Cickovski *et al. BMC Bioinformatics* 2019, **20**(Suppl 11):278
https://doi.org/10.1186/s12859-019-2820-7

METHODOLOGY Open Access

MATria: a unified centrality algorithm

Trevor Cickovski[1]*, Vanessa Aguiar-Pulido[2] and Giri Narasimhan[1]

From 7th IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS 2017) Orlando, FL, USA. 19-21 October 2017

Abstract

Background: Computing *centrality* is a foundational concept in social networking that involves finding the most "central" or important nodes. In some biological networks defining *importance* is difficult, which then creates challenges in finding an appropriate centrality algorithm.

Results: We instead generalize the results of any *k* centrality algorithms through our iterative algorithm MATRIA, producing a single ranked and *unified* set of central nodes. Through tests on three biological networks, we demonstrate evident and balanced correlations with the results of these *k* algorithms. We also improve its speed through GPU parallelism.

Conclusions: Our results show iteration to be a powerful technique that can eliminate spatial bias among central nodes, increasing the level of agreement between algorithms with various importance definitions. GPU parallelism improves speed and makes iteration a tractable problem for larger networks.

Keywords: Centrality, Networks, Iteration, Graphics Processing Unit (GPU)

Background

The concept of *centrality* is fundamental to social network theory and involves finding the most important or *central* nodes in a social network. There are three core types of *path-based* centrality, each with different definitions of *importance*. *Betweenness* centrality [1] bases importance on the number of shortest paths over all pairs of nodes that run through a node (finding hubs in a network), *closeness* [2] on the overall length of the shortest paths towards all other nodes that start from a node (finding nodes in the "center" of a network), and *degree* [3] on the number of connections. There are also *eigenvector-based* approaches, which solve a system of n equations with n unknown centrality values for a graph of n nodes, applying an eigensolver that eventually converges to the centrality values. *PN*-centrality [4] takes into account a node's local degree and that of its "friends" and "enemies". Google's PageRank [5] models centrality by a random walker which probabilistically either moves to a neighbor or someplace random, with centrality values reflecting how often this walker lands upon a node. PageTrust [6] extends PageRank to handle signed networks by incorporating *distrust* between nodes.

Many real-world networks (i.e., airports, search engines) have a clear definition of "importance", enabling the appropriate centrality algorithm to be chosen. When studying biological networks this can also be true, as has been shown with phylogenetically older metabolites tending to have larger degree in a metabolic network [7], and the removal of highly connected proteins within yeast protein interaction networks tending to be lethal [8]. Other times this is not so certain, as when studying properties such as transitivity in protein interaction networks [9], robustness against mutations in gene networks [10], and finding global regulators in gene regulatory networks [11]. This latter study in particular showed large amounts of disagreement between centrality algorithms in uncovering global regulators in an *E. Coli* gene regulatory network, and along with other studies [12, 13] indicates it is necessary to apply multiple centrality algorithms in situations where "importance" is difficult to define.

The challenge in these situations then becomes how to unify results over multiple centrality algorithms that

*Correspondence: tcickovs@fiu.edu
[1]Bioinformatics Research Group (BioRG) & Biomolecular Sciences Institute, School of Computing & Information Sciences, Florida International University, 11200 SW 8th St, 33199 Miami, FL, USA
Full list of author information is available at the end of the article

differ in their definitions of "importance" and therefore also their results. Figure 1 shows application of the three path-based approaches to a signed and weighted bacterial co-occurrence network [14], with parts (a1-3) demonstrating minimal similarity between each algorithm's top 20% most central nodes. To be certain we also tested on the two less modular biological networks shown in Fig. 2, including a Pacific Oyster gene co-expression network (GEO:GSE31012, network B) and a more fully connected bacterial co-occurrence network C. Table 1 shows Spearman correlations between rank vectors from the three path-based approaches (network A is from Fig. 1). Correlation with betweenness and the other two approaches peaked for network B, but went to almost zero for network A (modular) and network C (well-connected). Correlation between degree and closeness was the opposite, peaking for the extremes but low for network B.

Figure 1**a1-3** makes it evident that spatial biases within each algorithm largely contribute to this disagreement. For network A all central nodes were mostly on the same path with betweenness (a1), in the "middle" with closeness (a2), and in the same strongly connected component with degree (a3). The network had 126 nodes, and the three algorithms agreed on only five central nodes (in black) within their top 20%. This naturally leads to the question, if we were to somehow remove spatial bias, would we have more consensus among the results?

We build on a prior algorithm called ATRIA [15], which reduced bias in closeness centrality by applying iteration to identify central nodes spread widely across the

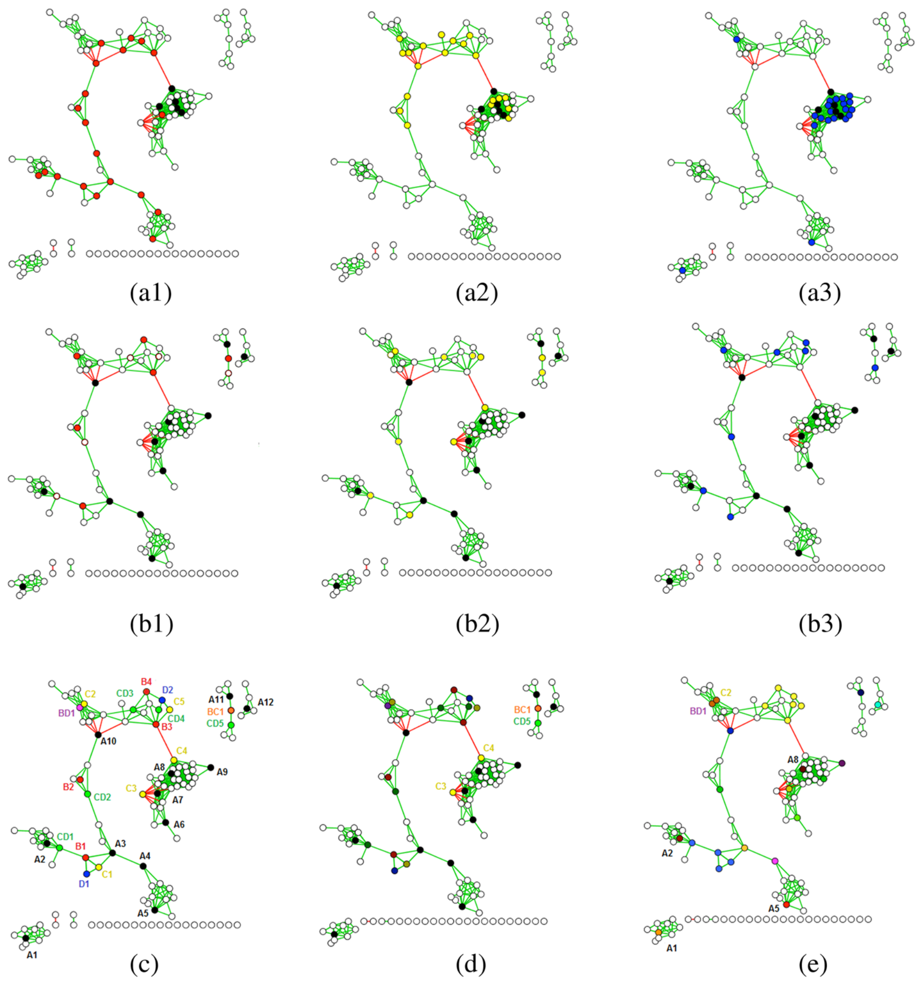

Fig. 1 Centrality results on a test microbial co-occurrence network. Top 20% most central nodes found by non-iterative betweenness (**a1**, red), closeness (**a2**, yellow) and degree (**a3**, blue) centrality in a correlation network, with mutual agreements in black. Central nodes found by iterative betweenness (**b1**), closeness (**b2**) and degree (**b3**) centrality on the same network, again with mutual agreements in black. **c** Same network with nodes found by all (black), betweenness only (red), closeness only (yellow), degree only (blue), betweenness and closeness (orange), closeness and degree (green), and betweenness and degree (violet). **d** Final network with all possible disagreements (dark) resolved. **e** Final centrality rankings of nodes and supernodes produced by MATRIA, red nodes are highly ranked, violet low, white zero

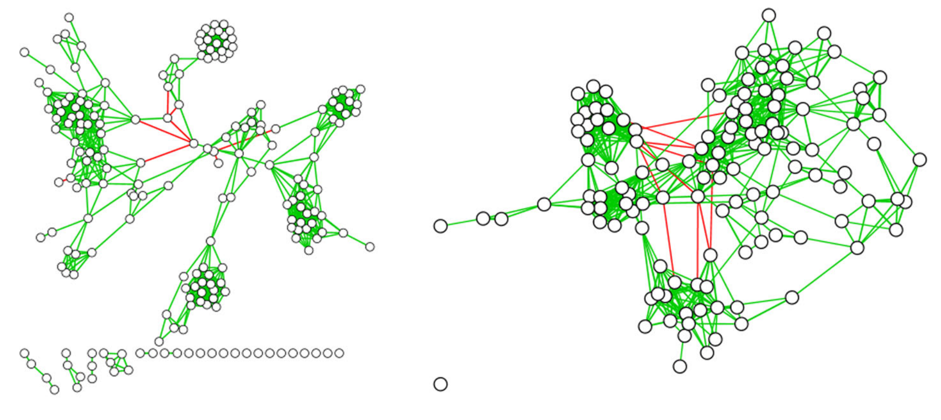

Fig. 2 Two other test biological networks. **b** Gene co-expression network from the Pacific Oyster; **c** Less modular microbial co-occurrence network

network. We used a socio-economic model with node pairs providing a "gain" and a "loss" to each other. We will now apply iteration to other centrality algorithms (which we refer to as *backbones*), and first illustrate stronger agreement between iterative backbones on our biological networks compared to their non-iterative counterparts. We next propose an algorithm MATRIA for unifying disagreements between these iterative backbones, producing a ranked set of central nodes and *supernodes* with multiple central node possibilities. This unified set had good coverage for our networks, with 90-100% of the nodes either in this set or universally agreed as unimportant. We also demonstrate that this rank vector correlates well with those from the iterative backbones, which by consilience [16] supports its reliability. Since iteration is computationally expensive we conclude with a discussion on improving efficiency for large biological networks through the GPU.

Background: iteration

With ATRIA we found spatial bias within closeness centrality could be fixed by iteratively finding and removing dependencies of the most central node, then recomputing centralities. We did this until all are zero ("unimportant"). Social network theory [17] states that two nodes connected by a mutual friend or enemy (known as a *stable triad*) will tend to become friends, and thus we defined a *dependency* of a node i as i itself plus any edges in a stable triad with i, illustrated by Fig. 3. In both cases if node A was most central we assumed edge BC to be *coincidental* and remove node A and edge BC before recomputing centralities. We first generalize iterative centrality using Algorithm 1, with X acting as a placeholder for some *backbone* algorithm.

Table 1 Rank vector correlations between non-iterative centrality algorithms on three signed/weighted biological networks

Algorithm Pairs	A	B	C
Betweenness/Closeness	0.071	0.396	-0.120
Betweenness/Degree	-0.170	0.106	-0.136
Closeness/Degree	0.699	0.256	0.863

```
ITERCENT_X(g)
begin
    repeat
        S = {};
        foreach node i from 0 to N − 1 do
            Compute Centrality_X(i);
        end
        Find node m with highest Centrality_X(m);
        Add m to the set S of central nodes;
        Remove dependencies of m ;
    until Centrality_X(m) = 0;
    return S
end
```

Algorithm 1: Generalized iterative centrality algorithm.

ATRIA also extended closeness centrality to operate on an undirected network with edge weights in the range $[-1, 1]$ by approaching centrality from the perspective of a node's benefit to the network. We used a simplified economic Payment Model [18], defining closeness (*CLO*) centrality $Centrality_{CLO}(i)$ of node i by Eq. 1.

$$Centrality_{CLO}(i) = |\sum_{j \neq i} G(i,j) + L(i,j)|, \quad (1)$$

where $G(i,j)$ is the maximum positive edge weight product over all paths between node i and node j, and $L(i,j)$ is the maximum negative edge weight product. We computed these paths using a modified Dijkstra's algorithm

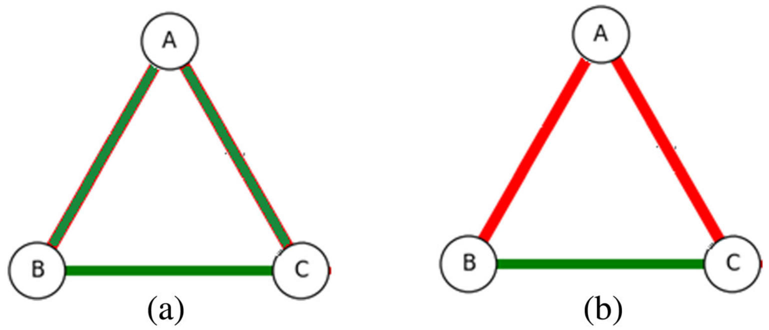

Fig. 3 Stable triads, with (**a**) zero and (**b**) two negative edges

MOD_DIJKSTRA that used edge products and chose maximum path magnitudes. This is just closeness centrality using maximum paths, with "path length" defined as $G(i,j) + L(i,j)$. Plugging *CLO* into *X* in Algorithm 1 represents our iterative closeness centrality algorithm ATRIA. We now define signed versions of other path-based backbones.

Signed versions of other path-based approaches

Degree centrality

Degree is easiest to define, with all local computations. For gains and losses we count incident positive and negative edges for a node i, producing:

$$Centrality_{DEG}(i) = |\sum_{j \neq i} W(i,j)|, \tag{2}$$

where $W(i,j)$ is the signed weight of edge (i,j).

Betweenness centrality

Betweenness is more challenging, but we can use the same *MOD_DIJKSTRA* algorithm to count the *number* of positive paths (call this $\gamma_{jk}(i)$) and negative paths (call this $\lambda_{jk}(i)$) that include i. The equation then becomes the sum of these terms:

$$Centrality_{BET}(i) = \sum_{j \neq i \neq k} \gamma_{jk}(i) + \lambda_{jk}(i). \tag{3}$$

We can then plug *BET* or *DEG* for *X* in Algorithm 1 to respectively produce iterative betweenness or degree centrality. Since non-iterative path-based approaches produced extremely different results on our networks, we will use these iterative versions ITERCENT$_{BET}$, ITERCENT$_{CLO}$, and ITERCENT$_{DEG}$ to demonstrate MATRIA. Other centrality algorithms can be substituted for *X*, and we will in fact show that MATRIA can support any *k* centrality algorithms.

Table 2 shows the updated rank vector correlations for iterative path-based algorithms on our biological networks, confirming improved performance for network A before any attempt to resolve disagreements (especially for betweenness). The less modular networks B and C do not show as much improvement and are sometimes worse. We now describe MATRIA, which produces a unified ranked set that correlates well with each iterative path-based approach.

MATria

Algorithm 2 shows our top-level MATRIA procedure that accepts a network g and produces the sets of central nodes S_{BET}, S_{CLO} and S_{DEG}, then resolves disagreements between these sets through a procedure UNIFY to produce a final set S.

```
MATRIA($g$)
begin
    $S_{BET}$ = ITERCENT$_{BET}(g)$;
    $S_{CLO}$ = ITERCENT$_{CLO}(g)$;
    $S_{DEG}$ = ITERCENT$_{DEG}(g)$;
    $S$ = UNIFY($S_{BET}, S_{CLO}, S_{DEG}$);
end
```

Algorithm 2: Top-level MATRIA algorithm.

Universal agreements

We define universal agreements as nodes discovered by all iterative backbones, or any $x : x \in S_{BET} \cap S_{CLO} \cap S_{DEG}$. On network A the iterative backbones agreed on twelve central nodes, colored black in Fig. 1**b1-3** and labeled $A1$-$A12$. Recall this is already an improvement upon the non-iterative versions, which agreed on only five central

Table 2 Rank vector correlations between iterative path-based centrality algorithms

Algorithm Pairs	A	B	C
ITERCENT$_{BET}$/ITERCENT$_{CLO}$	0.498	0.208	0.060
ITERCENT$_{BET}$/ITERCENT$_{DEG}$	0.569	0.538	0.298
ITERCENT$_{CLO}$/ITERCENT$_{DEG}$	0.717	0.189	0.209

nodes in the same vicinity. UNIFY first adds these twelve universal agreements to S.

Resolving disagreements

In Fig. 1**c** we label nodes found by one or two of the path-based backbones, but not all three (18 total). We use node color to indicate the backbone(s) that discovered them, with primary colors for nodes discovered by one backbone:

- **Betweenness** (4), colored **red**: $B1$-$B4$
- **Closeness** (5), colored **yellow**: $C1$-$C5$
- **Degree** (2), colored **blue**: $D1$, $D2$

We use secondary colors obtained by combining appropriate primary colors for nodes discovered by two backbones:

- **Betweenness & Closeness** (1), colored **orange**: $BC1$
- **Closeness & Degree** (5), colored **green**: $CD1$-$CD5$
- **Betweenness & Degree** (1), colored **violet**: $BD1$

We note patterns among these disagreements. Many times all three backbones are covered exactly once between two adjacent or three triad nodes. We argue that because of the fundamental properties of iteration, centrality is likely a "toss-up" in these situations. Take for example the triad $[x, y, z]$ in Fig. 4**a**. In this case x, y and z were found as central by iterative betweenness, closeness and degree respectively. However, suppose centrality is actually a "toss-up" between them, which would mean for example in iterative betweenness when x was found as most central, y and z had only slightly lower centrality values. In the next iteration x would be removed along with edge $y - z$, causing y and z to lose all contributions from paths involving this triad (which by definition are likely significant if x was central). The same thing would happen when y was found by iterative closeness, and z by iterative degree. Adjacencies like the one in Fig. 4**b** have the same issue for the same reason, with x (or y) losing contributions from its central neighbor upon its removal.

We define a *supernode* as any set of neighboring nodes such that each algorithm finds exactly one of them. In Fig. 1**c** we have two supernode triads: $[B1, C1, D1]$ and $[B3, C5, D2]$. UNIFY adds these to S (now 14 elements) as "toss-ups", and we also darken them in our updated Fig. 1**d** to indicate they have been resolved. For supernode adjacencies there are three types: red-green (betweenness, closeness/degree), yellow-violet (closeness, betweenness/degree), and blue-orange (degree, betweenness/closeness). We have a total of six supernode adjacencies in Fig. 1**c** and begin by adding them to S: $[B1, CD1]$, $[B2, CD2]$, $[B3, CD3]$, $[B3, CD4]$, $[B4, CD3]$, and $[C2, BD1]$.

We now have an issue, because two of these adjacencies also include supernode triad members ($B1$ and $B3$). Having supernodes that share members is not helpful, because each supernode should provide multiple options for a central node. We now describe how UNIFY merges supernodes with common members, and specifically address the triad and adjacency in detail to handle this network. Supernode triads can also overlap with each other, as can supernode adjacencies, and we later briefly describe how to merge those.

Merging overlapping supernodes

We first note that for a supernode adjacency x-y, if x is also a member of a supernode triad it is already a "toss up" with two nodes w and z, as shown in Fig. 5. We then note that w and z must be found by the same two algorithms that found y (since in a supernode triad all three algorithms must be covered). Thus, the "toss-up" becomes between (1) only x, (2) y and w, and (3) y and z. We merge these into one supernode triad $[x, \{y, w\}, \{y, z\}]$, now allowing a single node to represent a set of nodes as shown in the Figure. Although the edges from x to $\{y, w\}$ and $\{y, z\}$ now become ambiguous, their weights are no longer relevant because we already ran the backbones.

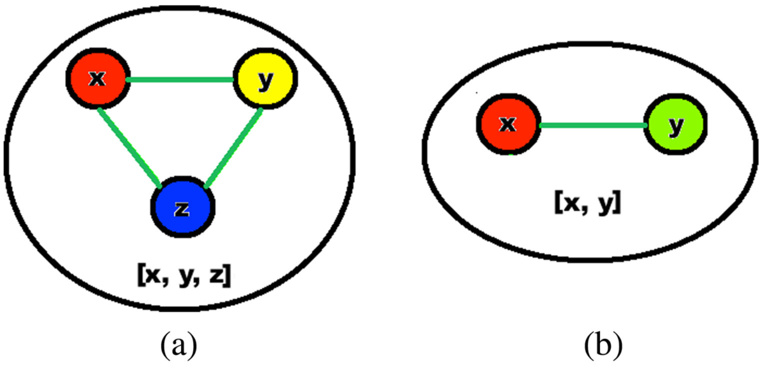

Fig. 4 Supernode examples; (**a**) triad, (**b**) adjacency

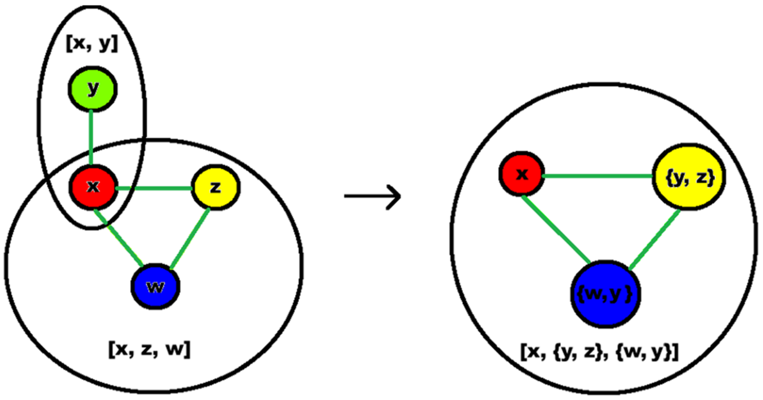

Fig. 5 Merging supernodes; in this case an overlapping triad and adjacency

We have several supernode adjacencies in our network where one of the two nodes is also in a supernode triad:

- **Central Triad** $[B1, C1, D1]$ with adjacency $[B1, CD1]$. We replace both elements in S by the supernode: $[B1, \{C1, CD1\}, \{D1, CD1\}]$.
- **Upper Triad** $[B3, C5, D2]$ with adjacencies $[B3, CD3]$ and $[B3, CD4]$. We replace all three elements in S by the supernode $[B3, \{C5, CD3, CD4\}, \{D2, CD3, CD4\}]$.
- **New Triad** $[B3, \{C5, CD3, CD4\}, \{D2, CD3, CD4\}]$ now has an overlap with adjacency $[B4, CD3]$. We similarly replace both elements in S by the supernode $[\{B3, B4\}, \{C5, CD3, CD4\}, \{D2, CD3, CD4\}]$.

Figure 1**d** shows all resolved disagreements darkened. In addition, Table 3 shows the other types of supernode merges performed by UNIFY, between triads that share one or two nodes or adjacencies that share one. Merging provides the final set S in UNIFY, which we now fully write as Algorithm 3.

Ranking Supernodes: The final step of UNIFY is to rank the elements of S. We do this as follows:

1. **Universal Agreements:** Mean ranking over backbones.
2. **Supernode Triads:** Mean ranking of each node using the backbone that found it. For example in Fig. 4**a** we would average the ranking of x in betweenness, y in closeness, and z in degree.

Table 3 Other types of supernode merges

Category	SN 1	SN 2	Replace Both By
Triad (2 Shared)	$[x, y, w]$	$[x, y, z]$	$[x, y, \{w, z\}]$
Triad (1 Shared)	$[x, v, w]$	$[x, y, z]$	$[x, \{v, y\}, \{w, z\}]$
Adjacency(1 Shared)	$[x, y]$	$[x, z]$	$[x, \{y, z\}]$

```
UNIFY($S_{BET}, S_{CLO}, S_{DEG}$)
begin
    $S = \{\}$ ;
    foreach ($x : x \in S_{BET} \cap S_{CLO} \cap S_{DEG}$) do
        add $x$ to $S$;                                  ▷ Universal Agreements
    end
    foreach ($x, y, z : x \in S_{BET} - (S_{CLO} \cup S_{DEG}), y \in S_{CLO} - (S_{BET} \cup S_{DEG}), z \in S_{DEG} - (S_{BET} \cup S_{CLO})$, and $W(x, y) * W(y, z) * W(x, z) > 0$) do
        add $[x, y, z]$ to $S$;                          ▷ Supernode Triads
    end
    foreach ($x, y : W(x, y) \neq 0$ and
    (($x \in S_{BET}$ and $y \in (S_{CLO} \cap S_{DEG}) - S_{BET}$) or
    ($x \in S_{CLO}$ and $y \in (S_{BET} \cap S_{DEG}) - S_{CLO}$) or
    ($x \in S_{DEG}$ and $y \in (S_{BET} \cap S_{CLO}) - S_{DEG}$)) do
        add $[x, y]$ to $S$;                             ▷ Supernode Adjacencies
    end
    Merge overlapping supernodes in $S$ ;
    Rank all elements in $S$ ;
    return S;
end
```

Algorithm 3: UNIFY procedure.

3. **Supernode Adjacencies**: Same as supernode triads, except one node will have rankings for two backbones.
4. **Merged Supernodes**: These have elements like $\{w, y\}$ where w and y were said to *both* be important by a backbone. In this case use the ranking of whichever of w and y was discovered first as the ranking of $\{w, y\}$, then apply the above logic for the supernode ranking. Our results, shown in Fig. 1**e** (red=high and violet=low rank), indicate that the top five entries ($A1, A2, A5, A8$, and the supernode $BD1$-$C2$) could correspond to leaders of the five most tightly connected components.

Unresolvable Disagreements: Although most disagreements in Fig. 1 were resolvable there are still two nodes $C3$ and $C4$ that were found by closeness and not involved in a resolvable disagreement. These are still colored yellow in Fig. 1**d**. Upon further investigation the disagreement resulted because iterative degree and betweenness found node $A7$ early (#2 and #7), but closeness found it later (#16, but more importantly after $C3$ and $C4$). With $A7$ directly connected to $C3$, removing it plummeted $C3$ in degree and betweenness centrality. But since $A7$ was also eventually discovered by closeness it became a universal agreement and could not be a supernode with $C3$. This seems to suggest forming supernodes on-the-fly, as opposed to waiting until the end. However the drop of $C4$ resulted from an indirect effect (removing $A7$ reduced many edges in that tight component), so that will not resolve all disagreements either. The other disagreement, $BC1$ and $CD5$, creates an interesting situation where two backbones each say one is important, but one (closeness) says both are important (i.e. not a "toss-up"). We leave this as unresolvable for now, though could potentially add another type of element in S which encapsulates this. We will see however that even with our current approach, these unresolvable disagreements are quite rare in our networks.

We also remark that UNIFY can be generalized to work with any k centrality algorithms. In our example ($k = 3$), we can view supernode adjacencies and triads as components of size 2 and 3. In general supernodes can be of sizes 2 to k.

Results

Coverage

We begin by evaluating the percentage of nodes for which UNIFY could reach an agreement on centrality. Table 4 shows that the number of agreed important nodes did not drop significantly as our networks became less modular. While the universal agreement (important and unimportant) percentage did drop, most of these nodes became involved in supernodes, all5owing us to still draw conclusions about their centrality. Only 3-7% of nodes were involved in unresolvable disagreements, demonstrating that MATRIA will generally produce a set with good coverage.

We also checked some of the agreed important genes discovered by MATRIA in network B. Although gene essentiality statistics are limited for the Pacific Oyster, the results show promise. The gene for the most abundant and fundamental eukaryotic protein, Actin [19], was found and ranked #2 by MATRIA. MATRIA also found genes for Death-Associated Protein 3 (DAP3) which has been marked essential in other eukaryotic organisms for its critical roles in respiration and apoptosis [20], and the Heat Shock Protein (HSP) which has also been marked essential for apoptosis in both prokaryotes and eukaryotes [21] and is involved in protein folding [22]. Additionally, MATRIA found genes for a member of the Sterile Alpha Motif (SAM) homology, which is known to have important roles in immunity [23] and its ability to bind to RNA [24], and also a Protein-Tyrosine Phosphatase Non-Receptor (PTPN, [25]) which has potential to affect multiple cellular functions through post-translational phosphorylation [26].

Table 4 MATRIA coverage of all three networks

	A	B	C
Agreed Important	12	11	7
Agreed Unimportant	96	127	50
Supernode Triad	6	20	3
Supernode Adj	8	31	57
Unresolvable	4	14	9
% Universal Agreement	86	68	45
% Resolvable	**97**	**93**	**93**

Correlations

We next verify that the rank vector for S correlates with the individual rank vectors S_{BET}, S_{CLO}, and S_{DEG}, plus those found when including PN-Centrality and PageTrust (thus $k = 5$). Table 5 shows that for all five examples we were able to produce a ranking with moderate and consistent correlations across all iterative backbones, with correlations tending to decrease as the network became less modular to just below 0.5 in the worst case (still demonstrating correlation).

Discussion

As we realize that iteration is computationally expensive, we parallelize MATRIA for the GPU using a four-step process demonstrated by Fig. 6. We can envision GPU threads as a jagged array indexed by two values i and j, where $i < j$. Each thread (i,j) first computes any maximum positive and negative paths between node i and node j in parallel. We then take N threads (for a network with N nodes), one per row, to compute the centrality of each element i. Next, we compute the most central node m on the CPU, followed by each thread (i,j) marking edge (i,j)

Table 5 MATRIA rank vector correlations

Algorithms	A	B	C
MATRIA/ITERCENT$_{BET}$	0.636	0.598	0.386
MATRIA/ITERCENT$_{CLO}$	0.708	0.606	0.404
MATRIA/ITERCENT$_{DEG}$	0.684	0.635	0.486
MATRIA/ITERCENT$_{PN}$	0.696	0.614	0.507
MATRIA/ITERCENT$_{PT}$	0.698	0.597	0.470

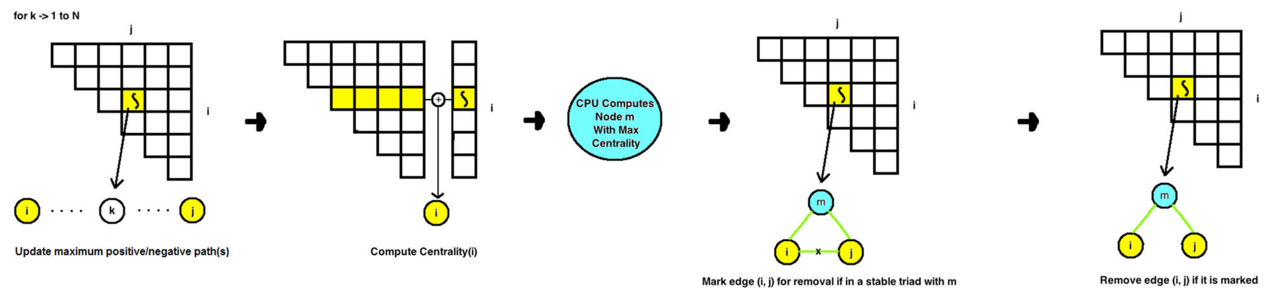
for k -> 1 to N
j
S
i
i
k
j
Update maximum positive/negative path(s)
S
i
i
Compute Centrality(i)
CPU Computes Node m With Max Centrality
j
S
i
m
i
x
j
Mark edge (i, j) for removal if in a stable triad with m
j
S
i
m
i
j
Remove edge (i, j) if it is marked

Fig. 6 Steps for our GPU multi-threaded code, and specific operations for each thread

if it (1) exists and (2) is in a stable triad with m. Finally each thread (i, j) removes edge (i, j) if it is marked. Table 6 shows the wall clock execution time of MATRIA on a Tesla K20 GPU, demonstrating that with this power MATRIA can practically produce results for networks in the low- to mid- thousands. Compared to serial execution on a 1.6 GHz CPU with 16 GB of RAM, this yielded 8- to 16- fold speedups on the first three networks and orders of magnitude speedups on the larger two (respectively over an hour and on pace for multiple days on the CPU). We continue to look for ways to run MATRIA on larger networks.

Conclusions

Our results illustrate that applying iteration to centrality algorithms with different definitions of "importance" and unifying their results gives more meaning to their computed central node sets. By resolving disagreements MATRIA produces a ranked list of central nodes and supernodes, with a cardinality much smaller than the size of the network and several mutually agreed unimportant nodes removed. Rank vectors correlate well between this set and the individual iterative backbones and are much more consistent compared to just the iterative or non-iterative backbones. While cases of unresolvable disagreements can still occur in this unified set, they are rare. Through GPU optimizations MATRIA is currently practical for medium-sized networks, and we are exploring ways to push this boundary. We also plan to experiment with weighted averages when computing overall rankings. Finally, applying MATRIA to directed (i.e. metabolic) biological networks will require an extension of iteration and supernodes to incorporate direction (i.e. adjacency $x \rightarrow y$ would now be different from $x \leftarrow y$), an interesting question that we plan to immediately pursue.

Table 6 MATRIA wall clock execution times

Sample Network	Nodes	GPU Wall Clock Time (s)
Lung Bacterial Co-Occurrence	126	0.48 +/- 0.00
Oyster Gene Co-Expression	203	1.60 +/- 0.00
Scale-Free Synthetic	500	28.52 +/- 0.03
Scale-Free Synthetic	1000	60.61 +/- 0.04
Fruit Fly Protein-Protein	2997	4357.94 +/- 3.97

Abbreviations

ATria: Ablatio Triadum; GPU: Graphics Processing Unit; MATria: Multiple Ablatio Triadum

Acknowledgements

We acknowledge Michael Campos, Mitch Fernandez, Wenrui Huang, Jingan Qu, Juan Daniel Riveros, Victoria Suarez-Ulloa, and Camilo Valdes for useful discussions.

Funding

TC was funded by a Faculty Development Grant from Eckerd College and resources provided by NVIDIA. VAP was supported by the College of Engineering and Computing at FIU. GN was partially supported by the Department of Defense (W911NF-16-1-0494), National Institute of Health (NIH 1R15AI128714-01), National Institute of Justice (NIJ 2017-NE-BX-0001), Florida Department of Health (FDOH 09KW-10), and the Alpha-One Foundation. Publication charges for this article will be paid through personal funds of the authors.

Availability of data and material

MATria is available as a plugin for the PluMA [27] analysis pipeline, available at http://biorg.cs.fiu.edu/pluma/ for download. All applicable input data is also available, along with sample executions from this paper at http://biorg.cs.fiu.edu/pluma/matria.

About this supplement

This article has been published as part of *BMC Bioinformatics Volume 20 Supplement 11, 2019: Selected articles from the 7th IEEE International Conference on Computational Advances in Bio and Medical Sciences (ICCABS 2017): bioinformatics*. The full contents of the supplement are available online at https://bmcbioinformatics.biomedcentral.com/articles/supplements/volume-20-supplement-11.

Authors' contributions

This work was conducted by the Bioinformatics Research Group (BioRG) at Florida International University managed by GN and spearheaded by TC. All authors contributed to all portions of this project, and have read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details
[1]Bioinformatics Research Group (BioRG) & Biomolecular Sciences Institute, School of Computing & Information Sciences, Florida International University, 11200 SW 8th St, 33199 Miami, FL, USA. [2]Center for Neurogenetics, Weill Cornell Medical College, 10021 New York, NY, USA.

Published: 6 June 2019

References

1. Freeman L. C. A set of measures of centrality based on betweenness. Sociometry. 1977;40(1):35–41.
2. Sabidussi G. The centrality index of a graph. Psychometrika. 1966;31(4): 581–603. https://doi.org/10.1007/BF02289527.
3. Havel V. A remark on the existence of finite graphs. Cas Pro Pestovani Matematiky. 1955;80:477–80.
4. Everett M. G., Borgatti S. P. Networks containing negative ties. Soc Networks. 2014;38:111–20.
5. Page L., Brin S., Motwani R., Winograd T. The PageRank citation ranking: bringing order to the web. Tech Rep. 1999. http://ilpubs.stanford.edu:8090/422/.
6. de Kerchove C., Van Dooren P. The PageTrust algorithm: How to rank web pages when negative links are allowed?. In: Proceedings SIAM Data Mining Conference (SDM2008). Atlanta: SIAM; 2008. p. 346–52.
7. Fell D. A., Wagner A. The small world of metabolism. Nat Biotechnol. 2000;18(11):1121–2.
8. Jeong H., Mason S. P., Barabasi A. L., Oltvai Z. N. Lethality and centrality in protein networks. Nature. 2001;411(6833):41–2.
9. Wuchty S. Interaction and domain networks of yeast. Proteomics. 2002;2(12):1715–23.
10. Hahn M. W., Conant G. C., Wagner A. Molecular evolution in large genetic networks: Does connectivity equal constraint?. J Mol Evol. 2004;58(2):203–11. https://doi.org/10.1007/s00239-003-2544-0.
11. Koschutzki D., Schreiber F. Centrality analysis methods for biological networks and their application to gene regulatory networks. Gene Regul Syst Bio. 2008;2:193–201.
12. Wuchty S., Stadler P. F. Centers of complex networks. J Theor Biol. 2003;223(1):45–53.
13. Koschutzki D., Schreiber F. Comparison of centralities for biological networks. In: Proc. German Conf Bioinforma (GCB'04); 2004. p. 199–206.
14. Kim P.-J., Price N. D. Genetic co-occurrence network across sequenced microbes. PLoS Comput Biol. 2011;7(12):1002340.
15. Cickovski T., Peake E., Aguiar-Pulido V., Narasimhan G. ATria: A novel centrality algorithm applied to biological networks. BMC Bioinforma. 2017;18(S8):239–48.
16. Wilson E. O. Consilience: The Unity Of Knowledge. New York, NY: Knopf; 1998.
17. Easley D., Kleinberg J. Networks, Crowds and Markets: Reasoning About a Highly Connected World. Cambridge, UK: Cambridge University Press; 2010.
18. Jackson M. O., Wolinsky A. A strategic model of social and economic networks. Economic Theory. 1996;71(8):44–74.
19. Elzinga M., Collins J. H., Kuehl W. M., Adelstein R. S. Complete amino-acid sequence of actin of rabbit skeletal muscle. Proc Natl Acad Sci. 1973;70(9): 2687–91. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC427084/.
20. Tang T., Zheng B., Chen S.-h., Murphy A. N., Kudlicka K., Zhou H., Farquhar M. G. hNOA1 interacts with complex I and DAP3 and regulates mitochondrial respiration and apoptosis. J Biol Chem. 2009;284(8): 5414–24. https://doi.org/10.1074/jbc.M807797200.
21. Lanneau D., Brunet M., Frisan E., Solary E., Fontenay M., Garrido C. Heat Shock Proteins: Essential proteins for apoptosis regulation. J Cell Mol Med. 2008;12(3):743–61.
22. Morano K. A. New tricks for an old dog. Ann NY Acad Sci. 2007;1113(1): 1–14. https://doi.org/10.1196/annals.1391.018.
23. Zhang L., Li L., Zhu Y., Zhang G., Guo X. Transcriptome analysis reveals a rich gene set related to innate immunity in the eastern oyster. Mar Biotechnol (NY). 2014;16(1):17–33.
24. Kim C. A., Bowie J. U. SAM domains: uniform structure, diversity of function. Trends Biochem Sci. 2013;28(12):625–8. https://doi.org/10.1016/j.tibs.2003.11.001.
25. Gurzov E. N., Stanley W. J., Brodnicki T. C., Thomas H. E. Protein Tyrosine Phosphatases: Molecular switches in metabolism and diabetes. Trends Endocrinol Metab. 2014;26(1):30–9. https://doi.org/10.1016/j.tem.2014.10.004.
26. Denu J. M., Dixon J. E. Protein tyrosine phosphatases: mechanisms of catalysis and regulation. Curr Opin Chem Biol. 1998;2(5):633–41.
27. Cickovski T., Aguiar-Pulido V., Huang W., Mahmoud S., Narasimhan G. Lightweight microbiome analysis pipelines. In: Proceedings of International Work Conference on Bioinformatics and Biomedical Engineering (IWBBIO16). Granada: Springer; 2016.