Community landscapes: an integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics

István A. Kovács, Robin Palotai, Máté S. Szalay, Péter Csermely*
Department of Medical Chemistry, Semmelweis University, Tűzoltó str. 37-47, H-1094 Budapest, Hungary

Revised Suppl. Table 2 of the Electronic Supplementary Material S1

This is a revised version of the Supplementary Table S2 of our paper appeared in PLoS ONE 5, e12528. We marked the revised sections with yellow highlights, and ask our colleagues to send us any further comments to the email of csermely@eok.sote.hu

Table S2. Comparison of network module determination methods

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Agglomerative methods										
Hierarchical agglomeration (clustering)	+	+	−	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Johnson, 1967; Aldenfelder and Blashfield, 1984; Wasserman and Faust, 1994; Slater 2008; Rivera et al., 2010
Clustering with previous overlap 'distribution'	+	−	−	parameter dependent	refined	+	~O(n^3), local version: O(n log$n$)	N.A.	N.A.	Gregory 2009
Shortest path-similarity and hierarchical agglomeration	+	−	−	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Rives and Galitsky 2003; Arnau et al., 2005
Neighborhood similarity and hierarchical agglomeration	+	−	−	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Andreopoulos et al., 2007
Topological overlap methods	+	+	−	parameter dependent	yes/no	+	≥O(n^3)	N.A.	N.A.	Ravasz et al., 2002; Zhang and Horvath, 2005; Li and Horvath, 2007; Yip and Horvath, 2007
Restricted neighborhood search/flow Markov clustering	+	+	−	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Enright et al., 2002; Spirin and Mirny, 2003; Dorow et al., 2004; King et al., 2004
Clustering and a random walk process	+	+	+	parameter dependent	yes/no	−	~O(n^2)	N.A.	1	Delvenne et al., 2010; E et al., 2008; Li et al., 2008b
Clustering and a flocking process	+	+	+	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Li et al., 2008c
Markov clustering with noise	+	−	−	parameter dependent	yes/no	+	N.A.	N.A.	(1)	Gfeller et al., 2005; 2007

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Agglomerative methods (continuation)										
Clustering similarity- and distance similarity-based method	+	–	–	parameter dependent	yes/no	+	N.A.	N.A.	N.A.	Poyatos and Hurst, 2004
Dissimilarity matrix reordering-based visual clustering	+	–	–	parameter dependent	yes/no	–	$O(n^2)$	N.A.	N.A.	Yang et al., 2006
Multilevel clustering (coarsening/de-coarsening)	+, –	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Oliviera and Seok, 2006
Network Information Bottleneck (NIB) clustering	+	–	–	automatic	with 'soft clustering': refined	With 'soft clustering: +	N.A.	0.4, 0.45	N.A.	Ziv et al., 2005
Bayesian clustering	+	–	–	automatic	yes/no	–	$O(n^2)$	N.A.	0	Hofman and Wiggins, 2008
Expectation-maximization, kernel-derived method	+	+	–	automatic	refined	+	N.A.	N.A.	3 overlaps	Ren et al., 2009
Potts-model	+	+	+	parameter dependent	yes/no	–	~O(nlogn)	0.88	N.A.	Blatt et al., 1996; Spirin and Mirny, 2003; Guimera et al., 2004; Ronhovde and Nussinov, 2009, 2010
Fuzzy Potts-model	+	+	–	parameter dependent	yes/no	+	parameter dependent	0.7	N.A.	Reichardt and Bornholdt, 2004; 2006b; Ispolatov et al., 2006; Heimo et al., 2008b
Informational coherence and fuzzy Potts-model	+	–	–	parameter dependent	yes/no	+	$\geq O(n^3)$	N.A.	N.A.	Shalizi et al., 2007
Laplacian clustering	+	+	–	parameter dependent	yes/no	–	N.A.	N.A.	N.A.	Kim et al., 2008

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Agglomerative methods (continuation)										
Enforced frustration method	+	+	–	automatic	yes/no	–	O($n^{3.2}$)	N.A.	3	Son et al., 2006
Dynamical clustering (sync-dependent hierarchy detection)	+	–	–	parameter dependent	yes/no	–	O(n^2)	0.45	0 (3 modules)	Arenas et al., 2006; Boccaletti et al., 2007; Pluchino et al., 2008
Dynamical simplex evolution method	+	–	–	automatic	yes/no	–	O(n^2)	0.83	N.A.	Gudkov et al., 2008
Symmetric connectivity and noise/continuous dynamics	+	+	–	automatic	yes/no	–	O(n^2)	N.A.	N.A.	Krawczyk and Kulakowski, 2008
k-means (*k*-median, *p*-median) of cluster distance minimization	+	–	–	parameter dependent	yes/no	–	~O(n^4)	0.26, 0.9	0	MacQueen, 1967; Gustafsson et al., 2006; Angelini et al., 2007a; Brusco and Köhn, 2008; Kumar and Kannan, 2010
fuzzy *k*-means clustering	+	–	–	parameter dependent	yes/no	–	N.A.	N.A.	N.A.	Bezdek, 1981; Dunn, 1974; Schwammle, 2010
Principal component analysis-based clustering	+	+	–	automatic	refined	+	N.A.	N.A.	N.A.	Asur et al., 2007
Network vectorization clustering	+	+	+	parameter dependent	yes/no	–	N.A.	0.87	0	Ren et al., 2008
Co-occurrence methods	+	–	–	automatic	refined	+	N.A.	N.A.	N.A.	Papin et al., 2004; Jin et al., 2008
Highly connected subgraph spectral analysis method	–, +	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Kleinberg, 1997; Gibson et al., 1998; Hartuv and Shamir, 2000; Bu et al., 2003
Distance-based *k*-clique clustering methods	+	–	–	automatic	yes/no	–	O(n^3)	N.A.	N.A.	Edachery et al., 1999
Game-based clustering	+	+	+	parameter dependent	yes/no	–	N.A.	N.A.	N.A.	Li et al., 2010b

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Network walk-based agglomerative methods[i]										
Clique percolation methods	–, +	+	+	automatic	Refined	+	O(*exp(n)*) – in real applications it runs faster, $O(n^2)$	N.A.	N.A.	Alba and Moore, 1978; Palla et al., 2005; Adamcsek et al., 2006; Zotenko et al., 2006; Farkas et al., 2007; Palla et al., 2007a; Du et al., 2008; Kumpula et al., 2008; Lehmann et al., 2008; Shen et al., 2008; Zubcsek et al., 2008
Bipartite cliques	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Tanay et al., 2004
Local community-based methods (basins of attraction)	–, +	+	–	automatic	yes/no	+	$O(n\log n)$, $O(n^3)$	0.18	1	Altaf-Ul-Amin et al., 2006; Luo et al., 2006; Bagrow, 2008; Carmi et al., 2008; Hu et al., 2008b
Local fitness optimization	+	–	–	automatic	refined	+	$>O(n^2)$, fast	0.6	5 overlaps	Baumes et al., 2005a; 2005b; Lancichinetti et al., 2009
Local community with fuzzy clustering	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Hu et al., 2007
k-core-based methods	–	–	–	automatic	yes/no	+	$\sim O(n^3)$	N.A.	N.A.	Bader et al., 2003; Wuchty and Almaas, 2005; Alvarez-Hamelin et al., 2006; Dorogovtsev et al., 2006; Baskerville et al., 2007; Carmi et al., 2007
Local communities with *l*-shell label propagation	–, +	–	–	automatic	yes/no	+	$>O(n^3)$	N.A.	3	Bagrow and Bolt, 2005; Porter et al., 2007
Local communities with *t*-shell (initial triangles) label propagation	–, +	–	–	automatic	yes/no	+	N.A.	N.A.	4 overlaps	Eckmann and Moses, 2002; Kelsic, 2005
Local communities with bridge-bounding	+	–	–	automatic	yes/no	–	$\sim O(n)$	N.A.	N.A.	Papadopoulos et al., 2009

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Network walk-based agglomerative methods[i] (continuation)										
Local communities with (hub-based) label propagation/hub duplication	–, +	+	+	automatic	yes/no	–	~O(n)	N.A.	1	Gibson et al., 1998; da Fontoura Costa, 2004; Ucar et al., 2006; Zhang et al., 2008
Similarity-based network evolution methods	+	+	+ (signed network)	automatic	yes/no	–	≤O(n^3)	0.82	1	Yang, 2006; Xiang et al., 2009
Maximum flow, minimum cut methods (electric circuit models w/o or with spectral analysis)	+	+	–	parameter dependent	yes/no	–	O(n), >O(n^3)	0.57	0,1	Elias et al., 1956; Ahuja et al., 1993; Flake et al., 2002; Eriksen et al., 2003; Simonsen et al., 2004; Wu and Huberman, 2004; Alves, 2007; Slater 2008
Community profile plot assessment method	+	+	–	parameter dependent	yes/no	–	N.A.	N.A.	three modules	Leskovec et al., 2008
Communication-related Green's function with spectral analysis	+	–	–	parameter dependent	yes/no	+	>O(n^3)	N.A.	overlaps	Estrada and Hatano, 2008
Communicability graph and clique identification	+	–	–	parameter dependent	yes/no	+	exp(n) but in reality faster	N.A.	0 or overlaps	Estrada and Hatano, 2009
Random walk-based similarity distance and hierarchical agglomeration	+	+	+	automatic	yes/no	–	~O(n^3)	N.A.	N.A.	Pons and Latapy, 2005
Diffusion-based hierarchical communities	+	+	–	automatic	yes/no	–	O(n^3)	~0.75	0	Zhou 2003a; 2003b; Zhou and Lipowski, 2004
Diffusion kernel-based similarity method	+	–	–	automatic	yes/no	–	N.A.	0.68	1	Zhang et al., 2007b

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Network walk-based agglomerative methods[i] (continuation)										
Agent propagation, voting, mergers	+	−	−	automatic	yes/no	−	N.A.	N.A.	0	Young et al., 2004
Opinion propagation with decaying confidence	+	−	−	automatic	yes/no	−	N.A.	N.A.	1	Morarescu and Girard, 2009
Autonomous agent-based method for dynamic networks	+	−	−	automatic	yes/no	−	>O(n^2)	N.A.	1	Yang et al., 2010
Belief propagation Bayesian method to solve the Potts-model	+	−	+	automatic	yes/no	−	O($nlog^{\alpha}n$)	0.95	1	Hastings et al., 2006; Sulc and Zdeborova, 2010
Label propagation with random link removal	+	+	−	automatic	yes/no	+	~O(n)	N.A.	3 solutions (1 correct)	Raghavan et al., 2007; Tibély and Kertész, 2008; Xiaodong et al., 2008; Barber and Clark, 2009; Leung et al., 2009; Gregory, 2009; Liu and Murata, 2009
Affinity propagation methods	+	+	+	automatic	yes/no	−	>O(n^2)	N.A.	N.A.	Frey and Dueck, 2007; 2008; Leone et al., 2008; Wang et al., 2007; Brusco and Köhn, 2008
Information flow-based methods	+	+	−	automatic	refined	+	O(n^2logn)	N.A.	N.A.	Cho et al., 2006; 2007; Hwang et al., 2006; 2008
Signal propagation with F-statistics and fuzzy C-means clustering	+	+	−	automatic	yes/no	−	>O(n^2)	0.8	0	Hu et al., 2008a
Neuronal activation propagation times with principal component analysis	+	−	+	automatic	yes/no	−	N.A.	N.A.	N.A.	da Fontoura Costa, 2008a; 2008b

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Network walk-based agglomerative methods[i] (continuation)										
All shortest paths with principal component analysis	+	–	–	automatic	yes/no	–	N.A.	N.A.	0	da Fontoura Costa and Rodrigues, 2008a; Zhang et al., 2008
Message percolation method	+	+	–	automatic	refined	+	N.A.	0.15	N.A.	Meng Muntz and Rezaei, 2006
Structure-connected clusters (common neighbors)	+	–	–	automatic	yes/no	yes/no	O(n^2)	N.A.	N.A.	Mete et al., 2008
ModuLand method family	+	+	+	automatic	refined	+	implementation-dependent, ≤O($n3$) possible	N.A.	3 modules with overlaps	Kovács et al., 2006; current paper
Methods (both agglomerative and divisive) based on modularity (Q) optimization[j]										
Hierarchical agglomeration with (greedy) optimization	+	–	–	parameter dependent	yes/no	–	~O(n^2), O($nlog^2n$)	0.33	0,1	Newman, 2004a; Clauset et al., 2004; Gustafsson et al., 2006
Hierarchical agglomeration with a random walk	+	–	–	parameter dependent	yes/no	–	~O(n)	N.A.	1	Pujol et al., 2006
Multi-step greedy algorithm with vertex mover refinement	+	+	–	parameter dependent	yes/no	–	≥O($nlogn$)	N.A.	N.A.	Schuetz and Caflish, 2008; Noack and Rotta, 2009; Sun et al., 2009
Extremal division optimization	+	+	+	parameter dependent	yes/no	–	O(n^2logn)	0.82	5 modules	Duch and Arenas, 2005
Simulated annealing	+	+	+	automatic	yes/no	–	≥O(n^3)	0.9	N.A.	Guimera and Amaral, 2005; Massen and Doye, 2006
Mean field annealing	+	–	–	automatic	yes/no	–	≥O(n^2)	N.A.	N.A.	Lehmann and Hansen, 2007
Basin hopping	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Massen and Doye, 2005

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Methods (both agglomerative and divisive) based on modularity (Q) optimization[j] (continuation)										
Convex optimization	+	–	–	automatic	yes/no	–	N.A.	0.29	1	Hildebrand, 2008
Linear or vector programming relaxation methods	+	–	–	automatic	yes/no	–	$O(n^2)$, $O(n^3)$	N.A.	1 (4 modules)	Agarwal and Kempe, 2008
Hierarchical community detection based on affinity matrices	+	–	–	automatic	refined	+	$>O(n^3)$	N.A.	N.A.	Sales-Pardo et al., 2007
Genetic algorithm method	+	–	–	automatic	yes/no	–	$O(n)$	N.A.	0,1	Tasgin et al., 2007
Mixed integer mathematical programming	+	–	–	automatic	yes/no	–	N.A. (high)	N.A.	N.A.	Xu et al., 2007
Modularity preserving pre-modularization size-reduction	–	+	–	automatic	yes/no	–	speed increase up to a factor of 4.27	N.A.	0	Arenas et al., 2007
Local modularity algorithm	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Hinne, 2007
Multiscale Q-optimization Using self-loops	+	+	+	automatic	yes/no	–	N.A.	N.A.	0	Arenas et al., 2008b
Methods using spectral properties of the network	+	+	+	automatic	yes/no	–	$\geq O(n^2)$	0.57, 0.72	0	Donetti and Munoz, 2004; 2005; White and Smyth, 2005; Newman, 2006a; 2006b; Barber, 2007; Leicht and Newman, 2008; Richardson et al., 2008; Ruan and Zhang; 2008

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Methods (both agglomerative and divisive) based on modularity (Q) optimization[j] (continuation)										
Combinatorial approach to Modularity	+	+	+	automatic	yes/no	–	N.A.	N.A.	N.A.	Radicchi et al., 2010
Multilevel partitioning using minimum weight cut of a derived complete graph	+	–	–	automatic	yes/no	–	≥ O($nlog^2n$)	0.7	1	Djidjev, 2008
Optimization methods using alternative modularity definitions										
Hierarchical agglomeration for heterogeneous modules	+	–	–	parameter dependent	yes/no	–	O($nlog^2n$)	>0.33	0	Danon et al., 2006
Cluster density-based optimization	+	–	–	parameter dependent	yes/no	–	N.A.	N.A.	N.A.	Spirin and Mirny, 2003
Maximal clique-based Q optimization	+	–	–	parameter dependent	refined	+	N.A.	N.A.	4 modules + overlaps	Shen et al., 2009
Modularity for directed graph and overlaps	+	–	+	automatic	yes/no	+	N.A.	N.A.	2 overlaps	Nicosia et al., 2009
Modularity for positive and negative links	+	+	+ (signed network)	automatic	yes/no	–	N.A.	N.A.	N.A.	Bansal et al., 2004; Gómez et al., 2009; Kaplan and Forrest, 2008; Traag and Bruggeman, 2009
Local modularity optimization	–, +	–	+	automatic	yes/no	+	~O(n^2)	~0.5	N.A.	Clauset, 2005; Muff et al., 2005; Rodrigues et al., 2007
Local modularity optimization and hierarchical agglomeration	+	+	–	automatic	yes/no	–	N.A. (fast)	0.67	4 modules	Blondel et al., 2008; Wallace and Gingras, 2008
Multi-scale modularity with combined resolution parameters	+	+	+	automatic	yes/no	–	N.A.	N.A.	N.A.	Lambiotte, 2010
Multiscale (local → global) modularity refinement	+	+	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Pons, 2006

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Optimization methods using alternative modularity definitions (continuation)										
Community strength modularity optimization	+	−	−	automatic	yes/no	−	N.A.	N.A.	3 modules	Medus and Dorso, 2009
Local influence-based modularity optimization	+	+	+	automatic	yes/no	−	N.A.	N.A.	0	Ghosh and Lerman, 2008
Motif-based modularity maximizing methods	+	+	+	parameter dependent	yes/no	−	N.A.	N.A.	0	Arenas et al., 2008a
Triangle-based modularity optimization	+	+	+	automatic	yes/no	−	$O(n^3)$	N.A.	4 modules	Serrour et al., 2010
Centrality-based modularity optimization	+	−	−	parameter dependent	yes/no	−	N.A.	N.A.	4 modules	Ghosh and Lerman, 2009; Lerman and Ghosh, 2009
Statistical distribution-based modularity	+	−	−	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Pei and Zhang, 2007
Blockmodeling-based modularity	+	−	+	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Reichardt and White, 2007
Partition-coverage starting modularity generalization	+	−	−	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Gaertler et al., 2007
Mutual information-based modularity maximizing methods (link to hierarchical clustering)	+	+	+	parameter dependent	yes/no	−	N.A.	N.A.	N.A.	Angelini et al., 2007b; Bickel and Chen, 2009
Facility location theory-based method using strongly local modularity	+	−	−	automatic	yes/no	−	$O(n log^2 n)$	N.A.	0	Berry et al., 2007
Fuzzy modularity optimization using *c*-means clustering	+	+	−	automatic	refined	+	~$O(n)$	N.A.	4 modules	Zhang et al., 2007a

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Optimization methods using alternative modularity definitions (continuation)										
Fuzzy c-means modularity optimization based on improved Shared Nearest Neighbor method	+	+	–	Automatic	Refined	+	N.A.	N.A.	2 modules	Xie et al., 2009
Markov random walk and oscillator-synchronization-based modularity	+	+	+	automatic	yes/no	–	N.A.	N.A.	N.A.	Lambiotte et al., 2008; Li et al., 2008a
K-means clusters of node synchronization correlation matrix	+	+	+	automatic	yes/no	–	N.A.	N.A.	2 modules	Li et al., 2010a; Shen et al., 2010
Synchronization-based evolutionary subnetworks	+	–	–	automatic	yes/no	+	$O(n^2)$	N.A.	4 modules	Li et al., 2009
Random walk link partition on weighted line graph (link-to-vertex dual)	+	+	+	automatic	yes/no	+	N.A.	N.A.	4 modules	Evans and Lambiotte 2009a
Time-dependent multiscale and multiplex modularity	+	+	+	parameter dependent	yes/no	–	depends on choice of computational heuristic	N.A.	2–4 modules	Mucha et al., 2010

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Optimization methods using alternative modularity definitions (continuation)										
Q optimization on weighted line graph (link-to-vertex dual)	+	+	−	automatic	yes/no	+	N.A.	N.A.	N.A.	Evans and Lambiotte 2009b
Community detection based on community extraction criterion	−	+	+	automatic	yes/no	−	N.A.	N.A.	3	Zhao et al., 2010
Self-organization-based modularity	+	−	−	automatic	yes/no	−	N.A.	N.A.	N.A.	Ahnert et al., 2009
Other divisive methods										
Centrality-based methods (betweenness random walk, current-based and information centralities)	+	−	−	parameter dependent	yes/no	−	O(n^3), O(n^4)	0.3, 0.16	0, 1,3	Girvan and Newman, 2002; Fortunato et al., 2004; Newman, 2004c; Newman and Girvan, 2004; Andrade et al., 2009
Fuzzy betweenness centrality methods	+	−	−	parameter dependent	refined	+	≤O(n^3)	N.A.	N.A.	Wilkinson and Huberman, 2004; Pinney and Westhead, 2006; Gregory, 2007
Conductance optimization	+	−	−	parameter dependent	yes/no	−	(NP-hard)	N.A.	N.A.	Bollobas, 1998; Cheng and Shen, 2010
Cut-size optimization	+	−	−	parameter dependent	yes/no	−	(NP-hard)	N.A.	N.A.	Wei and Cheng, 1989
Division optimization with branch and bound method	+	+	−	automatic	yes/no	−	N.A.	N.A.	N.A.	Hager et al., 2009
Division optimization based on link-clustering (loops)	+	+	+	parameter dependent	yes/no	−	≥O(n2)	0.82	5 modules	Radicchi et al., 2004; Vragović and Louis, 2006

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Other divisive methods (continuation)										
Mutual information-based methods	+	–	–	automatic	yes/no	–	>O(n3)	N.A.	0	Strehl and Ghosh, 2002; Rosvall and Bergstrom, 2007; 2008; Sun et al., 2007; Zhang et al., 2008; Kraskov and Grassberger, 2009
Bootstrap resampling and significance clustering	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Rosvall and Bergstrom, 2010
Maximum likelihood method	+	+	+	automatic	refined	+	fast	N.A.	0,1	Čopič et al., 2009; Newman and Leicht, 2007; Mitrovic and Tadic, 2008; Mungan and Ramasco, 2010; Ramasco and Mungan, 2008; Vazquez, 2008a; 2008b; Wang and Lai, 2008; Zanghi et al., 2008
Dendrogram maximum likelihood method	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Clauset et al., 2008
Tree mapping	+	+	–	automatic	yes/no	–	N.A.	N.A.	N.A.	da Fontoura Costa and Rodrigues, 2008b
Similarity optimization with simulated annealing	+	+	–	automatic	refined	+	N.A.	N.A.	N.A.	Nepusz et al., 2008

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Other divisive methods (continuation)										
Hypergraph mutual information	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Strehl and Ghosh, 2002
Random walk-based heat kernel pagerank and Cheeger inequality local cut	–	–	–	parameter dependent	yes/no	–	~O($n^{1.5}$)	N.A.	N.A.	Chung, 2007
Random walk-based LinkRank	+	+	+	parameter dependent	yes/no	–	N.A.	N.A.	N.A.	Kim et al., 2010
Clustering (eigenvector) centrality and repetitive matrix bipartition	+	+	+ (signed network)	automatic	yes/no	–	~O(n)	N.A.	N.A.	Yang and Liu, 2007
Methods using spectral properties of the network or its local communities	–, +	+	+	automatic	yes/no	–	≥O(n)	N.A.	N.A.	Barnes, 1982; Donath and Hoffman, 1972; Kleinberg, 1997; Gibson et al., 1998; Hartuv and Shamir, 2000; Capocci et al., 2005; Tibély et al., 2006; Heimo et al., 2008a; Sahai et al., 2009
Division optimization based on community sub-matrix eigenvalue maximization	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Chauhan et al., 2009
Truncated singular value decomposition of the modular contribution matrix	+	–	–	automatic	yes/no	–	N.A.	N.A.	N.A.	Arenas et al., 2010

Table S2. Comparison of network module determination methods (continuation)

Name of method	Complete Data-set[a]	Weighted graph[b]	Directed graph[b]	Number of modules[c]	Assignment to modules[d]	Overlaps[e]	Polynomial complexity (speed)[f]	Test-modules[g]	Zachary-network[h]	References
Other divisive methods (continuation)										
Matrix factorization (+ semi supervised clustering)	+	–	–	automatic	yes/no	–	N.A. (slow)	0.53	3	Zhang et al., 2007c; Wang et al., 2008b; Ma et al., 2010
Spectral properties of the complement graph	+	+	+	automatic	yes/no	–	N.A.	N.A.	N.A.	Zarei and Samani, 2009; Zarei et al., 2009
Symmetric community measurement (on graph and its complement)	+	–	–	automatic	yes/no	–	N.A.	N.A.	3 modules	Wang and Lai, 2009
Methods for finding overlaps after any given non-overlapping clustering method										
Find overlappings by paralel genetic algorithms	+	+	–	automatic	yes/no	+	N.A.	N.A.	2 modules with overlaps	Carchiolo et al., 2009
Overlapping modularity measurement	+	–	–	automatic	yes/no	+	N.A.	N.A.	N.A.	Lázár et al., 2010

Data-assembly of the Table was closed on July 29th 2010. While we have taken a considerable effort to detect and read a large number of modularization methods, obviously the above list is extremely far from being complete. We would like to deeply apologize to all respectful colleagues, whose methods and significant efforts have been inadvertedly omitted from this list in this voluminous and extremely fast-growing field. Comparison of the methods was helped by the reviews of Newman (2004b), Danon et al. (2005), Fortunato and Castellano (2009), Habib and Paul (2010), Li et al. (2010c) and Fortunato (2010). 35 clustering algorithms have been nicely reviewed and clustered to a network by Jain et al. (2004). Several algorithms were compared by Lancichinetti and Fortunato (2009b), Leskovec et al. (2010) and Tibély et al. (2010). The modularity maximization methods were critically assessed by Good et al. (2010). Where more than one references are given, notes and numbers in the columns may refer to only one or a few of them. In case of multiple values separated with commas, each of them is referring to different reference(s). N.A. = data not available.

[a]At the column "Complete data-set" a "+" sign means that the method used all data from the original data set. The "–" sign denotes that some of the original data were deleted or comprised (like at coarse-graining methods), or the analysis was a fully local method using only a sub-segment of the original network. If both signs are present, some of the methods used all original data, while others not. Note, that many currently available network data are, in fact, only samples of a larger data-set, and in this sense, even the "+" methods use only a partial information. A detailed elucidation of the effects of sampling biases on modularization awaits further analysis (for an initial study see Lusseau et al., 2008).

[b]"Weighted graph" and "Directed graph" notes, if the method uses the additional information of weights or directedness for determining the modules. Some of the methods marked as "–" in these columns were applied to weighted and directed networks, but did not use these properties for the refinement of the modular structure.

[c]"Number of modules" is "parameter dependent", if the maximal number of modules does not derived "automatic"-ally from the method.

[d]"Assignment of modules" is "refined", if the method gives a continuous scale for all links and nodes as their assignment to various modules (fuzzy clustering/partition), and "yes/no", if only a decisive "yes" or "no" answer (hard/crisp community clustering/partition) is given.

[e]"Overlaps" notes, if the method calculates overlaps between modules. If the method had a "yes/no" assignment, the existence of overlaps means that certain nodes/links were assigned to multiple modules simultaneously with an equal weight.

[f]"Speed" refers to the computational speed (computational complexity) of the method, where symbol n denotes the number of nodes. To simplify the formulas and help comparison, many times it was assumed that n is roughly equal to the number of links, i.e. the method is applied to sparse networks.

[g]Numbers in the "Test modules" column refer to the fraction of correctly identified nodes in the model network proposed by Girvan and Newman (2002) having four communities with 32 links each having 16 number of neighbors in average, half of them being intra-modular and the other half of the nodes pointing towards other modules.

[h]The "Zachary network" column gives the number of misplaced members of Zachary's karate club (Zachary, 1977), where the hidden modular division was later exposed by a real split, which made this network a gold-standard for the assessment of module determination methods (Girvan and Newman, 2002). Values in parentheses refer to the number of overlapping values, which are not real misplacements.

[i]Network walk-based methods also contain a large variety of methods, where the mutual information of the local network environment has been assessed and used for module determination.

[j]The modularity function (Q) has been suggested by Newman (2004a) as a measure of the "statistically surprising fraction of the links in a network fall within the chosen communities" (Leicht and Newman, 2008) meaning that the link-density larger than that of an appropriate model system. The exhaustive optimization of this function is an NP-complete problem (Brandes et al., 2007), which makes this straightforward method computationally untractable with larger size real networks. Therefore, several heuristic, optimum-search strategies have been applied to find the global optimum and to circumvent the traps of local optima using an algorithm with a reasonably low computational complexity.

Additional Supplementary References

Asur S, Ucar D, Parthasarathy S. An ensemble framework for clustering protein-protein interaction networks. *Bioinformatics* **23,** i29–i40 (2007).