

Detecting modules in biological networks by edge weight clustering and entropy significance

Original

Detecting modules in biological networks by edge weight clustering and entropy significance / Lecca, Paola; Re, Angela.
- In: FRONTIERS IN GENETICS. - ISSN 1664-8021. - ELETTRONICO. - 6:Aug(2015), p. 265.
[10.3389/fgene.2015.00265]

Availability:

This version is available at: 11583/2970530 since: 2022-08-08T09:43:51Z

Publisher:

Frontiers media

Published

DOI:10.3389/fgene.2015.00265

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Detecting modules in biological networks by edge weight clustering and entropy significance

Paola Lecca^{1†} and Angela Re^{2*}

¹ Centre for Integrative Biology, University of Trento, Italy, ² Laboratory of Translational Genomics, Centre for Integrative Biology, University of Trento, Trento, Italy

OPEN ACCESS

Edited by:

Spyros Petrakis,
Aristotle University
of Thessaloniki, Greece

Reviewed by:

Monika Heiner,
Brandenburg Technical University
Cottbus-Senftenberg, Germany
Qiangfeng Cliff Zhang,
Stanford University, USA

*Correspondence:

Angela Re,
Laboratory of Translational Genomics,
Centre for Integrative Biology,
University of Trento,
Via Sommarive 9 - 38123 Povo,
Trento, Italy
angela.re@unitn.it

† Present Address:

Paola Lecca,
Member of Association for Computing
Machinery, New York, USA

Specialty section:

This article was submitted to
Systems Biology,
a section of the journal
Frontiers in Genetics

Received: 25 March 2015

Accepted: 30 July 2015

Published: 27 August 2015

Citation:

Lecca P and Re A (2015) Detecting
modules in biological networks by
edge weight clustering and entropy
significance. *Front. Genet.* 6:265.
doi: 10.3389/fgene.2015.00265

Detection of the modular structure of biological networks is of interest to researchers adopting a systems perspective for the analysis of omics data. Computational systems biology has provided a rich array of methods for network clustering. To date, the majority of approaches address this task through a network node classification based on topological or external quantifiable properties of network nodes. Conversely, numerical properties of network edges are underused, even though the information content which can be associated with network edges has augmented due to steady advances in molecular biology technology over the last decade. Properly accounting for network edges in the development of clustering approaches can become crucial to improve quantitative interpretation of omics data, finally resulting in more biologically plausible models. In this study, we present a novel technique for network module detection, named WG-Cluster (Weighted Graph CLUSTERing). WG-Cluster's notable features, compared to current approaches, lie in: (1) the simultaneous exploitation of network node and edge weights to improve the biological interpretability of the connected components detected, (2) the assessment of their statistical significance, and (3) the identification of emerging topological properties in the detected connected components. WG-Cluster utilizes three major steps: (i) an unsupervised version of k-means edge-based algorithm detects sub-graphs with similar edge weights, (ii) a fast-greedy algorithm detects connected components which are then scored and selected according to the statistical significance of their scores, and (iii) an analysis of the convolution between sub-graph mean edge weight and connected component score provides a summarizing view of the connected components. WG-Cluster can be applied to directed and undirected networks of different types of interacting entities and scales up to large omics data sets. Here, we show that WG-Cluster can be successfully used in the differential analysis of physical protein–protein interaction (PPI) networks. Specifically, applying WG-Cluster to a PPI network weighted by measurements of differential gene expression permits to explore the changes in network topology under two distinct (normal vs. tumor) conditions. WG-Cluster code is available at <https://sites.google.com/site/paolaleccapersonalpage/>.

Keywords: protein–protein network, weighted network, node weight, edge weight, clustering, connected component, entropy

1. Introduction

With biology increasingly becoming a data-rich field, objectives of systems biology research include organizing molecular interactions as networks and characterizing their structure, dynamics, and controllability. Since the turn of the century, high-throughput interaction mapping has emerged as an extremely useful approach to identify the constituents and connections of these networks. For instance the systematic identification of pairwise protein interactions (Rual et al., 2005; Petschnigg et al., 2014) or protein associations into complexes (Havugimana et al., 2012) has been enormously valuable both for understanding the function of individual proteins and for elucidating the organizing principles of the cellular physical architecture. Additional types of interactions have been charted including protein-DNA (Chen et al., 2015), protein-RNA (Moore et al., 2014; Re et al., 2014) and kinase-substrate (Linding et al., 2007; Varjosalo et al., 2013) interactions. Many of the molecular interaction data generated find their way into database resources that are available online (Turner et al., 2010; Horn et al., 2014; Orchard et al., 2014). The ability to generate, process and integrate omics data is instrumental to increasingly faithful reconstructions of the information flow in biological systems. In this vein, the conceptualization of biological systems as networks and the subsequent reconstruction of their modular organization acquire great interest (Barabási and Oltvai, 2004; Barabási et al., 2011; Ideker and Krogan, 2012). The notion of a module refers to a discrete entity whose constituent elements are similar in some quantifiable (e.g., chemical, physical, or functional) property and/or in the profile of their relationships. Biology displays many examples of modules which generally accomplish relatively separable functions such as nucleic acid synthesis, DNA replication, mitotic spindle assembly and protein degradation (Hartwell et al., 1999; Barabási and Oltvai, 2004).

In recent years, a rich collection of computational approaches has emerged for module detection in weighted networks, where weights can be constrained by topological or alternative numerical properties of nodes (for example, node molecular activity extracted from transcriptomics profiling) and edges (for example, edge confidence). Aside from weight assignment either to nodes (Ideker et al., 2002; Bader and Hogue, 2003) or to edges (Tanay et al., 2004; Liu et al., 2009; Pandey et al., 2014), clustering algorithms differ in the procedures for finding modules including, for example, simulated annealing (Ideker et al., 2002), greedy (Chuang et al., 2007; Nacu et al., 2007), genetic (Klammer et al., 2007), and network propagation (Vandin et al., 2011; TCGA Research Network, 2013) algorithms. Despite all of this exciting research in network clustering, some limitations stand out as remarkable. First, processing tens of thousands of nodes and the edges among them is hard to accomplish in fast timescales. Second, albeit equally interesting properties, it remains unclear how to meaningfully account for both node and edge weights in a module detection procedure.

Here, we present a novel algorithm for modular structure detection, named WG-Cluster (Weighted Graph CLUSTERing), which seeks to address previous shortcomings to detect modules. Within WG-Cluster, a module is defined as a connected

component where nodes are characterized by homogeneous weights and are connected by edges of homogeneous weights. To this aim, WG-Cluster combines an edge-based network clustering with a fast-greedy algorithm. The treatment of network edge weights within WG-Cluster represents a novelty compared to most clustering algorithms since, by the initial edge-based network clustering, network edge weights underlie the subsequent detection and prioritization of the connected components. Furthermore, the procedural choice adopted by WG-Cluster permits to obtain modules, homogeneous not only in node weights but also in edge weights, without discernible additional cost in computational efficiency. Module prioritization can become particularly useful in applications related to differential network analysis where the primary goal is to identify modules changing across different conditions. Finally, it is worth mentioning here also the introduction of a measure of the significance of the returned connected components which is based on node weights. WG-Cluster is here applied for the analysis of a differential network, i.e., a network where node and edge weights are defined by the changes observed in node and edge numerical properties between two conditions. Differential network analysis is useful to tackle the dynamic nature of molecular interactions, for instance as a consequence of environmental shifts. Computational integration of a network with molecular profiles acquired in different contexts has shown a popular approach to extract context-dependent responsive modules, which mark strikingly changed regions of the network. The input network for the current WG-Cluster application is a differential network, which was obtained by integrating a physical protein-protein interaction (PPI) network with changes in gene expression between a normal and tumor conditions. Our analysis showed that WG-Cluster is useful for comprehensively analysing the quantitative changes affecting nodes or interactions in the network and for recognizing modules which link to functional properties.

2. Materials and Methods

2.1. Data Description and Pre-processing

We gathered multi-assay omics data to define the weighted network which is the primary input to WG-cluster. We collected PPIs from the open-access IntAct database which adopts a merging algorithm and a scoring system to provide richly annotated molecular interaction data. IntAct PPIs are described in the controlled vocabulary specified by the Proteomics Standards Initiative for Molecular Interaction (PSI-MI) data (Hermjakob et al., 2004) and adhere to the guidelines (Orchard et al., 2007) about the Minimum Information required for reporting a Molecular Interaction Experiment, which were supplied by the International Molecular Exchange (IMEx) consortium. PPIs involving human protein entities were selected and downloaded along with their confidence scores. Protein identifiers defined by the Universal Protein Resource (Uniprot) protein accessions (<http://www.uniprot.org/>) were mapped to gene identifiers defined by the HUGO Gene Nomenclature Committee (HGNC) gene symbols (<http://www.genenames>).

org/). We next integrated the IntAct PPIs with tumor-dependent changes in messenger RNA (mRNA) expression profiles. Processed gene expression data related to colon adenocarcinoma were downloaded from The Cancer Genome Atlas (TCGA) (<http://cancergenome.nih.gov/>). mRNA profiles were generated from 155 tumor and 19 normal tissue samples. Processed data were lowess normalized and collapsed by gene symbol (\log_2 scale). A differential co-expression score was computed for each gene pair, by subtracting the pairwise Pearson's correlation coefficient in the tumor condition from that in the normal condition. Next, the IntAct PPI confidence scores were multiplied by the differential co-expression scores to estimate the change in the interaction strength resulting from the differential co-expression of the mRNAs encoding the interacting proteins. The product between the IntAct score and the differential co-expression score defines the final weight of an edge in the differential network. The weight of a node in the differential network was obtained by computing the ratio between the average values of mRNA expression across samples in the normal and tumor conditions (mRNA fold change). This differential network, where both nodes and edges were weighted, was the primary input to the WG-Cluster algorithm.

2.2. WG-Cluster

The WG-Cluster algorithm is implemented in R (R software available at <http://www.r-project.org>), which provides one of the most widely used, most flexible and mature open source environments. For the most computationally intense tasks WG-Cluster employs built-in R functions implemented as a C(++) or Fortran code, that are optimized and faster than functions coded in R from scratch. The input data consist of the network edges reported in Simple Interaction File (SIF) format (Cytoscape, 2015) and of node weights reported in tabular format (node, weight). The algorithm sequentially executes three computational modules. First, it estimates the optimal number of clusters (sub-graphs) that split up the graph (i.e., network) and executes a Lloyd's K-means clustering (Du et al., 2006) of the edge weights to detect sub-graphs with edges of similar weights. Second, a fast-greedy modularity optimization procedure (Clauset et al., 2004) finds (if any) the connected components (i.e., modules) in each sub-graph. An entropy score is computed for each connected component and is used as a measure of the statistical significance of the connected component. Finally, an analysis of the convolution between sub-graph mean edge weight and connected component entropy allows for a summarizing view of both properties in the detected connected components (Figure 1). In the following, we give the details about each computational module of WG-Cluster. Hereafter, we will denote with V the number of vertices and with NE the number of edges in the input graph.

2.2.1. Detection of Sub-graphs

The optimal number of sub-graphs which partition the input graph is estimated by minimizing the total within-clusters sum of squares (WCSS) obtained with a K-means procedure. For a set of

edge weights $\mathbf{w} = (w_1, w_2, \dots, w_{NE})$, K-means clustering tries to find a set of K sub-graphs $S = (S_1, S_2, \dots, S_K)$ that is a solution to the minimization problem:

$$WCSS = \sum_{i=1}^K \sum_{\mathbf{w} \in S_i} \|\mathbf{w} - \mu_i\|^2$$

where μ_i is the mean of the edge weights \mathbf{w} in the sub-graph S_i .

An elbow in the curve interpolating the points ($n_{\text{sub-graphs}}$, WCSS) suggests the appropriate number of sub-graphs n_{optimal} . In our implementation, n_{optimal} is estimated as the minimum value of n_{clusters} at which the first derivative of WCSS w.r.t. $n_{\text{sub-graphs}}$ is null within a tolerance $0 < \epsilon \ll 1$, i.e.,

$$\left| \frac{d \text{WCSS}}{dn_{\text{sub-graphs}}} \right| \leq \epsilon.$$

The first derivative of the curve ($n_{\text{sub-graphs}}$, WCSS) is calculated by the Stineman algorithm (Johannesson and Bjornsson, 2012). Algorithm 1 reports the pseudo-code of the first module of WG-Cluster.

The problem of WCSS minimization is known to be NP-hard, implying long running times, that can become unacceptable in case of biological networks with thousands of nodes and tens of thousands of edges. Furthermore, if the input data do not have a strong clustering structure, the procedure may not converge. For this reason, WG-Cluster adopts the Lloyd's algorithm whose complexity is linear in the number of edges and number of sub-graphs, and is recommended in case of data poorly clustered (Du et al., 2006). Algorithm 2 presents the pseudo-code of the Lloyd's K-means. Those iterations are repeated until the centroids stop changing, within a tolerance quantified by the parameter `threshold` (see the pseudo-code 2).

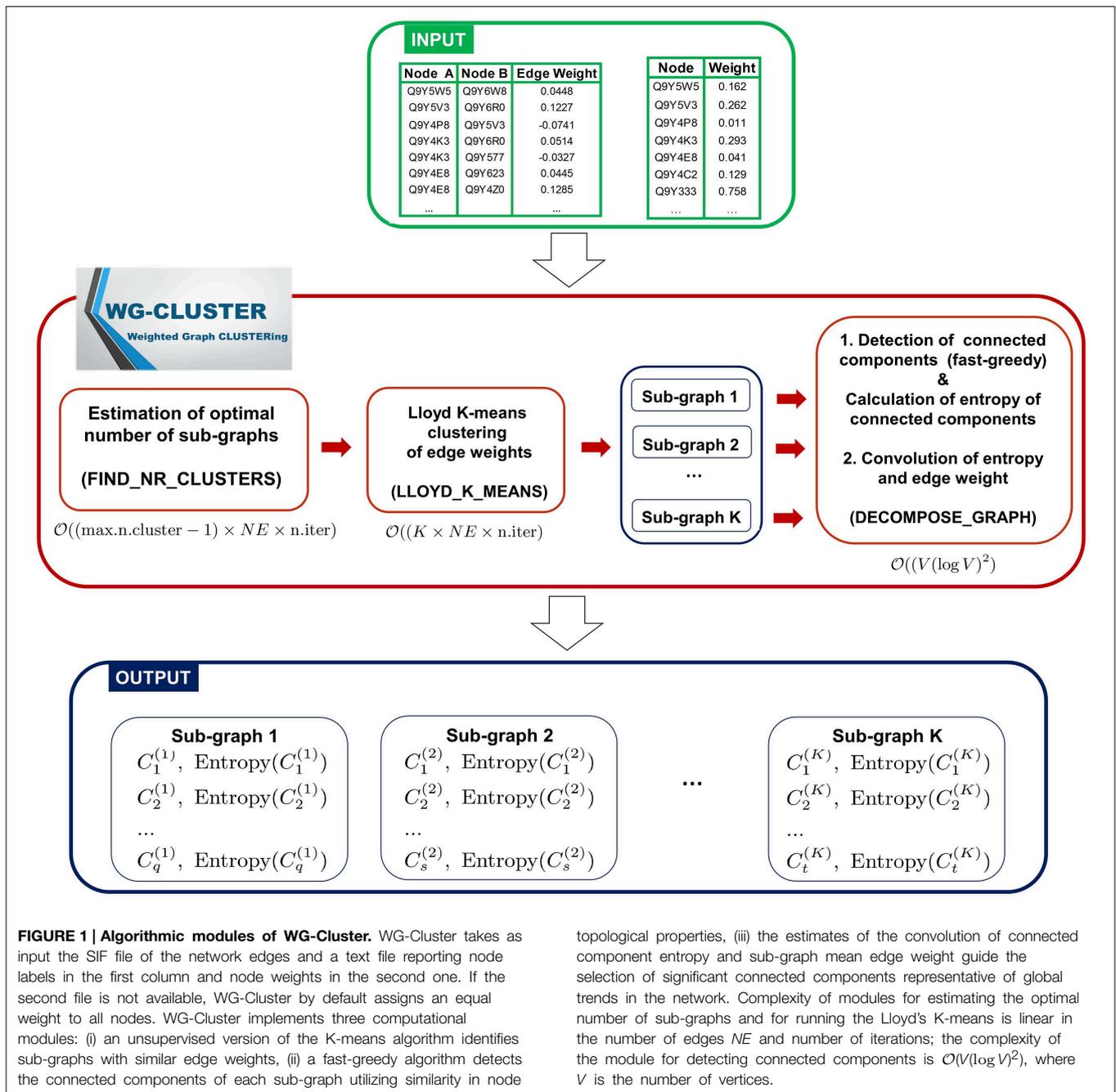
In Supplementary Material (Section 1.1) we present the exploratory analysis of other clustering approaches and the motivation of the choice of the K-means algorithm in WG-Cluster.

2.2.2. Detection of Connected Components

Each sub-graph S_i ($i = 1, \dots, K$) returned by the K-means clustering is decomposed into connected components $C_l^{(i)}$ (with $l = 1, 2, \dots, L_i$, where L_i is the number of connected components in the sub-graph S_i) via a fast-greedy optimization procedure (Clauset et al., 2004), as illustrated in Figure 2. The entropy of each connected component is calculated as follows:

$$E_{C_l^{(i)}} = - \sum_{j=1}^{N(C_l^{(i)})} \frac{p_j \log_2 p_j}{d_j} \tag{1}$$

where $N(C_l^{(i)})$ is the number of nodes in the connected component $C_l^{(i)}$, p_j is the fold change of the expression level (from normal to tumor condition) of gene j (normalized between 0 and 1) and d_j is the sum of the weights of the edges adjacent to the node representing gene j (known as *node strength*). Denoting



with $D^{(j)}$ the number of nodes directly connected to node j , d_j is thus defined as

$$d_j = \sum_{h=1}^{D^{(j)}} w_{jh}$$

where w_{jh} is the edge weight between the node j and its directly connected node h .

The entropy is used as a measure of significance of the connected components. In order to establish a threshold on

the entropy significance, we generated for each connected component $C_l^{(i)}$ an ensemble of 100 random connected components with the same degree distribution of the reference connected component $C_l^{(i)}$.

A connected component is considered significant, and retained, if its entropy value is more than three standard deviations far from the mean entropy of the corresponding ensemble of random connected components. Let denote with $\{C_l^{(i')}\}$, where $l' \in \{1, 2, \dots, L_i\}$ with $L_i' \leq L_i$, and $i' \in \{1, 2, \dots, K'\}$ with $K' \leq K$.

Algorithm 1 | Compute the optimal number of sub-graphs K

```

1: procedure FIND_NR_SUB_GRAPHS(edge.weights,
   max.n.sub.graphs, seed)
2:
3:    $NE \leftarrow$  Number of edges of the graph
4:
5:   1. Calculate the within-cluster sum of squares (wcss) via a
   K-means solution.
6:
7:    $wcss[1] \leftarrow (NE - 1) \times \text{Variance}(\text{edge.weights})$ 
8:
9:   set.seed(seed)
10:  for (i in 2:max.n.sub.graphs) do
11:     $wcss[i] \leftarrow \sum_1^i (\text{calculate.wcss}(\text{K-means}(\text{edge.weights}, \text{centroids} = i)))$ 
12:  end for
13:
14:
15:  2 Estimate  $d$  WCSS/ $dn.sub.graphs$  with the Stineman
   algorithm.
16:
17:   $n.sub.graphs \leftarrow 1:\text{max.n.sub.graphs}$ 
18:   $wcss.derivative \leftarrow \text{Stineman.derivative}(n.sub.graphs,$ 
    $wcss)$ 
19:
20:  3. Set a tolerance value.
21:
22:   $\text{tolerance} \leftarrow \epsilon$ 
23:
24:  4. Find the first local minimum of  $d$  WCSS/ $dn.sub.graphs$ 
25:
26:   $wcss.derivative.null \leftarrow \{-\epsilon \leq wcss.derivative \leq \epsilon\}$ 
27:   $K \leftarrow wcss.derivative.null[1]$ 
28:
29:  5. Return the optimal number of sub-graphs  $K$ .
   Return  $K$ 
30:
31: end procedure

```

2.2.3. Convolution of Mean Edge Weight and Entropy

Both the connected component entropy and the mean weight of the edges of the sub-graph to which a connected component belongs are considered to classify the connected components.

The convolution of the entropy of selected connected components (E_{selected}) with the mean edge weight MW of the sub-graphs to which they belong is performed as follows:

$$E_{\text{selected}}[h] * MW[h] = \sum_q E_{\text{selected}}[q] \cdot MW[q - h] \quad (2)$$

where $E_{\text{selected}} = \{E_{C_i^{(i')}}\}$ is the vector of the entropies of the significant connected components, and $MW =$

Algorithm 2 | Lloyd's K-means algorithm

```

1: procedure LLOYD_K_MEANS(edge.weights, K, distance)
2:
3:   1. Randomly choose  $K$  items from the edge weights
   vector and use these as the initial means.
4:
5:   2. Iterations of assignments and centroid
   recalculation.
6:   while distance(centroids, edge.weights) > threshold do
7:     a. Assign edge weights to the centroids
8:     for  $i \leq NE$  do
9:       Assign edge.weights[i] to closest sub-graph
       according to the distance measure.
10:    end for
11:     b. Recalculate centroids.
12:   end while
13: end procedure

```

$\{(1/NE^{(i')}) \sum_{l=1}^{NE^{(i')}} w_l\}$ is the mean edge weight of the sub-graph to which they belong.

The convolution in Equation (2) calculates the area overlap between the probability distributions of the entropy and of the mean edge weight as a function of the amount by which one of the distribution is translated. The area of the overlap of the two distribution measures the similarity between the entropy and mean edge weight distribution. The density of the convolution is a spectrum of the frequency of this similarity score and offers a way to classify the connected components by their membership to intervals of frequency corresponding to local maxima or minima of the convolution density. Maxima of the convolution density correspond to the most frequent values of similarity between entropy and mean edge weight, whereas local minima correspond to the least frequent values of similarity. Then, connected components can be classified according to the frequency of the convolution between their entropy and the mean edge weight of the sub-graph to which they belong. Algorithm 3 provides the steps of the pseudo-code implementing the procedure of detection and selection of significant connected components.

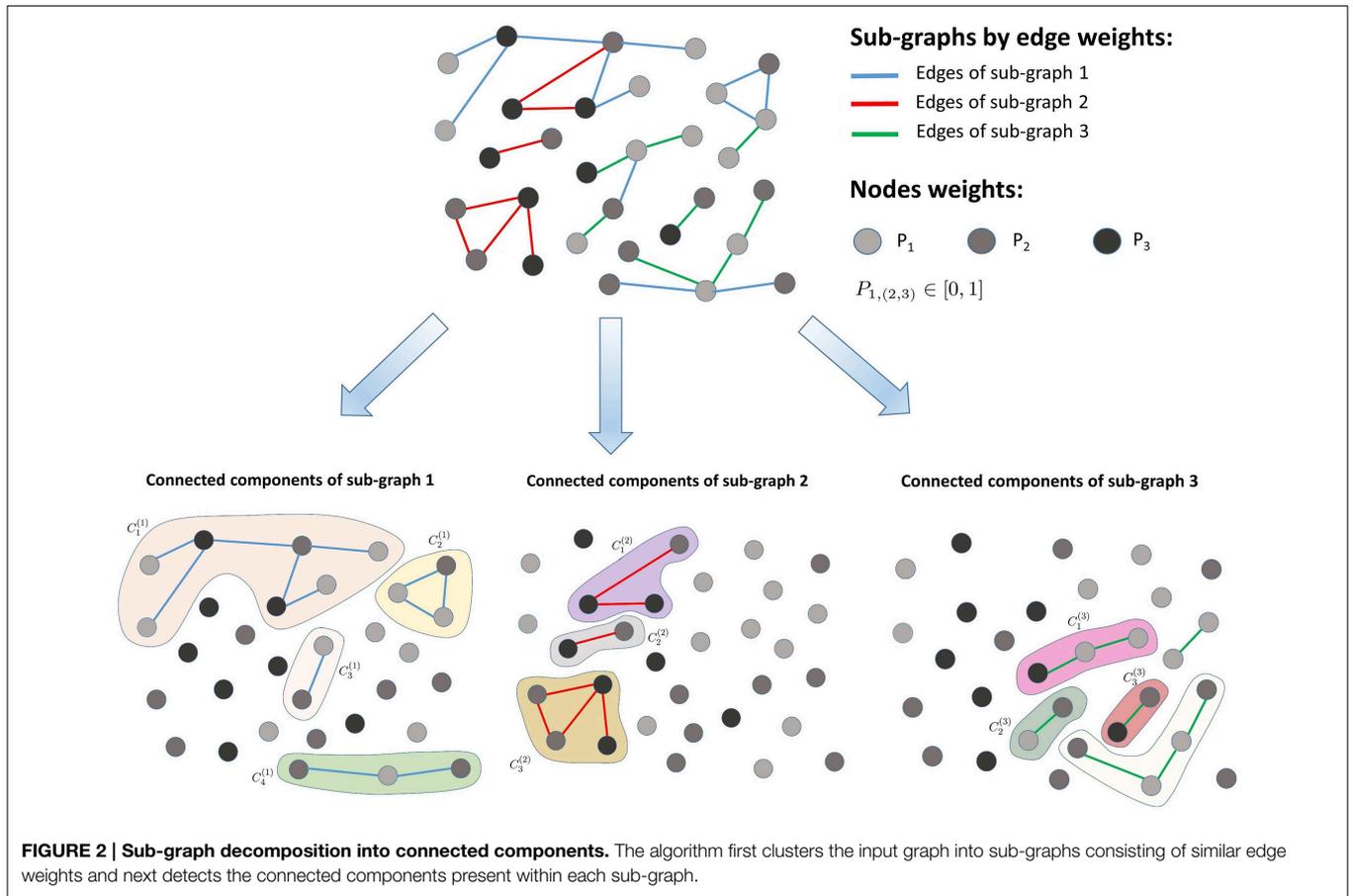
2.3. Functional Analysis of Connected Components

Enrichment analysis based on the generic Gene Ontology (GO) slim (<http://geneontology.org/>), a cut-down version of the Gene Ontology annotations, was conducted for each retained connected component (hypergeometric test). GO enrichment p -values were transformed in Benjamini-Hochberg false discovery rate (FDR) values and retained at the significance level of 0.05.

3. Results

3.1. Performances on synthetic data

We evaluated the performances of WG-Cluster in processing Erdős-Rényi random graphs, consisting of 500 nodes and an increasing number of edges, in terms of user CPU running time.



Algorithm 3 | Detection and selection of connected components

```

1: procedure DECOMPOSE_GRAPH(sub-graphs, node.weights)
2:
3: 1. Detection of connected components and calculation of their entropy.
4:
5:   for (i in 1:K) do
6:
7:     a. Fast-greedy decomposition of the i-th sub-graph into connected components.
8:     connected.components[[i]] ← fast.greedy.decomposition(sub-graph[i])
9:
10:    b. Entropy calculation.
11:
12:    for (l in 1:Li) do
13:      (b.1) entropy of connected components with Equation (1).
14:
15:      connected.components.entropy[l] ←
16:      entropy(connected.components[[i]][l], node.weights)
17:

```

```

18:      (b.2) Generate an ensemble of random weighted Erdős-Renyi connected components.
19:
20:      for (v in 1:100) do
21:        random.cc.component ←
22:        erdos.renyi.graph(nr.of.nodes= $N(C_l^{(i)})$ , nr.of.edges= $NE(C_l^{(i)})$ )
23:        edge.weights.random.cc.component ← Unif(0, 1)
24:        node.weights.random.cc.component ← Unif(0, 1)
25:        random.cc.entropies[v] ← calculate.entropy(random.cc.component, node.weights.random.cc.component, edge.weights.random.cc.component)
26:
27:      end for
28:
29:      (b.3) Calculate the mean of the entropies of the ensemble of random connected components.
30:      random.cc.entropy[l] ← calculate.mean.entropy(random.cc.entropies)
31:
32:      (b.4) Select connected components.

```

```

33:         if cc.entropy[l]  $\notin$   $[-3\sigma + \text{random.cc.entropy}[l],$ 
            $+3\sigma + \text{random.cc.entropy}[l]]$  then
34:             selected.connected.components  $\leftarrow C_l^{(i)}$ 
35:         else
36:             discard  $C_l^{(i)}$ 
37:         end if
38:     end for
39: end for
40:
41: 2. Convolution of Entropy (E) and mean edge weight
   (MW).
42:
43:     (a) Calculate the density of the convolution estimated
       by Equation (2).
44:     density.of.convolution  $\leftarrow$  density(convolve (E, MW))
45:     (b) Detect the maxima of the convolution density.
46:     (c) Select and return the connected components whose
       values of convolution of (E, MW) fall under convolution
       density maxima.
47: end procedure

```

Edge weights were drawn from a uniform distribution between 0 and 1 and clustered in 10 groups. A uniform distribution between 0 and 1 was also used to obtain node weights. We compared WG-Cluster running times to the running times of three widely used deterministic hierarchical approaches to graph clustering: (i) edge betweenness based clustering, (ii) label propagation, and (iii) InfoMap, which were selected because they handle directed (as well as undirected) and weighted networks as WG-Cluster does (see Table 1 for a summary of the currently available deterministic clustering methods implemented in R). Non-deterministic clustering algorithms [e.g., Walktrap (Pons and Latapy, 2005), Spinglass (Reichardt and Bornholdt, 2006), and label propagation (Raghavan et al., 2007)] were left out of this comparative analysis since they require the determination of the number of runs needed to build a consensus partition. This parameter often depends on the topological structure of the graph and can remarkably affect the performances (that are usually satisfactory on single runs). We also excluded from the comparison the algorithms that do not handle the processing of undirected networks [e.g., Leading eigenvectors, (Newman, 2006)]. From this analysis, WG-Cluster showed to outperform the alternative algorithms (Figure 3).

In Supplementary Material, Section 1.2, we provide a more comprehensive analysis of the time complexity of WG-Cluster applied to random graphs of increasing number of edges and number of nodes.

Finally, further improvements in efficiency will be tested in the next version of WG-Cluster by the usage of recent libraries developed specifically to perform an optimized memory-efficient management of large datasets. The input/output and data rearrangement operations on large datasets are computationally time consuming, and their speeding is one of the main research topic engaging the developers of the majority of programming languages. R proposed two major solutions to optimize the efficiency of massive dataset processing (Kane and Emerson, 2013; Adler et al., 2014). Using these solutions,

TABLE 1 | Summary of widely used hierarchical methods for module detection.

Method	Type of graph	Weighted edges	Weighted nodes
Edge-Betweenness (Girvan and Newman, 2001)	Directed and undirected	True	False
Fast-greedy (Clauset et al., 2004)	Directed and undirected	True	False
InfoMap (Rosvall and Bergstrom, 2008)	Directed and undirected	True	True

"True" and "False" in the two last columns stand for "the method can process also" and "the method does not process," respectively. For instance, edge-betweenness clustering method can process and take into account edge weights, but it does not handle information about node weights.

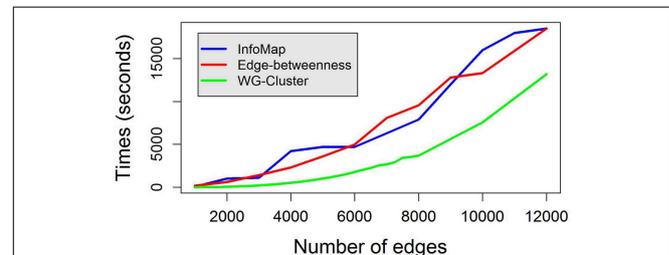


FIGURE 3 | Running times to cluster random weighted graphs with increasing number of edges.

WG-Cluster running time on a random weighted graph of 500 nodes and an increasing number of edges is compared with that achieved by the edge betweenness graph clustering algorithm (Girvan and Newman, 2001) and that of InfoMap (Rosvall and Bergstrom, 2008). Each algorithm was utilized in its R implementation on a desktop Windows 8.1 PC with a 3.1 GHz CPU. WG-Cluster ensured faster running time and a RAM usage inferior to 3Gb.

WG-Cluster could take advantage of the benefits of R (i.e., interactive data analysis and rich, flexible statistical programming environment), and, at the same time, of the benefit of C(++) language, i.e., an optimized memory-efficient management of big datasets.

3.2. Application

Biological systems are highly dynamical entities by depending on environment, tissue type, disease state or development. Nonetheless, relatively little effort has been spent in differential network analysis, i.e., the analysis of the changes occurring in a network in response to different conditions. Even though an increasing number of studies seek to analyse the dynamics of networks directly, through experimental mapping of networks across multiple conditions (Grossmann et al., 2015; Martin et al., 2015), a longstanding approach in differential network biology is to construct differential networks by integrating static (at standard laboratory conditions) molecular interaction networks (e.g., PPI networks) with changes observed in messenger RNA expression in different biological conditions (de Lichtenberg et al., 2005). The resulting differential network is a weighted network where node weights reflect the changes in mRNA expression levels and where edge weights reflect the changes in

interaction strengths due to differential mRNA co-expression levels under the two conditions. It is worth noting that the strongest differential interactions are not necessarily the strongest ones in the static networks. Since both node and edge properties are deeply ingrained in the clustering procedure, WG-Cluster can provide a unique view of the differences in network topology between any two biological conditions.

As a proof-of-principle, we applied the WG-Cluster approach to analyse the differential PPI network that arises when tumor and normal conditions are contrasted. The current application focused on the colorectal cancer which stands among the most common cancers with more than 1.2 million new cases and about 600,000 deaths per year worldwide (Jemal et al., 2011). Messenger RNA expression data were obtained from The Cancer Genome Atlas which, importantly, provides samples from tumor tissues and from matched normal tissues. We acquired PPIs from the IntAct database because it provides a heuristic scoring system which relies on the available annotation evidences associated with an interacting pair of proteins. The differential network was constructed as follows: a node was weighted by the mRNA fold change and an edge was weighted by multiplying the IntAct PPI confidence score with the difference of the mRNA co-expression scores between the normal/tumor conditions. In the Supplementary Material (Sections 1.3 and 1.4), we provide a full description of this network edge weight model. Furthermore, we show that this network edge weighting approach leads to improved clustering quality compared to the classical approach which is based only on differential co-expression. The differential network consisted of 5569 nodes and 18,078 edges, out of which 8880 were strengthened and 9198 weakened in the tumor condition compared to the normal one.

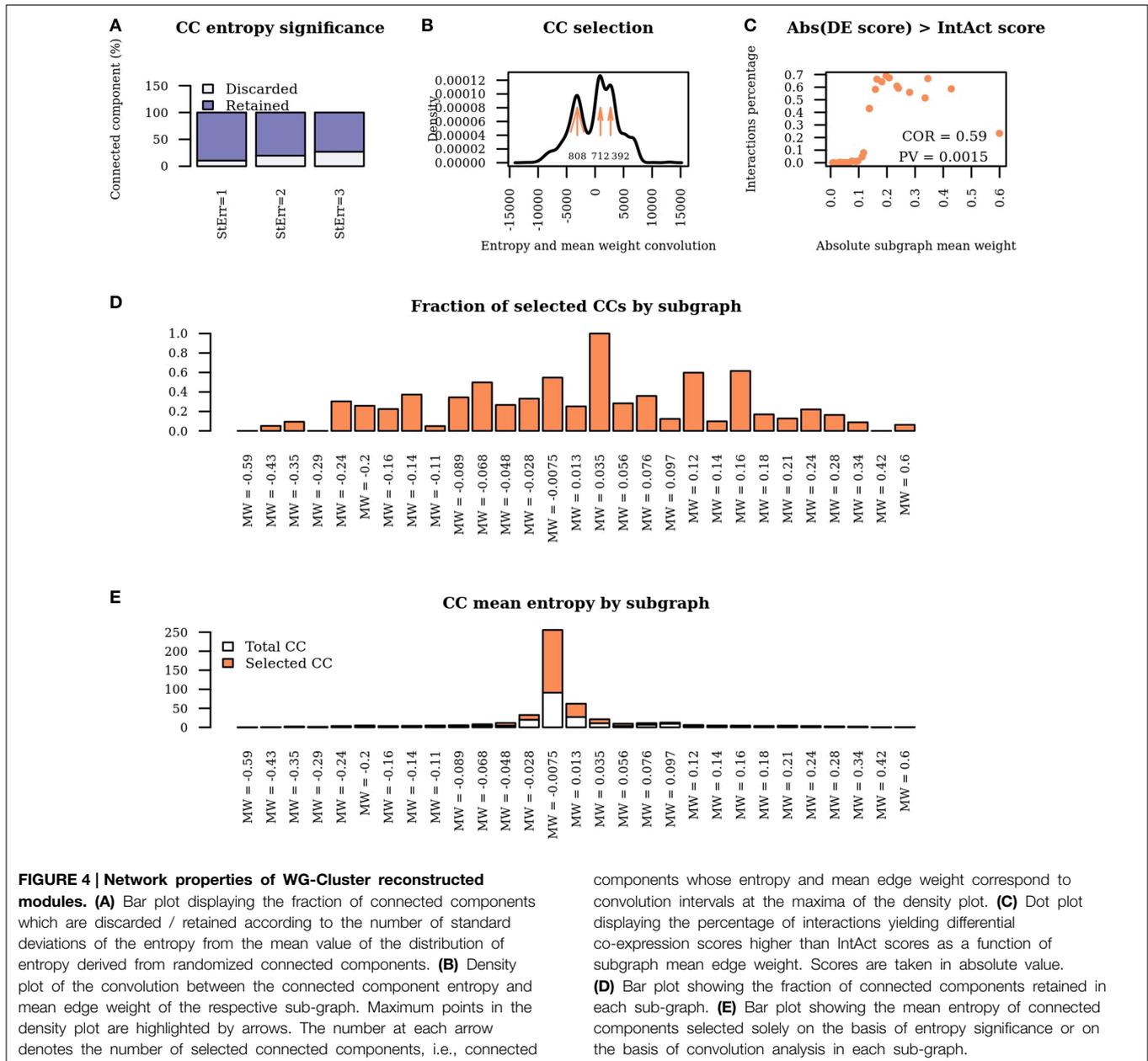
Applying WG-cluster to the differential network detected 6215 connected components which were arranged in 29 sub-graphs of distinct mean edge weights. Upon connected components detection, WG-Cluster allows the estimation of the statistical significance of the entropy of each connected component by comparing the observed value against the distribution of entropies obtained from appropriately randomized connected components. The rate of connected component exclusion appeared stably moderate when we incremented the number of standard deviations from the expected entropy value; setting this number at three resulted in the exclusion of 26.87% of connected components (Figure 4A). Statistically significant connected components can be prioritized by any sort of network property such as mean edge weight of the sub-graph, or entropy or number of nodes by connected component. It is noteworthy that the numerical features associated with each connected component provide complementary information. For instance, correlation between mean edge weight and entropy values was not statistically significant (Spearman's coefficient = -0.02 , $P = 0.15$). Since the mean weight of the edges in a sub-graph reflects the mean change in interactions strength and the entropy of a connected component reflects the mRNA expression changes, the observed lack of correlation is interesting because it is in agreement with previous data showing that the strongest differential interactions

do not necessarily involve the strongest differential genes (Ideker and Krogan, 2012).

The last WG-Cluster step implements the convolution of the probability distribution of the connected component entropy with that of the sub-graph mean edge weight. This operation offers an appealing way to classify connected components in terms of both of those properties which, in our vision, are of equal interest. Since we were interested into obtaining a summarizing view of the network clustering, we selected the connected components yielding the most frequent convolution values (Figure 4B). We then interpreted those convolution values in terms of the corresponding sub-graph mean edge weight and connected component entropy values.

The number of the connected components obtained was found to increase in sub-graphs yielding lower mean edge weight (Figure 4D); conversely, no trend was detectable by analysing the mean entropy of the selected connected components resulting from each sub-graph (Figure 4E). Since the edge scores in the differential network result from the product of the IntAct scores with the differential co-expression scores, we verified that a low mean edge weight depended on low differential co-expression score, which resulted to be the case; indeed, the percentage of interactions where the differential co-expression score was higher than the IntAct score positively correlated with the sub-graph mean edge weight (Figure 4C).

In summary, by a general survey of WG-Cluster outcome, the majority of the detected connected components were found to consist of moderately changing interactions. More interestingly, the arrangement of the detected connected components by decreasing sub-graph mean edge weight (as shown in Figure 4D), which is inherent to WG-Cluster, streamlined the identification of connected components of markedly changing interactions. Those connected components, albeit limited in number, are undoubtedly the most interesting for unveiling the most striking changes in network topology between tumor/normal conditions (Figure 4D). Gene Ontology enrichment analysis was conducted to broadly assess the functional significance of module selection since exploring the fine details of specific modules is out of the scope of our study. This analysis showed that sub-graph clustering by mean edge weight broadly corresponded to a clustering of GO biological processes (Figure 5). Genes involved in cell cycle, cell death, mRNA processing and protein modification processes were found to be overrepresented in modules of weakened interactions in the tumor compared to the normal condition (sub-graph positive mean edge weight). On the other hand, genes acting in cell adhesion, extracellular matrix organization and cell-cell signaling resulted overrepresented in modules of interactions which were found strengthened in the tumor vs normal condition (sub-graph negative mean edge weight). It is reassuring that the GO categories overrepresented in the connected components were largely found in agreement with a previous survey of pathways consistently overrepresented in a large collection of signatures of differentially expressed genes of prognostic value in colorectal cancer (Lascorz et al., 2011). This case study showed that WG-Cluster allows shedding light into the network organization by fast and statistically robust module detection. In the context of a differential network analysis, it



delivers emergent information about the quantitative changes of interaction strength and gene mRNA abundance between two conditions, and allows the user to pursue specific modules on the basis of any available biological rationale, including the extent of changes in interaction strength, the extent of mRNA fold change or the functional characterization of modules.

4. Discussion

Molecule interconnectivity in human cells is daunting with ~ 20,000 protein-coding genes and ~ 87,000 protein isoforms. Consequently, a network formalization of cellular processes is extremely useful to analyse the growing amount of data on many types of interactions, which include but are not

limited to physical PPIs. A rich array of methods is currently available to detect network modular organization (Andreopoulos et al., 2009; Chen et al., 2014). Major limitations of most clustering methods, in very general terms, include the high computational cost and the inefficiency in exploiting the knowledge on edge strength (Toubiana et al., 2013). These aspects appear increasingly limiting in the light of the steady increase in the size of interaction maps and of the efforts to achieve interaction scoring standards (Villaveces et al., 2015). In this work, a new algorithm for network clustering has been developed that leverages existing information on both network nodes and edges to efficiently provide statistically significant modules. The detected modules are allowed to overlap, which reflects a common biological scenario, where, for instance,

The case study presented here suggests WG-Cluster as a possible method for differential network analysis. A network of physical PPI interactions, which are scored utilizing community standards and are deposited in the IntAct database, was integrated with mRNA expression data acquired from colon adenocarcinoma tumor samples or from normal samples. Our integrative approach relied on the rationale that the strength of a protein–protein interaction depends on the extent of congruent protein levels and on their protein affinity. Under the assumption that protein expression can be approximated with mRNA expression and that the interaction score in IntAct reflects the interaction affinity, we specified nodes and edge weights of the differential network as follows. Node weights were defined by the mRNA level fold changes while edge weights were defined by the product of the IntAct scores with the differential mRNA co-expression scores between the two conditions. Applying WG-Cluster to the differential network permitted to prioritize modules in the PPI network representing regions of progressively decreasing changes between the tumor and normal conditions. Despite the fact that the majority of interactions changed moderately between the two conditions, the organization of the detection of weighted connected components by sub-graph, which is implemented in WG-Cluster, permitted to streamline the identification of modules of markedly changing interactions. Furthermore, it was possible to discern modules of interactions which get weakened or strengthened in the tumor compared to the normal condition. Interestingly, separating the modules by average increase or decrease in the strength of their interactions

reflected also on their functional enrichment into distinct GO categories.

WG-Cluster is available as an open-source tool at <https://sites.google.com/site/paolaleccapersonalpage/> for the community of computational biologists to encourage its further development and/or its integration in general analytical workflows.

Author Contributions

PL and AR equally contributed to the conception, design and testing of the WG-Cluster algorithm; both the authors equally contributed also to the selection of the data and the case study for WG-Cluster application and to the interpretation of the algorithm output. Both the authors contributed to writing the manuscript.

Acknowledgments

The authors thank Alessandro Quattrone of the Laboratory of Translational Genomics of Centre for Integrative Biology, University of Trento (Italy) for his useful suggestions and comments to this work.

Supplementary Material

The Supplementary Material for this article can be found online at: <http://journal.frontiersin.org/article/10.3389/fgene.2015.00265>

References

- Adler, D., Glser, C., Nenadic, O., Oehlschlagel, J., and Zucchini, W. (2014). *ff: Memory-Efficient Storage of Large Data on Disk and Fast Access Functions*. Available online at: <http://cran.r-project.org/web/packages/ff/index.html>
- Andreopoulos, B., An, A., Wang, X., and Schroeder, M. (2009). A roadmap of clustering algorithms: finding a match for a biomedical application. *Brief Bioinf.* 10, 297–314. doi: 10.1093/bib/bbn058
- Bader, G. D., and Hogue, C. W. (2003). An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4:2. doi: 10.1186/1471-2105-4-2
- Barabási, A. L., Gulbahce, N., and Loscalzo, J. (2011). Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.* 12, 56–68. doi: 10.1038/nrg2918
- Barabási, A. L., and Oltvai, Z. N. (2004). Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.* 5, 101–113. doi: 10.1038/nrg1272
- Chen, B., Fan, W., Liu, J., and Wu, F. X. (2014). Identifying protein complexes and functional modules—from static ppi networks to dynamic ppi networks. *Brief Bioinf.* 15, 177–194. doi: 10.1093/bib/bbt039
- Chen, H., Li, H., Liu, F., Zheng, X., Wang, S., Bo, X., et al. (2015). An integrative analysis of tfbs-clustered regions reveals new transcriptional regulation models on the accessible chromatin landscape. *Sci. Rep.* 5:8465. doi: 10.1038/srep08465
- Chuang, H. Y., Lee, E., Liu, Y. T., Lee, D., and Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* 3, 140. doi: 10.1038/msb4100180
- Clauset, A., Newman, M. E. J., and Moore, C. (2004). Finding community structure in very large networks. *Phys. Rev. E* 70:066111. doi: 10.1103/PhysRevE.70.066111
- Cytoscape. (2015). *SIF Simple Interaction Format*. Available online at: http://wiki.cytoscape.org/Cytoscape_User_Manual/Network_Formats.
- de Lichtenberg, U., Jensen, L. J., Brunak, S., and Bork, P. (2005). Dynamic complex formation during the yeast cell cycle. *Science* 307, 724–727. doi: 10.1126/science.1105103
- Du, Q., Emelianenko, M., and Ju, L. (2006). Convergence of the lloyd algorithm for computing centroidal voronoi tessellation. *SIAM J. Numer. Anal.* 44, 102–119. doi: 10.1137/040617364
- Girvan, M., and Newman, M. E. J. (2001). Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U.S.A.* 99, 7821–7826. doi: 10.1073/pnas.122653799
- Grossmann, A., Benlasfer, N., Birth, P., Hegele, A., Wachsmuth, F., Apelt, L., et al. (2015). Phospho-tyrosine dependent protein-protein interaction network. *Mol. Syst. Biol.* 11:794. doi: 10.15252/msb.20145968
- Guo, Z., Wang, L., Li, Y., Gong, Y., Yao, C., Ma, W., et al. (2007). Edge-based scoring and searching method for identifying condition-responsive protein-protein interaction sub-network. *Bioinformatics* 23, 2121–2128. doi: 10.1093/bioinformatics/btm294
- Hartwell, L. H., Hopfield, J. J., Leibler, S., and Murray, A. W. (1999). From molecular to modular cell biology. *Nature* 402(6761 Suppl.), C47–C52. doi: 10.1038/35011540
- Havugimana, P. C., Hart, G. T., Nepusz, T., Yang, H., Turinsky, A. L., Li, Z., et al. (2012). A census of human soluble protein complexes. *Cell* 150, 1068–1081. doi: 10.1016/j.cell.2012.08.011
- Hermjakob, H., Montecchi-Palazzi, L., Bader, G., Wojcik, J., Salwinski, L., Ceol, A., et al. (2004). The hupo psi's molecular interaction format - a community standard for the representation of protein interaction data. *Nat. Biotechnol.* 22, 177–183. doi: 10.1038/nbt926
- Horn, H., Schoof, E. M., Kim, J., Robin, X., Miller, M. L., Diella, F., et al. (2014). Kinomexplorer: an integrated platform for kinome biology studies. *Nat. Methods* 11, 603–604. doi: 10.1038/nmeth.2968

- Ideker, T., and Krogan, N. J. (2012). Differential network biology. *Mol. Syst. Biol.* 8:565. doi: 10.1038/msb.2011.99
- Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* 18(Suppl. 1), S233–S240. doi: 10.1093/bioinformatics/18.suppl_1.S233
- Jansen, R., Greenbaum, D., and Gerstein, M. (2002). Relating whole-genome expression data with protein-protein interactions. *Genome Res.* 12, 37–46. doi: 10.1101/gr.205602
- Jemal, A., Bray, F., Center, M. M., Ferlay, J., Ward, E., and Forman, D. (2011). Global cancer statistics. *CA Cancer J. Clin.* 61, 69–90. doi: 10.3322/caac.20107
- Johannesson, T., and Björnsson, H. (2012). *Stineman, A Consistently Well Behaved Method of Interpolation*. Available online at: <http://rpackages.ianhowson.com/cran/stinepack/>
- Kane, M. J., and Emerson, J. W. (2013). *Bigmemory: Manage Massive Matrices with Shared Memory and Memory-Mapped File*. Available online at: <http://cran.r-project.org/web/packages/bigmemory/index.html>
- Klammer, M., Godl, K., Tebbe, A., and Schaab, C. (2007). Identifying differentially regulated subnetworks from phosphoproteomic data. *BMC Bioinformatics* 11:351. doi: 10.1186/1471-2105-11-351
- Lascorz, J., Chen, B., Hemminki, K., and Förster, A. (2011). Consensus pathways implicated in prognosis of colorectal cancer identified through systematic enrichment analysis of gene expression profiling studies. *PLoS ONE* 6:e18867. doi: 10.1371/journal.pone.0018867
- Linding, R., Jensen, L. J., Ostheimer, G. J., van Vugt, M. A., Jørgensen, C., Miron, I. M., et al. (2007). Systematic discovery of *in vivo* phosphorylation networks. *Cell* 129, 1415–1426. doi: 10.1016/j.cell.2007.05.052
- Liu, G., Wong, L., and Chua, H. N. (2009). Complex discovery from weighted ppi networks. *Bioinformatics* 25, 1891–1897. doi: 10.1093/bioinformatics/btp311
- Martin, H., Shales, M., Fernandez-Pinar, P., Wei, P., Molina, M., Fiedler, D., et al. (2015). Differential genetic interactions of yeast stress response mapk pathways. *Mol. Syst. Biol.* 11:800. doi: 10.15252/msb.20145606
- Moore, M. J., Zhang, C., Gantman, E. C., Mele, A., Darnell, J. C., and Darnell, R. B. (2014). Mapping argonaute and conventional rna-binding protein interactions with rna at single-nucleotide resolution using hits-clip and cims analysis. *Nat. Protoc.* 9, 263–293. doi: 10.1038/nprot.2014.012
- Nacu, S., Critchley-Thorne, R., Lee, P., and Holmes, S. (2007). Gene expression network analysis and applications to immunology. *Bioinformatics* 23, 850–858. doi: 10.1093/bioinformatics/btm019
- Newman, M. E. J. (2006). Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E* 74:036104. doi: 10.1103/PhysRevE.74.036104
- Ochoa, D., and Beltrao, P. (2015). Kinase-two-hybrid: towards the conditional interactome. *Mol. Syst. Biol.* 11, 798. doi: 10.15252/msb.20156107
- Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., et al. (2014). The mintact project—intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* 42, D358–D363. doi: 10.1093/nar/gkt1115
- Orchard, S., Salwinski, L., Kerrien, S., Montecchi-Palazzi, L., Oesterheld, M., Stümpflen, V. et al. (2007). The minimum information required for reporting a molecular interaction experiment (mimix). *Nat. Biotechnol.* 25, 894–898. doi: 10.1038/nbt1324
- Pandey, G., Arora, S., Manocha, S., and Whalen, S. (2014). Enhancing the functional content of eukaryotic protein interaction networks. *PLoS ONE* 9:e109130. doi: 10.1371/journal.pone.0109130
- Petschnigg, J., Groisman, B., Kotlyar, M., Taipale, M., Zheng, Y., Kurat, C. F., et al. (2014). The mammalian-membrane two-hybrid assay (mamth) for probing membrane-protein interactions in human cells. *Nat. Methods* 11, 585–592. doi: 10.1038/nmeth.2895
- Pons, P., and Latapy, M. (2005). Computing communities in large networks using random walks. *Lect. Notes Comput. Sci.* 3733, 284–293. doi: 10.1007/11569596_31
- Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* 76:036106. doi: 10.1103/PhysRevE.76.036106
- Re, A., Joshi, T., Kulberkyte, E., Morris, Q., and Workman, C. T. (2014). Rna-protein interactions: an overview. *Methods Mol. Biol.* 1097, 491–521. doi: 10.1007/978-1-62703-709-9_23
- Reichardt, J., and Bornholdt, S. (2006). Statistical mechanics of community detection. *Phys. Rev. E* 74:016110. doi: 10.1103/PhysRevE.74.016110
- Rosvall, M., and Bergstrom, C. T. (2008). Maps of information flow reveal community structure in complex networks. *Proc. Natl. Acad. Sci. U.S.A.* 105, 1118–1123. doi: 10.1073/pnas.0706851105
- Rual, J. F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., et al. (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 437, 1173–1178. doi: 10.1038/nature04209
- Tanay, A., Sharan, R., Kupiec, M., and Shamir, R. (2004). Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl. Acad. Sci. U.S.A.* 101, 2981–2986. doi: 10.1073/pnas.0308661100
- TCGA Research Network. (2013). Comprehensive molecular characterization of clear cell renal cell carcinoma. *Nature* 499, 43–49. doi: 10.1038/nature12222
- Toubiana, D., Fernie, A. R., Nikoloski, Z., and Fait, A. (2013). Network analysis: tackling complex data to study plant metabolism. *Cell* 31, 29–36. doi: 10.1016/j.tibtech.2012.10.011
- Turner, B., Razick, S., Turinsky, A. L., Vlasblom, J., Crowdy, E. K., Cho, E., et al. (2010). irefweb: interactive analysis of consolidated protein interaction data and their supporting evidence. *Database* 2010:baq023. doi: 10.1093/database/baq023
- Vandin, F., Upfal, E., and Raphael, B. J. (2011). Algorithms for detecting significantly mutated pathways in cancer. *J. Comput. Biol.* 18, 507–522. doi: 10.1089/cmb.2010.0265
- Varjosalo, M., Sacco, R., Stukalov, A., van Drogen, A., Planyavsky, M., Hauri, S., et al. (2013). Interlaboratory reproducibility of large-scale human protein-complex analysis by standardized ap-ms. *Nat. Methods* 10, 307–314. doi: 10.1038/nmeth.2400
- Villaveces, J. M., Jimnez, R. C., Porras, P., Toro, N., Duesbury, M., Dumousseau, M., et al. (2015). Merging and scoring molecular interactions utilising existing community standards: tools, use-cases and a case study. *Database pii: bau131*. doi: 10.1093/database/bau131

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2015 Lecca and Re. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.