NetBone: A Python Package for Extracting Backbones of Weighted Networks

Ali Yassin, Abbas Haidar, Hocine Cherifi, Hamida Seba, Olivier Togni

Noname manuscript No.
(will be inserted by the editor)

NetBone: A Python Package for Extracting Backbones of Weighted Networks

Ali Yassin · Abbas Haidar · Hocine Cherifi · Hamida Seba · Olivier Togni

Received: date / Accepted: date

Abstract NetBone is a new open-source Python package designed to simplify analyzing complex networks. With a wide range of techniques available, NetBone allows researchers to extract the backbone of a network while preserving its essential structure. The package includes nine structural methods and five statistical techniques, offering users a comprehensive solution to network analysis. It is user-friendly and straightforward to use, with easy installation. The package accepts different types of inputs, including data frames or Networkx graphs, and provides evaluation measures for comparative purposes. Additionally, NetBone offers an option to generate plots. Its versatility makes it a valuable tool for data scientists and social scientists, significantly enhancing their research and data analysis capabilities.

Keywords Complex Networks · Backbone Filtering Techniques · Network Compression · Graph Summarization · Sparsification

This material is based upon work supported by the Agence Nationale de Recherche under grant ANR-20-CE23-0002.

A. Yassin
Laboratoire d'Informatique de Bourgogne, University of Burgundy, Dijon, France
E-mail: ali_yassin@etu.u-bourgogne.fr

A. Haidar
Computer Science Department, Lebanese University, Beirut, Lebanon

H. Cherifi
ICB UMR 6303 CNRS - Univ. Bourgogne - Franche-Comté, Dijon, France

H. Seba
Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne, France

O. Togni
Laboratoire d'Informatique de Bourgogne, University of Burgundy, Dijon, France

1 Introduction

Networks are increasingly important in scientific research across various fields, providing a valuable means of understanding relationships between different entities from different domains, including biological, infrastructural, and social sciences. Network analysis has been applied to diverse research questions, from studying the spread of diseases to understanding social hierarchies and beyond. However, as networks become more complex, analyzing them can be challenging, especially for those with a large number of nodes and edges. To address this issue, many approaches have been developed for reducing the size of networks while preserving their essential structure.

One of the most common methods for analyzing complex networks is to extract their backbone, which involves identifying the most significant edges and nodes while discarding extraneous information [1–9]. There are two primary categories of backbone extraction techniques: structural and statistical. Structural techniques focus on the network's topological properties and extract a backbone with specific topological features. Statistical techniques, on the other hand, assess the importance of edges and nodes based on hypothesis testing or empirical distribution, frequently used to eliminate noise in the network.

Despite the variety of approaches available for extracting the network backbone, there is currently no common framework in the field of network analysis that simplifies their application. To overcome this issue, we have introduced a new Python package called "NetBone" that offers a range of filtering techniques for extracting the network backbone.

2 Netbone presentation

The package provides nine structural techniques, such as the maximum spanning tree, global threshold, doubly stochastic, metric backbone, ultra-metric backbone, high salience skeleton, h-backbone, modularity backbone, and overlapping nodes and hubs backbone. Additionally, it offers five statistical techniques, including the disparity filter, marginal likelihood filter, noise corrected filter, locally adaptive network sparsification filter, and enhanced configuration mode filter.

The installation and usage of the NetBone package for network analysis are user-friendly and straightforward. With just a few simple steps, users can install the package via pip or directly from the repository [1]. Once installed, the user can simply import the package and call the desired method, giving it the appropriate input. NetBone accepts different types of inputs, including data frames or Networkx graphs. Then the user can then choose from a variety of filters, such as the boolean filter, fraction filter, or threshold filter. For instance, the boolean filter is suitable for methods that extract one subgraph that cannot be modified, such as the metric backbone or maximum spanning tree filter.

[1] `https://gitlab.liris.cnrs.fr/coregraphie/netbone/`

On the other hand, the threshold or fraction filter is useful for methods like the high salience skeleton or disparity filter that assign scores or p-values to the edges, allowing the user to set a threshold or keep only a specified fraction of edges or edges within the desired threshold. Listing 1 shows a sample code snippet on how to install and use the NetBone python package for network analysis.

In addition, NetBone not only offers various filtering techniques for extracting the backbone of a network, but it also includes evaluation measures utilized by Serrano [10] for comparing different methods. These measures encompass the fraction of edges, nodes, weights, and the number of components preserved in the backbone, as well as the cumulative weight and degree distribution. Moreover, NetBone provides an option for users to visualize the results by generating plots through the use of matplotlib and seaborn python libraries.

3 Conclusion

NetBone is a versatile and easy-to-use tool for network analysis that provides a comprehensive solution. Its open-source nature allows users to adapt and customize it to their specific needs. With its wide range of filtering techniques and evaluation measures, NetBone is suitable for researchers, data scientists, and social scientists alike. By providing an efficient way to analyze complex networks, NetBone has the potential to enhance your research and data analysis greatly.

```
# install NetBone
!pip install netbone
#######################################################

# import NetBone and used libraries
import NetBone as nb
import networkx as nx
import pandas as pd

#######################################################

# read the weighted edge list using networkx
G = nx.read_weighted_edgelist(filename)

# read the weighted edge list using pandas
G = pd.read_csv(filename)

#######################################################

# apply the Metric Backbone filter
m_backbone = nb.metric_backbone(G)

# apply the High Salience Skeleton filter
hss_backbone = nb.high_salience_skeleton(G)

#######################################################
```

```
27
28 # apply a boolean filter to extract the backbone
29 m_backbone = nb.boolean_filter(m_backbone)
30
31 # apply a threshold filter on the scores to keep all edges with
       scores higher than 0.5
32 hss_backbone = nb.threshold_filter(hss_backbone, threshold=0.5)
33
34 # apply a fraction filter on the scores to keep 30% of the network
35 hss_backbone = nb.fraction_filter(hss_backbone, fraction=0.3)
```

Listing 1 Here is a sample code snippet showing how to install and use the NetBone python package for network analysis

References

1. C.H. Gomes Ferreira, F. Murai, A.P. Silva, M. Trevisan, L. Vassio, I. Drago, M. Mellia, J.M. Almeida, Plos one **17**(9), e0274218 (2022)
2. Z. Ghalmane, C. Cherifi, H. Cherifi, M. El Hassouni, Scientific Reports **10**(1), 1 (2020)
3. Z. Ghalmane, C. Cherifi, H. Cherifi, M. El Hassouni, in *9th International Conference on Complex Networks and Their Applications* (2020), pp. p–3
4. V. Gemmetto, A. Cardillo, D. Garlaschelli, arXiv preprint arXiv:1706.00230 (2017)
5. Z. Ghalmane, C. Cherifi, H. Cherifi, M. El Hassouni, Information Sciences **576**, 454 (2021)
6. S. Rajeh, M. Savonnet, E. Leclercq, H. Cherifi, in *Network Science: 7th International Winter Conference, NetSci-X 2022, Porto, Portugal, February 8–11, 2022, Proceedings* (Springer International Publishing Cham, 2022), pp. 67–79
7. A. Yassin, H. Cherifi, H. Seba, O. Togni, in *2022 IEEE Workshop on Complexity in Engineering (COMPENG)* (IEEE, 2022), pp. 1–8
8. A. Yassin, H. Cherifi, H. Seba, O. Togni, in *Complex Networks and Their Applications XI: Proceedings of The Eleventh International Conference on Complex Networks and their Applications: COMPLEX NETWORKS 2022—Volume 2* (Springer International Publishing Cham, 2023), pp. 551–564
9. L. Dai, B. Derudder, X. Liu, Journal of Transport Geography **69**, 271 (2018)
10. M.A. Serrano, M. Boguna, A. Vespignani, Proceedings of the National Academy of Sciences **106**, 6483 (2009). DOI 10.1073/pnas.0808904106