

# Mapping flows on hypergraphs

Anton Eriksson (✉ [anton.eriksson@umu.se](mailto:anton.eriksson@umu.se))

Umeå University <https://orcid.org/0000-0001-5859-4073>

Daniel Edler

Umeå University

Alexis Rojas

Umeå University

Martin Rosvall

Umeå University <https://orcid.org/0000-0002-7181-9940>

---

## Article

**Keywords:** mapping flows, hypergraphs

**Posted Date:** January 7th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-134751/v1>

**License:**  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at Communications Physics on June 11th, 2021. See the published version at <https://doi.org/10.1038/s42005-021-00634-z>.

# 1 Mapping flows on hypergraphs

2 Anton Eriksson,\* Daniel Edler, Alexis Rojas, and Martin Rosvall

3 *Integrated Science Lab,*  
4 *Department of Physics,*  
5 *Umeå University, SE-901 87 Umeå,*  
6 *Sweden*

7 (Dated: December 23, 2020)

8 Hypergraphs offer an explicit formalism to describe multibody interactions in complex systems. To connect  
9 dynamics and function in systems with these higher-order interactions, network scientists have generalised  
10 random walk models to hypergraphs and studied the multibody effects on flow-based centrality measures.  
11 But mapping the large-scale structure of those flows requires effective community detection methods. We  
12 derive unipartite, bipartite, and multilayer network representations of hypergraph flows and explore how they  
13 and the underlying random walk model change the number, size, depth, and overlap of identified multilevel  
14 communities. These results help researchers choose the appropriate modelling approach when mapping  
15 flows on hypergraphs.

16 Researchers model and map flows on networks to identify impor-  
17 tant nodes and detect significant communities<sup>1,2,3,4</sup>. From small to  
18 large system scales, random walk-based methods help to uncover  
19 the inner workings of the systems the networks represent<sup>5,6</sup>. When  
20 standard network models fail to adequately represent a system's  
21 interactions, researchers turn to higher-order models of complex  
22 systems<sup>7,8</sup>, including multilayer networks<sup>9,10,11</sup> for multitype inter-  
23 actions, non-Markovian networks<sup>12,13,14</sup> for multistep interactions,  
24 and combinatorial models such as simplicial complexes<sup>15,16,17,18</sup>  
25 and hypergraphs<sup>19,20,21,22</sup> with nodes in hyperedges for multibody  
26 interactions.

27 While several methods can identify flow-based communities in  
28 multilayer<sup>9,23,24</sup> and memory<sup>12,13,14</sup> networks with non-Markovian  
29 dynamics, researchers have just begun to unravel the large-scale  
30 systemic effects of multibody interactions captured by hyper-  
31 graphs<sup>22</sup>. However, different systems and research questions call  
32 for different random walk and hypergraph models: Random walks  
33 can be lazy and able to visit the same node multiple times in a  
34 row or non-lazy and forced to move on. Hyperedges can have arbi-  
35 trary weights, and nodes can have hyperedge-dependent weights.  
36 Because these and other models can be represented with different  
37 network types – bipartite, unipartite, and multilayer networks –  
38 the questions multiply: How do different hypergraph random walk  
39 models combined with different network representations change  
40 the flow dynamics at scales captured by communities?

41 For example, random walks on hypergraphs can model flows  
42 of ideas in co-authorship networks. A node represents an author,  
43 and a hyperedge connects all authors of a paper. In the simplest  
44 dynamics, a random walker on a node picks a random hyperedge  
45 among those that contain the node and steps to a random node of  
46 the picked hyperedge. Then repeats. Excluding author self-links  
47 for non-lazy walks or including hyperedge weights from paper  
48 citations or using hyperedge-dependent node weights for varying  
49 author contributions are natural model variations that generate

50 different dynamics<sup>20,21</sup>. How does the organisation of authors in  
51 nested communities from research groups to research areas change  
52 with random-walk model and representation?

53 For lazy random walks on hypergraphs with self-links and  
54 hyperedge-independent node weights, random walks on weighted,  
55 undirected networks generate equivalent dynamics<sup>20</sup>. Each hyper-  
56 edge becomes a clique with properly adjusted link weights. This  
57 projection enables standard flow-based methods developed for  
58 weighted networks to identify communities where random walks  
59 stay for a long time. Non-lazy walks or walks with hyperedge-  
60 dependent node weights require directed networks<sup>20</sup>. A bipartite  
61 representation provides hyperedge assignments, and a multilayer  
62 representation enables overlapping communities.

63 Representing hypergraphs with bipartite networks requires  
64 weighted, directed links between two sets of nodes: one for the  
65 nodes and one for the hyperedges. Picking a random hyperedge  
66 becomes an explicit step to a hyperedge node. Non-lazy walks on  
67 the hypergraph require non-backtracking walks on the bipartite  
68 network<sup>25</sup>. With proper normalisation, the node-visit rates stay  
69 the same. Though unipartite and bipartite representations give  
70 identical node flows, the bipartite representation's link flows from  
71 nodes to hyperedge nodes and back to nodes can induce more  
72 flows between communities and alter the optimal community com-  
73 position. The community-detection algorithm must also assign  
74 more nodes, which implies more degrees of freedom and a larger  
75 search space.

76 Multilayer networks represent the hyperedges as layers with fully  
77 connected groups of nodes. Each node is present in each of its  
78 hyperedge layers. Hyperedge weights become layer weights, and  
79 hyperedge-dependent node weights become layer-dependent node  
80 weights. Though the node visit rates aggregated over layers remain  
81 the same, multilayer networks multiply the degrees of freedom  
82 and enable new models. Reducing the inter-layer link weights  
83 increases the time a random walker spends within a hyperedge  
84 before moving to another. Reducing the inter-layer link weights  
85 only between dissimilar layers reinforces flows within similar  
86 layers. The search space expands when nodes can belong to  
87 multiple overlapping communities.

---

\* anton.eriksson@umu.se

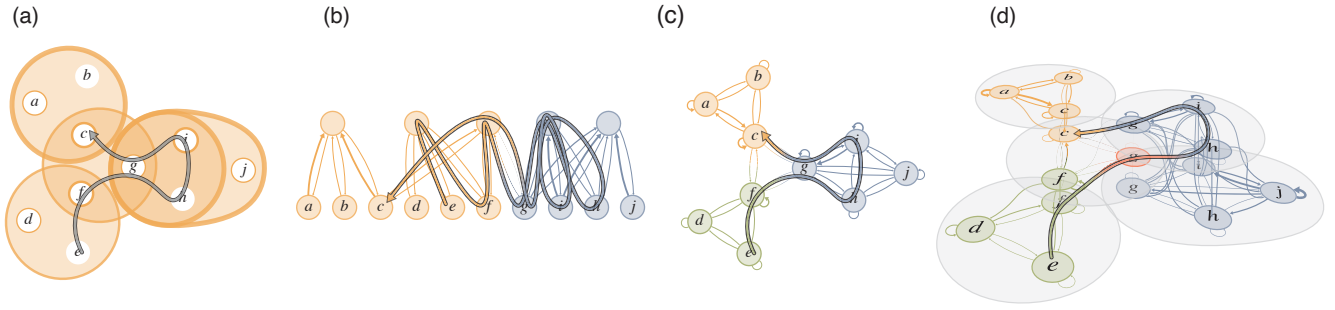


Fig. 1. A schematic hypergraph represented with three types of networks. (a) The schematic hypergraph with weighted hyperedges and hyperedge-dependent node weights. Thin borders for weight 1 and thick borders for weight 3. A lazy random walk on the schematic hypergraph represented on: (b) a bipartite network, (c) a unipartite network, and (d) a multilevel network. The colours indicate optimised module assignments, in (d) for hyperedge-similarity walks.

88 The many combinations of random walk models and represen- 127  
 89 tations available to address specific research problems call for 128  
 90 the question: For different data and questions, which model and 129  
 91 representation is best?

92 To address which combination of model and representation is 130  
 93 best for answering different questions about various hypergraph 131  
 94 data, we derive unipartite, bipartite, and multilayer network repre- 132  
 95 sentations of hypergraph flows with identical node-visit rates for 133  
 96 the same random-walk model. For unique node-visit rates when 134  
 97 a representation requires directed links, we apply an unrecorded 135  
 98 teleportation scheme robust to changes in the teleportation rate 136  
 99 and that preserves the node-visit rates when teleportation is super- 137  
 100 fluous in undirected networks<sup>26</sup>. The information-theoretic and 138  
 101 flow-based community detection method Infomap<sup>27</sup> allow us to 139  
 102 explore how different hypergraph random walk models and network 140  
 103 representation change the number, size, depth, and overlap 141  
 104 of identified multilevel communities.

105 By analysing schematic and real hypergraphs, we find that 142  
 106 the bipartite network representation requires the fewest links and 143  
 107 enables the fastest community detection. A multilayer network 144  
 108 representation that reinforces flows within similar layers give the 145  
 109 deepest modular structures with most overlapping communities but 146  
 110 at a high computational cost. The unipartite network representation 147  
 111 provides a trade-off with intermediate compactness, speed, and 148  
 112 detectable modular regularities.

## 113 Results and Discussion

114 **Modelling flows on hypergraphs.** We model flows on hyper- 151  
 115 graphs with random walks. We use hypergraphs with nodes  $V$ , 152  
 116 hyperedges  $E$  with weights  $\omega$ , and hyperedge-dependent node 153  
 117 weights  $\gamma$ . Each hyperedge  $e$  has a weight  $\omega(e)$ . Each node  $u$  154  
 118 with incident hyperedges  $E(u) = \{e \in E : u \in e\}$  has a weight 155  
 119  $\gamma_e(u)$  for each incident hyperedge  $e$ . To simplify the notation 156  
 120 when normalising weights into probabilities, we denote node  $u$ 's 157  
 121 total incident hyperedge weight  $d(u) = \sum_{e \in E(u)} \omega(e)$  and hy-  
 122 peredge  $e$ ' total node weight  $\delta(e) = \sum_{u \in e} \gamma_e(u)$ <sup>20</sup>. With these  
 123 weights, a lazy random walker moves from node  $u$  at time  $t$  to  
 124 node  $v$  at time  $t + 1$  in three steps by<sup>20</sup>:

125 1. Picking hyperedge  $e$  among node  $u$ 's hyperedges  $E(u)$  with  
 126 probability  $\frac{\omega(e)}{d(u)}$ .

2. Picking one of the hyperedge  $e$ 's nodes  $v$  with probability  
 $\frac{\gamma_e(v)}{\delta(e)}$ .  
 3. Moving to node  $v$ .

130 Variations include non-lazy walks, which never visit the same  
 131 node twice in a row with a modified second step

2. Picking one of the hyperedge  $e$ 's nodes  $v \neq u$  with proba-  
 bility  $\frac{\gamma_e(v)}{\delta(e) - \gamma_e(u)}$ ,

134 and teleporting walks, which jump to a random node at some  
 135 rate to ensure that all nodes can be reached from any node in a  
 136 finite number of moves, so-called ergodic walks. We pick the  
 137 next hyperedge based on the similarity with the previously picked  
 138 hyperedge in hyperedge-similarity walks, which are useful for  
 139 modelling flows that tend to stay among similar hyperedges such  
 140 as among research papers with similar author lists and likely similar  
 141 topics. These walks require memory and correspond to a higher-  
 142 order Markov chain model because they depend on the previously  
 143 picked hyperedge.

144 The bipartite, unipartite, and multilayer network representations  
 145 have different advantages and limitations (Fig. 1). A weighted,  
 146 undirected network suffices for memoryless lazy random walks  
 147 without hyperedge-dependent node weights, hyperedge-dependent  
 148 node weights require directed networks, and hyperedge-similarity  
 149 walks require multilayer networks.

150 Bipartite networks offer the most direct representation of the  
 151 three-step random walk process above. We represent the hyper-  
 152 edges with hyperedge nodes, and the three steps become a two-step  
 153 walk between the nodes at the bottom and the hyperedge nodes at  
 154 the top in Fig. 1b. For simplicity, we refer to them as nodes and  
 155 hyperedge nodes. First a step from a node  $u$  to a hyperedge node  $e$ ,

$$P_{ue} = \frac{\omega(e)}{d(u)}, \quad (1)$$

156 and then a step from the hyperedge node to a node  $v$ ,

$$P_{ev} = \frac{\gamma_e(v)}{\delta(e)}. \quad (2)$$

157 By starting the random walk on the nodes and taking two steps at  
 158 a time, corresponding to Markov time two<sup>28</sup>, hyperedge nodes are  
 159 only intermediate stops with zero flow when the random walk is  
 160 back on the nodes after two steps. The stationary distribution of  
 161 the random walk is concentrated to the nodes. For non-lazy walks  
 162 represented with bipartite networks, we use so-called state nodes<sup>27</sup>  
 163 in the hyperedge nodes. One state node for each incoming link  
 164 has out-links to all nodes in the hyperedge except the incoming  
 165 link's source ensures that the walks are not backtracking (Fig. 2).

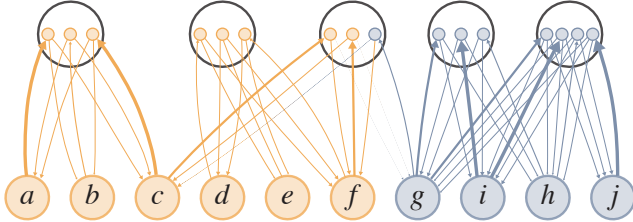


Fig. 2. Bipartite network with state nodes for non-lazy random walks. To prevent random walks on bipartite networks to visit the same node at the bottom twice in a row by backtracking from the hyperedge node at the top, we use state nodes in the hyperedge nodes. Each hyperedge node requires one state node for each node in the hyperedge. The state nodes have one incoming link from its source node and outgoing links to all other nodes in the hyperedge. Colours indicate the optimised partition in Fig. 3(b).

166 To represent the random walk on a unipartite network, we  
 167 project the three-step random walk process down to a one-step  
 168 process between the nodes and describe it with the transition rate  
 169 matrix

$$P_{uv} = \sum_{e \in E(u,v)} P_{ue} P_{ev} = \sum_{e \in E(u,v)} \frac{\omega(e)}{d(u)} \frac{\gamma_e(v)}{\delta(e)}, \quad (3)$$

170 where  $E(u, v) = \{e \in E : u \in e, v \in e\}$  is the set of hyperedges  
 171 incident to both nodes  $u$  and  $v$ . Each hyperedge forms a fully  
 172 connected group of nodes (Fig. 1c). Unipartite networks for  
 173 non-lazy walks have no self-links. Compared with the bipartite  
 174 representation, the unipartite representation with fully connected  
 175 groups of nodes requires more links.

176 To represent the random walk on a multilayer network, we  
 177 project the three-step random walk process down to a one-step  
 178 process on state nodes in separate layers  $\alpha$  for each hyperedge  $e$ .  
 179 A state node  $u^\alpha$  represents  $u$  in each layer  $\alpha \in E(u)$  that contains  
 180 the node. All state nodes in the same layer form a fully connected  
 181 set (Fig. 1d). The transition rate between state node  $u^\alpha$  in layer  $\alpha$   
 182 and state node  $v^\beta$  in layer  $\beta$  is

$$P_{uv}^{\alpha\beta} = \frac{\omega(\beta)}{d(u)} \frac{\gamma_\beta(v)}{\delta(\beta)} \text{ for } \beta \in E(u, v). \quad (4)$$

183 Node  $u$ 's state node visit rates in different layers sum to  $u$ 's visit  
 184 rate in the unipartite and bipartite representations. With one state  
 185 node per hyperedge layer that contains the node, the multilayer  
 186 representation requires the most nodes and links to describe the

187 walk. But this cost comes with benefits. The multilayer representa-  
 188 tion can describe higher-order Markov chains, which can capture  
 189 more regularities in the data.

190 For example, a useful variant of the basic hypergraph random  
 191 walk is to pick a hyperedge not only proportional to its weight but  
 192 also proportional to how similar it is to the hyperedge picked in  
 193 the previous step. To include hyperedge-dependent node weight  
 194 information in the similarity measure, we use one minus the Jensen-  
 195 Shannon divergence (JSD) between the transition rate vectors  $\mathbf{P}_{\alpha v}$   
 196 and  $\mathbf{P}_{\beta v}$  to nodes at layers  $\alpha$  and  $\beta$  as the hyperedge coupling  
 197 strength,

$$\begin{aligned} D_u^{\alpha\beta} &= \omega(\beta) [1 - \text{JSD}(\alpha, \beta)] \\ &= \omega(\beta) \left[ 1 - H \left( \frac{1}{2} \mathbf{P}_{\alpha v} + \frac{1}{2} \mathbf{P}_{\beta v} \right) \right. \\ &\quad \left. + \frac{1}{2} H(\mathbf{P}_{\alpha v}) + \frac{1}{2} H(\mathbf{P}_{\beta v}) \right] \end{aligned} \quad (5)$$

198 for  $\beta \in E(u, v)$ . With node  $u$ 's total incident hyperedge weight  
 199 in layer  $\alpha$

$$S_u^\alpha = \sum_{\beta \in E(u)} D_u^{\alpha\beta}, \quad (6)$$

200 the hyperedge-similarity walk has the transition rates

$$P_{uv}^{\alpha\beta} = \frac{D_u^{\alpha\beta}}{S_u^\alpha} \frac{\gamma_\beta(v)}{\delta(\beta)} \text{ for } \beta \in E(u, v). \quad (7)$$

201 Because the transition rates at a node depend on the current  
 202 layer, the random walks generate non-Markovian dynamics that a  
 203 unipartite or bipartite network representation cannot capture.

204 To ensure ergodic node-visit rates, we derived an unrecorded  
 205 teleportation scheme that leaves the node-visit rates unchanged  
 206 when teleportation is superfluous for hypergraphs with hyperedge-  
 207 independent node weights, robust to changes in the teleportation  
 208 rate when teleportation is needed<sup>26</sup>, and independent of the repre-  
 209 sentation (see Methods).

210 **Mapping flows on hypergraphs.** To identify flow-based commu-  
 211 nities or modules in hypergraphs, we seek to compress a modular  
 212 description of random walks on the network representations guided  
 213 by their links. We cast the problem of finding flow-based commu-  
 214 nities in hypergraphs as a minimum-description length problem  
 215 with the map equation framework<sup>3</sup>. With this compression-based  
 216 framework, we can compare how much the different representa-  
 217 tions compress modular flows.

218 When detecting communities, the representation matters be-  
 219 cause bipartite, unipartite, and multilayer networks provide the  
 220 community-detection algorithm Infomap with different degrees of  
 221 freedom<sup>27</sup>. Infomap assigns only nodes to communities in a uni-  
 222 partite network, but also hyperedge nodes in a bipartite network.  
 223 The multilayer network, with a state node for each hyperedge a

Table I. Optimal flow-based communities of the schematic hypergraph in Fig. 1 represented with different networks. The number of nodes includes state nodes for the multilevel representations and the bipartite non-lazy representation. We measure the overlap as the perplexity of the optimal solutions (see Methods).

Representation	Nodes	Links	Modules	Codelength (bits)	Overlap
<i>Lazy</i>					
Bipartite	15	32	2	2.90	–
Unipartite	10	40	3	2.35	–
Multilayer	16	98	3	2.35	1.00
Multilayer h-s <sup>a</sup>	16	98	4	2.28	1.09
<i>Non-lazy</i>					
Bipartite	26	52	2	3.00	–
Unipartite	10	30	3	2.63	–
Multilayer	16	68	3	2.62	1.10
Multilayer h-s <sup>a</sup>	16	68	4	2.32	1.29

<sup>a</sup> hyperedge-similarity

node belongs to, implies even more node assignments and possibly overlapping communities.

When mapping flows modelled by lazy and non-lazy random walks on the schematic network in Fig. 1, the optimal partitions of the bipartite networks have two communities, whereas the unipartite and multilayer networks have three communities (Table I and Fig. 3). The bipartite network favours fewer modules – using the optimal three-module partition of the unipartite network on the bipartite network gives code length 3.29 bits instead of 2.90 bits for two modules — because the random walker transitions more frequently between modules when they include hyperedges: Even if a hyperedge node contains no flows at the end of each two-step walk from node through hyperedge node to node, assigning it to a module costs extra bits when it has nodes in multiple modules. For example, if nodes *a*, *b*, and *c* in the bipartite network in Fig. 1(b) would belong to a third green module as in the optimal unipartite solution, and the random walker at node *c* would return to the hyperedge it comes from before revisiting node *c*, it would first need to exit the green module and enter the orange module, then exit the orange module and enter the green module. The corresponding walk on the unipartite network stays within the green module. As a result, the unipartite network representation favours more, smaller modules than the bipartite network representation for lazy and non-lazy walks (Table I).

Multilayer networks enable further compression with overlapping modules. But for this small network, only non-lazy walks give overlapping modules with 0.01 bits compression gain (Table I). With walks that preferentially move to similar hyperedges, the optimal partitions of the multilayer hyperedge-similarity network representations for lazy and non-lazy random walks both have more overlap in four modules (Table I and Fig. 3). The hyperedge-similarity walks favour these overlapping modules because they stay longer within them than the regular walks.

For a given random-walk model, the representations give equivalent node-visit rates but alter the link flows. And with different link flows, the optimal partition can change. The bipartite network

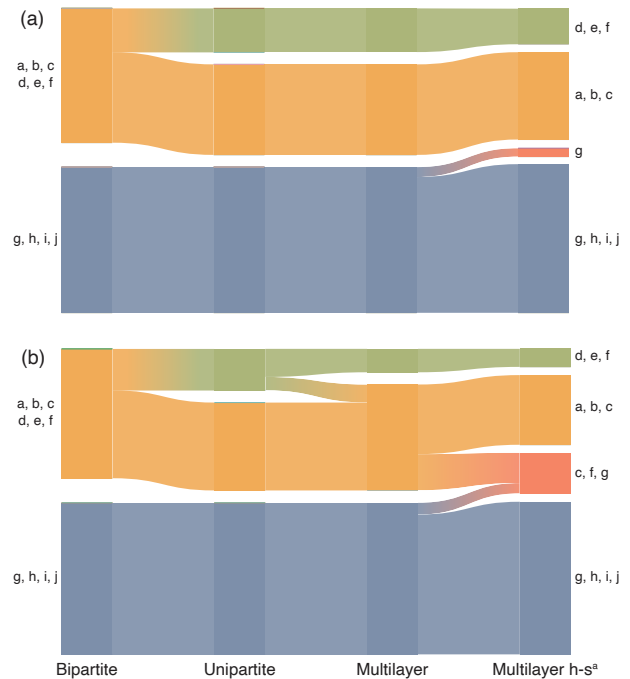


Fig. 3. Alluvial diagrams of optimal partitions for the schematic hypergraph in Fig. 1. (a) Optimal partitions for lazy walks represented with the networks in Fig. 1(b-d). (b) Optimal partitions for non-lazy walks.

representation favours partitions with fewer modules than the unipartite network representation because hyperedge nodes assigned to modules implies encoding more transitions between modules. Multilayer representations, especially with walks that spend longer time among similar hyperedges, favour more overlapping modules. The random-walk model determines how much the multilayer network modules overlap. Non-lazy and hyper-edge similarity walks favour overlap because they lead to longer persistence times among nodes in possibly overlapping groups.

**Experiments.** To illustrate how the network representation affects detected communities in real hypergraphs, we generated a collaboration hypergraph from the 734 references in *Networks beyond pairwise interactions: Structure and dynamics* by F. Battiston et al.<sup>8</sup> We modelled the referenced articles as hyperedges and their authors as nodes. Authors with multiple articles form connections between the hyperedges. We analysed the largest connected component with  $|V| = 361$  author nodes in  $|E| = 220$  hyperedges. The median number of authors in a hyperedge is 3, and the authors have contributed to 2.2 articles on average though most have only contributed to one.

We assigned the relative importance of references by their number of citations  $c$  in December 2020. Some references had no citations and some were highly cited. One such example is *Diffusion of innovations* by Everett M. Rogers, with more than 120,000 citations. To avoid disproportionately large or small hyperedge weights  $\omega(e)$ , we weighed the edges by the logarithm of the number of citations and added unit constants to avoid the zero citation

Table II. Optimised flow-based multilevel communities of the collaboration hypergraph represented with different networks. The number of nodes includes state nodes for the multilevel representations and the bipartite non-lazy representation. Shortest codelength of 100 trials with the variance in parenthesis. We measure the overlap as the perplexity of the optimised solutions (see Methods).

Representation	Nodes	Links	Modules				Codelength (bits)
			Top	Leaf	Levels	Overlap	
<i>Lazy</i>							
Bipartite	581	1,560	4	23	3	–	5.178(1)
Unipartite	361	2,607	9	69	4	–	3.82557(2)
Multilayer	780	17,193	9	76	4	1.003	3.82730(2)
Multilayer h-s <sup>a</sup>	780	17,193	8	90	4	1.127	3.54939(3)
<i>Non-lazy</i>							
Bipartite	1,141	3,548	5	25	3	–	5.1733(2)
Unipartite	361	2,246	7	49	4	–	4.25104(8)
Multilayer	780	12,843	7	54	4	1.098	4.16349(8)
Multilayer h-s <sup>a</sup>	780	12,843	9	66	4	1.181	3.70432(1)

<sup>a</sup> hyperedge-similarity

287 problem,

$$\omega(e) = \ln(c + 1) + 1. \quad (8)$$

288 We modelled the authors' different contributions to articles by  
289 assigning higher weights to the first and last author<sup>20</sup>. We used  
290 the edge-dependent node weights

$$\gamma_e(v) = \begin{cases} 2 & \text{if node } v \text{ is first or last author,} \\ 1 & \text{otherwise.} \end{cases} \quad (9)$$

291 We assumed equal contribution for alphabetically sorted authors,  
292 and assigned all of them weight  $\gamma(v) = 1$ . This model ranks  
293 co-corresponding author's contributions lower than corresponding  
294 authors.

295 To study how hypergraph representations and random walk  
296 models affect the community structure, we generated bipartite,  
297 unipartite, and multilayer representations for lazy and non-lazy  
298 random walks on the collaboration network. We identified nested  
299 hierarchical partitions in each network with Infomap, using 100  
300 independent searches for each network. Infomap's running time  
301 depends on the number of nodes, links, and solution levels: The  
302 bipartite and unipartite representations finished 3–7 times faster  
303 than the multilayer representations. The non-lazy bipartite repre-  
304 sentation with many state nodes ran almost as long.

305 The optimised partitions for the lazy and non-lazy representa-  
306 tions behave like the schematic example: The bipartite representa-  
307 tions have the fewest leaf modules and highest codelengths, and  
308 the multilayer hyperedge-similarity representations have the most  
309 leaf modules and shortest codelengths with the unipartite and the  
310 regular multilayer representations in between (Table II). Except  
311 for the non-lazy bipartite representation with its many state nodes,  
312 the lazy representations have more leaf modules and shorter code  
313 lengths than their corresponding non-lazy representations because  
314 the lazy random walk is more confined than the non-lazy random  
315 walk.

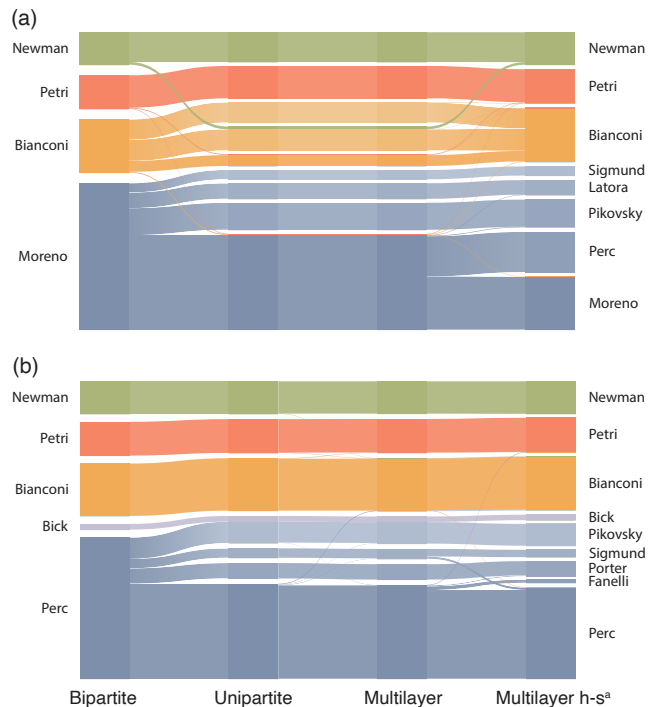


Fig. 4. Alluvial diagrams of optimised partitions for different representa-  
tions of the collaboration hypergraph. Lazy walks in (a) and non-lazy walks  
in (b). Module names from the top-ranked author within each module.

316 With more nodes than in the schematic example, the solutions  
317 have more depth. The bipartite solutions have three, and the uni-  
318 partite and multilayer solutions have four hierarchical levels. The  
319 unipartite and multilayer solutions also have more top modules.  
320 With non-lazy dynamics, they split the largest top module, and  
321 in the lazy dynamics, they split the two largest top modules. But  
322 the second-largest top module reunites in the hyperedge-similarity  
323 representation with stronger connections between similar hyper-  
324 edges (Fig. 4 and Fig. 7 in Appendix A). The unipartite and  
325 multilayer solutions are also most similar at the leaf level (Fig. 8  
326 in Appendix A).

327 In this larger example, the multilayer hyperedge-similarity rep-  
328 resentations give more overlap. The non-lazy representations  
329 result in higher average overlap because random walkers visit-  
330 ing a node must continue to other nodes, often in the same or a  
331 similar hyperedge layer. When random walkers from dissimilar  
332 hyperedges come together at a node, they tend to return to where  
333 they came from and favour overlapping modules. The non-lazy  
334 representations also result in higher max overlap with the same  
335 authors topping all representations (Fig. 5).

336 In line with the information-theoretic duality between finding  
337 regularities in data and compressing those data, representations  
338 that enable deeper solutions with more modules have shorter  
339 codelengths (Table II). The lazy multilayer representation is an  
340 exception. Its optimised codelength is bound above by the lazy  
341 unipartite representation's codelength – they have the same code-  
342 length for the same hard partition – and overlapping modules can  
343 potentially reduce the codelength. Infomap's best codelength was

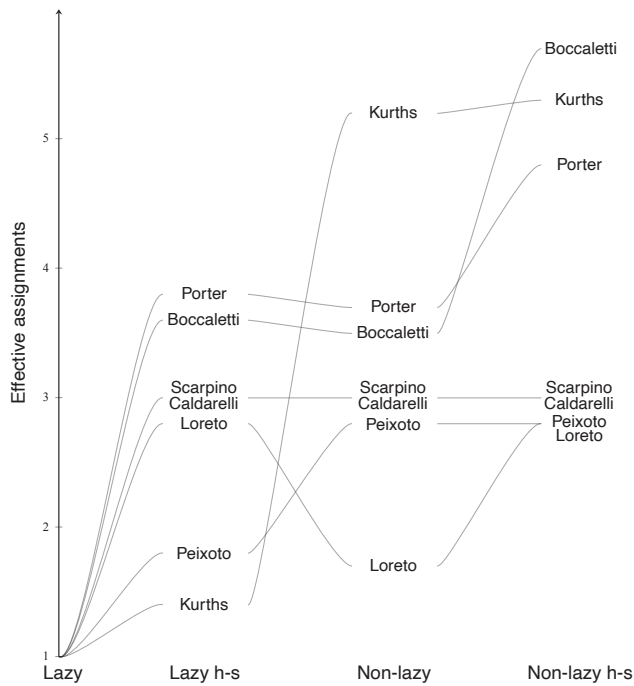


Fig. 5. Authors in the collaboration hypergraph with the highest average effective number of assignments in the lazy and non-lazy multilayer representations (see Methods).

344 instead 0.05 percent longer than for the lazy unipartite representa-  
 345 tion. Multilayer representations with their many state nodes and  
 346 links aggravate the search problem, and Infomap could not find a  
 347 better solution with 100 attempts. But the gain from overlapping  
 348 modules is higher for the non-lazy multilayer representation and  
 349 Infomap finds a solution with a significantly shorter codelength.

350 **A case study on fossil data.** Palaeontologists classify major  
 351 groups of marine animals archived in the fossil record into global-  
 352 scale faunas that change over time<sup>29</sup>. They have used different  
 353 network representations to understand the macroevolutionary pat-  
 354 tern of marine biodiversity<sup>30,31</sup>. However, it is still unclear how  
 355 such an organisation of marine animals into modules represent-  
 356 ing global faunas changes with random-walk model and network  
 357 representation. To illustrate how the network representation of  
 358 the underlying paleontological data affects empirical estimates of  
 359 this macroevolutionary pattern, we generated a hypergraph from  
 360 genus-level fossil occurrences presented in ref. 30 and retrieved  
 361 from the PaleoDB<sup>32</sup>. We restricted our analysis to fossil occur-  
 362 rences from the Cambrian (541 MY) to the Cretaceous period (66  
 363 MY) and modelled 77 geological stages as hyperedges and 13,276  
 364 genera as nodes. Genera occurring in multiple geological stages  
 365 form connections between hyperedges. We weighted the hyper-  
 366 edges by dividing the number of samples where a genus occurs in  
 367 a given geological stage by the total number of samples recorded  
 368 at the stage, a procedure modified from ref. 33. We generated bi-  
 369 partite, unipartite, and multilayer network representations for lazy  
 370 and non-lazy random walks from the underlying palaeontology  
 371 data and identified optimised partitions in the assembled networks

372 using Infomap.

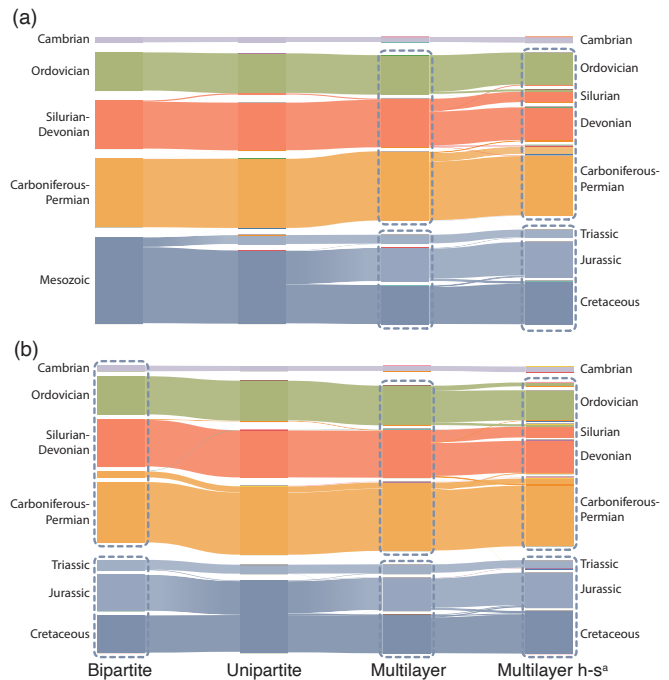


Fig. 6. Alluvial diagrams of optimised partitions for the fossil hypergraph represented with different networks. Lazy walks in (a) and non-lazy walks in (b). We show top modules when a partition lacks deeper levels and leaf modules marked with dashed lines when they exist. Module names follow the geological period or era represented by the fauna assemblage.

373 For lazy random walks, Infomap partitioned only the multilayer  
 374 representations into multilevel communities: three modules at the  
 375 first hierarchical level [Fig. 6(a)]. Similar to the schematic example  
 376 and the collaboration hypergraph, the bipartite representation for  
 377 the lazy random walks has the fewest leaf modules and the highest  
 378 codelength. The multilayer hyperedge-similarity representation  
 379 has the most leaf modules and the shortest codelength (Table III).

380 For non-lazy random walks, Infomap partitioned the bipartite  
 381 representation into a multilevel solution with shorter codelength  
 382 than the unipartite representation and the standard multilevel rep-  
 383 resentation [Fig. 6(b)]. The multilayer hyperedge-similarity repre-  
 384 sentation once more provides the most leaf modules and highest  
 385 overlap.

386 The multilayer network representations, including lazy and  
 387 non-lazy random walks, reproduce modules reminiscent of the  
 388 Cambrian, Paleozoic, and modern evolutionary faunas widely  
 389 used in macroevolutionary research<sup>29</sup>. Also, leaf modules in the  
 390 multilayer representations capture subfaunas from specific geolog-  
 391 ical periods as nested modules such as Silurian, Triassic, Jurassic,  
 392 and Cretaceous. Infomap applied to the bipartite representation of  
 393 the non-lazy random walks identified similar subfaunas but com-  
 394 bined Cambrian and Paleozoic faunas into a single top module,  
 395 obscuring the large-scale pattern. Overall, our results indicate  
 396 some advantages of using multilayer over bipartite and unipartite  
 397 representations of fossil occurrence data to quantify the marine  
 398 biodiversity's macroevolutionary patterns, with lazy and non-lazy  
 399 random walks providing similar solutions.

Table III. Optimised flow-based multilevel communities of the fossil hypergraph represented with different networks. The number of nodes includes state nodes for the multilevel representations and the bipartite non-lazy representation. The number of non-trivial top and leaf modules. Average number of levels weighted by the flow volume. We measure the overlap as the perplexity of the optimised solutions (see Methods). Shortest codelength of 20 trials with the variance in parenthesis.

Representation	Nodes ( $\times 10^3$ )	Links ( $\times 10^3$ )	Modules			Overlap	Codelength (bits)	Time (hh:mm:ss)
			Top	Leaf	Levels			
<i>Lazy</i>								
Bipartite	13	79	5	8	2.02	–	10.50927(5)	00:00:06
Unipartite	13	16,155	6	13	2.02	–	10.3953503(1)	00:13:24
Multilayer	40	174,490	3	17	3.00	1.011	10.39819(1)	09:08:43
Multilayer h-s <sup>a</sup>	40	174,490	3	19	3.28	1.135	9.84170(1)	14:19:39
<i>Non-lazy</i>								
Bipartite	53	25,937	2	15	3.02	–	10.34889(3)	01:14:25
Unipartite	13	16,141	6	12	2.02	–	10.4031798(6)	00:13:04
Multilayer	40	174,209	3	15	3.00	1.010	10.406141(9)	08:55:03
Multilayer h-s <sup>a</sup>	40	174,209	3	16	3.00	1.135	9.84912(1)	13:23:13

<sup>a</sup> hyperedge-similarity

## 400 Conclusions

401 We have derived unipartite, bipartite, and multilayer network rep-  
402 resentations of hypergraph flows with different advantages. We  
403 used the information-theoretic and flow-based community detec-  
404 tion method Infomap to explore how different hypergraph random  
405 walk models and network representation change the number, size,  
406 depth, and overlap of identified multilevel communities. By identi-  
407 fying flow-based communities in a schematic and real hypergraphs  
408 – a small collaboration hypergraph of researchers working on net-  
409 works beyond pairwise interactions and a large faunal hypergraph  
410 of sampled species across geological stages – we found that the  
411 bipartite network representation is most compact and enables the  
412 fastest community detection. A multilayer network representation  
413 that reinforces flows within similar layers – one for each hyperedge  
414 – gave the deepest modular structures with most module overlap.  
415 But the modular detection gain comes at a high computational cost:  
416 Combining fully connected layers with other layers requires many  
417 more nodes and links than the bipartite network representation. If  
418 the research question does not require hyperedge assignments or  
419 overlapping modules, the unipartite network representation pro-  
420 vides a trade-off with intermediate compactness, speed, and ability  
421 to reveal modular regularities. Among the random-walk models,  
422 lazy walks typically give more modules in deeper nested struc-  
423 tures, and non-lazy walks higher modular overlap. Our methods  
424 and results help researchers model and map flows on hypergraphs  
425 to study the effects of multibody interactions in complex systems.

## 426 Methods

427 **Unrecorded teleportation.** With hyperedge-independent node  
428 weights where  $\gamma_e(u) = \gamma(u)$  for all hyperedges  $e \in E(u)$ , undi-  
429 rected weighted networks can represent the dynamics, and the  
430 stationary distribution of the random walk  $\pi_u$  is proportional to  
431 the product of node  $u$ 's total incident hyperedge weight  $d(u)$  and  
432 edge weight  $\gamma(u)$ . With normalised node-visit rates<sup>20</sup>,

$$\pi_u = \frac{d(u)\gamma(u)}{\sum_{v \in V} d(v)\gamma(v)}. \quad (10)$$

433 For the multilayer network representation, the node-visit rates  
434 split between layers based on the node  $u$ 's incident hyperedge  
435 weight per layer state node

$$\pi_u^\alpha = \frac{\omega(\alpha)\gamma(u)}{\sum_{v \in V} d(v)\gamma(v)}. \quad (11)$$

436 With hyperedge-dependent node weights  $\gamma_e(u)$ , only directed  
437 weighted networks can represent the dynamics. We use random  
438 teleportation to ensure ergodic walks when deriving the node-visit  
439 rates with the power-iteration method. Unrecorded teleportation  
440 to links minimises the distortion<sup>26</sup>: In each iteration of the power-  
441 iteration method, we distribute a fraction  $\tau = 0.15$  of each node's  
442 flow volume among all nodes proportional to their out-link weights.  
443 The remaining flow volume moves on the links proportional to  
444 their weights. In the last iteration, we move all flows on the  
445 links proportional to their weights and record all flows on links  
446 and nodes to obtain the ergodic node- and link-visit rates with  
447 unrecorded teleportation. This procedure gives equivalent visit  
448 rates as simulating a random walker that only records moves on  
449 links: With probability  $1 - \tau$ , the random walker moves to a node  
450 by following the links proportional to their weights and records the  
451 link and the target node. With probability  $\tau$ , the random walker  
452 teleports without recording to the link's source node proportional  
453 to the link weight. The normalised number of recordings of each  
454 node and link gives the visit rates.

455 We want teleportation applied to undirected networks – where it  
456 is unnecessary – to leave the node- and link-visit rates unchanged.  
457 We achieve this smooth teleportation by scaling the transition rates  
458 from nodes by the node-visit rates: Then unrecorded teleportation  
459 proportional to nodes' total out-link weights followed by recorded  
460 moves on the links proportional to their weights distribute on the



461 nodes according to the ergodic visit rates on undirected networks<sup>26</sup>.  
 462 For the general case when the node weights can depend on the  
 463 hyperedge, and the network may be directed, we use Eq. 10 without  
 464 assuming  $\gamma_e(u) = \gamma(u)$  as an approximation of the node-visit  
 465 rates:

$$\tilde{\pi}_u = \frac{\sum_{e \in E(u)} \omega(e) \gamma_e(u)}{\sum_{v \in V, e \in E(v)} \omega(e) \gamma_e(v)} \quad (12)$$

466 for nodes and

$$\tilde{\pi}_u^\alpha = \frac{\omega(\alpha) \gamma_\alpha(u)}{\sum_{v \in V, e \in E(v)} \omega(e) \gamma_e(v)} \text{ for } \alpha \in E(u) \quad (13)$$

467 for state nodes. With exact node-visit rates, we would obtain the  
 468 stationary flow volumes on links by multiplying the transition rates  
 469 by the source nodes' visit rates. With approximate node-visit rates,  
 470 instead, we obtain the link weights

$$w_{ue} = \tilde{\pi}_u P_{ue} \quad (14)$$

471 for bipartite networks,

$$w_{uv} = \tilde{\pi}_u P_{uv} \quad (15)$$

472 for unipartite networks, and

$$w_{uv}^{\alpha\beta} = \tilde{\pi}_u^\alpha P_{uv}^{\alpha\beta} \text{ for } \beta \in E(u, v) \quad (16)$$

473 for multilayer networks. With unrecorded teleportation propor-  
 474 tional to these link weights, modelling flows on hypergraphs give  
 475 node-visit rates robust to changes in the teleportation rate and  
 476 independent of the representation.

477 **Overlap metric.** Modules overlap when Infomap assigns a node's  
 478 state nodes in the multilayer network representations to different  
 479 modules. Measuring the overlap through the absolute number of  
 480 assignments is misleading because the overlap is 2 regardless of  
 481 the number of state nodes assigned to a different module than the  
 482 rest. Instead, we used the effective number of assignments. If a  
 483 fraction  $f$  of node  $u$ 's state nodes is assigned to the  $m$ th module in  
 484  $u$ 's module assignment set, the  $m$ th element of  $u$ 's assignment vec-  
 485 tor is  $a_m^u = f$  and the effective number of assignments measured  
 486 by the perplexity of  $u$ 's module assignments is

$$o_u = 2^{H(\mathbf{a}^u)}. \quad (17)$$

487 The effective number of assignments is one if all  $u$ 's state nodes are  
 488 in one module, and it is equal to the number of assignments when  
 489 the state nodes are divided evenly among  $u$ 's module assignments.  
 490 We averaged over all nodes for the partition overlap.

#### 491 Data and code availability

492 All data and source code are available on GitHub: <http://github.com/mapequation/mapping-hypergraphs>.

## 494 References

- 495 1. Brin, S. & Page, L. The anatomy of a large-scale hypertextual web  
 496 search engine. *Comput. Networks ISDN* **30**, 107–117 (1998).
- 497 2. Simonsen, I., Eriksen, K. A., Maslov, S. & Sneppen, K. Diffusion  
 498 on complex networks: a way to probe their large-scale topological  
 499 structures. *Physica A* **336**, 163–173 (2004).
- 500 3. Rosvall, M. & Bergstrom, C. T. Maps of random walks on complex  
 501 networks reveal community structure. *Proceedings of the National  
 502 Academy of Sciences* **105**, 1118–1123 (2008).
- 503 4. Delvenne, J., Yaliraki, S. & Barahona, M. Stability of graph  
 504 communities across time scales. *Proc. Natl. Acad. Sci. USA* **107**,  
 505 12755–12760 (2010).
- 506 5. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U.  
 507 Complex networks: Structure and dynamics. *Physics reports* **424**,  
 508 175–308 (2006).
- 509 6. Fortunato, S. Community detection in graphs. *Physics reports* **486**,  
 510 75–174 (2010).
- 511 7. Lambiotte, R., Rosvall, M. & Scholtes, I. From networks to optimal  
 512 higher-order models of complex systems. *Nature physics* **15**, 313–320  
 513 (2019).
- 514 8. Battiston, F. *et al.* Networks beyond pairwise interactions: structure  
 515 and dynamics. *Physics Reports* (2020).
- 516 9. Mucha, P. J., Richardson, T., Macon, K., Porter, M. A. & Onnela, J.-P.  
 517 Community structure in time-dependent, multiscale, and multiplex  
 518 networks. *science* **328**, 876–878 (2010).
- 519 10. Kivela, M. *et al.* Multilayer networks. *Journal of complex networks*  
 520 **2**, 203–271 (2014).
- 521 11. De Domenico, M., Granell, C., Porter, M. A. & Arenas, A. The  
 522 physics of spreading processes in multilayer networks. *Nature Physics*  
 523 **12**, 901–906 (2016).
- 524 12. Rosvall, M., Esquivel, A. V., Lancichinetti, A., West, J. D. &  
 525 Lambiotte, R. Memory in network flows and its effects on spreading  
 526 dynamics and community detection. *Nature communications* **5**, 1–13  
 527 (2014).
- 528 13. Scholtes, I. *et al.* Causality-driven slow-down and speed-up of diffu-  
 529 sion in non-markovian temporal networks. *Nature communications*  
 530 **5**, 1–9 (2014).
- 531 14. Xu, J., Wickramaratne, T. L. & Chawla, N. V. Representing higher-  
 532 order dependencies in networks. *Science Advances* **2**, e1600028  
 533 (2016).
- 534 15. Parzanchevski, O. & Rosenthal, R. Simplicial complexes: spectrum,  
 535 homology and random walks. *Random Structures & Algorithms* **50**,  
 536 225–261 (2017).
- 537 16. Salnikov, V., Cassese, D. & Lambiotte, R. Simplicial complexes and  
 538 complex systems. *European Journal of Physics* **40**, 014001 (2018).
- 539 17. Iacopini, I., Petri, G., Barrat, A. & Latora, V. Simplicial models of  
 540 social contagion. *Nature communications* **10**, 1–9 (2019).
- 541 18. Schaub, M. T., Benson, A. R., Horn, P., Lippner, G. & Jadbabaie, A.  
 542 Random walks on simplicial complexes and the normalized hodge  
 543 1-laplacian. *SIAM Review* **62**, 353–391 (2020).
- 544 19. Zhou, D., Huang, J. & Schölkopf, B. Learning with hypergraphs:  
 545 Clustering, classification, and embedding. In *Advances in neural  
 546 information processing systems*, 1601–1608 (2007).
- 547 20. Chitra, U. & Raphael, B. J. Random walks on hypergraphs with  
 548 edge-dependent vertex weights. In *36th International Conference on  
 549 Machine Learning, ICML 2019, 2002–2011* (International Machine  
 550 Learning Society (IMLS), 2019).
- 551 21. Carletti, T., Battiston, F., Cencetti, G. & Fanelli, D. Random walks  
 552 on hypergraphs. *Physical Review E* **101**, 022308 (2020).
- 553 22. Carletti, T., Fanelli, D. & Lambiotte, R. Random walks and  
 554 community detection in hypergraphs. *arXiv:2010.14355* (2020).
- 555 23. De Domenico, M., Lancichinetti, A., Arenas, A. & Rosvall, M.  
 556 Identifying modular flows on multilayer networks reveals highly  
 557 overlapping organization in interconnected systems. *Physical Review*

- 558 X **5**, 011027 (2015).
- 559 24. Jeub, L. G., Mahoney, M. W., Mucha, P. J., Porter, M. A. *et al.*  
 560 A local perspective on community structure in multilayer networks.  
 561 *Network Science* **5**, 144–163 (2017).
- 562 25. Alon, N., Benjamini, I., Lubetzky, E. & Sodin, S. Non-backtracking  
 563 random walks mix faster. *Communications in Contemporary Mathe-*  
 564 *matics* **9**, 585–603 (2007).
- 565 26. Lambiotte, R. & Rosvall, M. Ranking and clustering of nodes in  
 566 networks with smart teleportation. *Phys. Rev. E* **85**, 056107 (2012).
- 567 27. Edler, D., Bohlin, L. *et al.* Mapping higher-order network flows in  
 568 memory and multilayer networks with infomap. *Algorithms* **10**, 112  
 569 (2017).
- 570 28. Kheirkhahzadeh, M., Lancichinetti, A. & Rosvall, M. Efficient  
 571 community detection of network flows for varying markov times and  
 572 bipartite networks. *Physical Review E* **93**, 032309 (2016).
- 573 29. Sepkoski, J. J. A factor analytic description of the  
 574 Phanerozoic marine fossil record. *Paleobiology* **7**, 36–53  
 575 (1981). URL [https://www.cambridge.org/core/product/](https://www.cambridge.org/core/product/identifier/S0094837300003778/type/journal_article)  
 576 [identifier/S0094837300003778/type/journal\\_article](https://www.cambridge.org/core/product/identifier/S0094837300003778/type/journal_article).
- 577 30. Rojas, A., Calatayud, J., Kowalewski, M., Neuman, M. & Rosvall,  
 578 M. A multiscale view of the phanerozoic fossil record reveals the  
 579 three major biotic transitions. preprint, *Paleontology* (2019). URL  
 580 <http://biorxiv.org/lookup/doi/10.1101/866186>.
- 581 31. Muscente, A. D. *et al.* Quantifying ecological impacts of mass extinc-  
 582 tions with network analysis of fossil communities. *Proceedings of the*  
 583 *National Academy of Sciences* **115**, 5217–5222 (2018). URL [http://](http://www.pnas.org/lookup/doi/10.1073/pnas.1719976115)  
 584 [www.pnas.org/lookup/doi/10.1073/pnas.1719976115](http://www.pnas.org/lookup/doi/10.1073/pnas.1719976115).
- 585 32. Peters, S. E. & McClennen, M. The Paleobiology  
 586 Database application programming interface. *Paleobiology* **42**,  
 587 1–7 (2016). URL [http://www.journals.cambridge.org/](http://www.journals.cambridge.org/abstract_S0094837315000391)  
 588 [abstract\\_S0094837315000391](http://www.journals.cambridge.org/abstract_S0094837315000391).
- 589 33. Rojas, A., Patarroyo, P., Mao, L., Bengtson, P. &  
 590 Kowalewski, M. Global biogeography of Albian am-  
 591 monoids: A network-based approach. *Geology* **45**, 659–662  
 592 (2017). URL [https://pubs.geoscienceworld.org/geology/](https://pubs.geoscienceworld.org/geology/article/45/7/659-662/207876)  
 593 [article/45/7/659-662/207876](https://pubs.geoscienceworld.org/geology/article/45/7/659-662/207876).

#### 594 Acknowledgments

595 We thank Christopher Blöcker, Manlio De Domenico, Michael Schaub,  
 596 and Jelena Smiljanić for valuable comments that helped us improve the  
 597 manuscript. A.E was supported by the Swedish Foundation for Strategic  
 598 Research, Grant No. SB16-0089. A.R., D.E. and M.R. were supported  
 599 by the Swedish Research Council, Grant No. 2016-00796.

#### 600 Author contributions

601 A.E. and M.R. conceived the study. A.E., A.R. and D.E. performed the  
 602 numerical experiments and analysed the results. A.E. and M.R. wrote  
 603 the manuscript.

#### 604 Competing interests

605 The authors declare no competing interests.

606 A. Appendix

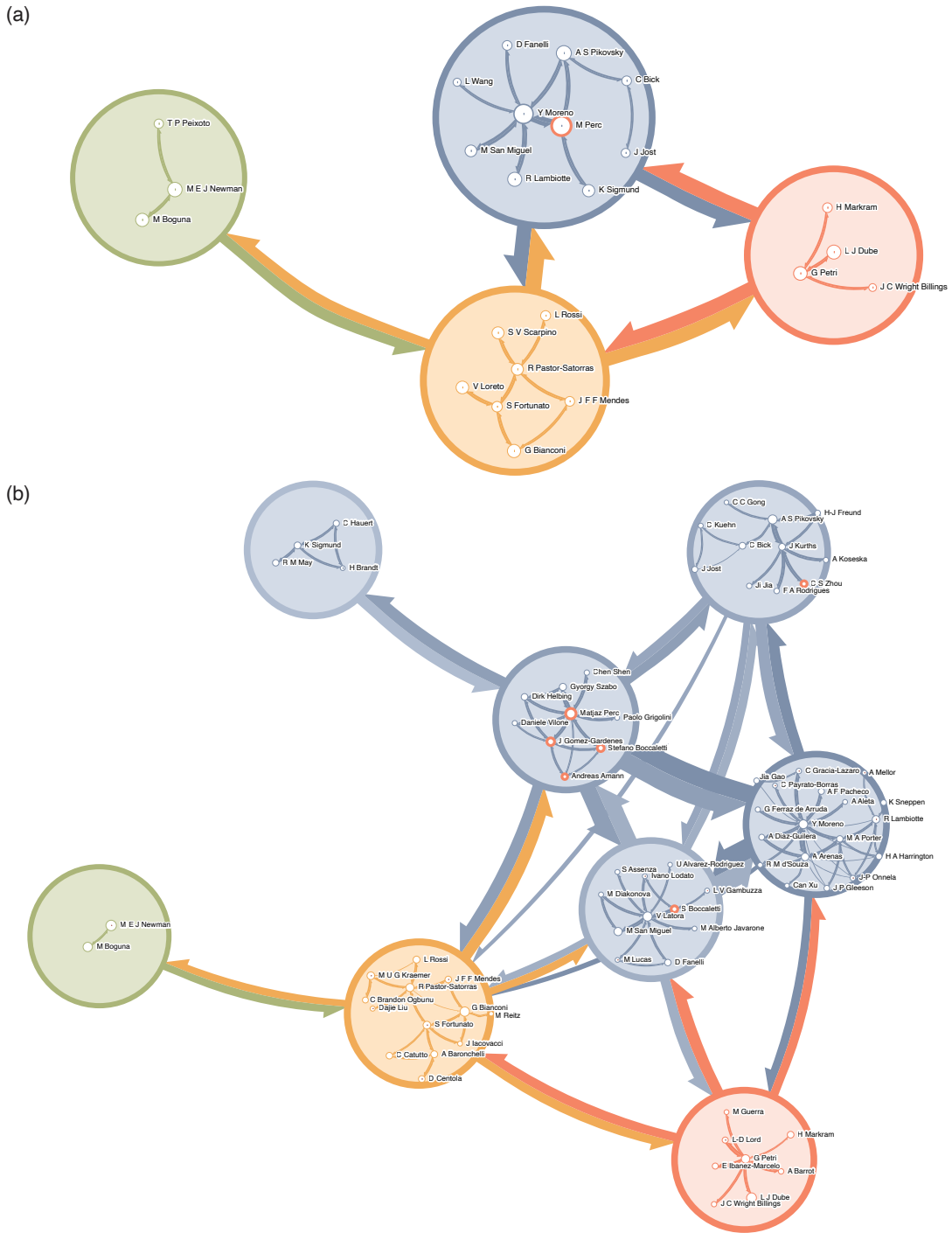


Fig. 7. Hierarchical maps of the collaboration hypergraph using (a) the bipartite representation and (b) the multilayer hyperedge-similarity representation. Module colours are the same as in Fig. 4(a). Aggregated inter-module links with sizes proportional to the exiting flow volume and length inversely proportional to the flow volume. White sub-modules are labelled with the top-ranked author. The largest blue top module in (a) contains ten sub-modules. In (b), the partition assigns those nodes to five top modules containing more sub-modules. S. Boccaletti, one of the most overlapping authors and highlighted in red, is assigned to one module in (a) and three top modules and six sub-modules in (b).

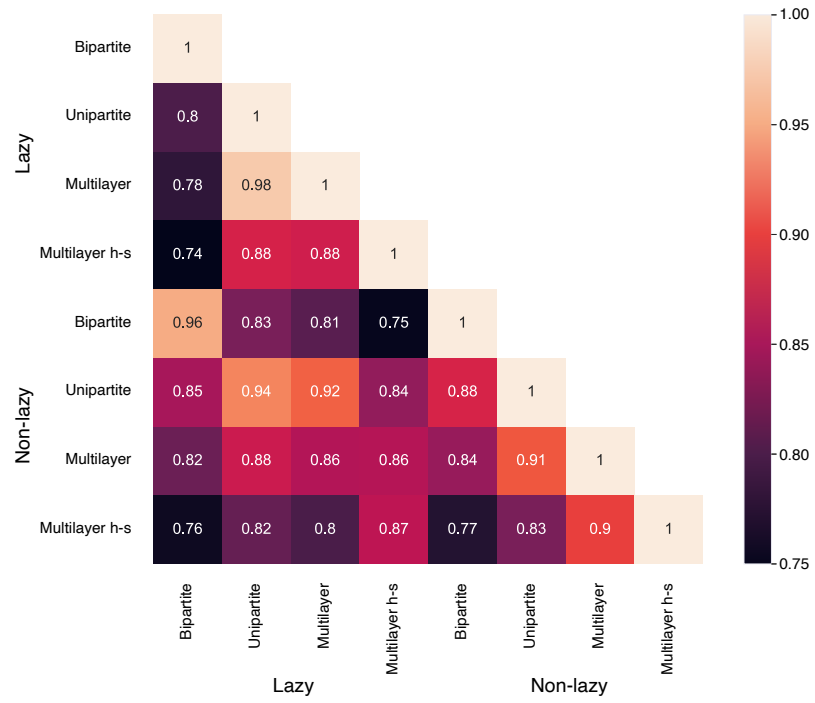
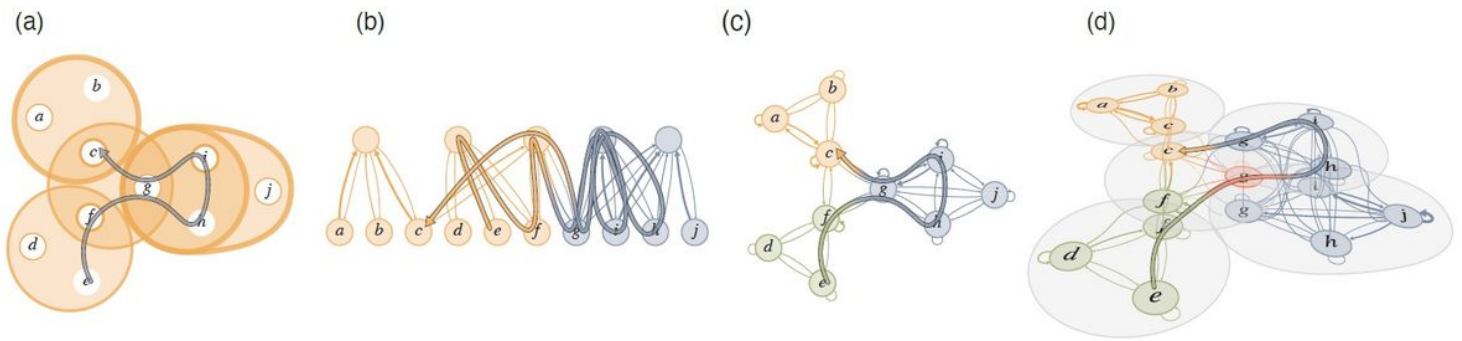


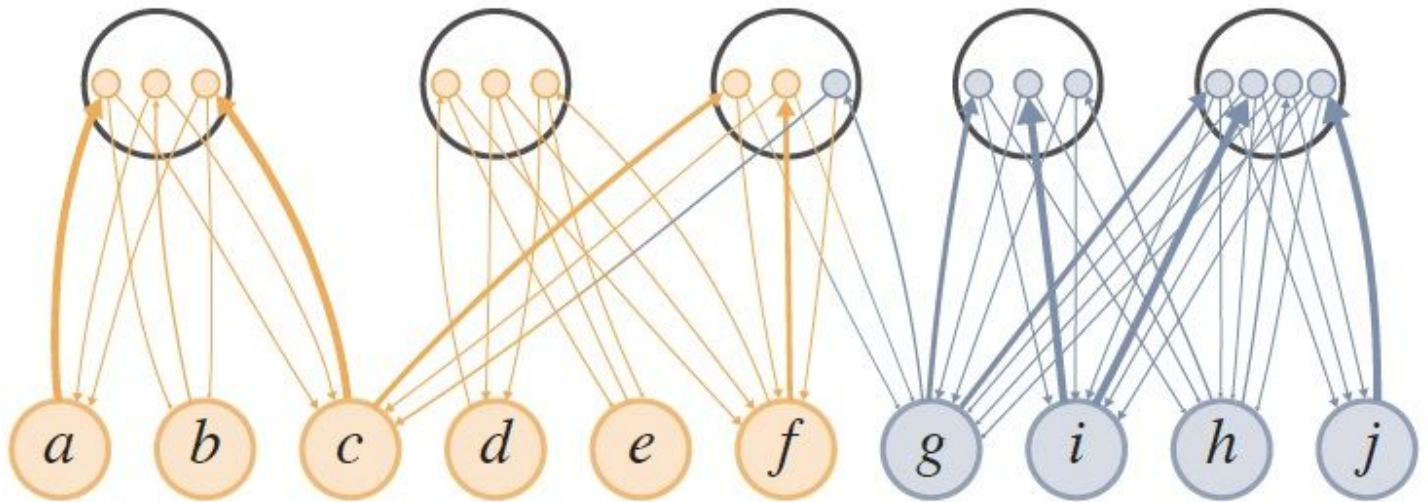
Fig. 8. Leaf module assignments' adjusted mutual information for different random walk dynamics and hypergraph representations. The bipartite representations differ the most from the other representations, and the unipartite and multilayer representations are most similar at the leaf level.

# Figures



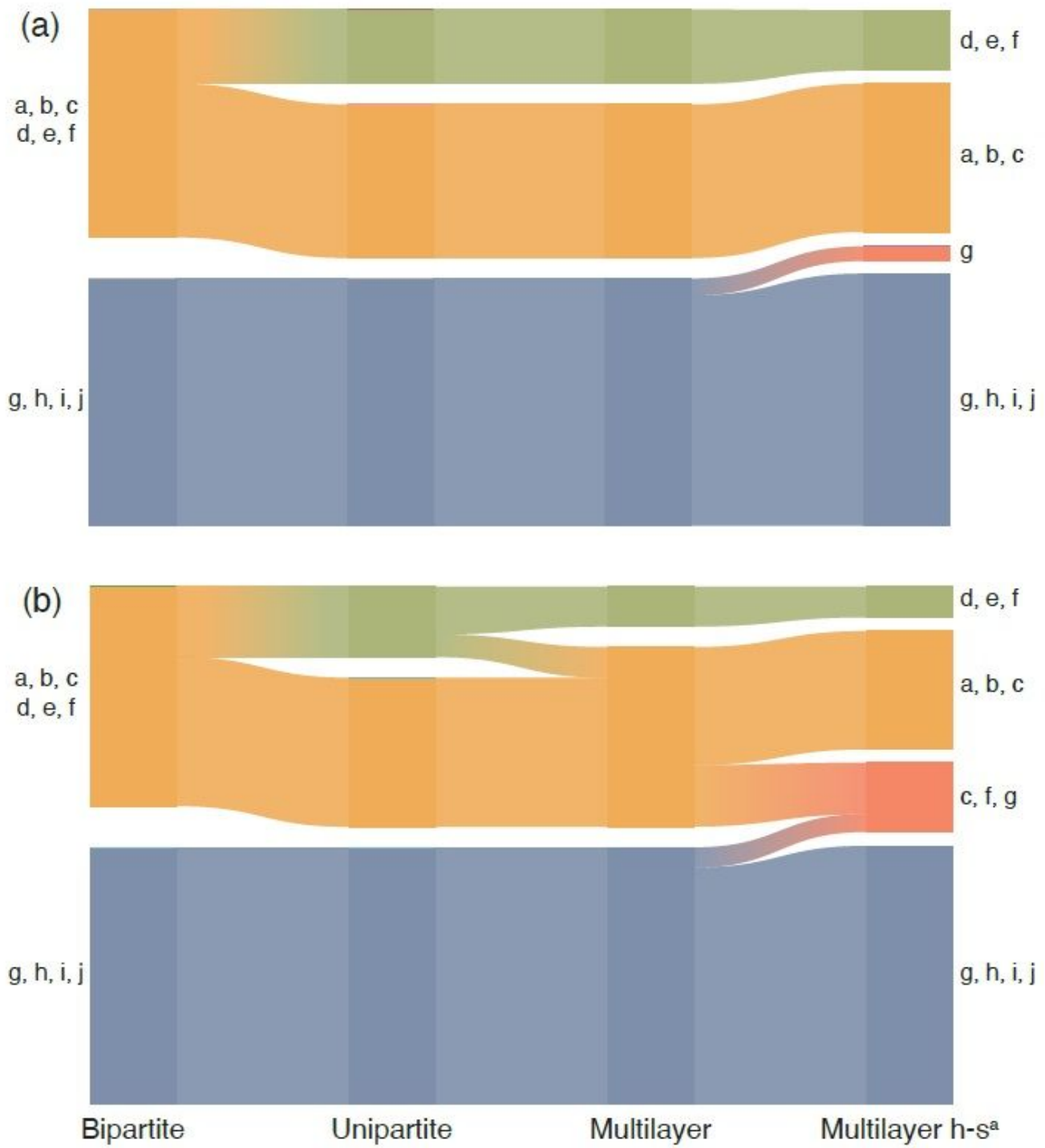
**Figure 1**

A schematic hypergraph represented with three types of networks. (a) The schematic hypergraph with weighted hyperedges and hyperedge-dependent node weights. Thin borders for weight 1 and thick borders for weight 3. A lazy random walk on the schematic hypergraph represented on: (b) a bipartite network, (c) a unipartite network, and (d) a multilevel network. The colours indicate optimised module assignments, in (d) for hyperedge-similarity walks.



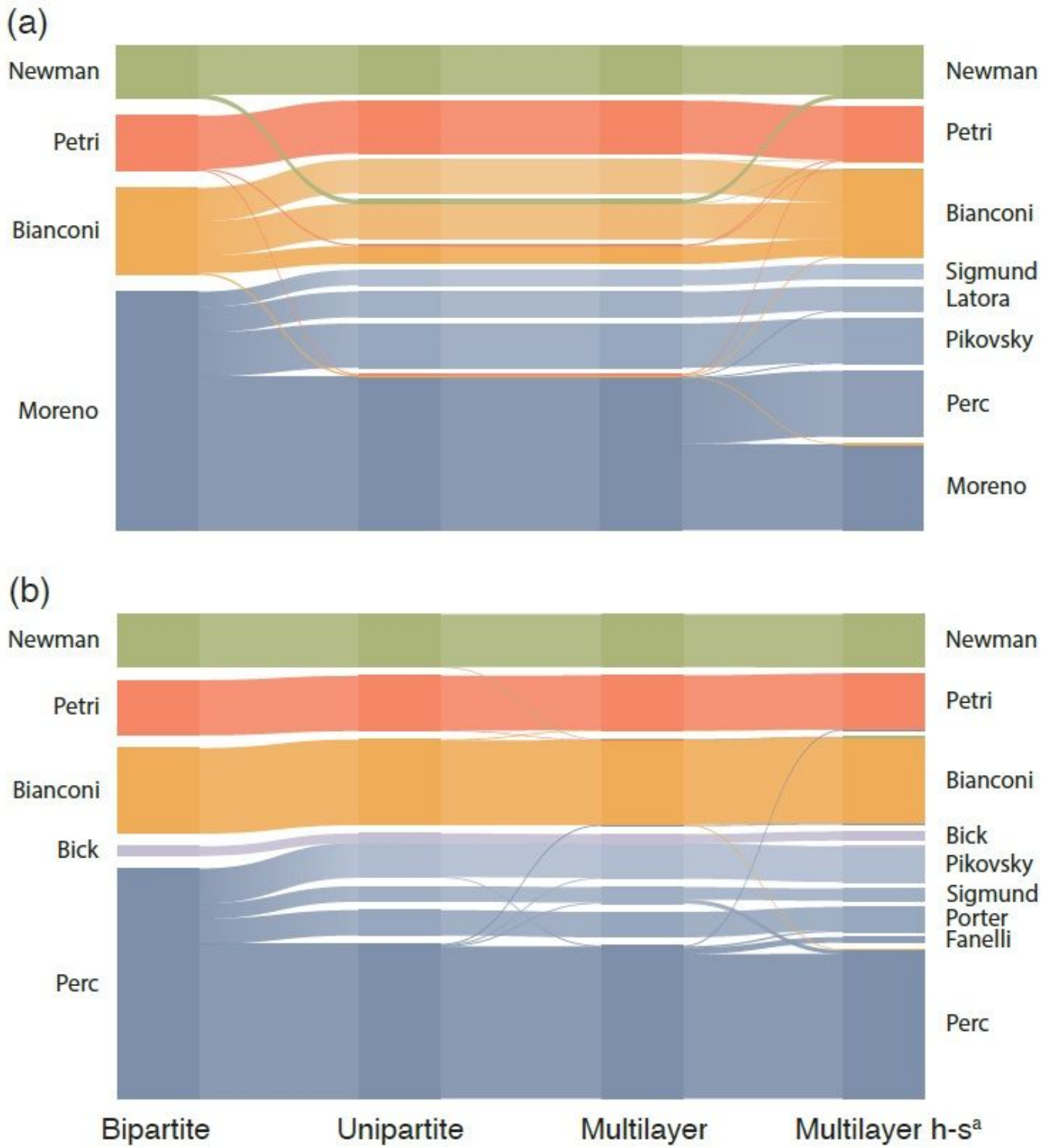
**Figure 2**

Bipartite network with state nodes for non-lazy random walks. To prevent random walks on bipartite networks to visit the same node at the bottom twice in a row by backtracking from the hyperedge node at the top, we use state nodes in the hyperedge nodes. Each hyperedge node requires one state node for each node in the hyperedge. The state nodes have one incoming link from its source node and outgoing links to all other nodes in the hyperedge. Colours indicate the optimized partition in Fig. 3(b).



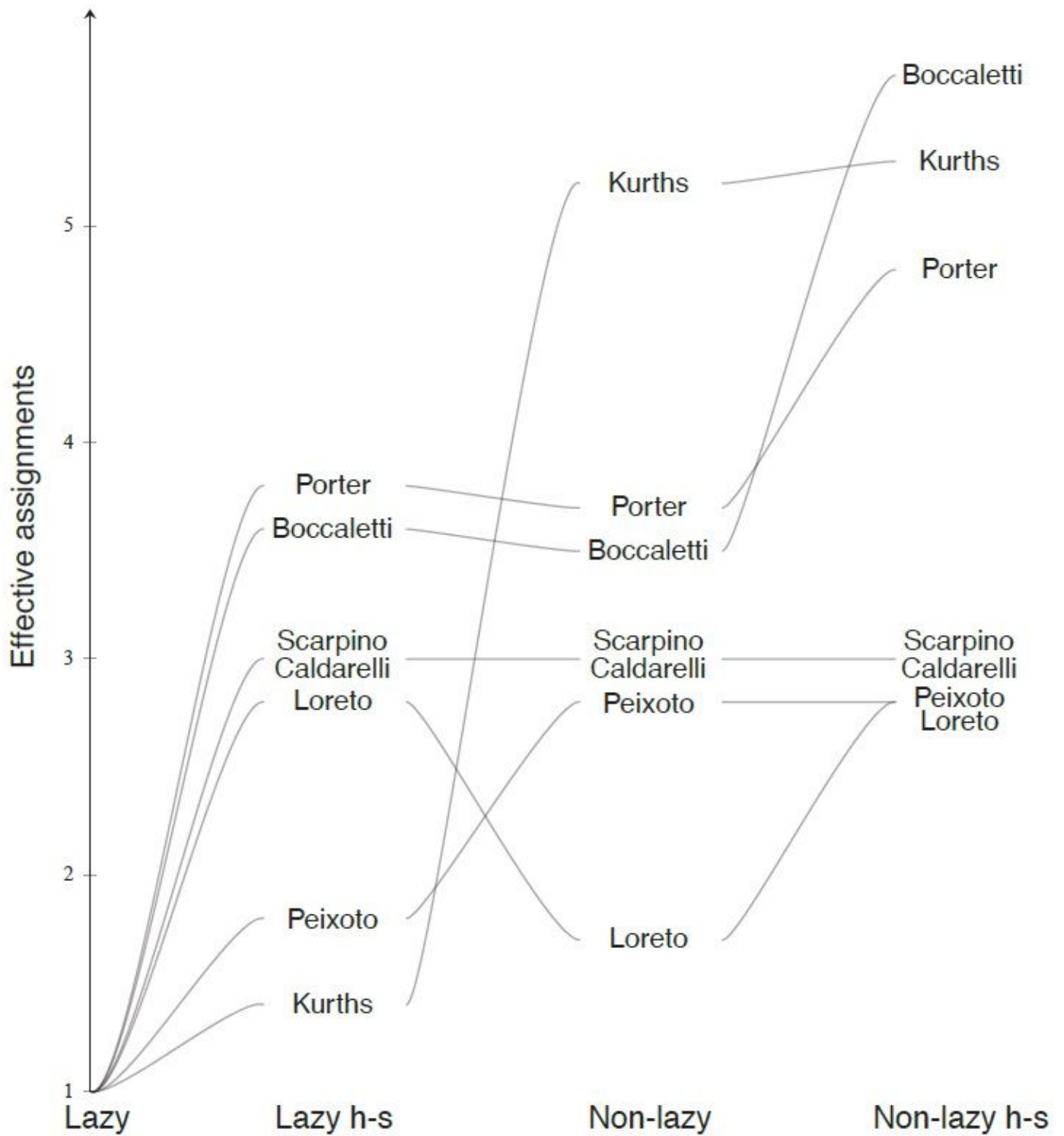
**Figure 3**

Alluvial diagrams of optimal partitions for the schematic hypergraph in Fig. 1. (a) Optimal partitions for lazy walks represented with the networks in Fig. 1(b-d). (b) Optimal partitions for non-lazy walks.



**Figure 4**

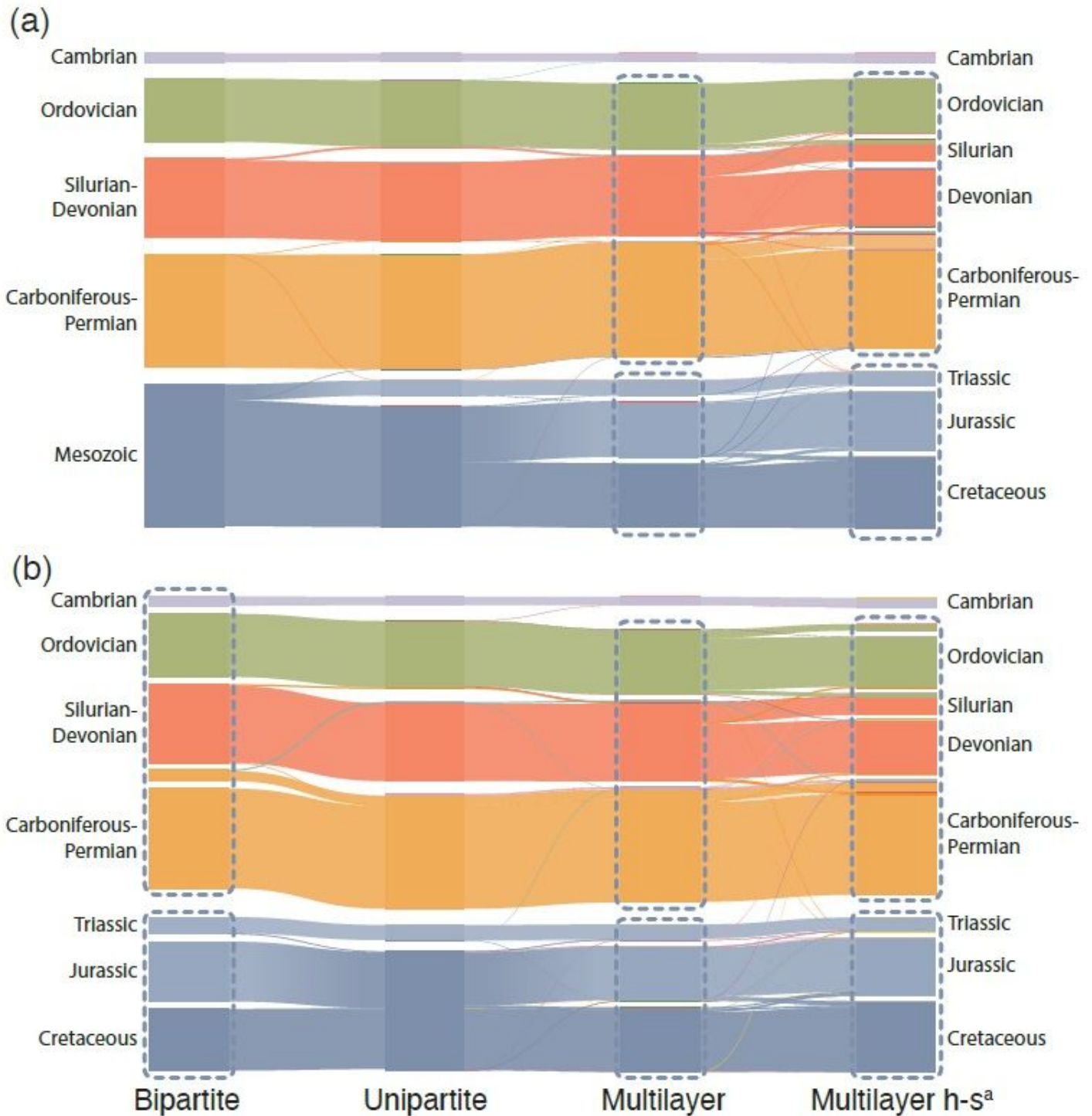
Alluvial diagrams of optimised partitions for different representations of the collaboration hypergraph. Lazywalks in (a) and non-lazywalks in (b). Module names from the top-ranked author within each module.



**Figure 5**

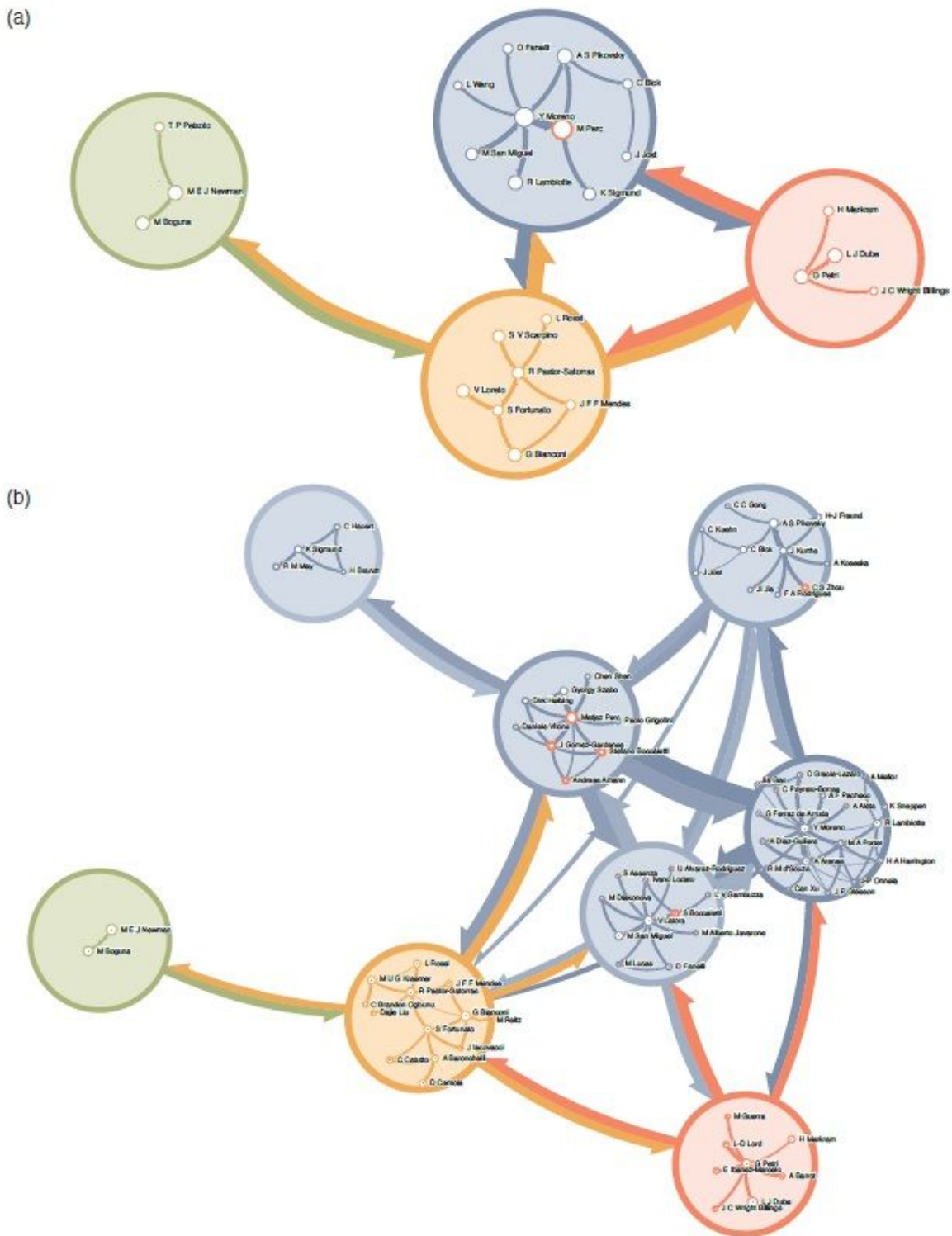
Authors in the collaboration hypergraph with the highest average effective number of assignments in the lazy and non-lazy multilayer representations (see Methods).





**Figure 6**

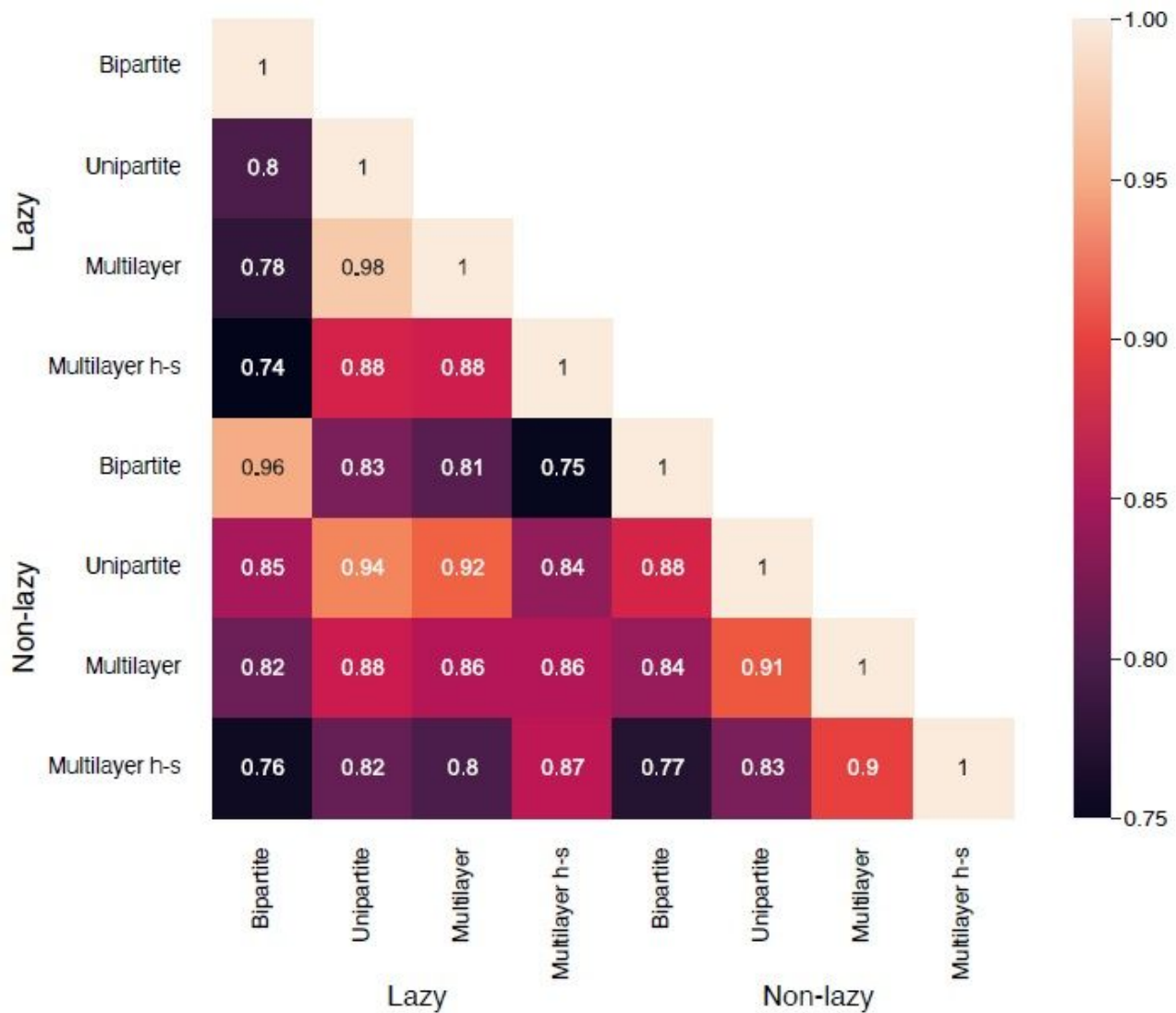
Alluvial diagrams of optimised partitions for the fossil hypergraph represented with different networks. Lazy walks in (a) and non-lazy walks in (b). We show top modules when a partition lacks deeper levels and leaf modules marked with dashed lines when they exist. Module names from the geological period or era represented by the fauna assemblage.



**Figure 7**

Hierarchical maps of the collaboration hypergraph using (a) the bipartite representation and (b) the multilayer hyperedge-similarity representation. Module colours are the same as in Fig. 4(a). Aggregated inter-module links with sizes proportional to the exiting flow volume and length inversely proportional to the flow volume. White sub-modules are labelled with the top-ranked author. The largest blue top module in (a) contains ten sub-modules. In (b), the partition assigns those nodes to five top modules containing

more sub-modules. S. Boccaletti, one of the most overlapping authors and highlighted in red, is assigned to one module in (a) and three top modules and six sub-modules in (b).



**Figure 8**

Leaf module assignments' adjusted mutual information for different random walk dynamics and hypergraph representations. The bipartite representations differ the most from the other representations, and the unipartite and multilayer representations are most similar at the leaf level.