METHOD ARTICLE

Recursive module extraction using Louvain and PageRank

[version 1; peer review: 2 approved]

Dimitri Perrin , Guido Zuccon

School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, QLD, 4001, Australia

V1 First published: 14 Aug 2018, 7:1286
<https://doi.org/10.12688/f1000research.15845.1>
Latest published: 14 Aug 2018, 7:1286
<https://doi.org/10.12688/f1000research.15845.1>

Abstract

Biological networks are highly modular and contain a large number of clusters, which are often associated with a specific biological function or disease. Identifying these clusters, or modules, is therefore valuable, but it is not trivial. In this article we propose a recursive method based on the Louvain algorithm for community detection and the PageRank algorithm for authoritativeness weighting in networks. PageRank is used to initialise the weights of nodes in the biological network; the Louvain algorithm with the Newman-Girvan criterion for modularity is then applied to the network to identify modules. Any identified module with more than k nodes is further processed by recursively applying PageRank and Louvain, until no module contains more than k nodes (where k is a parameter of the method, no greater than 100). This method is evaluated on a heterogeneous set of six biological networks from the Disease Module Identification DREAM Challenge. Empirical findings suggest that the method is effective in identifying a large number of significant modules, although with substantial variability across restarts of the method.

Keywords

Network biology, Module identification, Community detection, DREAM challenge

Open Peer Review**Approval Status**

	1	2
version 1		
14 Aug 2018	view	view

1. **Raghvendra Mall** , Hamad Bin Khalifa University (HBKU), Doha, Qatar

2. **Alina Sirbu**, University of Pisa, Pisa, Italy

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Dimitri Perrin (dimitri.perrin@qut.edu.au)

Author roles: **Perrin D:** Conceptualization, Data Curation, Formal Analysis, Methodology, Software, Writing – Original Draft Preparation, Writing – Review & Editing; **Zuccon G:** Conceptualization, Data Curation, Formal Analysis, Methodology, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

Copyright: © 2018 Perrin D and Zuccon G. This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Perrin D and Zuccon G. **Recursive module extraction using Louvain and PageRank [version 1; peer review: 2 approved]** F1000Research 2018, 7:1286 <https://doi.org/10.12688/f1000research.15845.1>

First published: 14 Aug 2018, 7:1286 <https://doi.org/10.12688/f1000research.15845.1>

Introduction

Biological functions emerge from interactions at the molecular level. For instance our circadian clock relies on the interactions between a large number of genes and proteins^{1,2}, and many cancer types are typically associated with specific genetic³ and epigenetic⁴ modifications. Unsurprisingly, biological networks such as protein-protein interaction (PPI) or regulatory networks therefore have a high degree of modularity (a measure of strength of the division of the network into subgroups, or clusters, called *modules* in our context) where the ‘modules’ often correspond to genes or proteins that are involved in the same biological functions. Diseases are also rarely associated with a single gene: disease genes have a high propensity to interact with each other, forming disease modules⁵. The identification of these disease modules is a valuable tool to identify disease pathways, but also to predict other disease genes.

This task is sometimes also known as community detection or graph clustering. This is a well established problem in network science. A large number of methods exist (see e.g., 6), but there was a lack of common evaluation on relevant biological networks.

The Disease Module Identification DREAM Challenge aimed to comprehensively assess module identification methods across six diverse, unpublished molecular networks⁷. Participating teams were tasked to predict disease-relevant modules both within individual networks (subchallenge 1) and across multiple, layered networks (subchallenge 2). The modules were defined as non-overlapping subsets containing 3 to 100 nodes. This is not a graph partition task, as not all nodes necessarily have to be assigned to a module.

In this article, we detail our solution for subchallenge 1. Next, we introduce the six networks and how we preprocessed them, then we describe our recursive algorithm, and discuss its performance across each network.

Methods

Networks

The human molecular networks used in the challenge are described in the challenge overview paper⁷. For convenience, we summarise their main characteristics in [Table 1](#). On top of capturing different types of biological information, they also vary in terms of size, link density and structural properties.

Table 1. Challenge networks.

ID	Type	# nodes	# edges	Directed
1	PPI	17,397	2,232,405	No
2	PPI	12,420	397,309	No
3	Signalling	5,254	21,826	Yes
4	Co-expression	12,588	1,000,000	No
5	Cancer	14,679	1,000,000	No
6	Homology	10,405	4,223,606	No

For the duration of the challenge, networks were only provided in anonymised form, without any gene names, details on the underlying data or how the networks were constructed. In the experiment in this article, we also considered networks in their anonymised form.

While protein interaction and homology networks, for instance, are obviously very different in nature, we opted to develop a method that could be applied to any network, independently of its type (although some preprocessing, described next, may be required, along with network-specific parameter tuning). This was because of the constraints of the challenge, in terms of both time and limited number of submissions.

Pre-processing

To have a method that works across network types, we decided to focus only on undirected networks. We also assumed that edge weights are in the range [0, 1]. Most networks in the challenge satisfy these requirements; pre-processing was applied to the remaining networks.

Network 3 is a directed network and as such needed to be converted to an undirected representation. This was achieved by simply assigning to all undirected $\langle u, v \rangle$ edges the average of the weights of the directed (u, v) and (v, u) edges (see [Figure 1](#)).

Networks 3 and 6 required normalisation of their weights. This was achieved by dividing all the original weights in each network by the maximum weight in that network.

These standardised networks are used as an input to our method. In what follows, any mention of a network refers to its standardised version.

Algorithm

The core of our method is the greedy Louvain algorithm⁸. This algorithm is a well-established method for community detection in networks⁶, it is applicable to weighted networks, and it provides better modularity maxima than other available greedy techniques⁶. In addition, the algorithm is computationally efficient and even large networks can be analyzed in reasonable runtime.

The algorithm starts by creating communities of size 1 where each node in the network forms a community. Then the algorithm proceeds by executing two steps. In the first step, the algorithm attempts to assign a node v to a community of a neighbor u , such that the modularity of the partition is increased. This process is repeated for as long as the modularity can be improved. This process generates an initial partition of the network. In the second step of the algorithm, each community of the partition is treated as a supernode. Supernodes are connected if at least one edge exists between nodes of each community the supernodes represents. Once this second step is concluded, the algorithm iterates and stops when the modularity cannot increase anymore.

As part of our methods, we rely on the implementation of [Louvain](#) (v0.2) by Blondel *et al.*⁸. The Louvain algorithm does

Figure 1. Conversion of a directed network into an undirected one.

not explicitly identify which modularity criterion is required: indeed, the algorithm can be instantiated using a number of modularity criteria. Their implementation supports ten modularity criteria; in all our submissions we used the default Newman-Girvan criterion⁹.

By default, in the Louvain algorithm, the initial partition assigns each node to a module that contains only the node itself. This creates a lot of variability in the results, which we reduced by modifying the algorithm. An idealised module is similar to a clique: it would contain nodes that are highly connected to other nodes, which are highly connected to similar nodes, etc. In other words, a node is important if it is linked to other nodes that are important. This closely matches the intuition of the PageRank algorithm developed to score web pages¹⁰. PageRank has been widely used in settings other than web search, including in bioinformatics¹¹. Our solution is therefore to calculate the PageRank for each node of the network, and to create an initial partition where each node is allocated to the module corresponding to its highest-scored neighbor (or itself, if that neighbor is scored lower). This has the advantage of both reducing the variability and ‘seeding’ Louvain with a promising partition. Here, we used a modified PageRank score that takes into account the edge weights.

Given that the task was to find modules with 3 to 100 nodes, a simple approach could be to run Louvain, process layer 1 from the hierarchical output generated by the algorithm, and extract all modules with a suitable size. This is, of course, far from optimal: Louvain generates modules of any size, and there may be interesting modules ‘hiding’ in a module containing more than 100 nodes (which would not be a valid submission to the challenge).

Initial tests on trimming or splitting large modules did not yield any useful results, so we implemented a recursive approach. For any network of size greater than k (for instance, $k = 100$), we run Louvain and process all modules. If a module contains between 3 and k nodes, it is saved. If it contains less than 3 nodes, it is discarded. If it contains more than k nodes, we extract the corresponding network and add it to a list of networks to which Louvain is recursively applied. The recursion terminates when this list is empty. PageRank-based initialisation is used for all recursion levels.

The overall algorithm is summarised in [Figure 2](#).

Evaluation

During the challenge, modules extracted from the anonymised networks were submitted to the online platform and evaluated

Require: G, k where G is an undirected graph where edges have weights in $[0, 1]$, and $3 \leq k \leq 100$ is the threshold indicating the maximum module size

Ensure: M is a list of modules of size in $[3, k]$

```

1: Run the PageRank algorithm on  $G$ 
2:  $L = \{G\}$ 
3:  $M = \emptyset$ 
4: while  $L \neq \emptyset$  do
5:    $l = \text{next graph in } L$ 
6:   Remove  $l$  from  $L$ 
7:   Run Louvain on  $l$ , with initialised allocation based on PageRank
8:   for each module  $m$  returned by Louvain do
9:     if  $|m| < 3$  then
10:      Discard  $m$ 
11:     else if  $|m| \leq k$  then
12:       Add  $m$  to  $M$ 
13:     else
14:       Extract  $G'$  the subgraph that contains all the nodes in  $m$  (and their corresponding edges)
15:       Add  $G'$  to  $L$ 
16:     end if
17:   end for
18: end while

```

Figure 2. Overall algorithm.

by the organisers. Modules were scored using the Pascal tool for pathway scoring¹². For each submission, the organisers would then communicate the number of significant modules that were identified for each of the six networks, but without providing any information on which submitted modules were significant. In the challenge leaderboard, submissions were ranked by the total number of significant modules identified. In this article, we analyse additional runs of our algorithm, evaluated locally using the code and GWAS data released by the organisers. Running the evaluation locally allows us to know which modules are significant.

The two parameters of our algorithm are the network being processed, and the value of the threshold k for the recursion. One configuration is a pair of a network and a threshold. For each configuration, we performed 10 runs of our algorithm.

Results

On the final challenge leaderboard, our solution ranked 12th overall with 44 significant modules identified across the six networks (when the winning team found 60). Relative to other teams, it performed best on network 2 (10 modules found, best score 13) and network 3 (7 modules found, best score 9).

Here, we analyse the performance over 100 new runs (10 per threshold value) for each network. The results are shown in [Figure 3](#).

(a) Network 1 **(b) Network 2** **(c) Network 3** **(d) Network 4** **(e) Network 5** **(f) Network 6**

Figure 3. Results on each network as a function of the value for k . White and red dots represent the median and mean values for each configuration, respectively. The blue line indicates our performance in the challenge leaderboard for that network, and the red line that of the best submission for that network.

Louvain is non-deterministic, and even after initialising it using PageRank, the results for any given configuration have high variability. It is also worth noting that, for five of the six networks, there is at least one configuration for which our algorithm matches or outperforms the best system submitted to the challenge. Only network 6 leads to poor results. If we combine the best result for each network, we obtain a theoretical total of 81 significant modules, close to double our final score and 35% better than the best-performing solution in the challenge.

For most networks the performance is robust to changes of k , but there still appears to be an optimal configuration for each network. For networks 1, 3 and 4, our method produces better results with large values of k . For network 5, aiming for smaller modules produced better results, while for network 2 mid-range values of k are preferable.

Discussion

The results from these 600 additional runs show the potential of our approach. Under the same conditions as the challenge, our algorithm can match or improve the best results from the competition phase.

Evaluating all the modules from a given solution against all the GWAS data using Pascal takes hours, and it is therefore not practical to use this evaluation to guide the creation of the modules. Even outside the challenge, it is more realistic for the extraction method to be purely driven by the network itself.

However, now that the challenge is completed, it is possible to evaluate thousands of modules. Using this data, future work will focus on developing a module ‘score’ that would be a good predictor of whether that module is significant. If this can be achieved, we would then add a local optimisation step at the end of our algorithm, to fine tune the extracted modules.

Another direction for future work is to study the consensus between restarts. How many times do we identify the same modules, or does this correlate with their significance? We believe there is potential for voting/fusion approaches to extend our algorithm.

Conclusions

Network-based approaches are an important tool in biomedical research, as they can lead to the identification of clusters of genes (modules) involved in the same molecular function or the same disease.

Identifying these modules is not trivial, and the Disease Module Identification DREAM Challenge was an important

initiative to benchmark various approaches. We developed a recursive method based on the Louvain and PageRank algorithms, which performed reasonably well in the challenge.

Here, we showed that this method can actually match or exceed the best results from the competition challenge. Further work will focus on exploiting the high variability between restarts, and on developing a module score that can guide optimisation of the identified modules.

Data availability

The dataset associated with the Disease Module Identification DREAM Challenge is available for registered participants at <http://www.synapse.org/#!Synapse:syn6156761/wiki/400659>.

Challenge results and scoring scripts are available at <http://www.synapse.org/#!Synapse:syn6156761/wiki/400647>.

Software availability

Source code implementation for the recursive method presented in this article and used in the Disease Module Identification DREAM Challenge is available from GitHub: <https://github.com/bmds-lab/DMI/tree/v0.1>

Archived source code at time of publication <https://doi.org/10.5281/zenodo.1330835>¹³.

Source code is available under a GPL 3.0 license

Competing interests

No competing interests were disclosed.

Grant information

The author(s) declared that no grants were involved in supporting this work.

Acknowledgements

The authors acknowledge the Disease Module Identification DREAM Challenge and Sage Bionetworks-DREAM for the provision of the six networks and of the evaluation method (code and GWAS studies).

The authors also wish to acknowledge the work from Charlie Shaw-Feather, who helped configure and run local Pascal evaluations. These evaluations also relied on computational resources and services provided by the HPC and Research Support Group, Queensland University of Technology, Brisbane, Australia.

References

1. Ukai-Tadenuma M, Yamada RG, Xu H, *et al.*: **Delay in feedback repression by cryptochrome 1 is required for circadian clock function.** *Cell*. 2011; 144(2): 268–281.
[PubMed Abstract](#) | [Publisher Full Text](#)
2. Jolley CC, Ukai-Tadenuma M, Perrin D, *et al.*: **A mammalian circadian clock model incorporating daytime expression elements.** *Biophys J*. 2014; 107(6): 1462–1473.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

3. McLean MH, El-Omar EM: **Genetics of gastric cancer.** *Nat Rev Gastroenterol Hepatol.* 2014; **11**(11): 664–674.
[PubMed Abstract](#) | [Publisher Full Text](#)
4. Perrin D, Ruskin HJ, Niwa T: **Cell type-dependent, infection-induced, aberrant DNA methylation in gastric cancer.** *J Theor Biol.* 2010; **264**(2): 570–577.
[PubMed Abstract](#) | [Publisher Full Text](#)
5. Barabási AL, Gulbahce N, Loscalzo J: **Network medicine: a network-based approach to human disease.** *Nat Rev Genet.* 2011; **12**(1): 56–68.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
6. Fortunato S: **Community detection in graphs.** *Phys Rep.* 2010; **486**(3–5): 75–174.
[Publisher Full Text](#)
7. Choobdar S, Ahsen ME, Crawford J, *et al.*: **Open community challenge reveals molecular network modules with key roles in diseases.** *bioRxiv.* 2018.
[Publisher Full Text](#)
8. Blondel VD, Guillaume JL, Lambiotte R, *et al.*: **Fast unfolding of communities in large networks.** *Phys Rep.* 2008; **2008**(10): P10008.
[Publisher Full Text](#)
9. Newman ME, Girvan M: **Finding and evaluating community structure in networks.** *Phys Rev E Stat Nonlin Soft Matter Phys.* 2004; **69**(2 Pt 2): 026113.
[PubMed Abstract](#) | [Publisher Full Text](#)
10. Page L, Brin S, Motwani R, *et al.*: **The pagerank citation ranking: Bringing order to the web.** *Technical report.* Stanford InfoLab, 1999.
[Reference Source](#)
11. Iván G, Grolmusz V: **When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks.** *Bioinformatics.* 2011; **27**(3): 405–407.
[PubMed Abstract](#) | [Publisher Full Text](#)
12. Lamparter D, Marbach D, Rueedi R, *et al.*: **Fast and Rigorous Computation of Gene and Pathway Scores from SNP-Based Summary Statistics.** *PLoS Comput Biol.* 2016; **12**(1): e1004714.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
13. Perrin D: **bmbs-lab/DMI: Initial release (Version v0.1).** *Zenodo.* 2018.
<http://www.doi.org/10.5281/zenodo.1330835>

Open Peer Review

Current Peer Review Status:

Version 1

Reviewer Report 14 September 2018

<https://doi.org/10.5256/f1000research.17297.r37207>

© 2018 Sîrbu A. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Alina Sîrbu

Department of Computer Science, University of Pisa, Pisa, Italy

The paper presents a recursive algorithm to find modules in biological networks. The authors evaluate the modules using a DREAM dataset. The algorithm is promising and the paper is well written. The results are comparable and sometimes better than the best performance of the DREAM Challenge. I think some changes could improve the paper:

1. In the definition of the algorithm, the intuition between the concept of modularity could be included, and maybe a high level definition of the chosen modularity index.
2. "By default, in the Louvain algorithm, the initial partition assigns each node to a module that contains only the node itself. This creates a lot of variability in the results" Do you mean variability from one run to another?
3. "For each configuration, we performed 10 runs of our algorithm." Here you are evaluating each run and showing all 10 results. I wonder what would happen if one combined modules from different runs. To create an 'ensemble' module extractor. One could just pool together the modules found. This could be done over the 10 runs with the same K, but also maybe over runs with different k? Or even changing the modularity criterion and combining the results. And then selecting the most frequently found modules...
4. For figure 3: is there information on how many significant modules are actually known in the networks? One could think of adding another line that would show the number of significant modules known as an upper bound for the performance.
5. "If we combine the best result for each network, we obtain a theoretical total of 81 significant modules, close to double our final score and 35% better than the best-performing solution in the challenge". How is this combination made?

Is the rationale for developing the new method (or application) clearly explained?

Yes

Is the description of the method technically sound?

Yes

Are sufficient details provided to allow replication of the method development and its use by others?

Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Complex systems modelling

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 10 September 2018

<https://doi.org/10.5256/f1000research.17297.r37208>

© 2018 Mall R. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Raghvendra Mall

Qatar Computing Research Institute (QCRI), Hamad Bin Khalifa University (HBKU), Doha, Qatar

The article proposes usage of a recursive form of Louvain method while including the PageRank of nodes to make graph partitions or detect biological 'modules' which are then evaluated through the DREAM Challenge evaluation tool PASCAL to determine how many of the identified modules were significant w.r.t. GWAS experiments conducted by the DREAM Challenge organisers.

The paper has been well written and all the technical details have been elaborated quite well by the authors, thereby suggesting that the method is reproducible and can be extended as per the suggestion of the authors to have a consensus disease module identification technique.

The authors provide a good introduction to Louvain method explaining its non-deterministic nature and limitations such as resolution limit for which it needs to be used in a recursive fashion

to detection modules of length k ($[3,100]$). Moreover, they explain well how PageRank is used along with the Louvain method.

The only issue that I have is with the experiment section where the authors perform an additional 100 new runs and claim that they can obtain theoretically 81 significant modules. This is not correct way of evaluation as the authors are using the test set and tuning their hyper-parameters on the test set. In order to have a generic model, the authors can tune their model parameters on the training set and use the same for each test set network rather than tuning the results on test set. The authors do indicate this when they say that in future work they will focus on developing a module 'score' to predict if a module is significant or not.

A major issue here is the non-deterministic nature of Louvain method which will result in different partitions every time the code is run. Hence the idea of having a 'consensus between restarts' is also interesting.

Finally, it would have been better if the authors add information about the biological content of the modules that they have discovered and for which GWAS traits were the modules enriched in a given population. That analysis would complete the paper from a biological standpoint also.

Is the rationale for developing the new method (or application) clearly explained?

Yes

Is the description of the method technically sound?

Yes

Are sufficient details provided to allow replication of the method development and its use by others?

Yes

If any results are presented, are all the source data underlying the results available to ensure full reproducibility?

Yes

Are the conclusions about the method and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research