen G-Z, Lee C-J, Chiu C-M, Chien C-H, Wu M-C, Huang C-Y, Tsou A-P, Huang H-D. miRTarBase: a database curates experimentally validated microRNA-target interactions. Nucleic Acids Res. 2010;39(suppl–1):163–9. https://doi.org/10.1093/nar/gkq1107.
31. Ru Y, Kechris KJ, Tabakoff B, Hoffman P, Radcliffe RA, Bowler R, Mahaffey S, Rossi S, Calin GA, Bemis L, Theodorescu D. The multiMiR R package and database: integration of microRNA-target interactions along with their disease and drug associations. Nucleic Acids Res. 2014;42(17):133. https://doi.org/10.1093/nar/gku631.
32. Griffiths-Jones S, Grocock RJ, van Dongen S, Bateman A, Enright AJ. miRBase: microRNA sequences, targets and gene nomenclature. Nucleic Acids Res. 2006;34(suppl–1):140–4. https://doi.org/10.1093/nar/gkj112.
33. Binns D, Dimmer E, Huntley R, Barrell D, O'Donovan C, Apweiler R. QuickGO: a web-based tool for gene ontology searching. Bioinformatics. 2009;25(22):3045–6. https://doi.org/10.1093/bioinformatics/btp536.

34. Liberzon A, Subramanian A, Pinchback R, Thorvaldsdóttir H, Tamayo P, Mesirov JP. Molecular signatures database (MSigDB) 3.0. Bioinformatics. 2011;27(12):1739–40. https://doi.org/10.1093/bioinformatics/btr260.
35. Reimand J, Isserlin R, Voisin V, Kucera M, Tannus-Lopes C, Rostamianfar A, Wadi L, Meyer M, Wong J, Xu C, Merico D, Bader GD. Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. Nat Protoc. 2019;14(2):482–517. https://doi.org/10.1038/s41596-018-0103-9.
36. Dunn OJ. Multiple comparisons among means. J Am Stat Assoc. 1961;56(293):52–64. https://doi.org/10.1080/01621459.1961.10482090.
37. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B (Methodological). 1995;57(1):289–300.
38. Merico D, Isserlin R, Stueker O, Emili A, Bader GD. Enrichment map: a network-based method for gene-set enrichment visualization and interpretation. PLoS One. 2010;5(11):1–12. https://doi.org/10.1371/journal.pone.0013984.
39. Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B, Ideker T. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 2003;13(11):2498–504. https://doi.org/10.1101/gr.1239303.
40. Glažar P, Papavasileiou P, Rajewsky N. circBase: a database for circular RNAs. RNA. 2014;20(11):1666–70.
41. Fagerberg L, Hallström BM, Oksvold P, Kampf C, Djureinovic D, Odeberg J, Habuka M, Tahmasebpoor S, Danielsson A, Edlund K, Asplund A, Sjöstedt E, Lundberg E, Szigyarto CA-K, Skogs M, Takanen JO, Berling H, Tegel H, Mulder J, Nilsson P, Schwenk JM, Lindskog C, Danielsson F, Mardinoglu A, Sivertsson Å, von Feilitzen K, Forsberg M, Zwahlen M, Olsson I, Navani S, Huss M, Nielsen J, Ponten F, Uhlén M. Analysis of the human tissue-specific expression by genome-wide integration of transcriptomics and antibody-based proteomics. Mol Cell Proteom. 2014;13(2):397–406. https://doi.org/10.1074/mcp.M113.035600.
42. Liu S, Li B, Li Y, Song H. Circular rna circ_0000228 promotes the malignancy of cervical cancer via microrna-195-5p/lysyl oxidase-like protein 2 axis. Bioengineered. 2021;12(1):4397–406. https://doi.org/10.1080/21655979.2021.1954846.
43. Pareto V. Cours Deconomie Politique. Geneva: Librairie Droz; 1964. p. 299–345.
44. Zhao B-W, Hu L, Hu P-W, You Z-H, Su X-R, Li D-X, Chen Z-H, Zhang P. MRLDTI: a meta-path-based representation learning model for drug-target interaction prediction. In: Huang D-S, Jo K-H, Jing J, Premaratne P, Bevilacqua V, Hussain A, editors. Intelligent computing theories and application. Cham: Springer; 2022. p. 451–9. https://doi.org/10.1007/978-3-031-13829-4_39.
45. Zhang M-L, Zhao B-W, Hu L, You Z-H, Chen Z-H. Predicting drug-disease associations via meta-path representation learning based on heterogeneous information net works. In: Huang D-S, Jo K-H, Jing J, Premaratne P, Bevilacqua V, Hussain A, editors. Intelligent computing theories and application. Cham: Springer; 2022. p. 220–32. https://doi.org/10.1007/978-3-031-13829-4_18.
46. Vural H, Kaya M, Alhajj R. A model based on random walk with restart to predict circRNA-disease associations on heterogeneous network. In: Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining. ASONAM '19, pp. 929–932. Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3341161.3343514. https://doi.org/10.1145/3341161.3343514
47. Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems. 1998;30(1):107–17. https://doi.org/10.1016/S0169-7552(98)00110-X (**Proceedings of the Seventh International World Wide Web Conference**).
48. Jiang Q, Wang Y, Hao Y, Juan L, Teng M, Zhang X, Li M, Wang G, Liu Y. miR2Disease: a manually curated database for microRNA deregulation in human disease. Nucleic Acids Res. 2008;37(suppl–1):98–104. https://doi.org/10.1093/nar/gkn714.
49. Huang Z, Shi J, Gao Y, Cui C, Zhang S, Li J, Zhou Y, Cui Q. HMDD v3.0: a database for experimentally supported human microRNA-disease associations. Nucleic Acids Res. 2018;47(D1):1013–7. https://doi.org/10.1093/nar/gky1010.
50. Piñero J, Queralt-Rosinach N, Bravo A, Deu-Pons J, Bauer-Mehren A, Baron M, Sanz F, Furlong LI. DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. Database, 2015; 2015. https://doi.org/10.1093/database/bav028. bav028. https://academic.oup.com/database/article-pdf/doi/10.1093/database/bav028/16975988/bav028.pdf
51. Lan W, Zhu M, Chen Q, Chen B, Liu J, Li M, Chen Y-PP. CircR2Cancer: a manually curated database of associations between circRNAs and cancers. Database, 2020; 2020. https://doi.org/10.1093/database/baaa085. baaa085. https://academic.oup.com/database/article-pdf/doi/10.1093/database/baaa085/34283838/baaa085.pdf
52. Yao D, Zhang L, Zheng M, Sun X, Lu Y, Liu P. Circ2Disease: a manually curated database of experimentally validated circRNAs in human disease. Sci Rep. 2018;8(1):11018. https://doi.org/10.1038/s41598-018-29360-3.
53. Fan C, Lei X, Fang Z, Jiang Q, Wu F-X. CircR2Disease: a manually curated database for experimentally supported circular RNAs associated with various diseases. Database. 2018. https://doi.org/10.1093/database/bay044.
54. Ghosal S, Das S, Sen R, Basak P, Chakrabarti J. Circ2Traits: a comprehensive database for circular RNA potentially associated with disease and traits. Front Genet. 2013. https://doi.org/10.3389/fgene.2013.00283.
55. Lei X, Fang Z, Chen L, Wu F-X. PWCDA: path weighted method for predicting circRNA-disease associations. Int J Mol Sci. 2018;19(11):3410. https://doi.org/10.3390/ijms19113410.
56. Zhao Q, Yang Y, Ren G, Ge E, Fan C. Integrating bipartite network projection and KATZ measure to identify novel circRNA-disease associations. IEEE Trans Nanobiosci. 2019;18(4):578–84. https://doi.org/10.1109/TNB.2019.2922214.
57. Lei X-J, Fang Z, Guo L. Predicting circRNA-disease associations based on improved collaboration filtering recommendation system with multiple data. Front Genet. 2019. https://doi.org/10.3389/fgene.2019.00897.
58. Lu C, Zeng M, Wu F-X, Li M, Wang J. Improving circRNA-disease association prediction by sequence and ontology representations with convolutional and recurrent neural networks. Bioinformatics. 2020;36(24):5656–64. https://doi.org/10.1093/bioinformatics/btaa1077.
59. Zhang H-Y, Wang L, You Z-H, Hu L, Zhao B-W, Li Z-W, Li Y-M. iGRLCDA: identifying circRNA-disease association based on graph representation learning. Brief Bioinform. 2022. https://doi.org/10.1093/bib/bbac083.

60. Zhao B-W, Hu L, You Z-H, Wang L, Su X-R. HINGRL: predicting drug-disease associations with graph representation learning on heterogeneous information networks. Brief Bioinform. 2021. https://doi.org/10.1093/bib/bbab515.
61. Li G, Luo J, Wang D, Liang C, Xiao Q, Ding P, Chen H. Potential circRNA-disease association prediction using DeepWalk and network consistency projection. J Biomed Inform. 2020;112: 103624. https://doi.org/10.1016/j.jbi.2020.103624.
62. Li G, Yue Y, Liang C, Xiao Q, Ding P, Luo J. NCPCDA: network consistency projection for circRNA-disease association prediction. RSC Adv. 2019;9(57):33222–8. https://doi.org/10.1039/C9RA06133A.
63. Zhang Y, Lei X, Fang Z, Pan Y. Circrna-disease associations prediction based on metapath2vec++ and matrix factorization. Big Data Min Anal. 2020;3(4):280–91. https://doi.org/10.26599/BDMA.2020.9020025.
64. Deepthi K, Jereesh AS. An ensemble approach for circrna-disease association prediction based on autoencoder and deep neural network. Gene. 2020;762:145040. https://doi.org/10.1016/j.gene.2020.145040.
65. Wang L, You Z-H, Huang D-S, Li J-Q. MGRCDA: Metagraph recommendation method for predicting circRNA-disease association. IEEE Trans Cybern. 2021. https://doi.org/10.1109/TCYB.2021.3090756.
66. Zheng K, You Z-H, Li J-Q, Wang L, Guo Z-H, Huang Y-A. iCDA-CGR: Identification of circRNA-disease associations based on chaos game representation. PLoS Comput Biol. 2020;16(5):1–22. https://doi.org/10.1371/journal.pcbi.1007872.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.