REGULAR PAPER

Exact and sampling methods for mining higher-order motifs in large hypergraphs

Quintino Francesco Lotito[1] · Federico Musciotto[2] · Federico Battiston[3] ·
Alberto Montresor[1]

Abstract

Network motifs are recurrent, small-scale patterns of interactions observed frequently in a system. They shed light on the interplay between the topology and the dynamics of complex networks across various domains. In this work, we focus on the problem of counting occurrences of small sub-hypergraph patterns in very large hypergraphs, where higher-order interactions connect arbitrary numbers of system units. We show how directly exploiting higher-order structures speeds up the counting process compared to traditional data mining techniques for exact motif discovery. Moreover, with hyperedge sampling, performance is further improved at the cost of small errors in the estimation of motif frequency. We evaluate our method on several real-world datasets describing face-to-face interactions, co-authorship and human communication. We show that our approximated algorithm allows us to extract higher-order motifs faster and on a larger scale, beyond the computational limits of an exact approach.

✉ Quintino Francesco Lotito
   quintino.lotito@unitn.it

   Federico Musciotto
   federico.musciotto@unipa.it

   Federico Battiston
   battistonf@ceu.edu

   Alberto Montresor
   alberto.montresor@unitn.it

1 Department of Information Engineering and Computer Science, University of Trento, via Sommarive 9, 38123 Trento, Italy

2 Dipartimento di Fisica e Chimica Emilio Segrè, Università di Palermo, Viale delle Scienze, 90128 Palermo, Italy

3 Department of Network and Data Science, Central European University, Quellenstraße 51, 1100 Vienna, Austria

1 Introduction

Network motifs are recurring patterns of interactions among a small set of nodes that appear in an observed network at a significant frequency. Motif analysis has established itself as an important tool for investigating networks at their microscale, highlighting the interdependence between the topology and dynamics of real-world networked systems [1, 2]. In fact, interacting systems with close functionalities tend to display similar over- and under-represented patterns of interactions [3].

Network motifs have found a vast set of applications in a number of different domains, such as biology [4–6], neuroscience [7], medicine [8], social network analysis [9], finance [10] and ecology [11, 12].

Given their multiple real-world applications, it is not surprising that the notion of network motifs has been extended to a variety of richer and more flexible network models, including weighted [13], temporal [14, 15] and multilayer [16, 17] networks. Recently, growing interest has been devoted to modelling real-world systems with group interactions [18, 19], from co-authorship [20] to face-to-face [21] interactions, by exploiting more complex mathematical tools such as hypergraphs [22], where hyperedges encode relationship among an arbitrary number of units. In hypergraphs, it is possible to identify higher-order motifs, i.e., connected sets of nodes interacting not only through pairwise edges but also through hyperedges encoding association among three or more nodes. In particular, higher-order motifs can be defined in terms of the overlapping patterns of hyperedges of a fixed size [23], or, more traditionally, by investigating all possible patterns of connected sub-hypergraphs for a given number of nodes [24].

Independently on the underlying network model, all the algorithms for motif analysis involve the following steps: (i) counting the occurrences of each motif in an observed network, (ii) counting the occurrences of each motif in suitable randomizations of the network and (iii) evaluating the over- and under-expression of each motif. The problem of counting the frequency of each motif in a target large network is inherently computationally challenging since it is equivalent to the problem of subgraph isomorphism, a well-known NP-complete problem. Moreover, for the statistical evaluation of motif frequencies, the counting step is repeated on each sample of the randomized graph model. Thus, exact algorithms for motif analysis tend to scale very poorly with graph and pattern sizes. A common algorithm design pattern to speed up the computation, albeit sacrificing the quality of the solutions, is relying on approximated algorithms. In particular, sampling methods are a popular choice for the task of motif discovery.

The increasing availability of large-scale real-world datasets with group interactions calls for the development of more efficient and usable algorithms for higher-order motif discovery. In this work, we build on our preliminary results regarding counting all the possible patterns of higher-order interactions involving a given number of nodes [24], and propose:

- an efficient exact algorithm for performing higher-order motif analysis with motifs involving 3 and 4 nodes, including efficiently solving the hypergraph isomorphism problem for small instances and constructing vertex-induced sub-hypergraphs;
- an approximated method based on hyperedge sampling that overcomes the scalability issues of the exact algorithm at the expense of only a very limited decrease of accuracy;
- the application of the approximated method to the study of higher-order motifs of order 5 (which are generally intractable for exact algorithms) in several networks of interest.

This paper is organized as follows. In Sect. 2 we survey related work. In Sect. 3 we introduce basic definitions and formalize the problem of interest. In Sect. 4 we outline our proposals for solving exactly the problem of higher-order motif discovery in hypergraphs. In Sect. 5 we propose a sampling algorithm for the same problem. In Sect. 6 we evaluate our proposals. In Sect. 7 we conclude the paper.

2 Related work

Network motifs have been extensively investigated:

- in the field of network science, for their relevance in the study of the local structure and the interplay between the topology and the dynamics of complex networks;
- in the field of data mining, due to the complexity of the problem of enumerating connected subgraphs up to a certain size from large graphs.

In this section, we propose a brief survey of the relevant prior work in both areas and highlight our contributions to both fields.

2.1 Network science

Network motifs describe complex networks by their preferential patterns of interactions at the microscale. They can be interpreted as fundamental circuits that have a role in the functionality of a system [1, 2], and are therefore able to discriminate networks that represent systems from different domains or with different functionalities [3]. The notion of network motifs has been extended to a variety of generalized network models, to encode and quantify a pattern of interactions with a richer set of features. Onnela et al. generalize network motifs to weighted networks, characterizing subgraphs in terms of their intensity and coherence [13]. In temporal networks, the topological and temporal microscale is described in terms of patterns of interactions inside restricted time windows [14, 15]. Battiston et al. extend motif analysis to multilayer networks by considering over-expressed subgraphs spanning across several layers [16]. More recently, motif analysis has been extended also to consider

patterns of higher-order interactions, i.e., interactions involving an arbitrary number of nodes. In particular, Lee et al. characterize real-world hypergraphs at their microscale in terms of patterns of connected hyperedges of fixed size [23]. However, they do not follow the traditional approach to network motifs first proposed by Milo et al. [1].

2.2 Data mining

Being the task of motif discovery of practical utility but computationally expensive (equivalent to the subgraph isomorphism problem, which is known to be NP-complete), a vast amount of literature has been developed around this problem. In particular, the development of faster algorithms for motif analysis has been historically motivated mainly by the ever-growing size of biological datasets to be analyzed. The algorithms for motif discovery can be clustered in two big groups: *exact counting* and *sampling methods*. Each type of algorithm comes with advantages and drawbacks: exact algorithms allow extracting the correct number of occurrences of each motif; however, they are usually slow and memory-intensive. Sampling methods are generally faster, but they need to be designed to avoid biases and only offer an approximation of the count of each motif. Historically, the interest of the community has moved from developing exact counting algorithms to developing approximated algorithms and heuristics able to tackle larger networks and extract larger motifs in an acceptable time, in spite of losing accuracy. The very first motif discovery algorithm was the exhaustive enumeration method proposed by Milo et al. [1]. The enumeration of all the possible patterns of subgraphs of a given number of nodes is computationally expensive, and only really small subgraphs of size 3 and 4 could be analyzed. After that, a plethora of improvements have been proposed. Among those, we highlight the ESU and RAND-ESU algorithms [25]. ESU is an efficient algorithm that enumerates the set of all induced subgraphs of a given size from a large graph exploiting a tree-like data structure. RAND-ESU is an unbiased sampling method built on top of ESU that samples the branches of the tree to visit. For a more in-depth overview of these algorithms, the interested reader can refer to [26].

While the amount of research for classic network motifs is huge, the data mining literature for generalized network motifs is far from being as complete, with some exceptions for temporal networks [27, 28]. In particular, the computational aspects associated with motif discovery in hypergraphs are largely overlooked. In their seminal paper [29], Horváth et al. propose an incremental polynomial time algorithm for the related problem of mining frequent sub-hypergraphs in hypergraph databases. In their setting, the frequency of a sub-hypergraph corresponds to the number of hypergraphs in the database containing a query sub-hypergraph. In the case of mining motifs, in which only one input hypergraph is considered, this algorithm can only find if a certain pattern of sub-hypergraph is present or not in the input hypergraph. Therefore, the algorithm proposed by Horváth et al. is not suited for mining motifs in a hypergraph, since we want to compute the exact number of occurrences of each possible pattern of sub-hypergraph in a given input hypergraph. Preti et al. tackle the problem of mining frequent patterns in simplicial complexes [30]. Albeit

both hypergraphs and simplicial complexes can represent higher-order interactions, the two mathematical objects present an important difference: simplicial complexes respect the downward closure property (i.e., in the case of hypergraphs, if a hyperedge e exists in a hypergraph H then also all the proper subsets of e exists in H). Lee et al. propose a set of algorithms, including an exact, a parallel and an approximated algorithm for mining motifs in hypergraphs [23, 31]. However, they focus only on the problem of extracting patterns of overlaps between hyperedges of fixed size.

2.3 Contributions

In our previous work [24], we adopt a traditional approach and extend to hypergraphs the original network motif definition proposed by Milo et al. [1], investigating all possible patterns of pairwise and group interactions among a given number of nodes. In [24], we extract fingerprints of real-world hypergraphs at their microscale and identify key motifs associated with families of hypergraphs belonging to different domains. Here, we build on our first work and further develop our algorithm for the efficient computation of frequencies of small patterns of sub-hypergraphs. While our proposal can outperform the baseline in every dataset, exact methods suffer from high computational costs and do not scale to large hypergraphs nor to patterns beyond four nodes. Hence, this paper introduces an approximated method for higher-order motif extraction based on hyperedge sampling as well. We show that our sampling method dramatically speeds up computations at the cost of only minimal errors in the estimation of motif frequency and allows us to analyse larger higher-order motifs beyond the computational limits of the exact approach.

3 Preliminaries and problem statement

Here we recall some basic definitions and give a formal description of the problem of our interest.

Definition 1 (Hypergraph) A hypergraph is a pair $H = (V, E)$ where V is the set of the vertices and E is the set of the *hyperedges*. A hyperedge e is a subset of V linking the vertices contained in it.

In Fig. 1 (left) we show an example of a hypergraph. We recall that a hypergraph in which every hyperedge links two vertices corresponds to the standard definition of a graph.

Definition 2 (Vertex-induced sub-hypergraph) The sub-hypergraph $H[V']$ induced by the subset $V' \subseteq V$ is the pair (V', E'), where $E' = \{e \in E : e \subseteq V'\}$.

Now we can define more formally the notions related to the *isomorphism* problem, which is the fundamental theoretical tool underlying network motif discovery.

Fig. 1 On the left, an example of a small higher-order network represented by a hypergraph. On the right, we show all the possible patterns of higher-order interactions involving three connected nodes
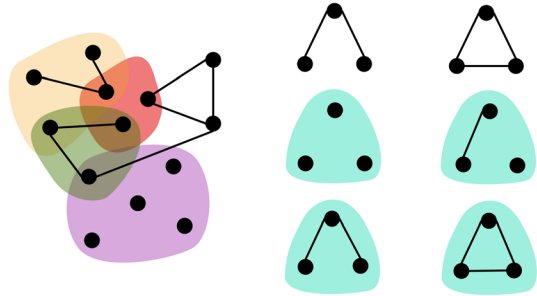
Definition 3 Two hypergraphs $H = (V, E)$ and $H' = (V', E')$ are isomorphic if they are identical modulo relabeling of the vertices. More formally, if there exists a bijection $f : V \rightarrow V'$ such that $e = \{u_1, ..., u_n\} \in E$ if and only if $e' = \{f(u_1), ..., f(u_n)\} \in E'$.

Definition 4 Given a hypergraph $H = (V, E)$ and a smaller query hypergraph $Q = (V', E')$, the occurrences of Q in H are all the sub-hypergraphs of H isomorphic to Q. We often refer to the number of occurrences of Q in H as the *frequency* of Q in H.

Finally, we can give a formal description of the problem of our interest.

Definition 5 Higher-order motifs are patterns of small connected sub-hypergraphs that occur in an observed hypergraph H at a frequency that is significantly higher with respect to a null model. We refer to the number of nodes involved in the pattern as the *order* of a higher-order motif.

As previously mentioned, to perform a higher-order motif analysis of a system, one needs to (i) count the frequency of each query higher-order motif in a hypergraph, (ii) compare the frequency of each query higher-order motif with that observed in a null model, and (iii) evaluate the over- or under-expression of query each higher-order motif. In this work we are mostly interested in the first step, i.e., motif discovery, therefore we will use higher-order motifs to refer to all the possible patterns of sub-hypergraphs involving a certain number of nodes. In Fig. 1 (right) we enumerate all the higher-order motifs of order 3.

Problem (Mining higher-order motifs). Given a hypergraph H and an integer $k > 2$, compute the frequency of every higher-order motif of order k.

4 Mining higher-order motifs

The enumeration of all the patterns of connected sub-hypergraphs of a given size is obviously the most expensive sub-task in motif analysis. The weight of this step is even more impactful considering that it must be repeated in randomized

networks as well. To solve this problem exactly, in this section, we propose a baseline algorithm based on projecting the hypergraph onto a graph and employing state-of-the-art motif analysis algorithms on it. Additionally, we present a more efficient method that directly leverages higher-order structures to construct sub-hypergraphs of a specified size.

4.1 Baseline algorithm

While traditional algorithms are not able to identify patterns of polyadic interactions, they can be used as a routine for more sophisticated algorithms. In our baseline, we consider the *projected graph* of a hypergraph.

Definition 6 (Projection of a hypergraph) The projection of a hypergraph $H = (V, E)$ is a graph $G = (V, E')$, defined on the same vertices of H and such that an edge between two vertices $a, b \in V$ exists if and only if a and b participate together in at least a hyperedge $e \in E$. In other words, every hyperedge $e \in E$ is replaced in G with a clique.

By running a classic algorithm (e.g., ESU [25]), we can efficiently enumerate connected subgraphs of size k in the projected graph. However, these subgraphs are only *candidate higher-order motifs* for two potential reasons: (i) they do not include higher-order interactions; (ii) even if a subgraph s of size k is connected in the projected graph, the sub-hypergraph induced by the vertices in s and the hyperedges E may be not connected. We highlight these pitfalls in Fig. 2.

In order to account for these issues, we construct the sub-hypergraph induced by the k nodes of the candidate motif (see Sect. 4.3) and check if this sub-hypergraph is connected. If it is, then one can simply update the frequency hash map, otherwise, the output is discarded. A more formal explanation of this method is reported in Algorithm 1. All in all, the baseline inherits the complexity of ESU [25]), plus the preprocessing cost of computing the clique projection of the hypergraph.
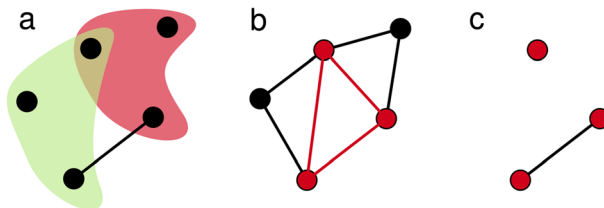

Fig. 2 **a** Example of a hypergraph H in which the baseline fails. **b** We highlight in red a connected subgraph s of size $k = 3$, one of the many possible outputs of a standard motif discovery algorithm applied on the projection of the previous hypergraph. **c** The sub-hypergraph induced by the vertices of s and the hyperedges of H is not connected

---

Input: a hypergraph $\mathcal{H} = (V, E)$ and an integer $k \in \{3, 4\}$.
Output: distribution of the frequency of the motifs of order k.
1: Let $G = (V, E')$ be the projection of $\mathcal{H}$
2: Let $\mathcal{M}$ be the motifs frequency dictionary
3: $\mathcal{S} \leftarrow \text{ESU}(G, k)$
4: **for each** $subgraph = (V^*, E^*) \in \mathcal{S}$ **do**
5:     $candidate\_motif \leftarrow$ sub-hypergraph (of $\mathcal{H}$) induced by V^*
6:     **if** $candidate\_motif$ is connected **then**
7:         Let $\mathcal{C}_m$ be the isomorphism class of $candidate\_motif$
8:         $\mathcal{M}[\mathcal{C}_m] \mathrel{+}= 1$
9:     **end if**
10: **end for**

Algorithm 1 Baseline: Counting higher-order motifs

Input: a hypergraph $\mathcal{H} = (V, E)$.
Output: distribution of the frequency of the motifs of order 3.
1: Let $\mathcal{M}$ be the motifs frequency hash map
2: **for each** hyperedge e of order 3 in E **do**
3:     $V^* \leftarrow$ vertices of e
4:     $motif \leftarrow$ sub-hypergraph induced by V^*
5:     Let $\mathcal{C}_m$ be the isomorphism class of $motif$
6:     $\mathcal{M}[\mathcal{C}_m] + = 1$
7: **end for**
8: $G \leftarrow$ Discard all hyperedges of order 3 from $\mathcal{H}$
9: $\mathcal{S} \leftarrow \text{ESU}(G, 3)$
10: **for each** $subgraph = (V^*, E^*) \in \mathcal{S}$ **do**
11:     **if** V^* not already visited **then**
12:         Let $\mathcal{C}_m$ be the isomorphism class of $subgraph$
13:         $\mathcal{M}[\mathcal{C}_m] + = 1$
14:     **end if**
15: **end for**

Algorithm 2 Efficient algorithm: Counting higher-order motifs of order 3

4.2 Efficient algorithms

The most expensive step in the previous algorithm is obviously the ESU subroutine. Moreover, the performance is widely impacted by the fact that hypergraph projections can be very dense and that lots of subgraphs are discarded for not satisfying the requirement of the induced sub-hypergraphs of being connected. To solve these problems, we work directly on hypergraphs, designing an efficient algorithm that exploits containment properties of higher-order structures in real-world systems. We optimize separately the two cases of 3- and 4-node motifs.

  3-*node motifs* As shown in Fig. 1, two of the motifs involving three nodes are composed only by pairwise relations, while the others involve one hyperedge of order 3.

Input: a hypergraph $\mathcal{H} = (V, E)$.
Output: distribution of the frequency of the motifs of order 4.
1: Let $\mathcal{M}$ be the motifs frequency hash map
2: **for each** hyperedge e of order 4 in E **do**
3:   $motif \leftarrow$ sub-hypergraph induced by e
4:   Let $\mathcal{C}_m$ be the isomorphism class of $motif$
5:   $\mathcal{M}[\mathcal{C}_m] += 1$
6:   Set vertices of $motif$ as visited
7: **end for**
8: $\mathcal{H} \leftarrow$ Discard all hyperedges of order 4 from $\mathcal{H}$
9: **for each** hyperedge e of order 3 in E **do**
10:   Let $\mathcal{E}$ be the set of hyperedges adjacent to e
11:   **for each** hyperedge e_i in $\mathcal{E}$ **do**
12:     **if** $|e \cup e_i| = 4$ and $e \cup e_i$ not already visited **then**
13:       $motif \leftarrow$ sub-hypergraph induced by $e \cup e_i$
14:       Let $\mathcal{C}_m$ be the isomorphism class of $motif$
15:       $\mathcal{M}[\mathcal{C}_m] += 1$
16:       Set vertices of $motif$ as visited
17:     **end if**
18:   **end for**
19: **end for**
20: $\mathcal{H} \leftarrow$ Discard all hyperedges of order 3 from $\mathcal{H}$
21: $\mathcal{S} \leftarrow \text{ESU}(\mathcal{H}, 4)$
22: **for each** $subgraph = (V^*, E^*) \in \mathcal{S}$ **do**
23:   **if** V^* not already visited **then**
24:     Let $\mathcal{C}_m$ be the isomorphism class of $subgraph$
25:     $\mathcal{M}[\mathcal{C}_m] += 1$
26:     Set V^* as visited
27:   **end if**
28: **end for**

Algorithm 3 Efficient algorithm: Counting higher-order motifs of order 4

To discover the latter, it is enough to iterate over all the hyperedges of order 3 and then recover the nested pairwise links to build the motif ("fill in" the hyperedges, see Sect. 4.3); the sub-hypergraph is trivially connected since its nodes are part of the same hyperedge. Then, we can ignore all the higher-order interactions and focus only on the pairwise links, since we are interested in counting the frequency of the first two motifs of Fig. 1. In this case, we can rely on ESU. This time, however, it will need to handle a lot fewer edges. Every time ESU returns an output, the triplet of nodes could have been counted already in the previous step (i.e., overlap between a pairwise motif and a hyperlink of order 3): in this case, the triplet is discarded. The first step has a complexity linear in the number of the hyperedges of size 3, while the second step inherits the complexity of the ESU algorithm. A formal description of the algorithm for higher-order motifs of order 3 is reported in Algorithm 2.

4-*node motifs* The algorithm for motifs of order 4 is similar, albeit there are some more details to take into account. One can still iterate on all the hyperedges of order 4, count the motifs by considering also the rich nested structures (one can observe that this time also hyperedges of order 3 can be nested), and discard all the 4-hyperedges. However, as a second step, one needs also to iterate over all the 3-hyperedges and consider all the possible neighbours; in fact, 3-hyperedges define only a sub-hypergraph with 3 nodes, while we are requesting 4 nodes. Neighbours can be listed by considering all the edges that add only 1 new node since 3 nodes are already fixed. The last step is to consider only pairwise interactions, and we rely again on ESU. Here, again, the first step has a complexity linear in the number of hyperedges of size 4, the second step has a complexity quadratic in the total number of hyperedges (linear in the number of hyperedges of size 3 and then linear for each hyperedge to explore its neighbourhood), and the final step inherits the complexity of the ESU algorithm. A formal description of the algorithm for higher-order motifs of order 4 is reported in Algorithm 3.

4.3 Algorithm details

Counting higher-order motifs can be interpreted as the enumeration of all the possible connected sub-hypergraphs of size k, assigning each of them to an isomorphism class. An efficient way to assign an isomorphism class to a connected sub-hypergraph of size k (for small values of k) is relying on a hash map. One can generate and hash every possible pattern of higher-order interactions involving k nodes, with all the possible relabelings. Relabelings are important because the same sub-hypergraphs can be stored with different labels on the vertices. For example, we have 6 different patterns of higher-order interactions with 3 nodes, each with 3! possible relabeling; eventually, the hash map will contain $6 \cdot 3! = 36$ entries. One can use the hash map as a counter since each observed sub-hypergraph is a key. After having enumerated all the sub-hypergraphs, the final count of each motif is simply the sum of all the entries of the hash map that belong to the same isomorphism class. We show a summary of this process in Fig. 3. Considering the sizes of the sub-hypergraphs involved, we can assume that this process incurs a constant time cost.

Another important routine in our algorithms is the construction of vertex-induced sub-hypergraphs. Given a set of vertices V', we are interested in querying the set of all the hyperedges to extract those who have all their endpoints in V'. This is what we referred to as "filling in" a set of vertices in the previous sections. For our specific case, this problem is efficiently solvable relying again on hash maps as follows. We can hash every hyperedge of a hypergraph: this ensures that we are able to check the existence of a hyperedge in constant time. Since we are only interested in solving this problem for a query set of vertices of size 3 or 4, we can easily generate all the possible 2^3 or 2^4 subsets of vertices (we can also ignore the empty set and the singletons) and check in constant time if each subset is an existing hyperedge. We show a summary of this process in Fig. 3. All in
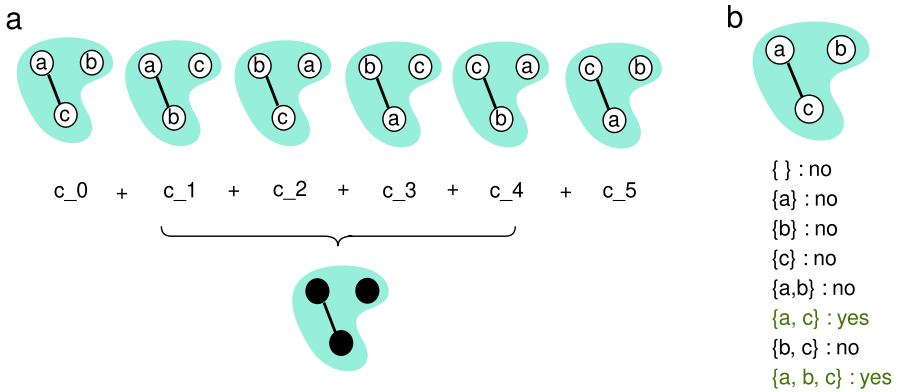
Fig. 3 **a** On the left, we show how to efficiently solve the problem of hypergraph isomorphism for small hypergraphs. We generate and hash every possible pattern of higher-order interactions involving k nodes with all the corresponding relabelings. Every observed sub-hypergraph will be equivalent to one and only one of the entries of the hash map. The final count of each motif is the sum of all the entries of the hash map that belong to the same isomorphism class. **b** On the right, we show how to construct vertex-induced sub-hypergraphs efficiently. As a preprocessing step, we hash every hyperedge in a hypergraph, allowing us to check for their existence in constant time. For a query set of 3 or 4 vertices, we generate all the possible 2^3 or 2^4 subsets of the query set and check in constant time if each subset is an existing hyperedge. Every time a subset is found to exist, we add it to the sub-hypergraph induced by the query set

all, given that we are interested in very small sets of vertices, we can construct vertex-induced sub-hypergraphs in constant time.

5 Sampling methods for counting higher-order motifs

Scalability is a persistent issue for exact motif discovery algorithms. Motif analysis has a number of real-world applications that require handling vast datasets. However, exact algorithms for motif discovery quickly become intractable for realistic inputs and motif sizes. To address this complexity, we propose an approximated method based on hyper-edge sampling.

Algorithm 4 samples with replacement S times a hyperedge e and enumerates all the connected sub-hypergraphs with a given number of nodes and containing e. The number of samples S controls the quality of the approximated results. However, directly sampling hyperedges from the hypergraphs leads to unreliable results. The distribution of hyperedge sizes is non-uniform, causing the algorithm to often sample hyperedges of size 2 while seldom sampling those of size 4. This skews the estimation of specific sub-hypergraph patterns. To mitigate this, we employ stratified sampling, segmenting the sampling process to guarantee a balanced consideration of hyperedges across different sizes. Let S_k be the number of samples assigned to hyperedges of size k, such that $S = \sum_k S_k$. We estimate appropriate values for S_k for every k empirically, exhaustively searching among different combinations of values and selecting those that maximize a defined quality function (see Section A).

Input: a hypergraph $\mathcal{H} = (V, E)$, number of hyperedge to sample S.
Output: approximated distribution of the frequency of the motifs of order 4.
1: Let $\mathcal{M}$ be the motifs frequency hash map
2: Let E_k represent the set of hyperedges of $\mathcal{H}$ of size k
3: Let S_k be the number of samples assigned to hyperedges of size k (ensure that the sum of the S_k for every possible k is equal to S)
4: $sampled\_edges \leftarrow$ sample S_2 hyperedges from E_2
5: $sampled\_edges \leftarrow sampled\_edges \cup$ sample S_3 hyperedges from E_3
6: $sampled\_edges \leftarrow sampled\_edges \cup$ sample S_4 hyperedges from E_4
7: **for each** hyperedge e in $sampled\_edges$ **do**
8:     Let $\mathcal{S}$ be the set of connected sub-hypergraphs of $\mathcal{H}$ containing e
9:     **for each** sub-$hypergraph$ in $\mathcal{S}$ **do**
10:         **if** sub-$hypergraph$ contains a hyperedge of cardinality bigger than e **then**
11:             continue
12:         **end if**
13:         Let $\mathcal{C}_m$ be the isomorphism class of sub-$hypergraph$
14:         $\mathcal{M}[\mathcal{C}_m] \mathrel{+}= 1$
15:     **end for**
16: **end for**
17: **for each** motif m with count c in $\mathcal{M}$ **do**
18:     Let $maxcard$ be the maximum size of the hyperedges in m
19:     Let $countmax$ be the number of the hyperedges in m of size $maxcard$
20:     $c \leftarrow \frac{c \cdot |E_{maxcard}|}{S_{maxcard} \cdot countmax}$
21: **end for**

Algorithm 4 Sampling algorithm: Counting higher-order motifs of order 4

The sampling algorithm proceeds in a way similar to the exact method (therefore we avoid explicitly repeating some details in the pseudocode). If we target the discovery of motifs of size k, then for all the sampled hyperedges of size k we have all the necessary vertices to build a target sub-hypergraph. No exploration of the neighbourhood is further required to add new nodes to the pattern. The complexity of this step is linear in the number of samples S_k. Then, for all the sampled hyperedges of size less than k, some exploration of possibly different levels of the neighbourhoods is required. The complexity of each of these steps is linear in the number of samples S_k multiplied, for each level of exploration, by a factor linear in the number of hyperedges. Moreover, for each pattern, the previously mentioned process of "filling in" the hyperedges is repeated to build vertex-induced sub-hypergraphs and count the right instances of the motifs. Again, these routines take constant time.

In order to estimate the exact count for each motif, the algorithm multiplies the observed count by a correction factor given by the probability of sampling a certain motif, as reported in the pseudocode. To simplify the computation of the correction factor, the algorithm discards all sub-hypergraphs encountered during the exploration of the neighbourhood of a hyperedge e that contains at least one hyperedge with

Table 1 Summary statistics of the datasets considered for our experiments. Each higher-order network is described by the domain, the number of nodes, and the total number of hyperedges of size 2, 3, 4 and 5

Dataset	Domain	N	E_2	E_3	E_4	E_5
hs	proximity	327	5498	2091	222	7
ps	proximity	242	7748	4600	347	9
EU	e-mail	998	12,753	4938	2294	1359
dblp	co-auth	1,924,991	693,363	667,291	419,434	205,970
history	co-auth	1,014,734	160,885	47,423	19,120	8775
geology	co-auth	1,256,385	275,736	227,950	159,509	99,140

a higher cardinality than e. In other words, a pattern of sub-hypergraph is only considered when the hyperedge of maximal cardinality is sampled. Given this approach, it is straightforward to prove that the estimator is unbiased.

6 Experimental evaluation

To assess the improvement in the performance of the algorithms for higher-order motif discovery when (i) exploiting higher-order structures instead of applying classic methods on the hypergraph projection and (ii) approximating motif frequency, we collected a variety of hypergraph datasets representing real-world systems with group interactions. Besides the evaluation of the performance, we also study the accuracy of the sampling algorithm and exploit sampling methods to study higher-order motifs of order 5. All the experiments have been carried out on a machine with an 8-core (2.2GHz) Intel Xeon CPU and 94GB of RAM, running Ubuntu 20.04.4 LTS. The algorithms presented in this paper are implemented in Python3. The code is publicly available [32]. Moreover, all the algorithms presented in this work are included in the Python library Hypergraphx for higher-order network analysis [33].

6.1 Datasets

We collected a variety of real-world datasets from different domains, describing face-to-face interactions, co-authorship relations and e-mail communications. Co-authorship data (dblp, history and geology) [34] is naturally encoded as a set of nodes (the authors) involved in higher-order interactions (the scientific papers). Also E-mail data (EU) [34] is naturally encoded as a set of higher-order interactions since e-mails can have multiple recipients at the same time. However, higher-order interactions need to be inferred from pairwise relations in data about face-to-face interactions (primary school and high school) [34]. In this case, cliques of size k are promoted to group interactions of order k, if the corresponding dyadic encounters happened at the same time. The summary statistics of the datasets are reported in Table 1. The datasets, as well as the preprocessing scripts, are publicly available [32].

Table 2 Comparison of the running time (s) of the exact algorithms with motifs of order 3 and 4

Dataset	Base-3	Eff-3	Base-4	Eff-4
hs	7	5	362	230
ps	25	18	1920	1339
EU	44	29	5286	2757
dblp	1185	134	> 24h	2885
history	42	19	4591	526
geology	207	36	32810	475

6.2 Performance evaluation

In Table 2 we compare our exact algorithm for higher-order motif discovery against the baseline algorithm in terms of their execution running time. We show that exploiting directly higher-order structures speeds up the computations. The efficient algorithm outperforms the baseline in every dataset. Moreover, it is worth mentioning that the analysis with motifs of order 4 of dblp with the baseline algorithm was not feasible in a reasonable amount of time. The larger gains are observed in co-authorship data. Co-authorship systems are proven to display a nested structure of hyperedges made up of a small number of hyperedges of large average size [24]. In fact, these kinds of systems are the ideal scenario for our algorithm. We can notice that the gains are not as noticeable in social datasets, which tend to be governed by dense patterns of lower-order interactions [24].

In Table 3 we show the execution running time in seconds of the sampling algorithm on the different datasets with multiple values of S, i.e., the parameter that controls the number of samples. The different size scales of the co-authorship and social datasets require different sample sizes to achieve results of comparable quality. Since the analysis of the motifs of order 3 was already easily doable, we consider only the task of motif discovery of order 4. We show that hyperedge sampling dramatically improves performance. As expected, the parameter S heavily affects the running time. As always, there is a trade-off between the accuracy of the results (higher values of S lead to more accurate estimates) and the execution running time.

6.3 Accuracy of sampling method

Besides evaluating the running time of the sampling method, it is also important to assess the output quality of the estimates compared to exact higher-order motif profiles. We compute motif profiles [24] comparing the observed frequencies of the motifs with those on a null model [35] to assess their statistical significance (we sample $N = 10$ times from the configuration model).

We evaluate the quality of the estimated motif profiles in terms of:

Table 3 Hyperedge sampling dramatically improves the performance with respect to the exact algorithm. The execution running time of the approximated algorithm heavily depends on the choice of the sample size S. The correlation coefficient ρ between the estimated and the exact motif profiles, the maximum absolute error MaxAE and the mean average error MAE improve with increases in the number of samples S. Due to their different size scale, co-authorship and social datasets require different sample sizes to achieve comparable results. We obtain reasonable results even with a very limited number of samples

Dataset	Exact exec. time (s)	S	Approx. exec. time (s)	ρ	MaxAE	MAE
hs	230	100	3	.914	.151	.015
		250	8	.953	.122	.011
		500	16	.978	.096	.007
		1K	29	.987	.065	.006
ps	1339	100	11	.918	.151	.017
		250	30	.950	.135	.012
		500	63	.977	.093	.008
		1K	118	.986	.071	.006
EU	2757	100	15	.804	.203	.028
		250	34	.887	.159	.020
		500	73	.923	.134	.016
		1K	144	.963	.098	.010
dblp	2885	1K	32	.495	.088	.055
		2.5K	41	.555	.088	.047
		5K	56	.610	.089	.038
		10K	85	.696	.087	.027
history	526	1K	5	.679	.176	.033
		2.5K	7	.804	.169	.022
		5K	11	.867	.170	.016
		10K	22	.913	.124	.012
geology	475	1K	12	.590	.107	.047
		2.5K	15	.661	.107	.039
		5K	19	.719	.106	.032
		10K	30	.784	.103	.024

- the *Pearson's correlation coefficient ρ* between the estimated and the exact higher-order motif profiles. The coefficient assigns values close to 1 to profiles in strong agreement and values close to -1 to profiles in strong disagreement.
- the *Maximum Absolute Error (MaxAE)* in estimating higher-order motif profiles.
- the *Mean Absolute Error (MAE)* in estimating higher-order motif profiles.

In Table 3, we show that we obtain good results even with a small number of samples with respect to the total size of the hypergraphs. The measures of output quality improve with increases in S. A good trade-off between the output quality, S and the execution running time will be critical for real-world applications. The measures of output quality are averaged across 10 repetitions for every value of S.

A second evaluation metric is the correlation matrix of the motif significance profiles [1, 24] of the different real-world hypergraphs. In Fig. 4, the correlation matrix shows the emergence of two "superfamilies" of real-world hypergraphs, in a way similar to [24]. Clustering tends to separate social and co-authorship data.

Fig. 4 The correlation matrix of the significance profiles built with sampling methods ($S = 1000$ for co-authorship data and $S = 100$ for social data) highlights the emergence of two clusters that separate social and co-authorship data
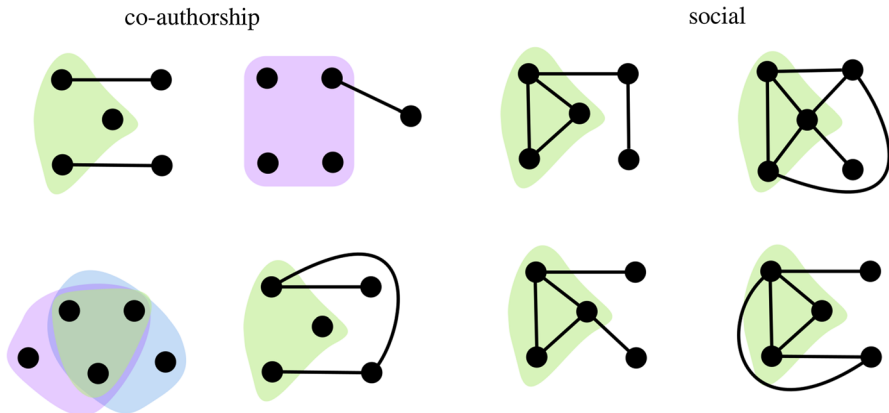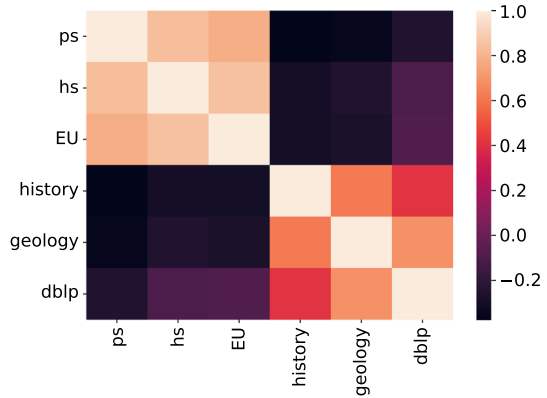

Fig. 5 Over-expressed patterns of higher-order interactions highlight structural principles of the different domains

This further proves that sampling methods are still able to capture and highlight patterns of higher-order interactions that are probably linked to the functionalities of the networks.

6.4 Applications: Mining larger higher-order motifs

Approximated methods not only speed up motif analysis for large datasets but also allow for the study of larger patterns of interactions. Exact counting algorithms are suited only for the extraction of motifs of order 3 and 4. Here we employ our proposed sampling algorithm and characterize two real-world hypergraphs, namely `history` and `high school`, in terms of their higher-order motifs of order 5. To evaluate the statistical significance of the results, we compare the results with those on a configuration model. We use the same statistical evaluation methods proposed in [24], i.e., we consider the relative abundance of each motif with respect to

a configuration model [35]. In Fig. 5 we show the most over-expressed higher-order motifs of order 5 in both the real-world hypergraphs. We can notice how also at this scale we still observe characteristic patterns of co-authorship data (low number of interactions of large average size) and face-to-face data (high number of interactions of small average size). This is in line with the results on the nested structure of real-world hyperedges proposed in [24].

7 Conclusion

In this paper, we have proposed a first algorithmic framework for the problem of higher-order motif discovery in hypergraphs as defined in [24]. We have developed an efficient exact algorithm for the analysis of the higher-order motif of order 3 and 4. Such an exact algorithm exploits hyperedges and their nested structure to efficiently enumerate sub-hypergraphs of a given size. We have proved that considering directly the hypergraph structure of data outperforms traditional computational frameworks for network motifs that work on projected data. We have developed an approximated method based on hyperedge sampling to overcome scalability issues of exact algorithms. We have proved that such an approximated algorithm allows for a huge gain in the running time at only a little expense on the accuracy of the results. Moreover, our sampling algorithm allows for the analysis of larger motifs, which were not computationally feasible with exact methods. We believe that faster algorithms for higher-order motif analysis, such as our proposed sampling algorithm, can pave the way for exciting applications. In this direction, important aspects for future work are the development of sampling algorithms with strong approximation guarantees and the investigation of more efficient sampling strategies for the different categories of patterns of higher-order interactions.

Appendix A: Parameters search

Our approximated algorithm requires different parameters. The parameter S controls the number of samples of hyperedges to be performed to estimate the count of the patterns of sub-hypergraphs. Without a careful design, directly sampling hyperedges from the hypergraphs leads to unreliable results. In fact, the distribution of the size of the hyperedges in a real-world hypergraph is not uniform, leading the algorithm to often sample hyperedges of size 2, and rarely, for example, hyperedges of size 4. This would result in poor estimations of higher-order motifs involving a group interaction of size 3 or 4. To address this issue, we stratify our sampling process, allocating specific sample budgets to hyperedges of different sizes. This ensures a balanced representation of hyperedges across all sizes. Let S_k be the number of samples assigned to hyperedges of size k. We fix the sum of S_k for every k to be equal to S. We estimate empirically good values for the parameters S_k, exhaustively searching among different combinations of values and selecting those that maximize a defined quality function (Pearson's correlation ρ between the exact higher-order motif profile and the estimated one). We perform the analyses on two datasets, one for each macro-domain: `high school` and `history`. We consider motifs of order 4, therefore we need to estimate S_2, S_3 and
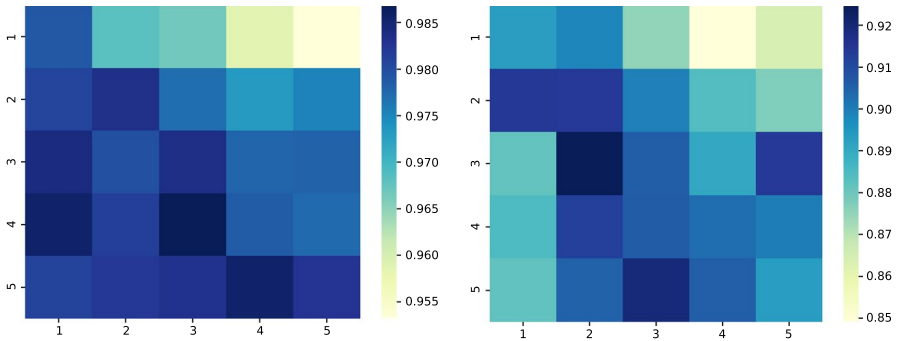
Fig. 6 We parametrize the number of samples of hyperedges of size 3 and 4 with respect to the number of samples of hyperedges of size 2 and search the values for which the correlation between the exact motif profile and the estimated one is maximized. The *x*-axis parametrizes the number of samples of hyperedges of size 4. The *y*-axis parametrizes the number of samples of hyperedges of size 3. Light squares exhibit lower levels of correlation, while dark squares show higher levels. On the left, we show the matrix for the high school dataset. On the right, is the matrix for the history dataset. We get the best parameters by averaging the two matrices

S_4, namely, respectively the number of samples from the hyperedges of order 2, 3 and 4. Given that $S_2 + S_3 + S_4 = S$, one can fix S_2, parametrize S_3 and S_4 to be multiple of S_2, and perform exhaustive search. We show the results in Fig. 6. Averaging the results of the two matrices, we get that our quality measure is maximized when $S_3 = 3S_2$ and $S_4 = 2S_2$. We use these parameters in our experiments.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U (2002) Network motifs: simple building blocks of complex networks. Science 298(5594):824–827
2. Schwarze AC, Porter MA (2021) Motifs for processes on networks. SIAM J Appl Dyn Syst 20(4):2516–2557
3. Milo R, Itzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M, Alon U (2004) Superfamilies of evolved and designed networks. Science 303(5663):1538–1542
4. Alon U (2007) Network motifs: theory and experimental approaches. Nat Rev Genet 8(6):450
5. Shen-Orr SS, Milo R, Mangan S, Alon U (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. Nat Genet 31(1):64
6. Dobrin R, Beg QK, Barabási A-L, Oltvai ZN (2004) Aggregation of topological motifs in the *Escherichia coli* transcriptional regulatory network. BMC Bioinformatics 5(1):1–8
7. Sporns O, Kötter R (2004) Motifs in brain networks. PLoS Biol 2(11):369
8. Chen L, Qu X, Cao M, Zhou Y, Li W, Liang B, Li W, He W, Feng C, Jia X et al (2013) Identification of breast cancer patients based on human signaling network motifs. Sci Rep 3(1):1–7
9. Hong-lin, X., Han-bing, Y., Cui-fang, G., Ping, Z.: Social network analysis based on network motifs. J Appl Math (2014)
10. Saracco F, Di Clemente R, Gabrielli A, Squartini T (2016) Detecting early signs of the 2007–2008 crisis in the world trade. Sci Rep 6(1):1–11
11. Bascompte J, Stouffer DB (2009) The assembly and disassembly of ecological networks. Philos Trans R Soc B: Biol Sci 364(1524):1781–1787
12. Simmons BI, Cirtwill AR, Baker NJ, Wauchope HS, Dicks LV, Stouffer DB, Sutherland WJ (2019) Motifs in bipartite ecological networks: uncovering indirect interactions. Oikos 128(2):154–170
13. Onnela J-P, Saramäki J, Kertész J, Kaski K (2005) Intensity and coherence of motifs in weighted complex networks. Phys Rev E 71:065103
14. Kovanen L, Karsai M, Kaski K, Kertész J, Saramäki J (2011) Temporal motifs in time-dependent networks. J Stat Mech: Theory Exp 2011(11):11005
15. Paranjape A, Benson AR, Leskovec J (2017) Motifs in temporal networks. In: Proceedings of the tenth ACM international conference on web search and data mining, pp 601–610 (2017). ACM
16. Battiston F, Nicosia V, Chavez M, Latora V (2017) Multilayer motif analysis of brain networks. Chaos Interdiscip J Nonlinear Sci 27(4):047404
17. Kivelä M, Porter MA (2018) Isomorphisms in multilayer networks. IEEE Trans Netw Sci Eng 5(3):198–211
18. Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J-G, Petri G (2020) Networks beyond pairwise interactions: structure and dynamics. Phys Rep 874:1–92
19. Battiston F, Amico E, Barrat A, Bianconi G, Ferraz de Arruda G, Franceschiello B, Iacopini I, Kéfi S, Latora V, Moreno Y, Murray MM, Peixoto TP, Vaccarino F, Petri G (2021) The physics of higher-order interactions in complex systems. Nat Phys 17(10):1093–1098
20. Patania A, Petri G, Vaccarino F (2017) The shape of collaborations. EPJ Data Sci 6(1):18
21. Cencetti G, Battiston F, Lepri B, Karsai M (2021) Temporal properties of higher-order interactions in social networks. Sci Rep 11:7028
22. Berge, C.: Graphs and hypergraphs (1973)
23. Lee G, Ko J, Shin K (2020) Hypergraph motifs: concepts, algorithms, and discoveries. Proc VLDB Endow 13(12):2256–2269
24. Lotito QF, Musciotto F, Montresor A, Battiston F (2022) Higher-order motif analysis in hypergraphs. Commun Phys 5(1):79
25. Wernicke S (2006) Efficient detection of network motifs. IEEE/ACM Trans Comput Biol Bioinf 3(4):347–359
26. Patra S, Mohapatra A (2020) Review of tools and algorithms for network motif discovery in biological networks. IET Syst Biol 14(4):171–189
27. Liu, P., Benson, A.R., Charikar, M.: Sampling methods for counting temporal motifs. In: Proceedings of the twelfth ACM international conference on web search and data mining. WSDM '19, pp 294–302. Association for Computing Machinery, New York, NY, USA (2019)
28. Jazayeri A, Yang CC (2020) Motif discovery algorithms in static and temporal networks: a survey. J Complex Netw 8(4):cnaa031

29. Horváth T, Bringmann B, De Raedt L (2007) Frequent hypergraph mining. In: Muggleton S, Otero R, Tamaddoni-Nezhad A (eds) Inductive logic programming. Springer, Berlin, pp 244–259
30. Preti G, De Francisci Morales G, Bonchi F (2022) Fresco: mining frequent patterns in simplicial complexes. In: Proceedings of the ACM web conference 2022. WWW '22, pp. 1444–1454. Association for Computing Machinery, New York, NY, USA (2022)
31. Lee, G., Shin, K.: Thyme+: temporal hypergraph motifs and fast algorithms for exact counting. In: 2021 IEEE international conference on data mining (ICDM), pp 310–319 (2021). IEEE
32. Lotito, Q.F.: Higher-order motif discovery sampling algorithm (2022). https://github.com/FraLotito/sampling-motifs
33. Lotito QF, Contisciani M, De Bacco C, Di Gaetano L, Gallo L, Montresor A, Musciotto F, Ruggeri N, Battiston F (2023) Hypergraphx: a library for higher-order network analysis. J Complex Netw 11(3):019
34. Benson AR, Abebe R, Schaub MT, Jadbabaie A, Kleinberg J (2018) Simplicial closure and higher-order link prediction. Proc Natl Acad Sci USA 115(48):11221–11230
35. Chodrow, P.S.: Configuration models of random hypergraphs. J Complex Netw 8(3) (2020)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.