A Unified Graph Clustering Framework for Complex Systems Modeling

Bruno Gaume, Ixandra Achitouv, David Chavalarias

A Unified Graph Clustering Framework for Complex Systems Modeling

Bruno Gaume[1,3*] Ixandra Achitouv[3†], David Chavalarias[2,3‡]

[1] Cognition, Langues, Langage, Ergonomie (CLLE, UMR 5263), CNRS, France

[2] Centre d'Analyse et de Mathématique Sociales (CAMS, UMR8557), CNRS, France

[3] Institut des Systemes Complexes de Paris IdF (ISC-PIF, UAR3611), CNRS, France

*Corresponding author: bruno.gaumeATiscpif.org

†ixandra.achitouvATcnrs.fr, ‡David.chavalariasATiscpif.fr

Abstract

Networks are pervasive for complex systems modeling, from biology to language or social sciences, ecosystems or computer science. Detecting communities in networks is among the main methods to reveal meaningful structural patterns for the understanding of those systems. Although dozens of clustering methods have been proposed so far, sometimes including parameters such as resolution or scaling, there is no unified framework for selecting the method best suited to a research objective. After more than 20 years of research, scientists still justify their methodological choice based on *ad-hoc* comparisons with 'ground-truth' or synthetic networks, making it challenging to perform comparative study between those methods. This paper proposes a unified framework, based on easy-to-understand measures, that enables the selection of appropriate clustering methods according to the situation. If required, it can also be used to fine-tune their parameters by interpreting them as *description scale* parameters. We demonstrate that a new family of algorithms inspired by our approach outperforms a set of state-of-the-art community detection algorithms, by comparing them on a benchmark dataset. We believe our approach has the potential to provide a fresh start and a solid foundation for the development and evaluation of clustering methods across a wide range of disciplines.

1 Rationale

From biology to social sciences, ecosystems or computer science, complex systems are defined as large sets of entities interacting in a decentralized ways. Networks (or graphs) are one of the main conceptual structures for modeling them [22, 9, 3, 20], where nodes $\{v \in V\}$ represent the basic entities and edges $\{e \in E\}$, defined as pairs of nodes, represent their interactions.

This simplified representation of a complex system has the advantage of revealing groups of densely connected nodes called *modules*, from the study of which we can infer or deduce particular characteristics or functions. A complex system can then be conceptualized as the interactions between these modules, an operation that *de facto* defines a *scale of description*. The counterpart of these modules in the conceptual representation are called *clusters* or *communities*.

This poses two legitimate questions: How to decompose a complex system? And which are the appropriate scales for this decomposition? The first question is a fast growing research field *per se* in mathematics and computer science with thousands publications per year. It is the art to define clustering or community detection methods on graphs. Some of these methods have been criticized for their inhability to adapt to different scales of observation [6], while others include explicitly a resolution parameter to address second question [5, 13]. There is however no unified framework to compare clustering methods two by two and consequently to choose one rather than the other. The lack of precise semantics in defining what constitutes a *'good clustering'* leads to incomparable outputs from different clustering methods, hindering constructive debates on their comparative advantages and slowing down research progress.

Hereafter, we provide such semantics as well as an associated unified framework to compare any clustering methods on undirected and unweighted graphs.

Types of graph clustering

For a set of vertices V (the entities), let's note $\mathcal{P}(V)$ the subsets of V and $\mathcal{P}_2(V) \subset \mathcal{P}(V)$ the pairs of elements from V. For $E \subset \mathcal{P}_2(V)$, $G = (V, E)$ defines an undirected graph on V.

By definition, a set $\mathcal{C} \in \mathcal{P}(\mathcal{P}(V))$ such that $\mathcal{C} = \{C_i | C_i \subset V, C_i \neq \emptyset, i \in I\}$ is a clustering of G with clusters C_i if and only if $\bigcup_{i \in I} C_i = V$. It is a

partitional clustering if clusters do not overlap ($\forall i \neq j \in I, C_i \cap C_j = \emptyset$), else it is an *overlapping clustering*. The number of partitional clustering of a set of size $n = |V|$ is equal to the n^{th} Bell number, a sequence known to grow exponentially [11]. Consequently, this definition tells us what a clustering is, but not what a *'good clustering'* is among the huge number of possible clustering. Therefore we need metrics to evaluate clustering according to our needs. The state of the art identifies more than 70 different metrics to evaluate the quality of a clustering [23, 4, 7], which fall into two categories:

> **Intrinsic metrics** aiming at finding clustering on a graph G according to some general principles (like the modularity of [16, 17] or the compressibility of [19]);
>
> **Extrinsic metrics** aiming at evaluating clustering (e.g. the Rand index [18]) in relation to *a priori* known structures or 'ground-truth' clustering, such as clusters of synthetic networks [12].

Then in each category, one can find metrics for *partitional clustering* or *overlapping clustering*. This leads to four kinds of metrics: *intrinsic* or *extrinsic*, for clusters with or without overlapping. Up to our knowledge (see also [23, 4, 7]), no framework so far has been proposed for the simultaneous evaluation of *intrinsic* and *extrinsic* metrics for *partition* and *overlapping* clustering.

2 Rethinking graph clustering

Graph clustering interpreted as graph compression

To overcome this problem, let's start giving some semantics to clustering methods. If we interpret a clustering as a means to describe a complex systems at some scale, or to 'compress' the graph that describes it, it's essential to note that the typical definition of clustering doesn't explicitly address the edges of the graph. Since edges play a crucial role in graph description, representing a graph through its clusters shouldn't disregard the edges. Instead, it involves approximating that all elements within a cluster are connected while assuming no connections exist between clusters. This simplifies the network description as a set of cliques, with some edges within cliques being falsely observed (false positives) and some edges between cliques being omitted (false negatives).

This approximation of a graph by a clustering can be formalized by defining $\widehat{\mathcal{C}} = \Big(U(\mathcal{C}),\ \Xi(\mathcal{C})\Big)$ the derived graph from the clustering $\mathcal{C}$ using the following two functions:

$$U : \mathcal{P}(\mathcal{P}(V)) \longrightarrow \mathcal{P}(V),\ U(\mathcal{C}) = \bigcup_{C_i \in \mathcal{C}} C_i \tag{1}$$

$$\Xi : \mathcal{P}(\mathcal{P}(V)) \longrightarrow \mathcal{P}_2(V),\ \Xi(\mathcal{C}) = \bigcup_{C_i \in \mathcal{C}} \mathcal{P}_2(C_i) \tag{2}$$

These functions satisfy the following properties:

- If $\mathcal{C}$ is a clustering of a graph $G = (V, E)$ then $U(\mathcal{C}) = V$;
- $\forall\ E \subset \mathcal{P}_2(V),\ \Xi(E) = E$;
- $\forall\ \mathcal{C} \subset \mathcal{P}(\mathcal{P}(V)),\ \Xi(\Xi(\mathcal{C})) = \Xi(\mathcal{C})$.

And for any clustering $\mathcal{C}$ on a graph G we can compute the following metrics which assess the capacity of $\widehat{\mathcal{C}}$ to approximate G:

Precision: $P(\widehat{\mathcal{C}}, G) = \frac{|\Xi(\mathcal{C}) \cap E|}{|\Xi(\mathcal{C})|}$. This is the probability that an edge drawn at random in $\Xi(\mathcal{C})$ actually belongs to E. It measures the ability of the clustering $\mathcal{C}$ not to include non-edges of the graph G in its clusters.

Recall: $R(\widehat{\mathcal{C}}, G) = \frac{|\Xi(\mathcal{C}) \cap E|}{|E|}$. This is the probability that an edge drawn at random in E belongs to $\Xi(\mathcal{C})$. It measures the ability of the clustering $\mathcal{C}$ to include edges of the graph G in its clusters.

Then for any clustering $\mathcal{C}$ on a graph $G = (V, E)$, precision and recall satisfy the following property:

$$\begin{gathered} P(\widehat{\mathcal{C}}, G) = 1\ \&\ R(\widehat{\mathcal{C}}, G) = 1 \\ \Updownarrow \\ U(\mathcal{C}) = V\ \&\ \Xi(\mathcal{C}) = E \\ \Updownarrow \\ \widehat{\mathcal{C}} = G \end{gathered}$$

For *overlapping clustering*, the set of maximal cliques $\mathcal{C}_{mc}$ on a graph G is such that $\widehat{\mathcal{C}_{mc}} = G$ and $\sum_{C_i \in \mathcal{C}_{cm}} |C_i| \leqslant |E|$ (it lossless compresses the graph G, see[10]). For *partitional clustering*, these two measures are antagonistic, unless the graph is reduced to a set of unconnected cliques: improving the *recall* decreases the *precision* and improving the *precision* decreases the *recall*.

Graph clustering as a bi-objective task

The antagonistic relation between precision and recall, as defined above, means that describing complex systems as sets of non overlapping clusters on networks (or as lossy compressed networks) should be envisioned as a bi-objective approach that generally does not have solutions optimizing both objectives at the same time. The evaluation of these methods should thus be parameterized by the desired trade-off between *precision* and *recall* of the corresponding description. One of the common way to do this parameterization is to use the F-score function:

$$F_s(P, R) = \frac{(1 + f(s)^2).(P.R)}{R + f(s)^2.P} \tag{3}$$

With $f(s) = tan(\frac{\pi.s}{2})$ and $F_1(P, R) = R$.

For $s = 0.5$ *precision* and *recall* are of the same importance (when you compress the graph, loosing a edge or adding one costs as much), for $s = 0$, only the *precision* counts, whereas for $s = 1$, only the *recall* counts.

This trade-off defines a scale of description of the system under study: for s values close to 0, precision will be higher with a greater number of smaller and denser clusters ; while for s values close to 1, recall will be higher with a fewer number of bigger but less dense clusters (eventually only one, with all nodes). The scale s can be used to adjust the granularity of our point of view on the real-world, as with the wheel of a telescope.

To illustrate the change in perspective of the proposed framework, we consider a toy graph G_{toy}, all its possible partitions and their respective precision and recall scores (cf. Fig. 1). Some of these partitions have the properties that no other partition exist that both increases the precision and the recall. These special partitions are called *Pareto front* – or Pareto optimal solutions, noted $Pos(Gtoy)$. To select a particular partition clustering from the Pareto front, we need to specify our priorities in terms of precision and recall, essentially determining a scale of description represented by a value s. This decision hinges on defining what constitutes a 'good clustering'. Only after deciding on a description scale can we compare two clusterings and assess the performance of different clustering methods.

Understanding complex systems necessitates comparing various scales of organization. Typically, a fundamental scale is chosen for measuring basic entity properties, alongside a separate scale for describing interactions between these entities. For example, the study of living systems can focus on the higher order structures (modules) built from different types of basic

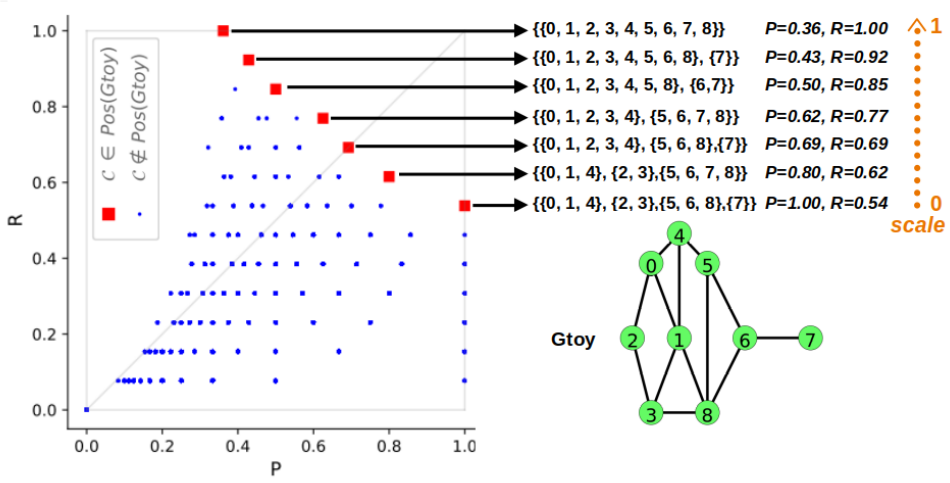

Figure 1: **The set of all the 21 147 partitional clustering of Gtoy in the precision (P) / recall (R) space. The Pareto front is highlighted in red.**

entities such as the genes, the cells, the organs, the individuals, etc. Each description is complementary but offers distinct insights. Thus, analyzing complex systems involves both *subjective* decisions (regarding the choice of the basic entities and scale of interaction description) ; and an *objective* methodology (determining the optimal system division to unveil modules at the chosen scale). If clustering methods are viewed as tools for defining these structures, *establishing a description scale precedes the selection of clustering methods*. Consequently, the approach outlined in this paper offers a unified framework for comparing clustering methods within a chosen interaction description scale.

3 Results

New clustering methods based on F_s optimization

We have shown so far that it is possible to propose a conceptual framework for comparing any type of clustering. Let's now show that it can also be used to define a family of new clustering methods, hereafter noted $nPnB$ clustering, that outperform existing ones.

Given an undirected and unweighted graph $G = (V, E)$ and a desired

description scale s_p, the trivial clustering $\mathcal{C} = \{\{i\} | \ i \in V\}$ where each vertice is assigned to its own cluster is a partitional clustering. Then $\forall s \in [0,1]$ its F_s score $F_s(P(\widehat{\mathcal{C}}, G), R(\widehat{\mathcal{C}}, G)) = 0$ since its recall $R(\widehat{\mathcal{C}}, G) = 0$.

We can then improve this trivial clustering by an agglomeration process that reviews each edge of G only once and merges the clusters of their vertices if this operation does not decrease $F_{s_p}(P(\widehat{\mathcal{C}}, G), R(\widehat{\mathcal{C}}, G))$ (cf. Sect. 4 algorithm 1 $nPnB^{s_p}$ for pseudo-code).

The order in which edges are traversed is essential. The proposed algorithm involves choosing an ordering function on E derived from a similarity measure $Sim(G, x, y)$ on the vertices of G. Edges $\{x, y\} \in E$ are then reviewed by descending order of $Sim(G, x, y)$.

The intuition is as follows: since graph classification aims to group vertices sharing certain structural properties, pairs of vertices that are most similar should be considered first in the clustering process.

The quality of the process strongly depends on the choice of the Sim similarity measure, and there is no guarantee of obtaining an optimal approximation $\widehat{\mathcal{C}}$ of G relatively to F_{s_p}. However, few trials are sufficient to find similarity measures such that the associated partitional clustering outperforms state-of-the-art partitional clustering methods.

We tested 84 state-of-the-art similarities [15]. One of the best scalable metrics was $CosP$ which has been chosen in the subsequent application (cf. Sect. 4):

$$\begin{array}{c} CosP(G = (V, E), x, y) \\ \| \\ Cosinus\left(\overrightarrow{(P^2_G(x \rightsquigarrow x), P^2_G(x \rightsquigarrow y))}, \overrightarrow{(P^2_G(y \rightsquigarrow x), P^2_G(y \rightsquigarrow y))}\right) \end{array}$$

This approach can be generalized to define families of overlapping clustering $nPnB^{s_p}_{s_o}$ (cf. Sect. 4 and algorithm 2) where s_p defines the desired scale of description and s_o defines the desired amount of overlapping.

Clustering methods comparison

In the following, we will distinguish the use of Eq. 3 as the function F_s to be optimized for the family $nPnB^s$ using the variable name s, and its use as the function F_σ in the context of the selection of the description scale using the variable name σ to evaluate the various clustering methods. Note that the metric $F_{\sigma=0.5}$ gives equal importance to *precision* and *recall* (Eq. 3) ;

and can be interpreted as a 'middle point of view'. It has both homogeneity and completeness, two fundamental properties for metrics intending compare clusterings [1]. On contrary, *precision* has only homogeneity property –it is the archetypal metric of homogeneity– and *recall* has only completness property –it is the archetypal metric of completness.

We now show how we can compare the performance of the proposed $nPnB$ clustering family and several hitherto incommensurable state-of-the-art algorithms: Louvain [2], Infomap [19] , Starling [8], and Spectral Graph Clustering [13] – for which we consider several resolutions– applied on a real-world network $G_{em} = (V_{em}, E_{em})$ [24].

This network G_{em} describes e-mail data from a large research institution composed of a set V_{em} of employees. This is a standard benchmark with $|V_{em}| = 1,005$, $|E_{em}| = 16,064$. The graph contains an undirected edge $\{i, j\}$ if employee i and employee j have exchanged at least one e-mail either way. The dataset (available at `https://snap.stanford.edu/data/email-Eu-core.html`) on which G_{em} is build, also contains the list of the 42 departments of the research institute that are often considered as a 'ground-truth' partition $\mathcal{C}_{Dep}$ on G_{em}.

We add to this clusterings comparison the Oracle method met_{Dep} returning the 'ground-truth' partition $\mathcal{C}_{Dep}$ itself and the omniscient overlapping clustering method met_E returning the edges of graph itself ($\mathcal{C} = E \subset \mathcal{P}_2(V)$).

For $nPnB$ clustering, we consider two families: the partitional clustering family $nPnB^{s_p}$, returning partitional clustering based on the optimization of F_{s_p} and the overlapping clustering $nPnB^{s_p}_{s_o}$, returning overlapping clustering based on gluantly extending the clusters produced by $nPnB^{s_p}$ through the optimization of F_{s_o}.

Spectral Graph Clustering (SGC, [13]) methods require to specify the number κ of clusters. Our comparison includes the SGC partitional clustering for $\kappa = 24$ (SGC_{24}) and $\kappa = 54$ (SGC_{54}). We select these two values because at scale $\sigma = 0.5$, which is a natural entry point to compare clustering given the two properties of *homogeneity* and *completeness* of $F_{\sigma=0.5}$, (i) SGC_{24} is the one with the best *extrinsic* score relatively to $\mathcal{C}_{Dep}$ and (ii) SGC_{54} is the one with the best *intrinsic* score ; i.e. $\forall \kappa \in \mathbb{N},\ 0 < \kappa \leq |V_{em}|$:

(i) $F_{0.5}(R^{Dep}_{\kappa}, P^{Dep}_{\kappa}) < F_{0.5}(R^{Dep}_{24}, P^{Dep}_{24})$ with $R^{Dep}_{\kappa} = R(\widehat{SGC_{\kappa}},\ \widehat{\mathcal{C}_{Dep}})$ and $P^{Dep}_{\kappa} = P(\widehat{SGC_{\kappa}},\ \widehat{\mathcal{C}_{Dep}})$;

(ii) $F_{0.5}(R^{em}_{\kappa}, P^{em}_{\kappa}) < F_{0.5}(R^{em}_{54}, P^{em}_{54})$ with $R^{em}_{\kappa} = R(\widehat{SGC_{\kappa}}, G_{em})$ and

$$P_{\kappa}^{em} = P(\widehat{SGC_{\kappa}}, G_{em}) \ ;$$

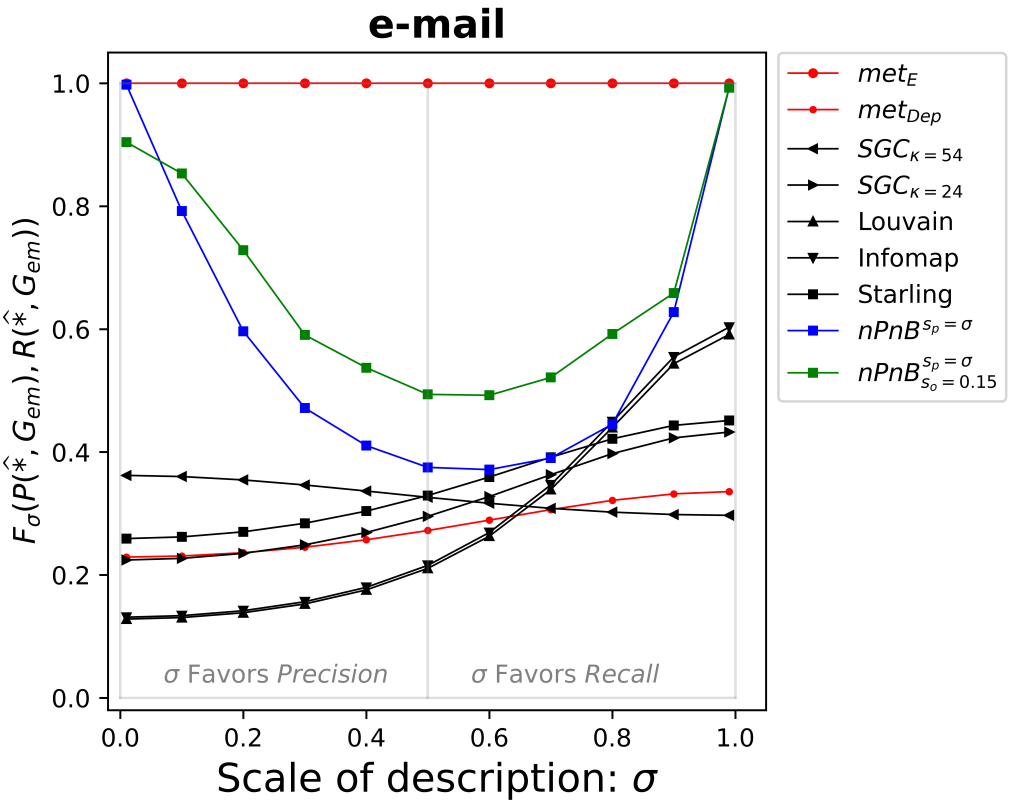

Figure 2: **Performance $\mathbf{F}_\sigma(\mathbf{P}(\hat{*}, \mathbf{G_{em}}), \mathbf{R}(\hat{*}, \mathbf{G_{em}}))$ of clustering methods as derived graphs $\hat{*} = \widehat{\mathbf{met}(\mathbf{G_{em}})}$ according to the description scale σ.**

Fig. 2 displays the performances of each clustering method according to the scale of description $\sigma \in [0, 1]$. A key result is that the family of overlapping clusterings $\{nPnB^{s_p=\sigma}_{s_o=0.15}\}_{\sigma\in[0\ 1]}$ outperforms all other methods when their scale parameter s_p is set to coincide with the desired scale of description σ. This result holds if we restrict ourselves to partitional clustering. For a given scale of description σ, $nPnB^{s_p=\sigma}$ outperforms the other partitional clustering methods tested. Removing those two families of clustering methods from the comparison, none of the methods tested outperforms the others for all scales of description σ.

Fig. 3 displays methods applied to G_{em} on the *precision-recall* plane. It highlights the trade-off made by each clustering methods in terms of *precision* and *recall*. Several lessons can be drawn from this visualization:

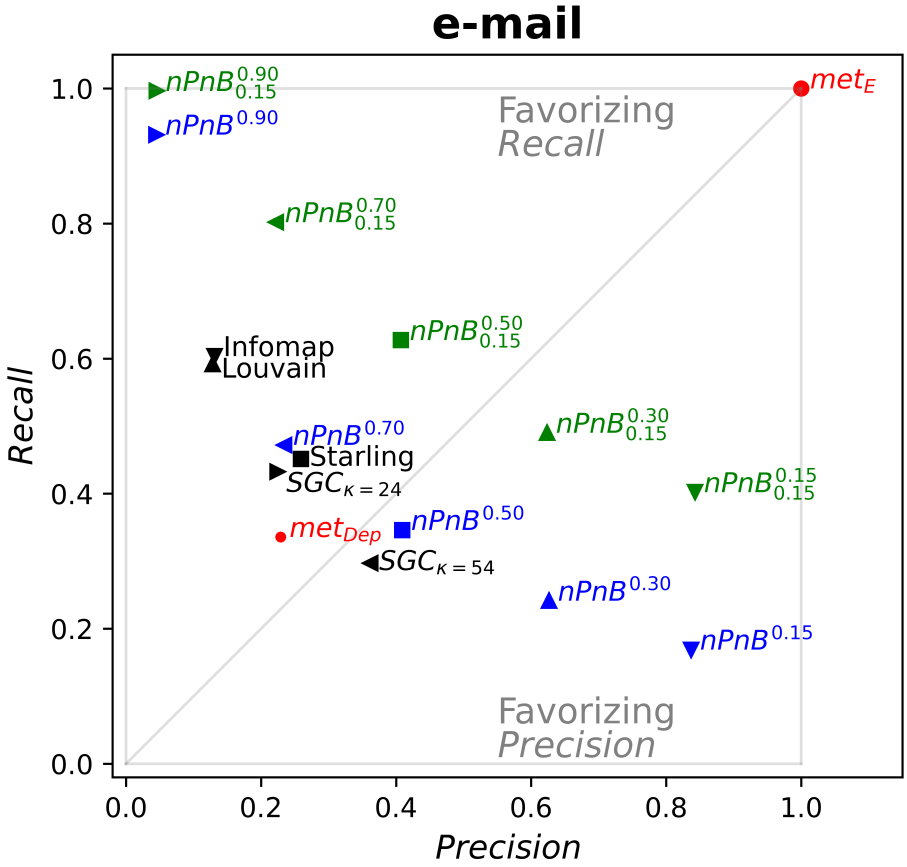

Figure 3: **Comparison in the precision-recall plane of the performances of different clustering methods when applied to the e-mail graph $\mathbf{G_{em}}$. The Oracle methods $\mathbf{met_{Dep}}$ and the Omniscient methods $\mathbf{met_E}$ are highligthed in red.**

First: Non parameterized methods like Louvain, Infomap or Starling differ in the trade-offs they make.

Second: Parameterized methods SGC perform less well on both dimensions than the family $nPnB$:

- Both *precision* and *recall* of $nPnB^{0.50}$ are greater than these of $SGC_{\kappa=54}$;
- Both *precision* and *recall* of $nPnB^{0.70}$ are greater than these of $SGC_{\kappa=24}$.

Last: The 'ground-truth' clustering $\mathcal{C}_{Dep}$ has poor *precision*/*recall* scores, which calls into question its relevance as a 'ground-truth' reference.

Table 1: **Intrinsic and extrinsic scores of clustering methods:** Each row gives the precision, recall and $F_{\sigma=0.5}$ score for the graphs derived from the clustering of G_{em} against both the original graph G_{em} and the 'ground-truth' departments clustering. Best scores are highlighted in red.

	Intrinsic	**Extrinsic**
Scores $\times 100$	G_{em}	$\widehat{\mathcal{C}_{Dep}}$
Omniscient met_E	100,100,100	34,23,27
Oracle met_{Dep}	23,34,27	100,100,100
SGC_{54}	36,30,33	56,31,40
SGC_{24}	22,43,30	46,60,52
Louvain	10,63,17	18,**80**,30
Infomap	13,60,21	22,70,33
Starling	26,45,33	51,61,**56**
$nPnB^{0.50}$	41,35,**(38)**	**59**,34,43
$nPnB^{0.50}_{0.15}$	**(41**,**63)**,**49**	40,42,41

Table 1 compares, at scale of description $\sigma = 0.5$, methods intrinsically against the original graph G_{em}, and extrinsically against the derived graph $\widehat{\mathcal{C}_{Dep}}$ from the 'ground-truth' $\mathcal{C}_{Dep}$.

Intrinsically: The best result –both best *precision* and *recall*– is obtained with $nPnB^{0.5}_{0.15}$ and the best $F_{\sigma=0.5}$ score for partitional clustering is not obtained with met_{Dep} but with $nPnB^{0.5}$. This could be understood by the fact that $nPnB^{0.5}$ is optimizing $F_{0.5}$ (s the optimized scale by $nPnB^s$ is equal to the description scale σ used for evaluation).

Extrinsically: The omniscient method strikingly presents the worst $F_{\sigma=0.5}$ score, which again calls into question the relevance of $\mathcal{C}_{Dep}$ as a 'ground-truth' reference: research departments of a research institution are apparently not the right structures to explain patterns in e-mail exchanges among its employees. Defining a proper 'ground-truth' reference is a difficult task. Expert often disagree with each other even when their judgements are based on the same protocol [14]. Clearly defining the desired scale of description and measuring quality with the F_σ function can help to define 'ground-truth' in a more consensual way in the future.

Prior to this work it was impossible to compute Table 1 due to the lack of a framework for comparing partitional and overlapping clusterings to both the original graph and a 'ground-truth' clustering. This is a key result of the existence of a unified graph clustering framework.

4 Conclusion

We have proposed a general framework to compare different clustering algorithms that where previously incommensurable. This unified framework naturally includes the notion of *description scale* found in many clustering algorithms in the form of a resolution or granularity parameter. Evaluation in this framework is based on meaningful metrics, the *precision* and the *recall*, widely used in science and therefore easily understandable by most users of real-world graphs. This framework is effective in the sense that it provides inspiration for new clustering algorithms that both outperform existing ones in the *precision*/*recall* dimensions and make sense when applied on real-world graph. It also makes it possible to assess the relevance of 'ground-truth' references that are sometimes proposed when studying complex networks. Further development of this framework could take into account the directionality of certain networks, edge weights where appropriate, and their temporal dimension.

Contributions

BG initiated the research, wrote the first draft and carried out the digital implementation. BG, IA and DC further developed the writing of the article and the analyses. All authors contributed to the final version of the manuscript.

Aknowledgments

This work was supported by the Complex Systems Institute of Paris Île-de-France (ISC-PIF) and the EU NODES project (LC-01967516).

References

[1] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. Information Retrieval Journal, 12(4):461–486, 2009.

[2] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008(10):P10008, 2008.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. U. Hwang. Complex networks: Structure and dynamics. Physics Reports, 424(4):175–308, Feb. 2006.

[4] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly. Metrics for community analysis: A survey. ACM Comput. Surv., 50(4), aug 2017.

[5] F. R. K. Chung. Spectral Graph Theory. American Mathematical Society, 1997.

[6] S. Fortunato and M. Barthélemy. Resolution limit in community detection. Proceedings of the National Academy of Sciences, 104(1):36–41, Jan. 2007. Publisher: Proceedings of the National Academy of Sciences.

[7] S. Fortunato and M. E. J. Newman. 20 years of network community detection. Nature Physics, 18(8):848–850, Aug. 2022. Number: 8 Publisher: Nature Publishing Group.

[8] B. Gaume. Starling: Introducing a mesoscopic scale with confluence for graph clustering. PLOS ONE, 18(8):1–30, 08 2023.

[9] B. Gaume, F. Mathieu, and E. Navarro. Building real-world complex networks by wandering on random graphs. Revue I3, 10(1):73–91, 2010.

[10] P. D. Grünwald. The minimum description length principle. MIT press, 2007.

[11] D. E. Knuth. The Art of Computer Programming: Fundamental algorithms. The Art of Computer Programming. Addison -Wesley, 1968.

[12] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. Physical Review E, 78(4):046110+, Oct. 2008.

[13] U. Luxburg. A tutorial on spectral clustering. Statistics and Computing, 17(4):395–416, dec 2007.

[14] G. C. Murray and R. Green. Lexical Knowledge and Human Disagreement on a WSD Task. Computer Speech & Language, 18(3):209–222, 2004.

[15] E. Navarro. Métrologie des graphes de terrain, application à la construction de ressources lexicales et à la recherche d'information. November 2013.

[16] M. E. J. Newman. The Structure and Function of Complex Networks. SIAM Review, 45:167–256, 2003.

[17] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. Physical Review E, 69(2), Feb 2004.

[18] W. M. Rand. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association, 66(336):846–850, 1971.

[19] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4):1118–1123, 2008.

[20] S. H. Strogatz. Exploring complex networks. Nature, 410(6825):268–276, Mar. 2001. Number: 6825 Publisher: Nature Publishing Group.

[21] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. Theoretical Computer Science, 363(1):28–42, 2006. Computing and Combinatorics.

[22] D. J. Watts and S. H. Strogatz. Collective Dynamics of Small-World Networks. Nature, 393:440–442, 1998.

[23] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. 2012 IEEE 12th International Conference on Data Mining, pages 745–754, 2012.

[24] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich. Local higher-order graph clustering. Proceedings of Conference on Knowledge Discovery and Data Mining, page 555–564, 2017.

ANNEX

The Families nPnB: binary classifier of **n**ode **P**airs by **n**ode **B**locks

Similarity between edges ends for the agglomerative strategy of nPnB^{s_P}

The Algorithm 1 $nPnB^{s_p}$ describes our method to find a partitional clustering $\mathcal{C}$ such that the graph $\widehat{\mathcal{C}}$ best approximates the graph G relatively to $F_{s_p}(P(\widehat{\mathcal{C}}, G), R(\widehat{\mathcal{C}}, G))$, where s_p defines the desired scale of description. In algorithm 1, $Sim(G, x, y)$ is a similarity between nodes which serves as an agglomerative strategy on the edges $\{x, y\} \in E$ to merge two clusters C_1 such that $x \in C_1$ and C_2 such that $y \in C_2$ if it is not decreasing $F_{s_p}(P(\widehat{\mathcal{C}}, G), R(\widehat{\mathcal{C}}, G))$ (`Line`$_2$). We tested 84 state-of-the-art similarities [15]. The best results (in decreasing performances) were obtained with the three similarities Cm, $Confluence$ and $CosP$:

$$Cm(G = (V, E), x, y) = Max\Big(\{0\} \cup \{|C| \text{ such } x, y \in C \in \text{Cliques of } G\}\Big)$$

$$Confluence(G = (V, E), x, y) = \begin{cases} 0 \text{ if } x = y, \\ \frac{P^3_{G^{x,y}}(x \rightsquigarrow y) - \frac{d_G{}^{x,y}(y)}{2(|E|-1)}}{P^3_{G^{x,y}}(x \rightsquigarrow y) + \frac{d_G{}^{x,y}(y)}{2(|E|-1)}} \text{ otherwise.} \end{cases}$$

$$CosP(G = (V, E), x, y) =$$
$$Cosinus\Big(\overrightarrow{(P^2_G(x \rightsquigarrow x), P^2_G(x \rightsquigarrow y))}, \overrightarrow{(P^2_G(y \rightsquigarrow x), P^2_G(y \rightsquigarrow y))}\Big)$$

Where: $G^{x,y} = \Big(V, \Big\{e \in E \text{ such } e \neq \{x, y\}\Big\}\Big)$*;* $d_G(x) = |\{y \in V / \{x, y\} \in E\}|$*; and* $P^{\varpi}_G(x \rightsquigarrow y)$ *is the probability that a random walker wandering on the graph* G *through its edges, reaches the node* y *after* ϖ *steps starting from the node* x*.*

For roughly equivalent performances, the worst-case time complexity of $CosP$, $Confluence$ and Cm (respectively $O(2|E|)$, $O(3|E|)$ and $O(3^{\frac{|V|-2}{3}})$, [21]) favors $CosP$. In order to place ourselves in the worst-case scenario of dealing with large graphs, the results in this paper are based on the $CosP$ similarity

both for algorithm 1 (partitional clustering) and 2 (overlapping clustering). Note that different edges $\{x_1, y_1\} \in E$ and $\{x_2, y_2\} \in E$ might happen to have the exact same Sim value ($Sim(G, x_1, y_1) = Sim(G, x_2, y_2)$), making the process non-deterministic in general, because of its sensitivity on the order in which the edges with identical Sim values are processed (`(Line1)` and `(Line2)` respectively in algorithms 1 and 2). To avoid such non-deterministic process, we can sort edges by first comparing their Sim values and then using the lexicographic order on the words $x_1\, y_1$ and $x_2\, y_2$ (with $x_1 < y_1$ and $x_2 < y_2$) when Sim values are strictly identical.

nPnB$^{\mathbf{s_p}}_{\mathbf{s_o}}$ defining overlapping clustering

The Algorithm 2 $nPnB^{s_p}_{s_o}$ describes our method to find an overlapping clustering $\mathcal{C}$ such that $|\mathcal{C}| \leqslant |V|$ and the graph $\widehat{\mathcal{C}}$ best approximates the graph G, where s_p defines the desired scale of description and s_o defines the desired amount of overlap. It is based on gluantly extending the clusters of $\mathcal{C}^{s_p} = nPnB^{s_p}(G)$ through the optimization of $F_{s_o}(P(\widehat{\mathcal{C}^{s_p}}, G), R(\widehat{\mathcal{C}^{s_p}}, G))$.

Let $\mathcal{C}^{s_p} = nPnB^{s_p}(G)$ and $\mathcal{C}^{s_p}_{s_o} = nPnB^{s_p}_{s_o}(G)$. With algorithms 1 and 2, it is then clear that for any graph $G = (V, E)$:

- $\forall\ s_p, s_o \in [0, 1], \Xi(\mathcal{C}^{s_p}) \subseteq \Xi(\mathcal{C}^{s_p}_{s_o})$ (Because **Line$_1$** & **Line$_3$** in Algo. 2). This has the direct consequence:

$$\forall\ s_p, s_o \in [0, 1],\ R(\widehat{\mathcal{C}^{s_p}}, G) \leqslant R(\widehat{\mathcal{C}^{s_p}_{s_o}}, G)$$

- $\forall\ s_p, s_o \in [0, 1], |\mathcal{C}^{s_p}_{s_o}| \leqslant |\mathcal{C}^{s_p}| \leqslant |V|$ (Because $\mathcal{C}^{s_p}$ is a partition of V and **Line$_4$** & **Line$_5$** in Algo. 2).

Algorithm 1 $\mathcal{C}_p = nPnB^{s_p}(G)$: To find Partitional Clustering

Input:

$G = (V, E)$ ▶ An undirected graph

$s_p \in [0, 1]$ ▶ For optimizing $F_{s_p}(P(\widehat{\mathcal{C}_p}, G), R(\widehat{\mathcal{C}_p}, G))$ with Blocks without overlaps

Output:

$\mathcal{C}_p$ ▶ A Partitional Clustering of G

$X \leftarrow \left\{\{i, j\} \in E \text{ such } i \neq j\right\}$

for $i \in V$ **do** ▶ Initialization

  $mod_i \leftarrow \{i\}$ ▶ One node per cluster

  $M_i \leftarrow i$ ▶ node i is in cluster i

$\Upsilon \leftarrow \varnothing$; $TP \leftarrow 0$; $FP \leftarrow 0$; $FN \leftarrow |E|$; $Fscore \leftarrow 0$

While $\Upsilon \neq X$ **do**

  $\{i, j\} \leftarrow \underset{\{x,y\} \in X - \Upsilon}{\arg\max}\ Sim(G,\ x,\ y)$ ▶ Strategy based on Sim of edges**(Line$_1$)**

  $\Upsilon \leftarrow \Upsilon \cup \{\{i,\ j\}\}$

  if $M_i \neq M_j$ **then** ▶ mod_i and mod_j have not yet been merged together

    $newTP \leftarrow TP$; $newFP \leftarrow FP$; $newFN \leftarrow FN$

    for $u \in mod_i$ **do**

      for $v \in mod_j$ **do**

        if $\{u, v\} \in E$ **then**

          $newTP \leftarrow newTP + 1$

          $newFN \leftarrow newFN - 1$

        else

          $newFP \leftarrow newFP + 1$

    $newFscore \leftarrow \frac{(1+(f(s_p)^2).newTP}{(1+(f(s_p)^2).newTP+f(s_p)^2.newFN+newFP}$

    if $Fscore \leqslant newFscore$ **then** ▶ **(Line$_2$)**

      $mod_i \leftarrow mod_i \cup mod_j$ ▶ mod_j merge with mod_i in mod_i

      $mod_j \leftarrow \varnothing$ ▶ mod_j is removed

      for $k \in V$ **do** ▶ Updating the membership list

        if $M_k = j$ ▶ node k was in mod_j

          $M_k \leftarrow i$ ▶ node k is now in mod_i

      $TP \leftarrow newTP$; $FP \leftarrow newFP$; $FN \leftarrow newFN$

      $Fscore \leftarrow newFscore$

$\mathcal{C}_p \leftarrow \varnothing$

for $i \in V$ **do**

  if $mod_i \neq \varnothing$ ▶ mod_i is alive

    $\mathcal{C}_p \leftarrow \mathcal{C}_p \cup \left\{mod_i\right\}$

Return $\mathcal{C}_p$

Algorithm 2 $\mathcal{C}_o = nPnB^{s_p}_{s_o}(G = (V, E))$: To find Clustering allowing overlaps such $|\mathcal{C}_o| \leqslant |V|$

Input:

$G = (V, E)$ ▶ `An undirected graph`

$s_p \in [0, 1]$ ▶ `For optimizing` $F_{s_p}(P(\widehat{\mathcal{C}_p}, G), R(\widehat{\mathcal{C}_p}, G))$ `with Blocks without overlaps`

$s_o \in [0, 1]$ ▶ `For gluantly exend, in` C_o`, the clusters of` $\mathcal{C}_p$ `by optimizing` $F_{s_o}(P(\widehat{\mathcal{C}_p}, G),$

Output:

$\mathcal{C}_o$ ▶ `A Clustering of` G `with Blocks allowing overlaps`

$\mathcal{C}_p \leftarrowtail nPnB^{s_p}(G)$ ▶ `Partitional Clustering optimizing` $F_{s_p}(P(\widehat{\mathcal{C}_p}, G), R(\widehat{\mathcal{C}_p}, G))$

$X \leftarrowtail \left\{\{i, j\} \in E \text{ such } i \neq j\right\}$

for $modI_i \in \mathcal{C}_p$ **do** ▶ `Initialization`

  $modO_i \leftarrowtail modI_i$ ▶ $modO_i$ `of` $\mathcal{C}_o$ `is equal to` $modI_i$ `of` $\mathcal{C}_p$ **(Line$_1$)**

  for $j \in modI_i$ **do**

    $M_i \leftarrowtail j$ ▶ `node` j `is in cluster` i `of` $\mathcal{C}_p$

$\Upsilon \leftarrowtail \varnothing$; $TP \leftarrowtail |\Xi(\mathcal{C}_p) \bigcap E|$; $FP \leftarrowtail |\Xi(\mathcal{C}_p) \bigcap \overline{E}|$; $FN \leftarrowtail |\overline{\Xi(\mathcal{C})} \bigcap E|$

$Fscore \leftarrowtail \frac{(1+f(s_o)^2).TP}{(1+f(s_o)^2).TP+f(s_o)^2.FN+FP}$

While $\Upsilon \neq X$ **do**

  $\{i, j\} \leftarrowtail \underset{\{x,y\} \in X-\Upsilon}{\arg\max}\ Sim(G,\ x,\ y)$ ▶ `Strategy based on` Sim `of edges` **(Line$_2$)**

  $\Upsilon \leftarrowtail \Upsilon \cup \{\{i,\ j\}\}$

  if $M_i \neq M_j$ **then** ▶ i `and` j `are not in a same cluster of` $\mathcal{C}_p$

    for $(x_1, x_2) \in \{(i, j), (j, i)\}$**do**

      if $x_1 \notin modO_{x_2}$ **then** ▶ x_1 `is not already added to` $modO_{x_2}$ `of` $\mathcal{C}_o$

        $newTP \leftarrowtail TP$; $newFP \leftarrowtail FP$; $newFN \leftarrowtail FN$

        for $u \in modO_{x_2}$ **do**

          if $\{x_1, u\} \in E$ **then**

            $newTP \leftarrowtail newTP + 1$

            $newFN \leftarrowtail newFN - 1$

          else

            $newFP \leftarrowtail newFP + 1$

        $newFscore \leftarrowtail \frac{(1+f(s_o)^2).newTP}{(1+f(s_o)^2).newTP+f(s_o)^2.newFN+newFP}$

        if $Fscore \leqslant newFscore$ **then**

          $modO_{x_2} \leftarrowtail modO_{x_2} \bigcup \{x_1\}$ ▶ $\Rightarrow modI_{x_2} \subsetneq modO_{x_2}$ **(Line$_3$)**

          $TP \leftarrowtail newTP$; $FP \leftarrowtail newFP$; $FN \leftarrowtail newFN$

          $Fscore \leftarrowtail newFscore$

$\mathcal{C}_o \leftarrowtail \varnothing$

for $modI_i \in \mathcal{C}_p$ **do**

  if $(\nexists j$ such $modO_i \subsetneq modO_j)$ **then** ▶ **(Line$_4$)**

    $\mathcal{C}_o \leftarrowtail \mathcal{C}_o \bigcup \{modO_i\}$ ▶ **(Line$_5$)**

Return $\mathcal{C}_o$