



HAL
open science

A Unified Graph Clustering Framework for Complex Systems Modeling

Bruno Gaume, Ixandra Achitouv, David Chavalarias

► **To cite this version:**

Bruno Gaume, Ixandra Achitouv, David Chavalarias. A Unified Graph Clustering Framework for Complex Systems Modeling. 2024. hal-04505654v2

HAL Id: hal-04505654

<https://hal.science/hal-04505654v2>

Preprint submitted on 12 Apr 2024 (v2), last revised 18 Apr 2024 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Unified Graph Clustering Framework for Complex Systems Modeling

Bruno Gaume^{1,3}, Ixandra Achitouv^{3,*}, and David Chavalarias^{2,3,*}

This manuscript was compiled on April 4, 2024

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.

community detection | complex networks | multi-scale

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 (1–4), 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 inability to adapt to different scales of observation (5), while others include explicitly a resolution parameter to address second question (6, 7). 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.

Significance Statement

Complex networks are one of the main conceptual tools for modeling large decentralized systems, whether natural or artificial. Clustering algorithms are used to identify sub-parts of these systems, from the study of which we can infer or deduce particular characteristics or functions of these systems. However, there is no unified framework for comparing clustering methods two by two, which makes it difficult to choose a clustering algorithm given a system under study and slows down research by preventing a constructive debate on the comparative advantages of different proposed clustering methods. This article proposes such a unified framework. Its effectiveness is demonstrated by the effective comparison of so far incommensurable state-of-the-art methods and its ability to inspire new clustering algorithms that outperform them. Applications are provided on real-world networks.

Author affiliations: ¹Cognition, Langues, Langage, Ergonomie (CLLE, UMR 5263), CNRS, France; ²Centre d’Analyse et de Mathématique Sociales (CAMS, UMR8557), CNRS, France; ³Institut des Systemes Complexes de Paris IdF (ISC-PIF, UAR3611), CNRS, France

Contributions: Bruno Gaume initiated the research, wrote the first draft and carried out the digital implementation. Bruno Gaume, Ixandra Achitouv and David Chavalarias further developed the writing of the article and the analyses. All authors contributed to the final version of the manuscript.

We have no one competing interest here.

¹Correspondence should be addressed. E-mail: bruno.gaume@iscpif.org

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 (8). 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 (9–11), which fall into two categories:

Intrinsic metrics aiming at finding clustering on a graph G according to some general principles (like the modularity of (12, 13) or the compressibility of (14));

Extrinsic metrics aiming at evaluating clustering (e.g. the Rand index (15)) in relation to *a priori* known structures or 'ground-truth' clustering, such as clusters of synthetic networks (16).

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 (9–11)), no framework so far has been proposed for the simultaneous evaluation of *intrinsic* and *extrinsic* metrics for *partition* and *overlapping* clustering.

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 $\widehat{C} = (U(\mathcal{C}), \Xi(\mathcal{C}))$ the derived graph from the clustering \mathcal{C} using the following two functions:

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

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

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{C} to approximate G :

Precision: $P(\widehat{C}, G) = \frac{|\Xi(\mathcal{C}) \cap E|}{|\Xi(\mathcal{C})|}$.

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: $R(\widehat{C}, G) = \frac{|\Xi(\mathcal{C}) \cap E|}{|E|}$.

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:

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

For *overlapping clustering*, the set of the edges E and the set of maximal cliques \mathcal{C}_{mc} on a graph G are such that $\widehat{E} = \widehat{\mathcal{C}_{mc}} = G$. 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:

$$F_s(P, R) = \frac{(1 + f(s)^2) \cdot (P \cdot R)}{R + f(s)^2 \cdot P} \quad [3]$$

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

For $s = 0.5$ *precision* and *recall* are of the same importance (when you compress the graph, losing 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 $Pos(G_{toy})$. 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.

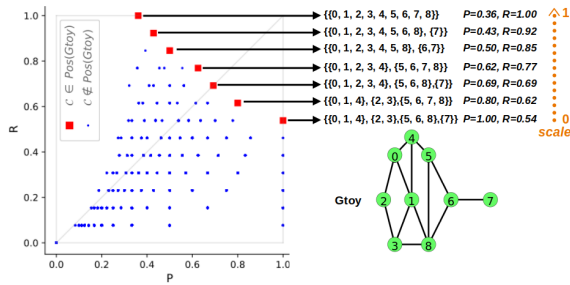


Fig. 1. The set of all the 21 147 partitional clustering of G_{toy} in the precision (P) / recall (R) space. The Pareto front is highlighted in red.

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 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.

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 vertex is assigned to its own cluster is a partitional clustering. Then $\forall s \in [0, 1]$ its F_s score $F_s(P(\hat{\mathcal{C}}, G), R(\hat{\mathcal{C}}, G)) = 0$ since its recall $R(\hat{\mathcal{C}}, G) = 0$.

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 $F_{s_p}(P(\hat{\mathcal{C}}, G), R(\hat{\mathcal{C}}, G))$ (cf. Sect. A 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 $\hat{\mathcal{C}}$ 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 (18). One of the best scalable metrics was $CosP$ which has been chosen in the subsequent application (cf. Sect. A) :

$$CosP(G = (V, E), x, y) \parallel \overrightarrow{Cosinus \left((P_G^2(x \rightsquigarrow x), P_G^2(x \rightsquigarrow y)), (P_G^2(y \rightsquigarrow x), P_G^2(y \rightsquigarrow y)) \right)}$$

This approach can be generalized to define families of overlapping clustering $nPnB_{s_o}^{s_p}$ (cf. Sect. B 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 (19). On contrary, *precision* has only homogeneity property –it is the archetypal metric of homogeneity– and *recall* has only completeness property –it is the archetypal metric of completeness.

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 (20), Infomap (14) , Starling (21), and Spectral Graph Clustering (7) – for which we consider several resolutions– applied on a real-world network $G_{em} = (V_{em}, E_{em})$ (22).

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

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

386 For $nPnB$ clustering, we consider two families: the
 387 partitional clustering family $nPnB^{s_p}$, returning partitional
 388 clustering based on the optimization of F_{s_p} and the overlapping
 389 clustering $nPnB_{s_o}^{s_p}$, returning overlapping clustering based on
 390 gradually extending the clusters produced by $nPnB^{s_p}$ through
 391 the optimization of F_{s_o} .

392 Spectral Graph Clustering (SGC, (7)) methods require to
 393 specify the number κ of clusters. Our comparison includes the
 394 SGC partitional clustering for $\kappa = 24$ (SGC_{24}) and $\kappa = 54$
 395 (SGC_{54}). We select these two values because at scale $\sigma = 0.5$,
 396 which is a natural entry point to compare clustering given the
 397 two properties of *homogeneity* and *completeness* of $F_{\sigma=0.5}$, (i)
 398 SGC_{24} is the one with the best *extrinsic* score relatively to
 399 \mathcal{C}_{Dep} and (ii) SGC_{54} is the one with the best *intrinsic* score ;
 400 i.e. $\forall \kappa \in \mathbb{N}$, $0 < \kappa \leq |V_{em}|$:

402 (i) $F_{0.5}(R_{\kappa}^{Dep}, P_{\kappa}^{Dep}) < F_{0.5}(R_{24}^{Dep}, P_{24}^{Dep})$ with $R_{\kappa}^{Dep} =$
 403 $R(\widehat{SGC_{\kappa}}, \widehat{\mathcal{C}_{Dep}})$ and
 404 $P_{\kappa}^{Dep} = P(\widehat{SGC_{\kappa}}, \widehat{\mathcal{C}_{Dep}})$;

405 (ii) $F_{0.5}(R_{\kappa}^{em}, P_{\kappa}^{em}) < F_{0.5}(R_{54}^{em}, P_{54}^{em})$ with $R_{\kappa}^{em} =$
 406 $R(\widehat{SGC_{\kappa}}, G_{em})$ and
 407 $P_{\kappa}^{em} = P(\widehat{SGC_{\kappa}}, G_{em})$;

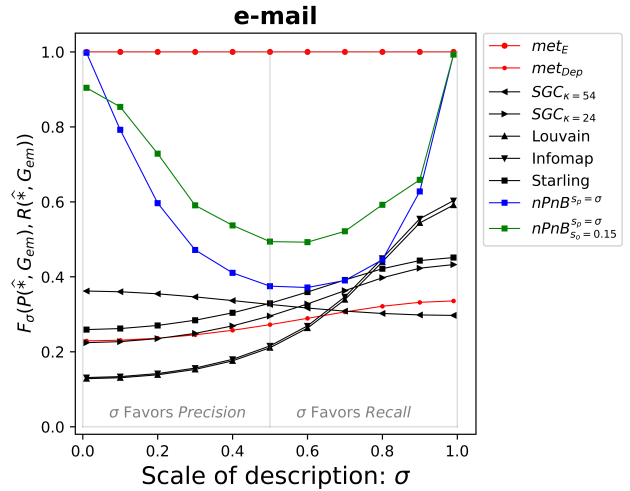
408 Fig. 2 displays the performances of each clustering
 409 method according to the scale of description $\sigma \in [0, 1]$.
 410 A key result is that the family of overlapping clusterings
 411 $\{nPnB_{s_o=0.15}^{s_p=\sigma}\}_{\sigma \in [0, 1]}$ outperforms all other methods when
 412 their scale parameter s_p is set to coincide with the desired
 413 scale of description σ . This result holds if we restrict ourselves
 414 to partitional clustering. For a given scale of description
 415 σ , $nPnB^{s_p=\sigma}$ outperforms the other partitional clustering
 416 methods tested. Removing those two families of clustering
 417 methods from the comparison, none of the methods tested
 418 outperforms the others for all scales of description σ .

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

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

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

- 427 • Both *precision* and *recall* of $nPnB^{0.50}$ are greater than
 428 these of $SGC_{\kappa=54}$;

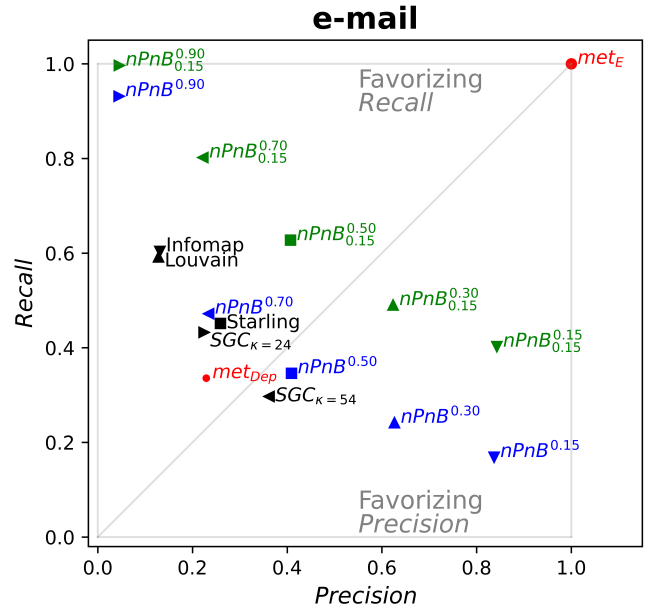


435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455

Fig. 2. Performance $F_{\sigma}(P(\widehat{*}, G_{em}), R(\widehat{*}, G_{em}))$ of clustering methods as derived
 graphs $\widehat{*} = met(\widehat{G_{em}})$ according to the description scale σ .

- 456 • Both *precision* and *recall* of $nPnB^{0.70}$ are greater than
 457 these of $SGC_{\kappa=24}$.

458 **Last:** The ‘ground-truth’ clustering \mathcal{C}_{Dep} has poor *precisi-*
 459 *on/recall* scores, which calls into question its relevance as a
 460 ‘ground-truth’ reference.



461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485

Fig. 3. Comparison in the precision-recall plane of the performances of different
 clustering methods when applied to the e-mail graph G_{em} . The Oracle methods
 met_{Dep} and the Omniscient methods met_E are highlighted in red.

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

489 **Intrinsically:** The best result –both best *precision* and
 490 *recall*– is obtained with $nPnB_{s_o=0.15}^{0.5}$ and the best $F_{\sigma=0.5}$ score
 491 for partitional clustering is not obtained with met_{Dep} but with
 492
493
494
495
496

497 $nPnB^{0.5}$. This could be understood by the fact that $nPnB^{0.5}$
 498 is optimizing $F_{0.5}$ (s the optimized scale by $nPnB^s$ is equal
 499 to the description scale σ used for evaluation).

500 **Extrinsically:** The omniscient method strikingly presents
 501 the worst $F_{\sigma=0.5}$ score, which again calls into question the
 502 relevance of \mathcal{C}_{Dep} as a ‘ground-truth’ reference: research
 503 departments of a research institution are apparently not the
 504 right structures to explain patterns in e-mail exchanges among
 505 its employees. Defining a proper ‘ground-truth’ reference is
 506 a difficult task. Expert often disagree with each other even
 507 when their judgements are based on the same protocol (23).
 508 Clearly defining the desired scale of description and measuring
 509 quality with the F_{σ} function can help to define ‘ground-truth’
 510 in a more consensual way in the future.

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

	Intrinsic	Extrinsic
Scores $\times 100$	G_{em}	$\widehat{\mathcal{C}}_{Dep}$
<i>Omniscient met_E</i>	100,100,100	34,23,27
<i>Oracle met_{Dep}</i>	23,34,27	100,100,100
<i>SGC₅₄</i>	36,30,33	56,31,40
<i>SGC₂₄</i>	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

528
 529 Prior to this work, it was impossible to compute Table 1
 530 due to the lack of a framework for comparing partitional
 531 and overlapping clusterings to both the original graph and a
 532 ‘ground-truth’ clustering. This is a key result of the existence
 533 of a unified graph clustering framework.

535 Conclusion

536
 537 We have proposed a general framework to compare different
 538 clustering algorithms that were previously incommensurable.
 539 This unified framework naturally includes the notion of
 540 *description scale* found in many clustering algorithms in the
 541 form of a resolution or granularity parameter. Evaluation in
 542 this framework is based on meaningful metrics, the *precision*
 543 and the *recall*, widely used in science and therefore easily
 544 understandable by most users of real-world graphs. This
 545 framework is effective in the sense that it provides inspiration
 546 for new clustering algorithms that both outperform existing
 547 ones in the *precision/recall* dimensions and make sense when
 548 applied on real-world graph. It also makes it possible to
 549 assess the relevance of ‘ground-truth’ references that are
 550 sometimes proposed when studying complex networks. Further
 551 development of this framework could take into account
 552 the directionality of certain networks, edge weights where
 553 appropriate, and their temporal dimension.

554
 555 **Acknowledgments.** This work was supported by the Complex
 556 Systems Institute of Paris Île-de-France (ISC-PIF) and the
 557 EU NODES project (LC-01967516).
 558

559 1. DJ Watts, SH Strogatz, Collective Dynamics of Small-World Networks. *Nature* **393**, 440–442 (1998). 560
 561 2. B Gaume, F Mathieu, E Navarro, Building real-world complex networks by wandering on random graphs. *Revue I3* **10**, 73–91 (2010). 562
 563 3. S Boccaletti, V Latora, Y Moreno, M Chavez, DU Hwang, Complex networks: Structure and dynamics. *Phys. Reports* **424**, 175–308 (2006). 564
 565 4. SH Strogatz, Exploring complex networks. *Nature* **410**, 268–276 (2001) Number: 6825 Publisher: Nature Publishing Group. 566
 567 5. S Fortunato, M Barthélemy, Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**, 36–41 (2007) Publisher: Proceedings of the National Academy of Sciences. 568
 569 6. FRK Chung, *Spectral Graph Theory*. (American Mathematical Society), (1997). 570
 571 7. U Luxburg, A tutorial on spectral clustering. *Stat. Comput.* **17**, 395–416 (2007). 572
 573 8. DE Knuth, *The Art of Computer Programming: Fundamental algorithms*, The Art of Computer Programming. (Addison-Wesley), (1968). 574
 575 9. J Yang, J Leskovec, Defining and evaluating network communities based on ground-truth. *2012 IEEE 12th Int. Conf. on Data Min.*, pp. 745–754 (2012). 576
 577 10. T Chakraborty, A Dalmia, A Mukherjee, N Ganguly, Metrics for community analysis: A survey. *ACM Comput. Surv.* **50** (2017). 578
 579 11. S Fortunato, MEJ Newman, 20 years of network community detection. *Nat. Phys.* **18**, 848–850 (2022) Number: 8 Publisher: Nature Publishing Group. 580
 581 12. MEJ Newman, The Structure and Function of Complex Networks. *SIAM Rev.* **45**, 167–256 (2003). 582
 583 13. MEJ Newman, M Girvan, Finding and evaluating community structure in networks. *Phys. Rev. E* **69** (2004). 584
 585 14. M Rosvall, CT Bergstrom, Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**, 1118–1123 (2008). 586
 587 15. WM Rand, Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**, 846–850 (1971). 588
 589 16. A Lancichinetti, S Fortunato, F Radicchi, Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**, 046110+ (2008). 590
 591 17. PD Grünwald, *The minimum description length principle*. (MIT press), (2007). 592
 593 18. E Navarro, Métrologie des graphes de terrain, application à la construction de ressources lexicales et à la recherche d’information. (2013). 594
 595 19. E Amigó, J Gonzalo, J Artiles, F Verdejo, A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Inf. Retr. J.* **12**, 461–486 (2009). 596
 597 20. VD Blondel, JL Guillaume, R Lambiotte, E Lefebvre, Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008). 598
 599 21. B Gaume, Starling: Introducing a mesoscopic scale with confluence for graph clustering. *PLOS ONE* **18**, 1–30 (2023). 600
 601 22. H Yin, AR Benson, J Leskovec, DF Gleich, Local higher-order graph clustering. *Proc. Conf. on Knowl. Discov. Data Min.*, p. 555–564 (2017). 602
 603 23. GC Murray, R Green, Lexical Knowledge and Human Disagreement on a WSD Task. *Comput. Speech & Lang.* **18**, 209–222 (2004). 604
 605 24. E Tomita, A Tanaka, H Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments. *Theor. Comput. Sci.* **363**, 28–42 (2006) Computing and Combinatorics. 606
 607
 608
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 620

621	683
622	684
623	685
624	686
625	687
626	688
627	689
628	690
629	691
630	692
631	693
632	694
633	695
634	696
635	697
636	698
637	699
638	700
639	701
640	702
641	703
642	704
643	705
644	706
645	707
646	708
647	709
648	710
649	711
650	712
651	713
652	714
653	715
654	716
655	717
656	718
657	719
658	720
659	721
660	722
661	723
662	724
663	725
664	726
665	727
666	728
667	729
668	730
669	731
670	732
671	733
672	734
673	735
674	736
675	737
676	738
677	739
678	740
679	741
680	742
681	743
682	744

A. The Families nPnB: binary classifier of node Pairs by node Blocks

A. 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 $F_{s_p}(P(\widehat{\mathcal{C}}, G), R(\widehat{\mathcal{C}}, G))$ (Line2). We tested 84 state-of-the-art similarities (18). The best results (in decreasing performances) were obtained with the three similarities Cm , $Confluence$ and $CosP$:

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

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

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

Where: $G^{x,y} = (V, \{e \in E \text{ such } e \neq \{x, y\}\})$; $d_G(x) = |\{y \in V / \{x, y\} \in E\}|$; and $P_G^\varpi(x \rightsquigarrow y)$ 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 $O(3^{\frac{|V|-2}{3}})$, (24)) 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.

B. $nPnB_{s_o}^{s_p}$ defining overlapping clustering. The Algorithm 2 $nPnB_{s_o}^{s_p}$ describes our method to find an overlapping clustering \mathcal{C} such that $|\mathcal{C}| \leq |V|$ 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 $\mathcal{C}_{s_o}^{s_p} = nPnB_{s_o}^{s_p}(G)$. With algorithms 1 and 2, it is then clear that for any graph $G = (V, E)$:

- $\forall s_p, s_o \in [0, 1], \Xi(\mathcal{C}^{s_p}) \subseteq \Xi(\mathcal{C}_{s_o}^{s_p})$ (Because Line1 & Line3 in Algo. 2). This has the direct consequence:

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

- $\forall s_p, s_o \in [0, 1], |\mathcal{C}_{s_o}^{s_p}| \leq |\mathcal{C}^{s_p}| \leq |V|$ (Because \mathcal{C}^{s_p} is a partition of V and Line4 & Line5 in Algo. 2).

869		931
870		932
871		933
872		934
873		935
874		936
875		937
876		938
877		939
878	Algorithm 1 $\mathcal{C}_p = nPnB^{s_p}(G)$: To find Partitional Clustering	940
879	Input:	941
880	$G = (V, E)$ ▶ An undirected graph	942
881	$s_p \in [0, 1]$ ▶ For optimizing $F_{s_p}(P(\widehat{\mathcal{C}}_p, G), R(\widehat{\mathcal{C}}_p, G))$ with Blocks without overlaps	943
882	Output:	944
883	\mathcal{C}_p ▶ A Partitional Clustering of G	945
884		946
885	1: $X \leftarrow \{\{i, j\} \in E \text{ such } i \neq j\}$	947
886	2: for $i \in V$ do ▶ Initialization	948
887	3: $mod_i \leftarrow \{i\}$ ▶ One node per cluster	949
888	4: $M_i \leftarrow i$ ▶ node i is in cluster i	950
889		951
890	5: $\Upsilon \leftarrow \emptyset; TP \leftarrow 0; FP \leftarrow 0; FN \leftarrow E ; Fscore \leftarrow 0$	952
891		953
892	6: While $\Upsilon \neq X$ do	954
893	7: $\{i, j\} \leftarrow \arg \max_{\{x, y\} \in X - \Upsilon} Sim(G, x, y)$ ▶ Strategy based on Sim of edges(Line1)	955
894		956
895	8: $\Upsilon \leftarrow \Upsilon \cup \{\{i, j\}\}$	957
896	9: if $M_i \neq M_j$ then ▶ mod_i and mod_j have not yet been merged together	958
897	10: $newTP \leftarrow TP; newFP \leftarrow FP; newFN \leftarrow FN$	959
898	11: for $u \in mod_i$ do	960
899	12: for $v \in mod_j$ do	961
900	13: if $\{u, v\} \in E$ then	962
901	14: $newTP \leftarrow newTP + 1$	963
902	15: $newFN \leftarrow newFN - 1$	964
903	16: else	965
904	17: $newFP \leftarrow newFP + 1$	966
905	18: $newFscore \leftarrow \frac{(1+(f(s_p)^2).newTP)}{(1+(f(s_p)^2).newTP+f(s_p)^2.newFN+newFP)}$	967
906	19: if $Fscore \leq newFscore$ then ▶ (Line2)	968
907	20: $mod_i \leftarrow mod_i \cup mod_j$ ▶ mod_j merge with mod_i in mod_i	969
908	21: $mod_j \leftarrow \emptyset$ ▶ mod_j is removed	970
909		971
910	22: for $k \in V$ do ▶ Updating the membership list	972
911	23: if $M_k = j$ ▶ node k was in mod_j	973
912	24: $M_k \leftarrow i$ ▶ node k is now in mod_i	974
913	25: $TP \leftarrow newTP; FP \leftarrow newFP; FN \leftarrow newFN$	975
914	26: $Fscore \leftarrow newFscore$	976
915		977
916	27: $\mathcal{C}_p \leftarrow \emptyset$	978
917	28: for $i \in V$ do	979
918	29: if $mod_i \neq \emptyset$ ▶ mod_i is alive	980
919	30: $\mathcal{C}_p \leftarrow \mathcal{C}_p \cup \{mod_i\}$	981
920		982
921	31: Return \mathcal{C}_p	983
922		984
923		985
924		986
925		987
926		988
927		989
928		990
929		991
930		992

993		1055
994		1056
995		1057
996		1058
997		1059
998		1060
999		1061
1000		1062
1001		1063
1002		1064
1003	Algorithm 2 $\mathcal{C}_o = nPnB_{s_o}^{s_p}(G = (V, E))$: To find Clustering allowing overlaps such $ \mathcal{C}_o \leq V $	1065
1004	Input:	1066
1005	$G = (V, E)$ ▶ An undirected graph	1067
1006	$s_p \in [0, 1]$ ▶ For optimizing $F_{s_p}(P(\widehat{\mathcal{C}}_p, G), R(\widehat{\mathcal{C}}_p, G))$ with Blocks without overlaps	1068
1007	$s_o \in [0, 1]$ ▶ For gluantly exend, in \mathcal{C}_o , the clusters of \mathcal{C}_p by optimizing $F_{s_o}(P(\widehat{\mathcal{C}}_p, G), R(\widehat{\mathcal{C}}_p, G))$	1069
1008	Output:	1070
1009	\mathcal{C}_o ▶ A Clustering of G with Blocks allowing overlaps	1071
1010		1072
1011	1: $\mathcal{C}_p \leftarrow nPnB^{s_p}(G)$ ▶ Partitional Clustering optimizing $F_{s_p}(P(\widehat{\mathcal{C}}_p, G), R(\widehat{\mathcal{C}}_p, G))$	1073
1012	2: $X \leftarrow \{\{i, j\} \in E \text{ such } i \neq j\}$	1074
1013	3: for $modI_i \in \mathcal{C}_p$ do ▶ Initialization	1075
1014	4: $modO_i \leftarrow modI_i$ ▶ $modO_i$ of \mathcal{C}_o is equal to $modI_i$ of \mathcal{C}_p (Line ₁)	1076
1015	5: for $j \in modI_i$ do	1077
1016	6: $M_i \leftarrow j$ ▶ node j is in cluster i of \mathcal{C}_p	1078
1017	7: $\Upsilon \leftarrow \emptyset$; $TP \leftarrow \Xi(\mathcal{C}_p) \cap E $; $FP \leftarrow \Xi(\mathcal{C}_p) \cap \overline{E} $; $FN \leftarrow \Xi(\mathcal{C}) \cap E $	1079
1018	8: $Fscore \leftarrow \frac{(1+f(s_o)^2).TP}{(1+f(s_o)^2).TP+f(s_o)^2.FN+FP}$	1080
1019	9: While $\Upsilon \neq X$ do	1081
1020	10: $\{i, j\} \leftarrow \arg \max_{\{x, y\} \in X - \Upsilon} Sim(G, x, y)$ ▶ Strategy based on Sim of edges (Line ₂)	1082
1021	11: $\Upsilon \leftarrow \Upsilon \cup \{\{i, j\}\}$	1083
1022	12: if $M_i \neq M_j$ then ▶ i and j are not in a same cluster of \mathcal{C}_p	1084
1023	13: for $(x_1, x_2) \in \{(i, j), (j, i)\}$ do	1085
1024	14: if $x_1 \notin modO_{x_2}$ then ▶ x_1 is not already added to $modO_{x_2}$ of \mathcal{C}_o	1086
1025	15: $newTP \leftarrow TP$; $newFP \leftarrow FP$; $newFN \leftarrow FN$	1087
1026	16: for $u \in modO_{x_2}$ do	1088
1027	17: if $\{x_1, u\} \in E$ then	1089
1028	18: $newTP \leftarrow newTP + 1$	1090
1029	19: $newFN \leftarrow newFN - 1$	1091
1030	20: else	1092
1031	21: $newFP \leftarrow newFP + 1$	1093
1032		1094
1033		1095
1034	22: $newFscore \leftarrow \frac{(1+f(s_o)^2).newTP}{(1+f(s_o)^2).newTP+f(s_o)^2.newFN+newFP}$	1096
1035		1097
1036	23: if $Fscore \leq newFscore$ then	1098
1037	24: $modO_{x_2} \leftarrow modO_{x_2} \cup \{x_1\}$ ▶ $\Rightarrow modI_{x_2} \subsetneq modO_{x_2}$ (Line ₃)	1099
1038	25: $TP \leftarrow newTP$; $FP \leftarrow newFP$; $FN \leftarrow newFN$	1100
1039	26: $Fscore \leftarrow newFscore$	1101
1040	27: $\mathcal{C}_o \leftarrow \emptyset$	1102
1041	28: for $modI_i \in \mathcal{C}_p$ do	1103
1042	29: if ($\nexists j$ such $modO_i \subsetneq modO_j$) then ▶ (Line ₄)	1104
1043	30: $\mathcal{C}_o \leftarrow \mathcal{C}_o \cup \{modO_i\}$ ▶ (Line ₅)	1105
1044	31: Return \mathcal{C}_o	1106
1045		1107
1046		1108
1047		1109
1048		1110
1049		1111
1050		1112
1051		1113
1052		1114
1053		1115
1054		1116