SOFTWARE TOOL ARTICLE

Finding the shortest path with PesCa: a tool for network reconstruction [version 1; peer review: 2 approved with reservations]

Giovanni Scardoni¹, Gabriele Tosadori¹, Sakshi Pratap², Fausto Spoto³, Carlo Laudanna⁴

¹Center for Biomedical Computing, University of Verona, Verona, Italy

²Birla Institute of Technology & Science, Goa, India

³Department of Computer Science, University of Verona, Verona, Italy

⁴Department of Pathology and Diagnostics,, University of Verona, Verona, Italy

V1 First published: 05 Aug 2015, 4:484
<https://doi.org/10.12688/f1000research.6769.1>

Latest published: 07 Apr 2016, 4:484
<https://doi.org/10.12688/f1000research.6769.2>

Abstract

Network analysis is of growing interest in several fields ranging from economics to biology. Several methods have been developed to investigate different properties of physical networks abstracted as graphs, including quantification of specific topological properties, contextual data enrichment, simulation of pathway dynamics and visual representation. In this context, the PesCa app for the Cytoscape network analysis environment is specifically designed to help researchers infer and manipulate networks based on the shortest path principle. PesCa offers different algorithms allowing network reconstruction and analysis starting from a list of genes, proteins and in general a set of interconnected nodes. The app is useful in the early stage of network analysis, i.e. to create networks or generate clusters based on shortest path computation, but can also help further investigations and, in general, it is suitable for every situation requiring the connection of a set of nodes that apparently do not share links, such as isolated nodes in sub-networks. Overall, the plugin enhances the ability of discovering interesting and not obvious relations between high dimensional sets of interacting objects.

Keywords

biological networks , shortest path , pesca , protein protein interaction networks , connect isolated node , cytoscape

This article is included in the **Cytoscape** gateway.

Open Peer Review

Approval Status ? ✓ ? ✓

	1	2	3	4
version 2				
(revision)		✓	?	✓
07 Apr 2016		view	view	view
		↑		
version 1	?	?		
05 Aug 2015	view	view		

1. **Ferenc Jordán**, Hungarian Academy of Sciences, Budapest, Hungary
2. **Ankush Sharma** , High Performance Computing and Networking Institute, National Research Council, Naples, Italy
3. **Anna Zhukova**, Institut Pasteur, Paris, France
4. **Salvatore Alaimo** , University of Catania, Catania, Italy

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

Corresponding author: Giovanni Scardoni (giovanni.scardoni@gmail.com)

Competing interests: No competing interest were disclosed.

Grant information: This work was supported by: Italian Association for Cancer Research (AIRC, IG 8690) (C.L.); Fondazione Cariverona; Nanomedicine project University of Verona and Fondazione Cariverona (C.L.). Part of the software was developed thanks to the Google Summer of Code 2014.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2015 Scardoni G *et al.* 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. Data associated with the article are available under the terms of the [Creative Commons Zero "No rights reserved" data waiver](#) (CC0 1.0 Public domain dedication).

How to cite this article: Scardoni G, Tosadori G, Pratap S *et al.* **Finding the shortest path with PesCa: a tool for network reconstruction [version 1; peer review: 2 approved with reservations]** F1000Research 2015, **4**:484 <https://doi.org/10.12688/f1000research.6769.1>

First published: 05 Aug 2015, **4**:484 <https://doi.org/10.12688/f1000research.6769.1>

Introduction

Network analysis is a hot area of investigation in different, apparently unrelated, research fields. In particular in biology, biotechnology, and biomedical research, consistent efforts are being carried out in order to investigate how complex biological processes work¹. In this scenario a disease, a metabolic pathway or a coexpression microarray could be analyzed by means of network theoretical formalisms such that the structural properties of such models can be quantified². The central point of this approach concerns the emergence of peculiar properties³ arising when a set of distinct objects reciprocally interact generating a functionally integrated system. In this context, the goal is to uncover the complex behaviors, hidden by system complexity, that are specific to a particular system. The interactions between objects are abstracted as graphs and analyzed by means of graph theory. Thus, it is possible to study the topological role of each network component (node), to uncover hidden structural patterns, find clusters, or even simulate the time evolution of specific network topologies.

In order to identify the hidden properties of a complex system, as a first step it is necessary to reconstruct a network representing the system under investigation. Cytoscape⁴ has several built-in tools allowing network reconstruction and analysis. Notably, in systems biology one general assumption implies that the informational flow follows a maximum parsimony principle (see Box 3 in 5). Consequently, computing the shortest path in a network can have direct functional implications. Such capability is, however, lacking in the basic Cytoscape core. Several algorithms have been developed in order to solve the shortest paths problem, such as Dijkstra⁶, Floyd-Warshall⁷, Bellman-Ford⁸. Here we describe PesCa, a novel Cytoscape app specifically designed to compute the shortest paths between two or more nodes in a network, thus permitting the reconstruction of sub-networks based on the maximum parsimony principle. The generated clusters allow focusing the analysis on sets of nodes characterized by reduced topological complexity. Many options are also implemented, enabling the users to investigate different aspects of network complexity.

Methods

Implementation and operation

PesCa is a Cytoscape app, thus it is not standalone, but only works in conjunction with the Cytoscape environment. The release of PesCa presented here is developed for the 3.x Cytoscape series. The version for the 2.x series is no longer updated and lacks the features of the new release. Since the new version of Cytoscape has a new structure and uses a different architecture the new PesCa is developed and maintained only for the 3.x platform.

The PesCa core is based on the All Pair Shortest Paths (APSP) version of the basic Dijkstra algorithm, single thread; it performs a modified version of the APSP search that finds all the shortest paths between each couple of nodes. Furthermore, PesCa offers further options: for example the *Multi Shortest Paths (S-P Cluster)* is an APSP version computing the shortest paths between all selected nodes in a specific network. The *Multi Shortest Paths Tree* is a modified version of an APSP searching all the shortest paths connecting a single selected node to all other nodes in a network. PesCa is also designed to extract a fully connected sub-network from a giant

network component, thus allowing connecting nodes that apparently do not interact with other network regions (isolated nodes).

Figure 1 shows the main panel and the tasks that can be accomplished with PesCa:

- *Multi Shortest Paths Tree* allows computing all the shortest paths connecting a node to all other nodes in a network.
- *Multi Shortest Paths (S-P Cluster)* allows computing the shortest paths connecting two or more selected nodes in the network. It allows generating network sub-clusters (modules) based on minimal cost.
- *Connect isolated nodes* allows finding all the connecting shortest paths between the isolated node and a group of other nodes in a network (the so called Giant component).

Notably, the *Connect isolated nodes* function connects a node to the nearest nodes in the selected sub-network: this means that the task does not return, as a result, all the shortest paths between the node and the selected subnetwork. Only the shortest paths between the node and the nearest node(s) in the selected cluster are given. It is important to note that the nodes that form the Giant component don't have to share links. The so called Giant component is a set of nodes that is considered as a unique target for this task: the shortest paths are found from the isolated node to this set of nodes. Upon selection of the *Connect isolated nodes* function, a wizard dialog opens guiding the user through the sequence of steps necessary to complete the task.

Figure 1. The main panel with all the options PesCa offers.

For each function PesCa has a button, indicated with a question mark, that opens a new window defining the characteristics and the steps concerning the selected task. The app has several windows that appear during usage, designed to help users. For instance, by selecting the *Multi Shortest Paths (S-P Cluster)* option and then clicking on the Start button without choosing the minimum number of nodes required, the app presents a window prompting to the user to select two or more nodes. Every time that the selected input does not correspond to the expected input, PesCa presents a dialog in order to help the user in selecting the appropriate entries.

It is also possible to analyze directed and undirected networks, depending on the characteristics of the edges. Analysis of networks with weighted edges is also allowed. Notably, edges can have positive and negative weights. If this option is selected, after clicking Start, a window will appear asking for the name of the attribute that stores the information about the weights (edge attribute). Notably, weighted edges can simply correspond to edge length but also provide information about the functional influence of a node on another node(s), such as, for instance, in transcriptomics networks. Thus, this PesCa function may introduce interesting possibilities in the analysis of gene expression networks.

With relatively small networks, e.g. the IntegrinActivation_FN.sif pre-loaded network, PesCa does not require high rates of memory nor long computational times; for instance, only a few seconds are necessary to perform a *Multi Shortest Paths Tree* on a Xubuntu 13.10 machine with an Intel®i5, 2.80GHz CPU and 4 GB of RAM. This network has 3091 nodes and 97115 edges and can be automatically loaded within PesCa. Indeed, the PesCa panel also offers a Select network scrollable menu that permits loading a set of pre-loaded biological networks, in different file formats. The description of these networks is provided at <http://dp.univr.it/~laudanna/LCTST/downloads/index.html>. These networks, and many others, are freely available for the download from this website.

Use cases

A few case studies are now provided, illustrating the functionality of PesCa. The first example describes how to perform a *Multi Shortest Paths (S-P Cluster)* retrieval: the goal is to find all the shortest paths that link two, or more, selected nodes. Figure 2 shows the analysed network (which is provided as **Supplementary material**): ten numbered nodes and 14 undirected edges. The nodes that were used to compute the shortest paths are in yellow: Node 1 and Node 9. After node selection, by clicking the Start button PesCa performed the search.

The result panel in Figure 3 shows the output. The table on the top of the panel lists the retrieved shortest paths, the source for each path and its size. The size, i.e. the length, of a shortest paths is given in terms of how many edges are needed to reach the target. PesCa found four shortest paths, two starting from Node 1 to Node 9 and two starting from Node 9 to Node 1: their length is four. The table below the one already described shows how many paths have a specific length: it groups the paths by their size. In this example PesCa found two shortest paths of length four. It states two shortest

Figure 2. The network we used to perform the *Multi Shortest Paths* retrieval. We used Node 6 as the source and Node 9 as the target.

Figure 3. The results panel that shows the result of our analysis with the *Multi Shortest Paths (S-P Cluster)* option.

paths because the network is undirected and, since the edges are bidirectional, PesCa considers the path “Node 1 to Node 9” equal to the path “Node 9 to Node 1”. Consequently only two paths are listed: one passes through Node 8, 7 and 4, the other one passes through the Node 8, 10 and 4. The last table, at the bottom, shows some characteristics of the network: the average path length is four, the number of unique short paths is two and two other parameters are not relevant to this example.

PesCa retrieved the shortest paths giving the sequence of the nodes that are involved which could be highlighted by selecting a specific path in the table. Furthermore, the button in the top left corner, pass through S-P, enables the user to highlight the shortest paths passing through a selected node.

The second example is used to describe the third table in the results panel, the one with the missing values. By using the network in Figure 2 a *Multi Shortest Path Tree* was computed. To carry out this analysis it is necessary to select a node; in this case Node 1 is used. In Figure 7 the results are shown. The interesting point here is the bottom table; all the options are now defined by a value: the average path length, the number of unique shortest paths, the number of expected paths, and the Connected column. The number of expected paths refers to the total number of shortest paths a network is supposed to develop if it is fully connected. The Connected column could be True or False and states if all the nodes are able to communicate together by means of a path. The network in Figure 2 is connected and the column states True. Now, if the edge between Node 4 and Node 7 is removed, then two different connected components will appear. The network is now disconnected and the value will be False. Furthermore, if a network is directed, see Figure 4, the returned value can be both True or False. In the example it will be True if the *Multi Shortest Path Tree* is computed from Node 1. It will be False if the *Multi Shortest Path Tree* is computed from Node 2 and 3 because neither are able to reach Node 1.

Figure 4. A directed network that has non communicating nodes.

The third example describes how to use the *Connect isolated component*. Figure 5 shows the network used for the analysis. Again, there are ten nodes, fourteen edges, and a few highlighted nodes in yellow. In this analysis, PesCa retrieved the paths from Node 6 to the cluster formed by Node 8, 9 and 10. After selecting the option, by clicking Start, a window shows up like the one in Figure 6, and guides the user in selecting the Giant component. The Giant component is the cluster to which PesCa will connect the node. In this example the component is represented by Node 8, 9 and 10. By selecting it and then clicking Ok the user is able to choose the Isolated Node: highlight Node 6 and then click on Ok. Finally by clicking Start, PesCa will run the algorithm.

The results show two shortest paths, one reaching Node 8 and one reaching Node 10; Node 9 is not considered as a target because it

Figure 5. The network we used to show how to connect Node 6 to the sub-network formed by Node 8, 9, 10.

Figure 6. The panel that will help in connecting disconnected nodes.

Figure 7. The results panel for the *All nodes Shortest Paths* option.

does not develop a shortest path with Node 6 since it is connected to Node 6 by means of Node 8 and 10.

Summary

We have briefly introduced the main functionalities of PesCa. We described a few application cases by using a very simple network in order to show how to setup the input and how the output panel works. Overall, PesCa is designed for sub-network retrieval and

shortest paths search and, in the Cytoscape context, it is the only app that performs this task. It can be used to enhance the predictive power of biological networks by reducing the complexity of the processes under investigation and, in conjunction with other apps, it permits the researcher to deeply investigate the properties of subsets of nodes.

Software availability

- 1. Software available from: <http://apps.cytoscape.org/apps/pesca30>
- 2. Latest source code: <https://bitbucket.org/giovanniscardoni/pes-careleasefor3public>
- 3. Link to archived source code as at time of publication: <http://dx.doi.org/10.5281/zenodo.211459>
- 4. Software license: Lesser GNU Public License 3.0: <https://www.gnu.org/licenses/lgpl.html>

Author contributions

GS designed and implemented the software, GT wrote the manuscript, SP implemented the software, FS participated in the design, CL participated in the design and in the revision of the manuscript.

Competing interests

No competing interest were disclosed.

Grant information

This work was supported by: Italian Association for Cancer Research (AIRC, IG 8690) (C.L.); Fondazione Cariverona; Nano-medicine project University of Verona and Fondazione Cariverona (C.L.). Part of the software was developed thanks to the Google Summer of Code 2014.

I confirm that the funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Supplementary material

Analysed network for use cases. Available as a simple interaction file.
[Click here to access the data](#)

References

1. Ideker T, Galitski T, Hood L: **A new approach to decoding life: Systems biology.** *Annu Rev Genomics Hum Genet.* 2001; **2**(1): 343–372.
[PubMed Abstract](#) | [Publisher Full Text](#)

2. Pavlopoulos GA, Secrier M, Moschopoulos CN, *et al.*: **Using graph theory to analyze biological networks.** *BioData Min.* 2011; **4**(1): 10.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

3. Bhalla US, Iyengar R: **Emergent properties of networks of biological signaling pathways.** *Science.* 1999; **283**(5400): 381–387.
[PubMed Abstract](#) | [Publisher Full Text](#)

4. Shannon P, Markiel A, Ozier O, *et al.*: **Cytoscape: a software environment for integrated models of biomolecular interaction networks.** *Genome Res.* 2003; **13**(11): 2498–2504.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

5. Barabasi AL, Gulbahce N, Loscalzo J: **Network medicine: a network-based approach to human disease.** *Nat Rev Gene.* 2011; **12**(1): 56–68.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

6. Dijkstra EW: **A note on two problems in connexion with graphs.** *Numer Math.* 1959; **1**(1): 269–271.
[Publisher Full Text](#)

7. Floyd RW: **Algorithm 97: Shortest path.** *Commun ACM.* 1962; **5**(6): 345.
[Publisher Full Text](#)

8. Bellman R: **On a routing problem.** *Quart Appl Math.* 1958; **16**(1): 87–90.
[Reference Source](#)

9. Scardoni G, Tosadori G, Pratap S, *et al.*: **Finding the shortest path with PesCa: a tool for network reconstruction.** *Zenodo.* 2015.
[Data Source](#)

Open Peer Review

Current Peer Review Status: ? ?

Version 1

Reviewer Report 01 September 2015

<https://doi.org/10.5256/f1000research.7273.r9885>

© 2015 Sharma 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.

Ankush Sharma

Laboratory for Genomics, Transcriptomics and Proteomics, High Performance Computing and Networking Institute, National Research Council, Naples, Italy

The authors describe the useful PesCa app for Cytoscape version 3.x for computing the shortest paths in networks and shortest paths connecting any two selected nodes and reconstruction of subnetworks based on maximum parsimony principle. The PesCa app scales well with efficiency in computation time on large networks. I agree with the point raised by reviewer 1 in version 1 of referee report.

Minor concern

1. Using terminology such as isolated nodes (nodes with no connection) and giant networks is making article confusing.
2. Maximum parsimony principle needs to be elaborated in details. and how it scores over other methods such as maximum likelihood used for reconstruction of weighted and unweighted networks along with one illustrative example will help in better understanding of the concept.
3. The representational figure 2 for demonstrating short paths from node 1 to node 9 can be improved by coloring edges involved in short paths with different color.

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, however I have significant reservations, as outlined above.

Reviewer Report 24 August 2015

<https://doi.org/10.5256/f1000research.7273.r9882>

© 2015 Jordán F. 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.

Ferenc Jordán

Centre for Ecological Research, Hungarian Academy of Sciences, Budapest, 1051, Hungary

The Authors present a useful Cytoscape app for calculating shortest paths in networks.

This is useful with the following comments to be considered:

I do not suggest to use the words "isolate" and "giant component" for the situation when the distance between a well-defined node a well-defined subnetworks is to calculated. In network science, there is enough confusion about certain concepts, it is better to avoid adding more. If you refer to a "selected node" and a "selected subgraph", everything is perfectly understandable and there is no confusion.

In the second paragraph (4th line) of the Introduction, you speak about "reconstructing" networks. Why not simply "constructing"?

The maximum parsimony principle may work well or may not work at all, depending on the actual problem. I do not suggest to use it as a general assumption. It would be more interesting (and a lot more useful) to dedicate a brief paragraph for this issue: when does it make sense (network flows flow shortest paths) and when it does not (network flows do not prefer shortest paths).

In the legend of Figure 2, you mention node 6 instead of node 1.

In Figure 5, I suggest to keep nodes 8, 9 and 10 yellow but to use another colour for node 6.

Please add a bit more explanation about how to manage information about link direction, link weight and link sign in the database. Can you convert (symmetrize, binarize) and if yes, how does it happen (e.g. how to determine weight if you symmetrize AB and BA with different weights?).

What happens if a node is really isolated (degree equals zero in the original graph)? Can you perform a calculation based on the reciprocal distance matrix?

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, however I have significant reservations, as outlined above.

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