ERGM parameter estimation of very large directed networks: implementation, example, and application to the geography of knowledge spillovers

Alex Stivala [1]
Alfons Palangkaraya [2]
Dean Lusher [2]
Garry Robins [3,2]
Alessandro Lomi [1,4]

[1]Università della Svizzera italiana, Lugano, Switzerland. [2]Swinburne University of Technology, Melbourne, Australia. [3]The University of Melbourne, Melbourne, Australia. [4]University of Exeter Business School.

INSNA Sunbelt XXXIX, Montréal, Canada, June 18–23, 2019

Introduction

- Estimation of two very large (over 1 million nodes) directed networks:
  - An online social network with approx. 1.6 million nodes.
  - A patent citation network with approx. 3.8 million nodes.
- Using a new implementation of the EE algorithm (Byshkin et al., 2018) for directed networks,
- and also implemented so that very large networks can be handled (earlier implementations run into limits of memory usage and data structure efficiency).
- A substantive contribution in examining geographical knowledge spillover effects using the patent citation network.

Previous work with on large network ERGM estimation

- Stivala et al. (2016) *Soc. Netw.* 47:167–188 estimate ERGM parameters from snowball samples (undirected networks only).
- Stivala et al. (2015) Sunbelt XXXV Brighton [unpublished manuscript] extension of snowball sampling method to directed networks.
- Byshkin et al. (2016) *J. Stat. Phys.* 165:740–754 introduced the IFD sampler which increases the ESS by an order of magnitude with corresponding decrease in estimation time.
- Byshkin et al. (2018) *Sci. Rep.* 8:11509 introduced the Equilibrium Expectation (EE) algorithm which is shown to be two orders of magnitude faster than existing methods, and applied to a 104 103 node social network (Livemocha) and several smaller biological networks.

Extension to directed networks

- IFD and EE were initially only implemented for undirected networks.
- The EE algorithm was extended to directed networks and reimplemented (only using the "basic" sampler, not IFD yet) and applied to some directed networks:
  - The 1490 node political blogs network (Stivala et al. (2018), unpublished)
  - A 3308 node hospital transfer network (Stivala et al. (2018) INSNA Sunbelt XXXVIII (Utrecht) [unpublished manuscript]).
  - Two biological networks with 423 and 1781 nodes (Stivala et al. (2018) PASC 2018 Basel poster presentation [unpublished manuscript].
- The IFD sampler is now also implemented for directed networks and applied in the EE algorithm to the very large directed networks described here.

Pokec

- Pokec is the most popular online social network in Slovakia.
- The network data (Takac & Zabovsky, 2012) is the entire anonymized network containing various attributes such as gender, age, region (187, in Slovakia or elsewhere), etc.
- "Friendships" in Pokec are directed (and not necessarily reciprocated).
- The network has 1 632 803 nodes and 30 622 564 arcs.
- The network is described as "scale-free" by Takac & Zabovsky (2012), but only based on eyeballing a degree–frequency plot.

Patent citations

- The NBER patent citation network (Hall, Jaffe, & Trajtenberg, 2001).
- Citation network of patents granted between 1975 and 1999.
- There are 3 774 768 nodes and 16 518 948 arcs (citations).
- There is data about the patents (category, subcategory, application date, inventor location, etc.) for patents from 1963 to 1999.
- There is such data for 2 755 865 patents in the citation network
- I.e. approx. 73% of the patents in the citation network have attribute data.
- There are 6 patent categories and 36 subcategories (derived from the original 400 classes from USPTO).

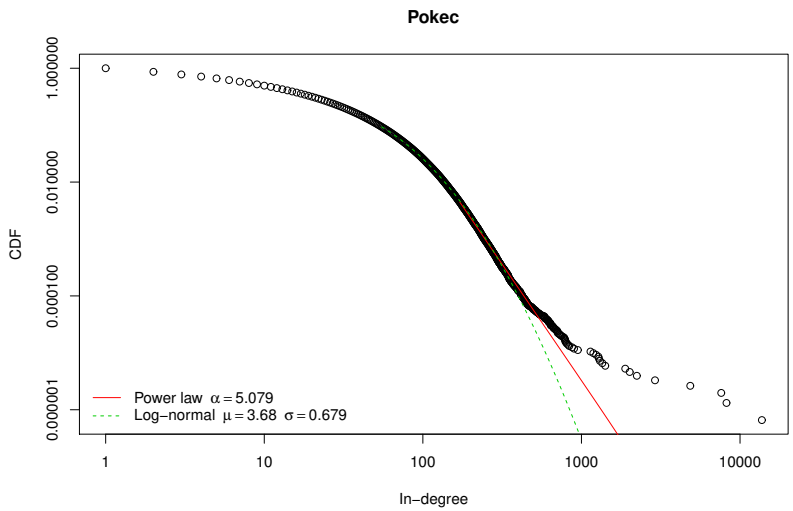
Graph summary statistics

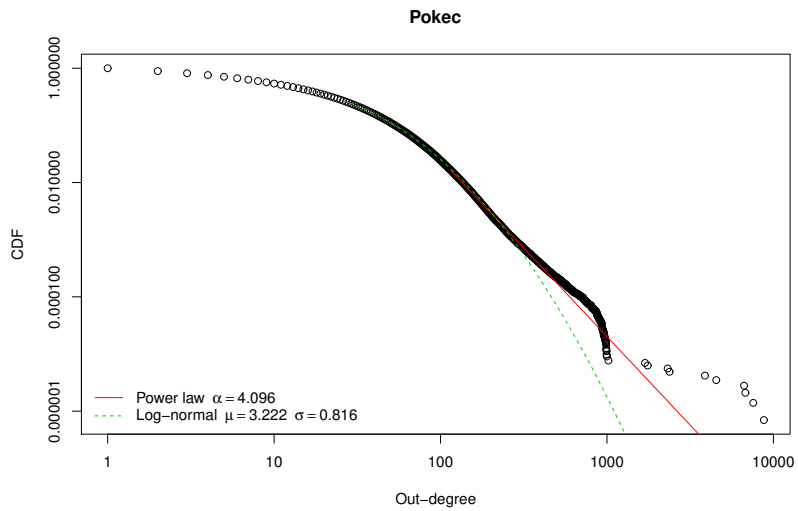
Description	N	Components	Mean degree	Density	Clustering coefficient	Assortativity coefficient
Pokec	1632803	1	37.51	0.000011	0.04682	-0.00049
Pokec (no hubs)	1632783	577	37.38	0.000011	0.05369	0.07867
Patents	3774768	3627	8.75	0.000001	0.06714	0.13317
Patents (NA removed)	2755865	13401	10.14	0.000002	0.07193	0.13397

Previous work on the patent citation network, particularly relating to "knowledge spillovers"

- Reviewed in Jaffe & de Rassenfosse (2017).
- Jaffe, Trajtenberg, & Henderson (1993) using patent class matching between citing and "control" patents, find that citations are more likely to be geographically proximate, evidence for knowledge spillover effect.
- Thompson & Fox-Kean (2005) argue that using the patent class is too coarse and use a finer-grained matching, finding the spillover effect is then reduced at the intra-national level (but not the international level).
- Henderson, Jaffe, & Trajtenberg (2005) respond that the question is how robust the localization effect is under different assumptions.
- However note all of this (in common with most economics work using networks) treats the network as exogenous. Using ERGM the network is instead endogenous.

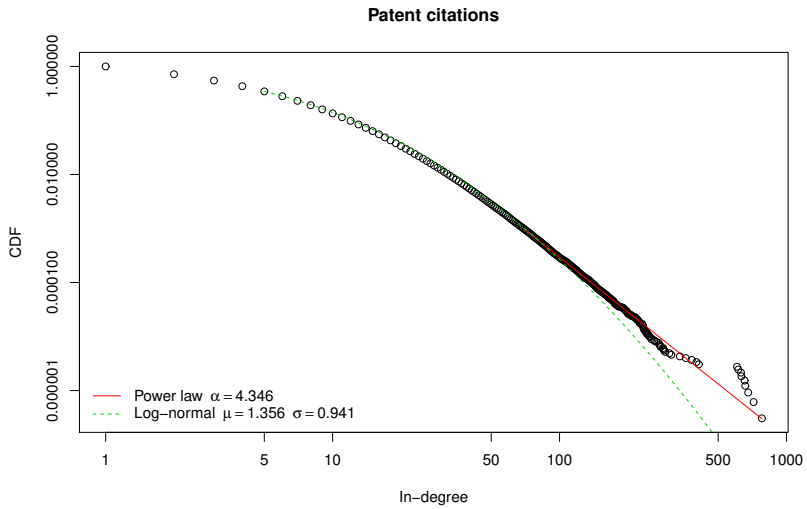
Pokec degree distribution

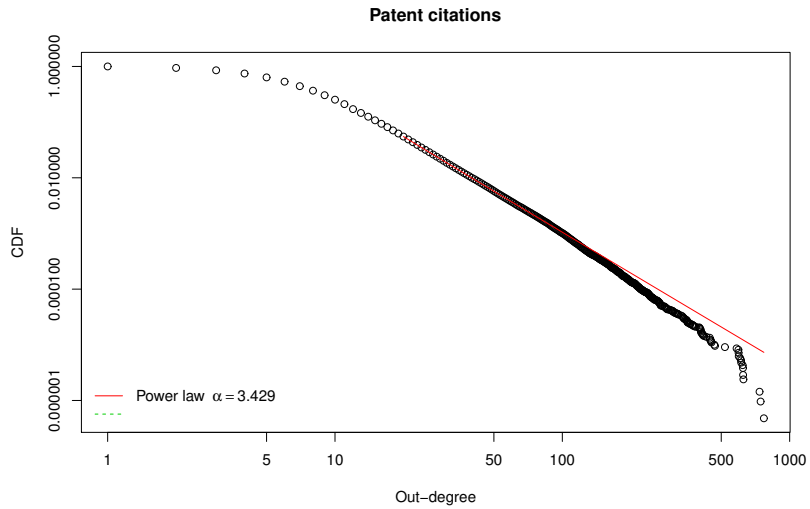




Consistent with neither power law nor log-normal ($p < 0.01$)

Patent citations degree distribution





In-degree distribution is consistent with power-law, but not log-normal ($p < 0.01$). Out-degree distribution is not consistent with power-law ($p < 0.01$), and log-normal could not be fitted.

Pokec initial results

- Models with the usual parameters Reciprocity, AltInStars, and AltOutStars did not converge, probably due to the strong "hubs" in the network.
- Even without them, the convergence of this model was not good.
- Also about 54% of the arcs are reciprocated, so lack of such arcs is definitely not the reason it does not converge with these parameters.
- So this indicates the model was badly specified or cannot be fitted for other reasons.

More on Pokec initial results

- Although reciprocity was not in the model, the simulated network reciprocity was 45% (54% for the observed network).
- So we try removing the hubs, which we define as nodes with in or out degree greater than 1000, based on the noticeable "break" there on the CDF plot on earlier slide.
- Note this results in removing only 20 nodes (0.001% of nodes), and furthermore Takac & Zabovsky (2012) note that "hubs in Pokec are not people but commercial companies which advertise through this network." (p. 5)

Estimation of Pokec network with 20 hub nodes removed

- 2 parallel tasks with 500 GB each ("bigMem" partition)
- Estimation took approx. 22 hours.
- IFD sampler acceptance rate was between approx. 21% and 26%.

Effect	Model 1	Model 2	Model 3	Model 4
Arc	−19.402 (−20.132, −18.672)	−20.670 (−21.308, −20.032)	−20.666 (−21.430, −19.901)	−20.642 (−21.405, −19.880)
Isolates	−6.599 (−11.140, −2.059)	−7.722 (−12.878, −2.566)	−7.717 (−12.499, −2.936)	−7.711 (−12.248, −3.173)
Reciprocity	—	34.195 (27.684,40.705)	34.234 (27.922,40.547)	33.473 (25.805,41.140)
Popularity spread (AinS)	1.430 (1.261,1.598)	1.684 (1.546,1.822)	1.683 (1.493,1.872)	1.720 (1.464,1.976)
Activity spread (AoutS)	1.566 (1.100,2.031)	1.876 (1.547,2.204)	1.875 (1.384,2.366)	1.900 (1.339,2.461)
Two-path (A2P-T)	−0.006 (−0.013,0.002)	−0.014 (−0.020, −0.008)	−0.014 (−0.019, −0.010)	−0.014 (−0.021, −0.007)
Shared popularity (A2P-D)	0.016 (0.011,0.021)	0.021 (0.014,0.027)	0.021 (0.014,0.028)	0.021 (0.016,0.026)
Shared activity (A2P-U)	0.017 (0.010,0.024)	0.022 (0.016,0.029)	0.022 (0.016,0.028)	0.022 (0.015,0.028)
Path closure (AKT-T)	2.179 (1.213,3.145)	2.335 (1.028,3.641)	2.179 (1.262,3.096)	2.151 (1.330,2.972)
Popularity closure (AKT-D)	2.007 (1.333,2.682)	2.054 (0.977,3.130)	2.281 (1.363,3.199)	2.270 (1.393,3.148)
Activity closure (AKT-U)	2.074 (1.064,3.084)	2.021 (0.989,3.052)	2.307 (1.483,3.131)	2.270 (1.456,3.083)
Cyclic closure (AKT-C)	1.303 (0.546,2.059)	0.635 (0.502,0.767)	—	—
Matching region	3.405 (2.185,4.625)	3.211 (1.498,4.924)	3.215 (2.123,4.307)	3.172 (2.086,4.258)
Matching gender	−1.047 (−1.636, −0.458)	−1.167 (−1.981, −0.352)	−1.168 (−1.913, −0.422)	−1.164 (−1.738, −0.591)
Sender age	0.025 (−0.001,0.051)	0.020 (−0.028,0.068)	0.020 (−0.019,0.060)	0.021 (−0.008,0.049)
Receiver age	0.026 (−0.003,0.055)	0.023 (−0.017,0.063)	0.023 (−0.020,0.066)	0.022 (−0.003,0.047)
Diff age	−0.115 (−0.142, −0.087)	−0.100 (−0.122, −0.077)	−0.100 (−0.129, −0.071)	−0.099 (−0.123, −0.074)
Sender completion %	0.002 (−0.010,0.015)	0.001 (−0.010,0.013)	0.001 (−0.010,0.012)	—
Receiver completion %	0.003 (−0.007,0.013)	0.001 (−0.010,0.013)	0.001 (−0.013,0.016)	—

Diff completion %	−0.003 (−0.017,0.011)	−0.002 (−0.021,0.017)	−0.002 (−0.021,0.017)	—
Sender public	0.080 (−0.139,0.300)	0.205 (0.102,0.308)	0.204 (0.087,0.321)	—
Receiver public	−0.099 (−0.203,0.005)	−0.110 (−0.244,0.025)	−0.109 (−0.274,0.056)	—

Pokec results interpretation [1]

- Centralization on both in- and out- degree.
- Positive activity closure: people who send ties to the same people tend also to have a tie (however shared activity is also positive, so can we conclude this?)
- Positive path closure: friends of friends also tend to be friends.
- Positive cyclic closure (non-hierarchical network closure). Note this is quite rare in friendship networks (although this apparent rarity may be spurious (Block, 2015)).
- However this apparent "generalized exchange" effect seems to be reduced when reciprocity is explicitly included in the model, and cyclic triads (030C) appear to have reasonable fit in the "pseudo-GoF" plots whether or not cyclic closure is included in the model with reciprocity included.
- (Note however triad 021C [directed line] does not have great fit, despite A2P-T being included in the model; when Reciprocity is not included, 021C is too high, and when it is included, 021C fits better, but (slightly) too low.)

Pokec results interpretation [2]

- There is significant homophily on age and region.
- We don't find any significant effects of profile completion percentage or profile "public" flag.
- There is significant heterophily on gender. (This seems interesting/unusual for a social network: is it being used for "dating"?)

Special properties of citation networks

- There are no reciprocated arcs (or nearly none; but there are exactly 0 in this data).
- So cannot include reciprocity in the model, and so a constraint is implemented in the sampler to prevent such arcs being created in the simulation.
- Things "in the future" can't be cited, i.e. a patent or publication can only cite one that is already published (or applied for) — although exceptions can also occur due to long review times, etc.
- This means that citation networks are (approximately) directed acyclic graphs.

The DiffSign parameter

- To account for this we introduced the "DiffSign" statistic on a continuous attribute a, which for an an arc $i \to j$ is $+1$ if $a_i > a_j$, -1 if $a_i < a_j$, and 0 if $a_i = a_j$
- So if the attribute is the patent date, the DiffSign parameter is significantly positive if patents tend to cite those in the past.
- This is similar to an ergm.userterm added to statnet for this purpose, described by Graham et al. at Utrecht Sunbelt (2018) and published as McLevey et al. (2018).
- Note we could also implement a constraint in the simulation to prevent this happening at all, but it can be preferable to do it this way as for example in this case the data does contain some patents that cite those "in the future".

Estimating the patent citation network

- 8 parallel tasks with 55 GB each (default "slim" partition).
- Elapsed time approx. 2 hours.
- IFD sampler acceptance rate approx. 28%.

Estimation results

- Full citation graph (3 774 768 nodes)
- Subgraph induced by nodes that have patent data (2 755 865 nodes).
- For the full graph, 45% of nodes are sinks (cite no other patents) and 14% of nodes are sources (are not cited).
- We include the Sink and Source parameters to control for this.
- For the subgraph only 26% of nodes are sinks and 19% are sources (many of the removed patents without attributes were sinks).
- Note base for sender year is 1975 but for receiver year is 1963 as citation data starts in 1975, but full patent attribute data starts in 1963 (see also "quasi-structural" model, Table 3 of Hall, Jaffe, & Trajtenberg, 2001).
- As will be seen on the next slide, results are not substantively different.

Effect	Full graph	Subgraph
Arc	−17.139 (−17.823, −16.456)	−18.542 (−19.161, −17.922)
Sink	3.876 (2.915,4.838)	1.735 (1.697,1.772)
Source	−0.867 (−0.890, −0.844)	−0.762 (−0.782, −0.741)
Popularity spread (AinS)	1.192 (0.985,1.399)	1.198 (0.939,1.457)
Activity spread (AoutS)	0.395 (0.368,0.423)	0.427 (0.406,0.447)
Two-path (A2P-T)	−0.018 (−0.037,0.001)	−0.015 (−0.035,0.006)
Shared popularity (A2P-D)	0.022 (0.001,0.043)	0.018 (−0.001,0.037)
Shared activity (A2P-U)	0.026 (0.007,0.045)	0.028 (0.005,0.050)
Sender APPYEAR [1975]	0.019 (0.002,0.035)	0.046 (0.032,0.059)
Receiver APPYEAR [1963]	−0.050 (−0.064, −0.036)	−0.044 (−0.059, −0.030)
DiffSign APPYEAR	1.743 (1.520,1.966)	2.333 (1.913,2.752)
Diff APPYEAR	−0.105 (−0.120, −0.090)	−0.101 (−0.116, −0.085)
Sender CLAIMS	0.010 (−0.007,0.027)	0.014 (−0.000,0.028)
Receiver CLAIMS	0.011 (−0.003,0.025)	0.010 (−0.004,0.024)
Diff CLAIMS	−0.005 (−0.018,0.009)	−0.007 (−0.018,0.004)
Matching CAT	0.799 (0.623,0.974)	1.025 (0.841,1.208)
Matching SUBCAT	2.895 (2.222,3.568)	2.897 (2.205,3.589)
Matching COUNTRY	0.222 (0.155,0.289)	0.457 (0.284,0.631)
Matching POSTATE	0.830 (0.341,1.320)	0.842 (0.340,1.344)
Matching ASSIGNEE	3.889 (1.414,6.363)	3.823 (1.578,6.068)

Patent citation estimation results

- Models with triangles (closure) included did not converge well, even with far more iterations.
- There are always exactly 0 triads with reciprocity due to the constraint.
- But a very small number of cyclic triads (030C) are created in the simulated networks, even though there are exactly 0 in the observed network.
- There is centralization on in-degree and out-degree.
- Positive shared activity, shared popularity, negative simple two-path. Models structure, but in absence of converged models with closure parameters, no real useful interpretation?

Patent citation results interpretation

- Positive DiffSign parameter on application year: as expected, patents are more likely to cite those applied for earlier in time.
- Positive sender and negative receiver on application year: more recent patents make more citations but receive fewer (control for fact that older patents have more time to get citations, and citation rates increase over time).
- Significant negative heterophily on application year: patents are more likely to cite patents closer in time (more recent).
- Significant homophily on category and subcategory.
- Significant homophily on assignee: "self-citation" indicating mostly internalized knowledge transfers.
- Significant homophily on country and state: **The geographic spillover effect hypothesis is confirmed at both the country and state levels.** (While also controlling for homophily on technology category and subcategory and assignee).

Conclusions / contributions

- Methodological
  - Freely available code for ERGM estimation of very large directed networks using the recently published EE algorithm and IFD sampler.
  - Demonstration ERGM parameter estimation of the largest (online) social network for which an ERGM has ever been estimated to date (1.6 million nodes).
  - ERGM parameter estimation of the largest network for which an ERGM has ever been estimated to date (over 3.7 million nodes).
  - The latter is a special case as it is a citation network and so special constraints and parameters were also introduced.
- Substantive (patent citation network and spillover effects)
  - Using more sophisticated statistical techniques to more closely examine claims about the degree distributions of the patent citation network, finding that only the in-degree distribution is consistent with a power law.
  - Treating the citation network as endogenous (not exogenous as in previous work) and finding the geographical spillover effect appears to be robust to this.

All data and codes are freely available

Pokec Stanford Network Analysis Project (SNAP) [Leskovec & Krevl, 2014]

`http://snap.stanford.edu/data/index.html`

Patents National Bureau of Economic Research (NBER)

`http://www.nber.org/patents/`

EstimNetDirected `https://github.com/stivalaa/EstimNetDirected`

These slides and manuscript preprint(s) such as

> Stivala, A., Robins, G., & Lomi, A. (2019). Exponential random graph model parameter estimation for very large directed networks. arXiv preprint arXiv:1904.08063. `https://arxiv.org/abs/1904.08063`

will be available via my website

`https://sites.google.com/site/alexdstivala/`

Acknowledgments

- We gratefully acknowledge the support of Swiss National Science Foundation NRP 75 Big Data project 167326 “The Global Structure of Knowledge Networks: Data, Models and Empirical Results”.
- We thank Dr Pavel Krivitsky (University of Wollongong) for useful discussions on ERGM MLE standard error estimation.
- We thank the administrators of the USI ICS cluster, and in particular Mr Hardik Kothari and Mr Radim Janalik, for technical assistance.

Hidden bonus slides

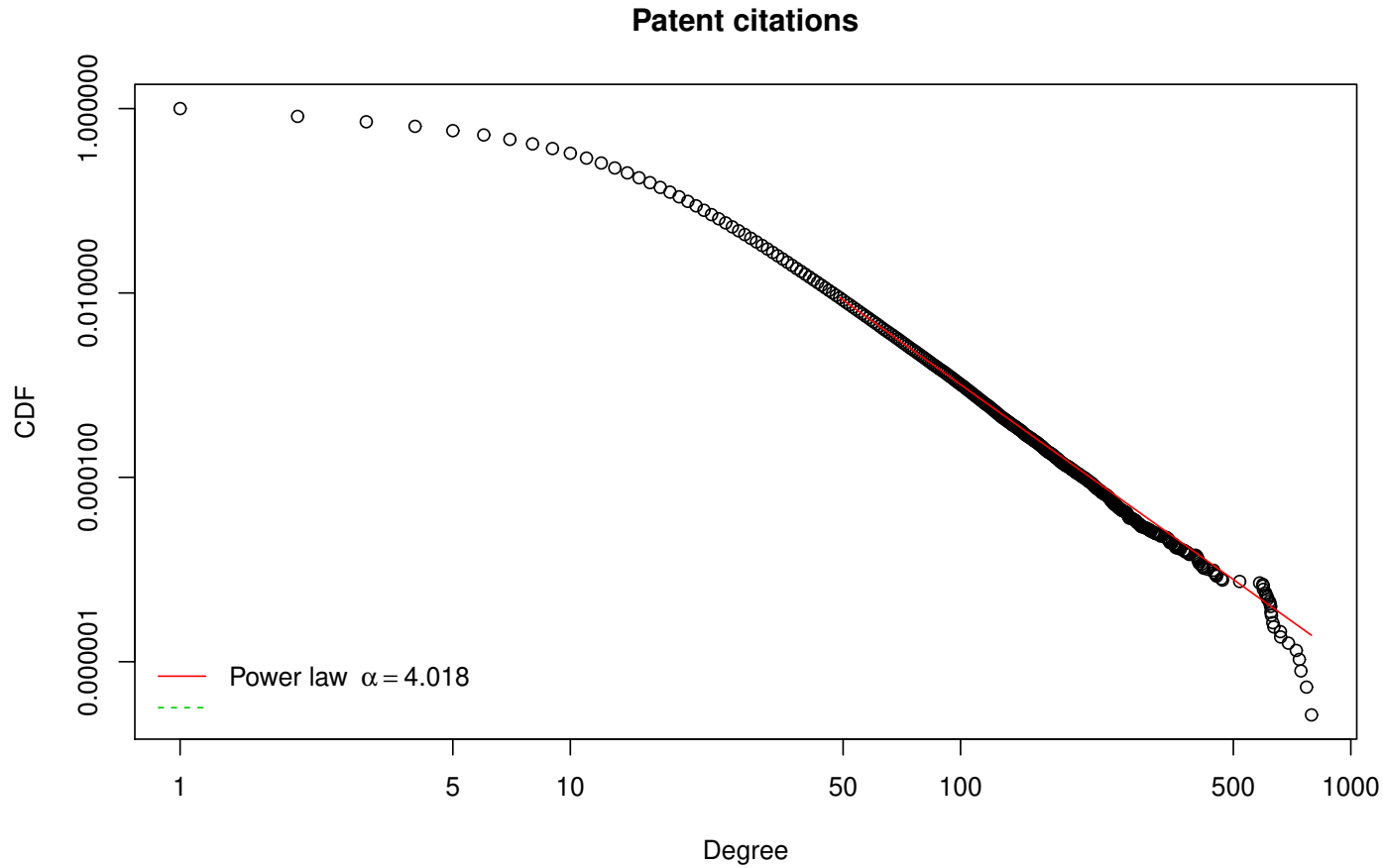
Previous work on the patent citation network, particularly relating to degree distribution

- Chen & Hicks (2004) find power law distribution in the patent citation network (with exponent 2.89)
- And further state that this means patent citation networks are "scale-free, not a real surprise".
- However this is using only the undirected degree, not treating in- and out- degree distributions separately.
- Further, is is done by the unsophisticated method of fitting a straight line to a degree–frequency double log plot
- Which is (now) known to be not a good technique (Clauset, Shalizi, & Newman, 2009; Stumpf & Porter, 2012)
- And perhaps finding a scale-free degree distribution should be more of a surprise than previously advertised (above citations, plus also Broido & Clauset, 2018)

Fitting power law distributions to empirical data

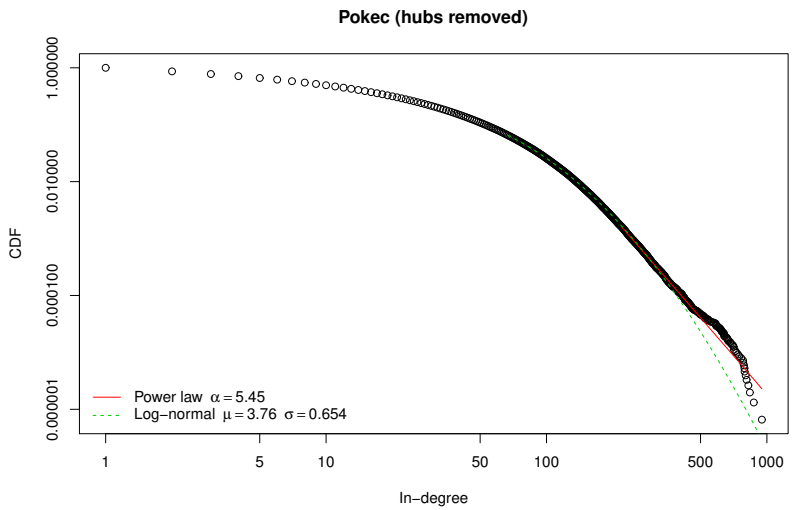
- Using unsophisticated methods (like fitting a least-squares line) can be inaccurate,
- And also gives no indication of whether or not the distribution is consistent with a power law.
- Clauset, Shalizi, & Newman (2005) give a principled statistical technique using MLE and GoF tests.
- We use these methods as implemented in the poweRlaw R package (Gillespie, 2015)
- To test for fits to both power law and log-normal distributions.

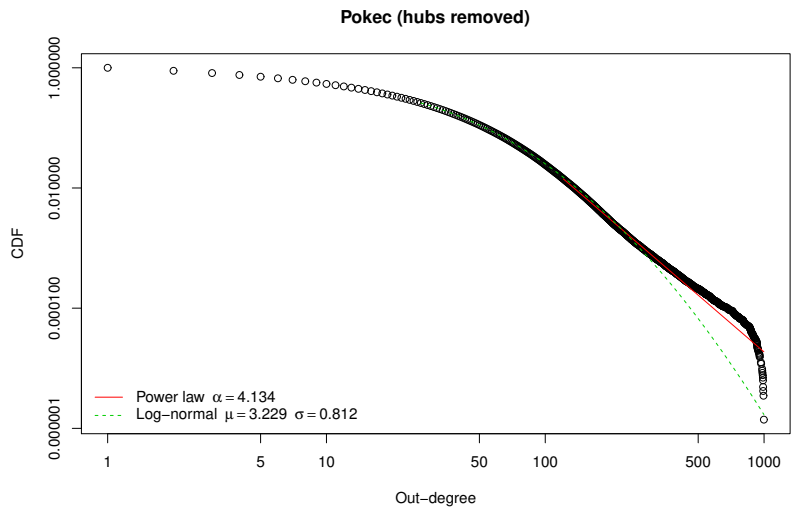
Patent citations degree distribution (undirected)



The degree distribution is not consistent with power law ($p < 0.01$), and log-normal could not be fitted.

Pokec degree distribution (20 highest degree nodes removed)





In-degree distribution is consistent with power law, but not log-normal ($p < 0.01$). Out-degree distribution is consistent with neither power law nor log-normal ($p < 0.01$).

Standard error estimation

- For each parallel MCMC run, the parameter and its standard error are estimated:
  - The point estimate (mean) and asymptotic covariance matrix (batch means method) for MCMC standard error are estimated using the `mcmcse` R package (Flegal et al., 2017).
  - The covariance matrix for the error in approximating the MLE is estimated as the inverse of the covariance matrix of the simulated statistics (Fisher information), also using `mcmcse`.
  - The total estimated covariance matrix is then estimated as the sum of these two covariance matrices, and from this we compute the standard error.
- The overall estimate and its standard error are then estimated as the inverse variance weighted average of these parallel runs.

Technical details

- Completely new implementation of the EE algorithm and IFD sampler in C, using MPI to run multiple estimations in parallel.
- Efficient data structures and algorithms so that computation of change statistics is fast and scalable:
  - Graphs stored as adjacency lists for memory efficiency, with both forward and reverse adjacency list for fast computation of change statistics, as well as flat list of all arcs for fast lookup of random arcs (needed by IFD sampler).
  - For fast computation of the "alternating" statistics, three two-path tables are built.
  - In PNet these are arrays but this is not scalable; so instead we take advantage of sparsity and use hash tables, with Bloom filter for fast misses. (Implemented in the "uthash" library `https://troydhanson.github.io/uthash/userguide.html`)
- For good pseudorandom number generation, the Random123 counter-based PRNG (Salmon et al., 2011) is used.

More technical details

- Using hash tables and not arrays for two-path lookup tables is vital.
- It is also very advantageous to use a Bloom filter so that the overwhelmingly more frequent case of looking up an entry that is not present is faster.
- During the MCMC process arcs are added and deleted, and it is also essential to delete from the hash tables any two-path values that fall to zero, to stop the tables growing indefinitely. (Unfortunately, however, this diminishes the effectiveness of the Bloom filter).
- For the Pokec network, these matrices are approx 0.06% nonzero, and so use approx. 200 GB instead of the nearly 10 TB they would occupy as arrays.
- For the patent citation network, they are approx. 0.001% nonzero, and so use approx. 10 GB instead of the 50 TB they would occupy as arrays.

Ongoing and future work and problems (methodological 1)

- The EE algorithm used here required "tuning" of hyper-parameters, but a new version was recently proposed which has only one hyper-parameter (Borisenko et al., 2019), potentially making it easier to obtain converged models.
- These were whole network estimations, but snowball sampling and conditional estimation are also implemented, to allow even larger networks via snowball samples.
- Also need to extend to bipartite (or more generally multilevel) networks.
- Some empirical networks of interest have two-path matrices that are not sparse enough or have the "wrong" structure for either the efficient two-paths data structure or snowball sampling to work well with, e.g. the approx. 1 million node physician referral network described by An et al. (2017). Why? What to do about it?

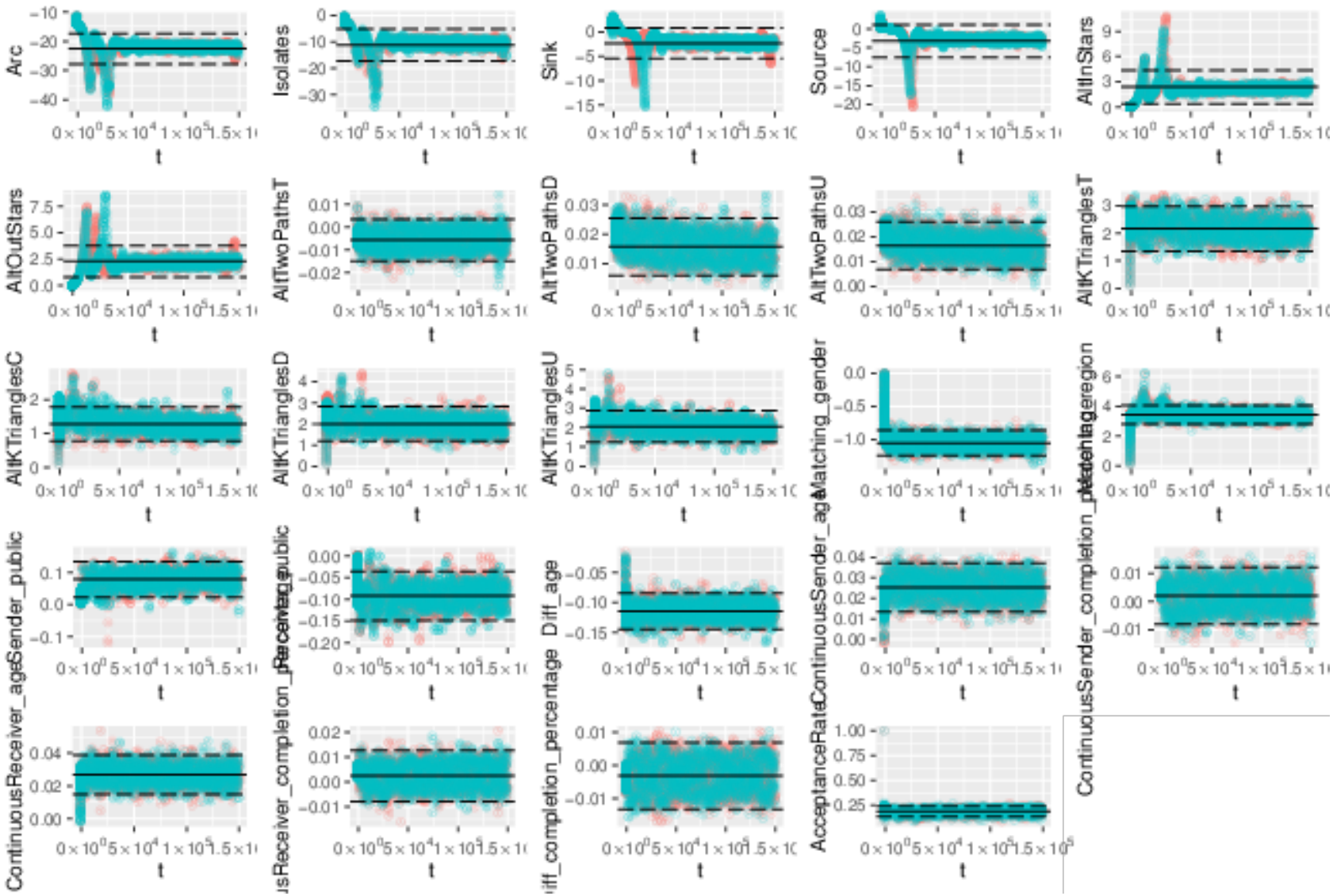
Ongoing or future work and problems (methodological 2)

- Contrastive Divergence and the EE algorithm are fast as we "cheat" in the MCMC and do not need burn-in.
- So our "pseudo-GoF" we use as a check is probably over-optimistic.
- But we can't practically simulate such large ERGM networks from scratch to do it properly.
- So we need some other way, for example by simulating snowball samples and comparing statistics to snowball samples from the observed network.
- Also, we know bias in the ERGM MLE for some parameters (in undirected case: Edge, Alt.K-Star) in these models gets worse for larger networks (Stivala, Byshkin, & Robins 2017 Sunbelt Beijing; unpublished manuscript).
- Does an ERGM even make sense for such large networks, especially the homogeneity assumption?

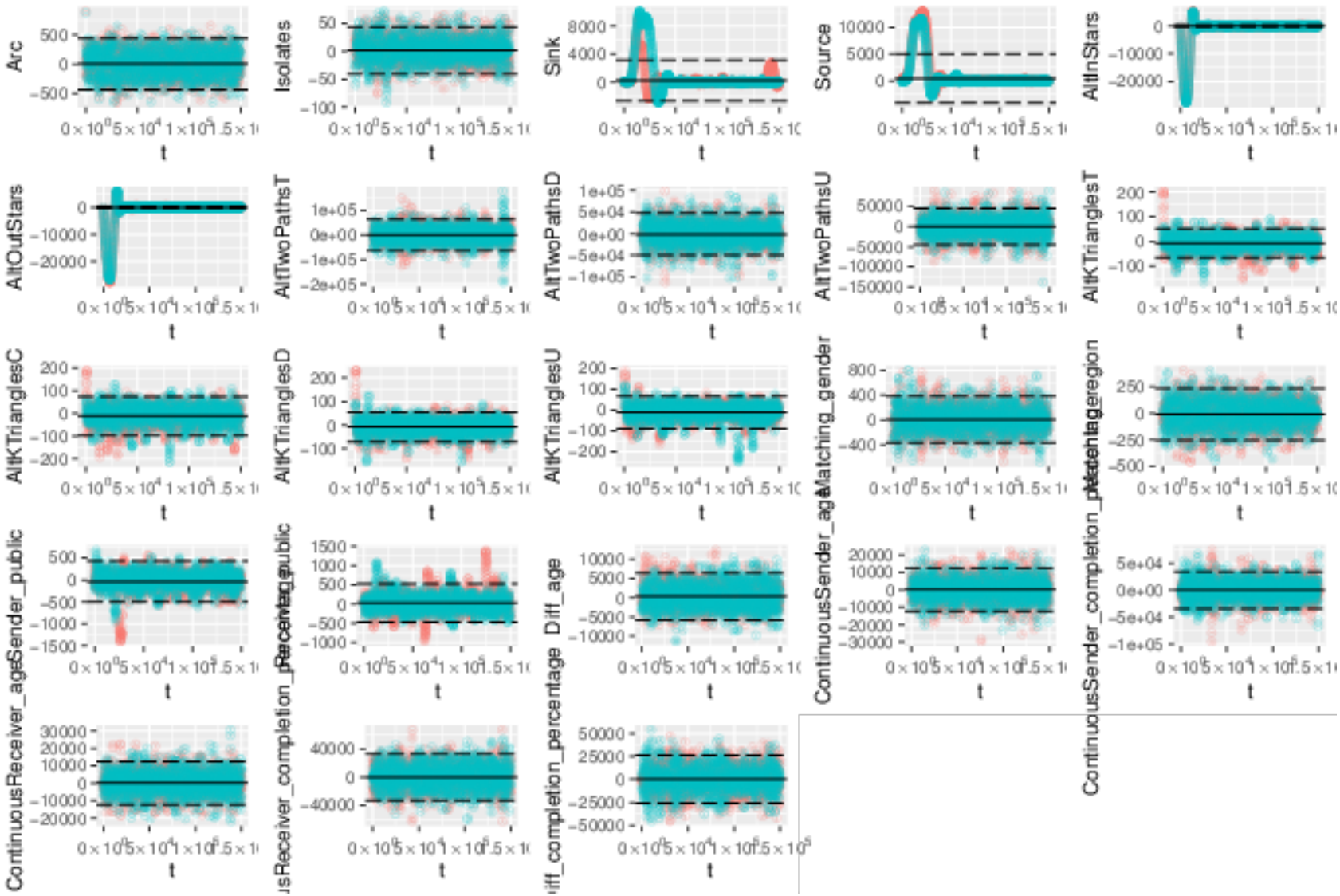
Future work and problems (substantive)

- As per Henderson, Jaffe, & Trajtenberg (2005), the question of what level of patent class /category / subcategory to match is not clear.
- In the matching techniques used by Jaffe, Trajtenberg, & Henderson (1993) and Thompson & Fox-Kean (2005) patents with the same assignee are excluded,
- but instead we include a model term to control for this, in the framework of ERGM terms which are nested (matching country, matching state, matching assignee).
- But perhaps as a robustness check it should be run on data with patents with the same assignees excluded?
- We do not include a matching term for same inventor, however, and perhaps this should be done (Henderson, Jaffe, & Trajtenberg, 2005)
- However this involves more "linking out" (data set matching).
- More importantly, perhaps patent citations do not even capture localization of knowledge transfers ("spillovers") at all (Arora, Belenzon, & Lee, 2018).

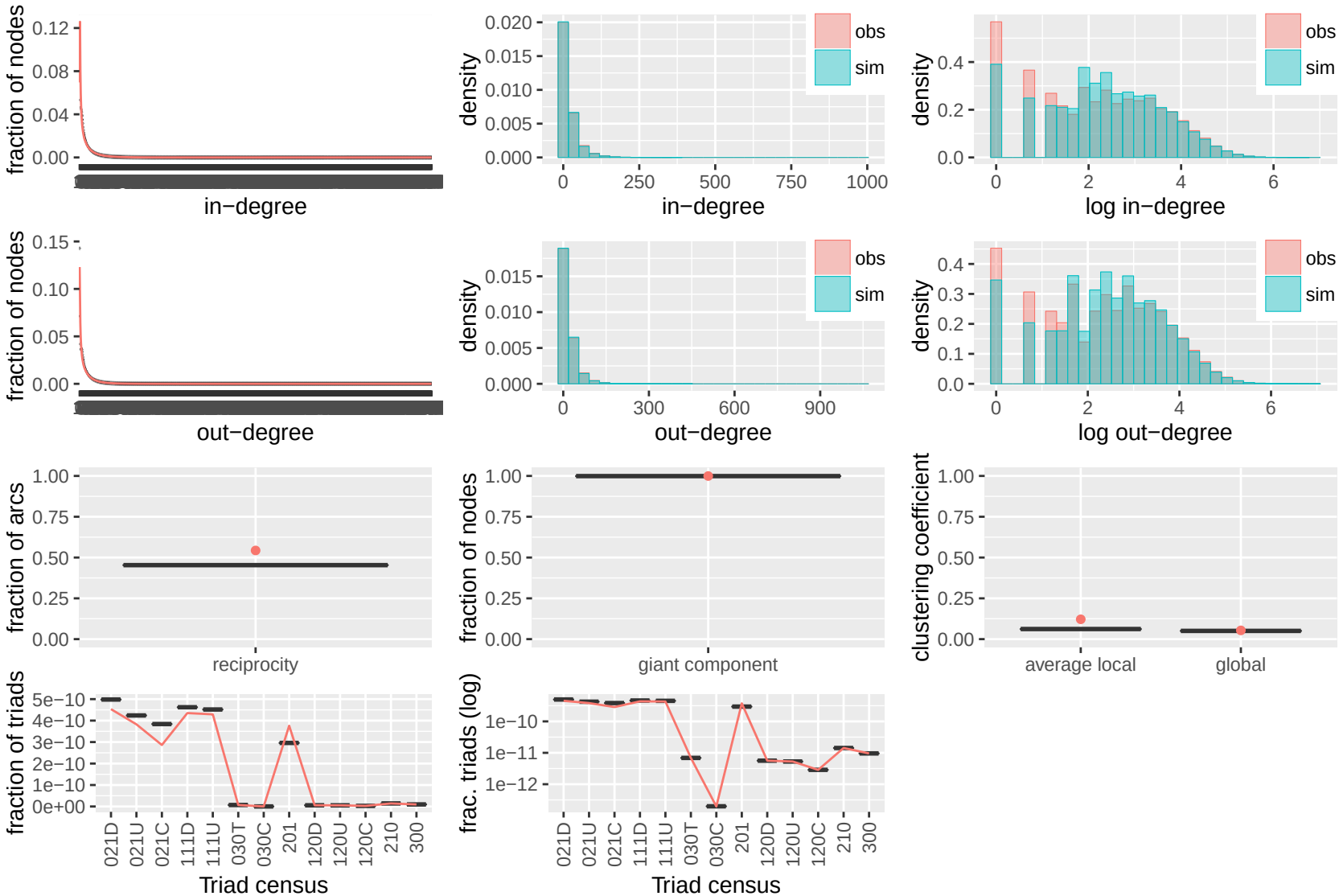
Pokec (no hubs) parameter trace



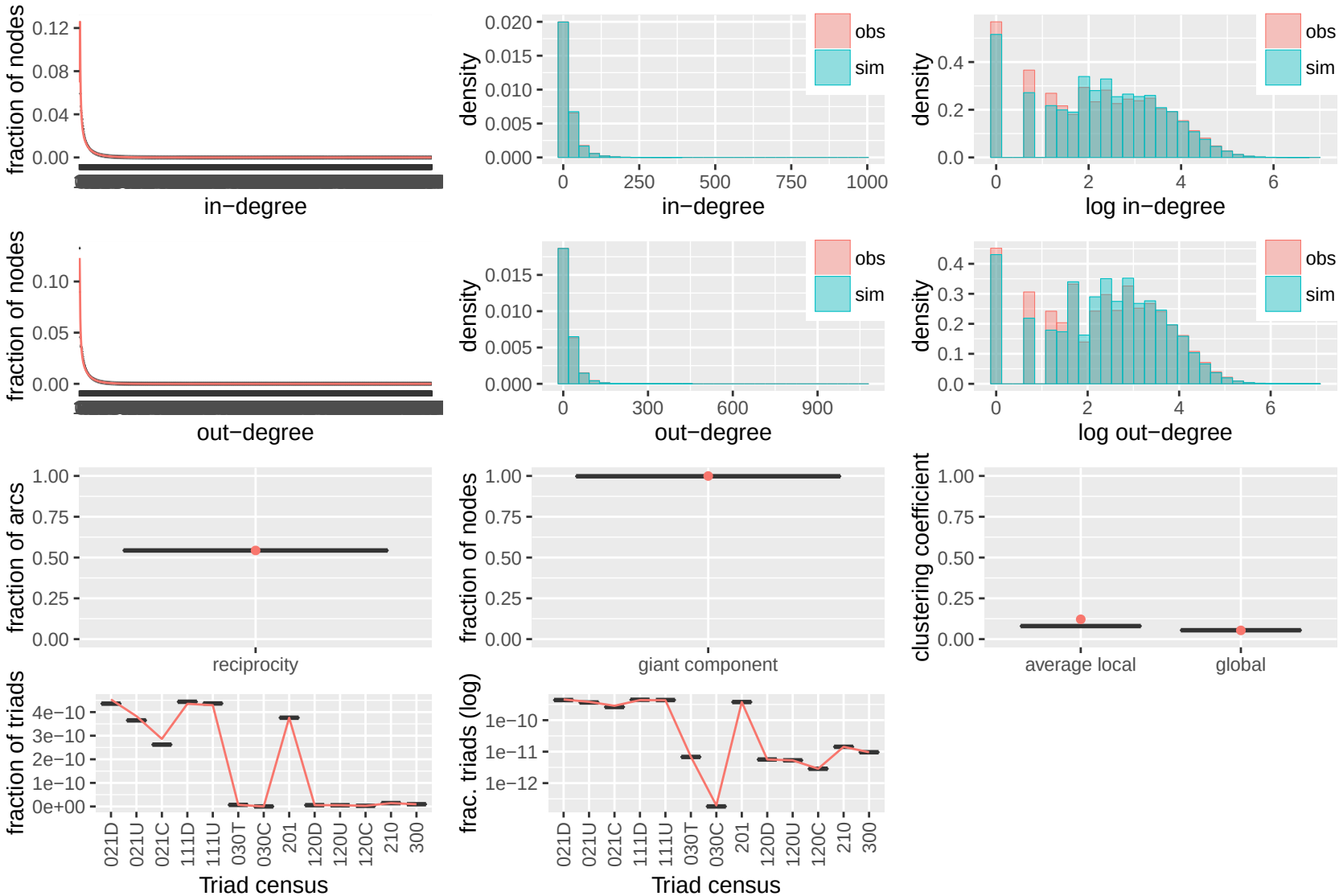
Pokec (no hubs) convergence trace



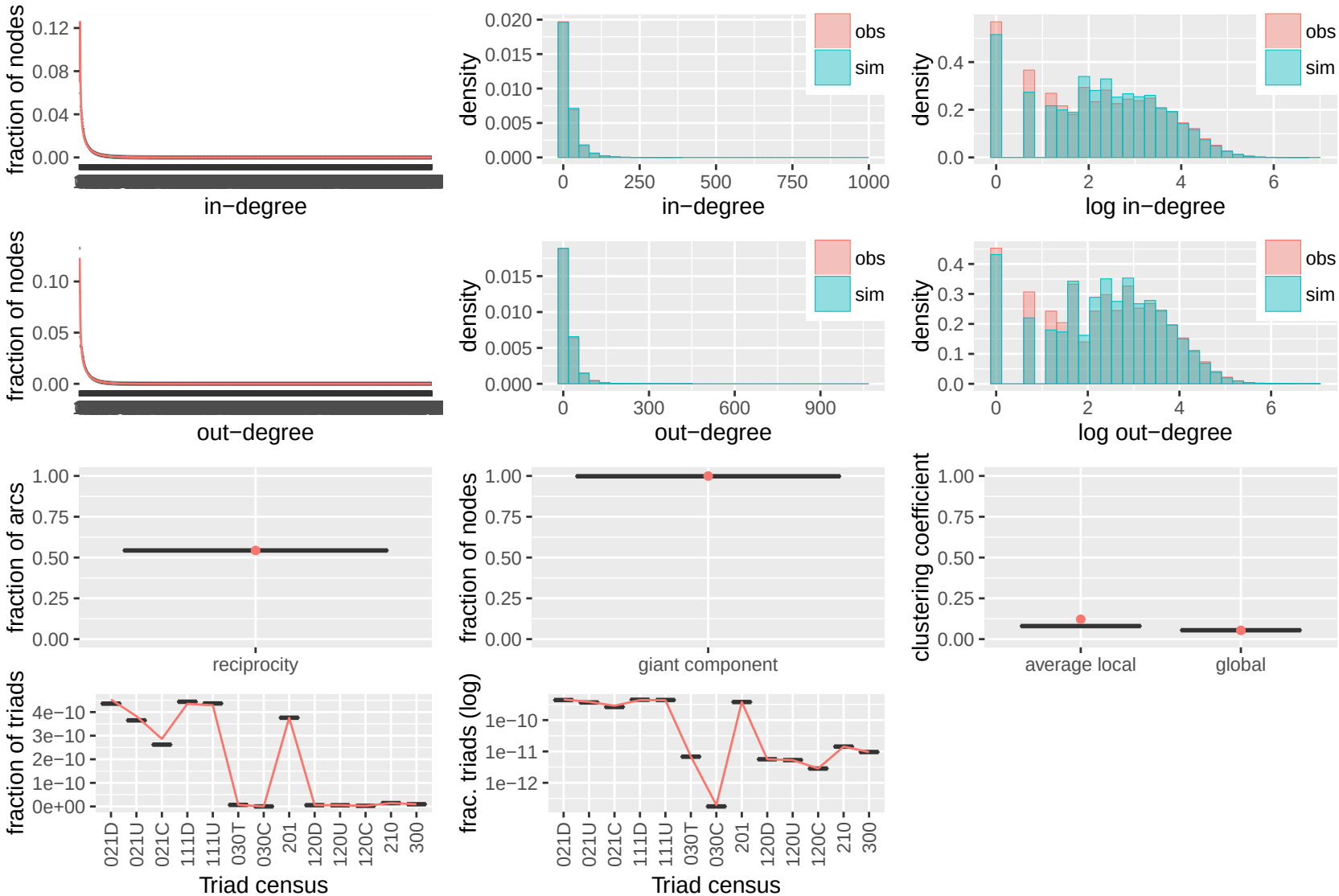
Pokec (no hubs) pseudo-GoF (model 1)



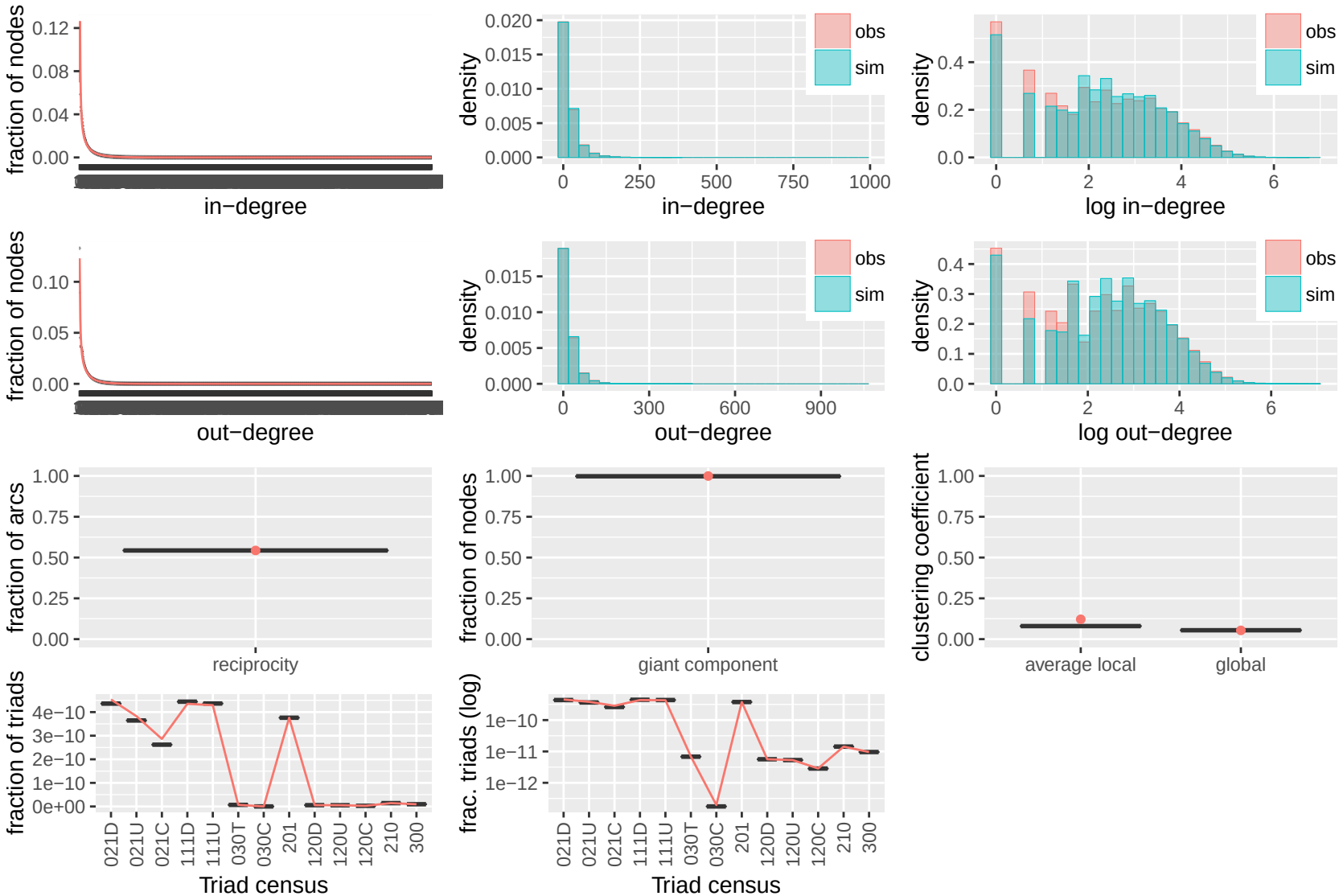
Pokec (no hubs) pseudo-GoF (model 2)



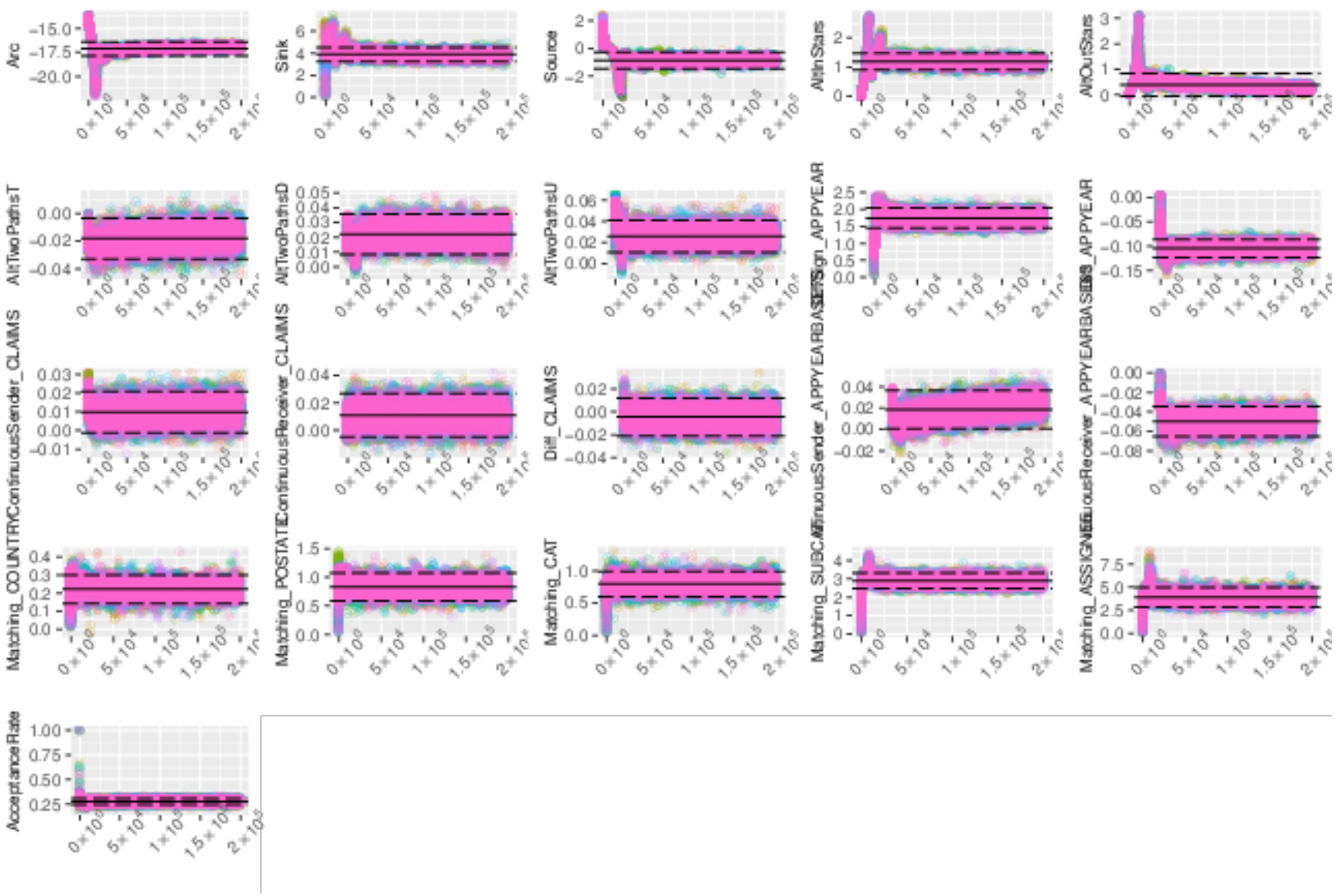
Pokec (no hubs) pseudo-GoF (model 3)



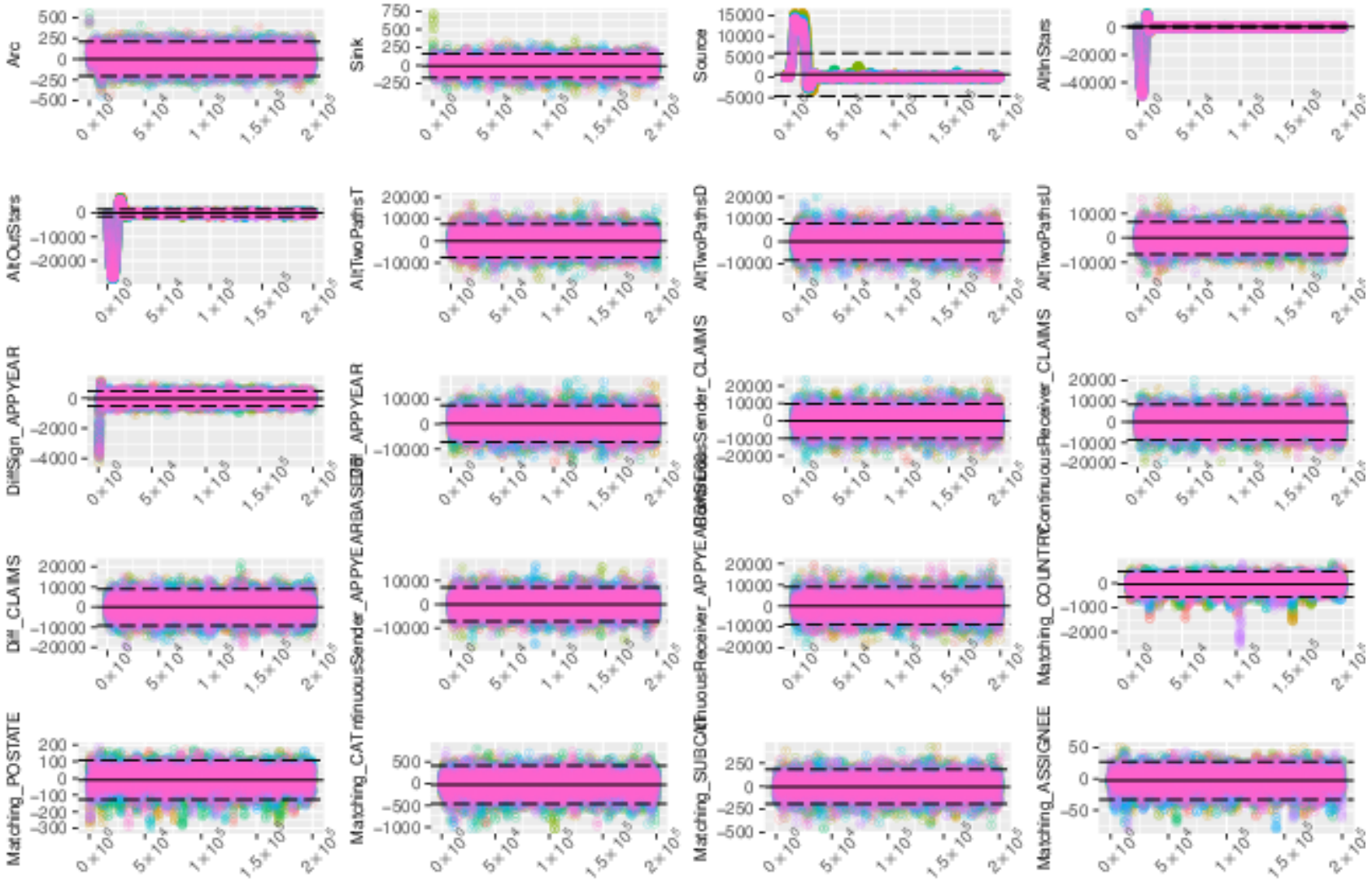
Pokec (no hubs) pseudo-GoF (model 4)



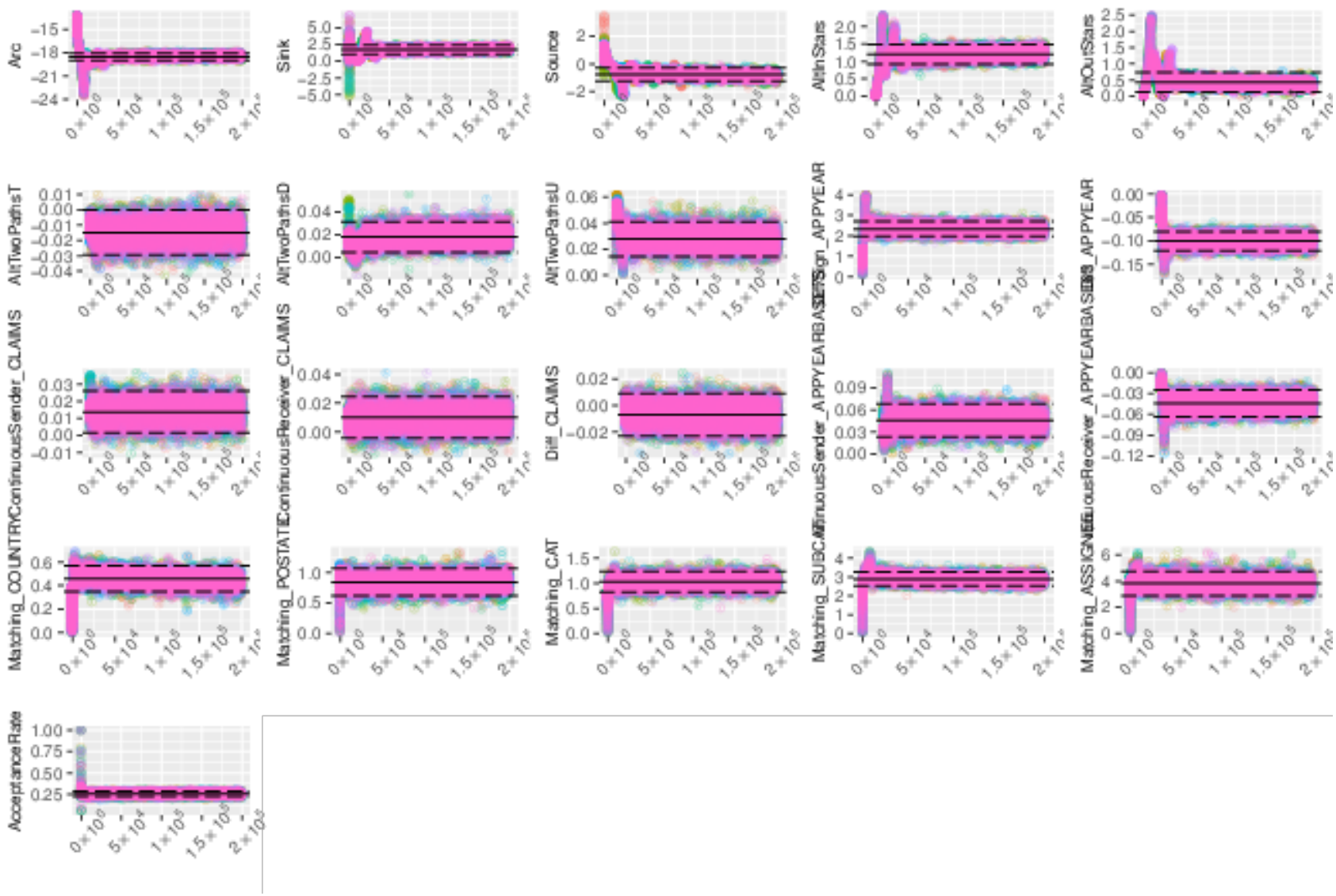
Patent parameter trace



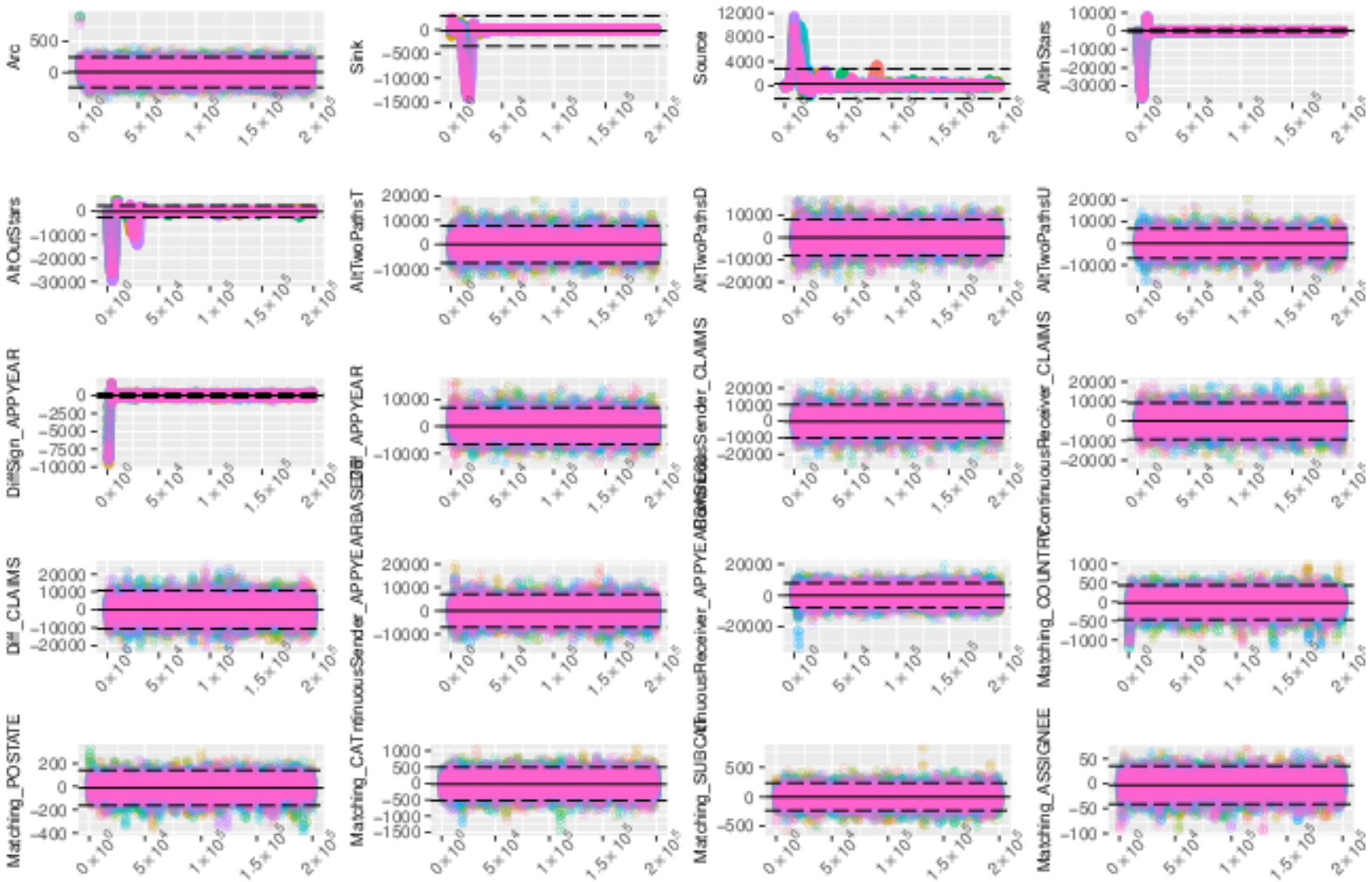
Patent convergence trace



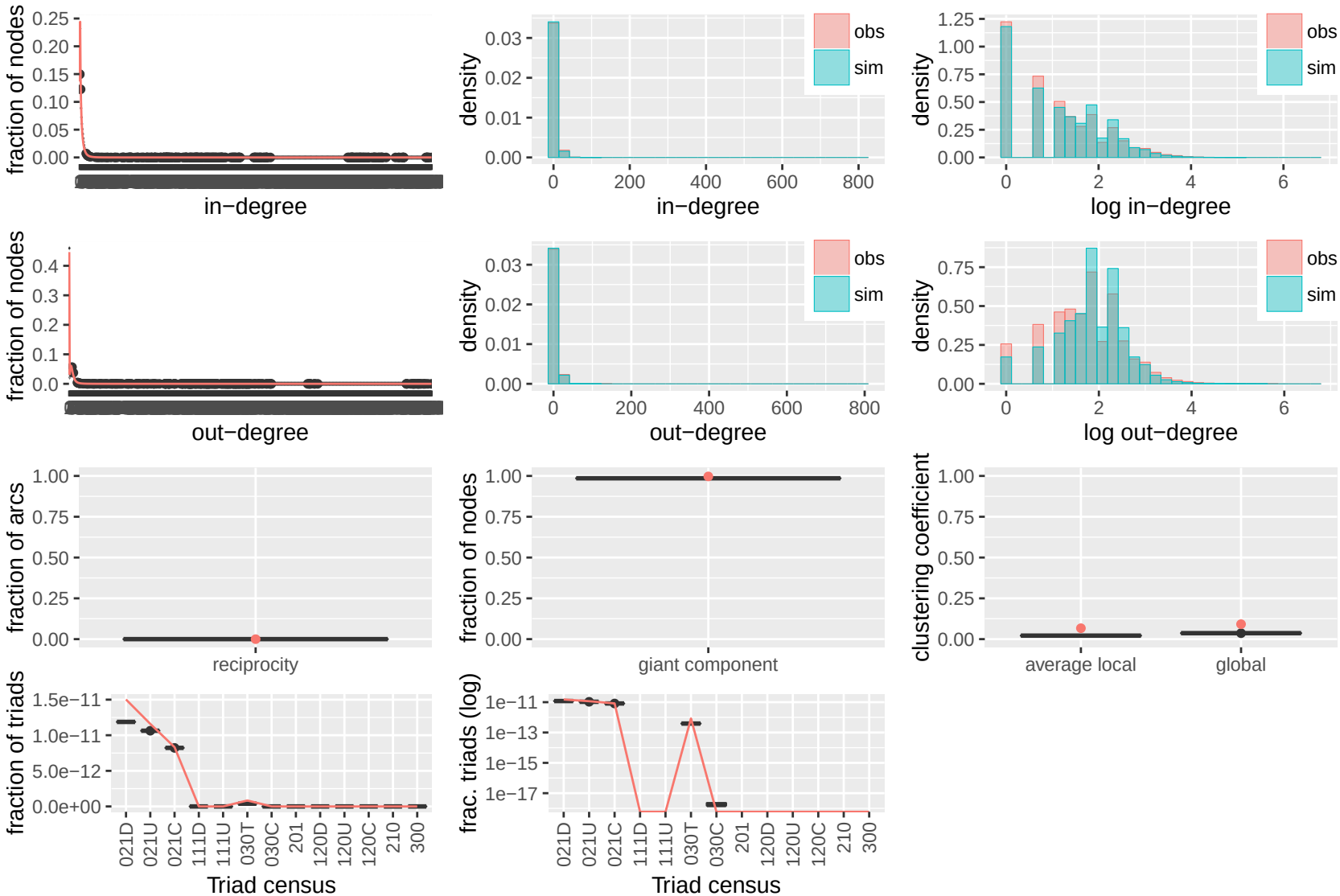
Patent parameter trace (subgraph)



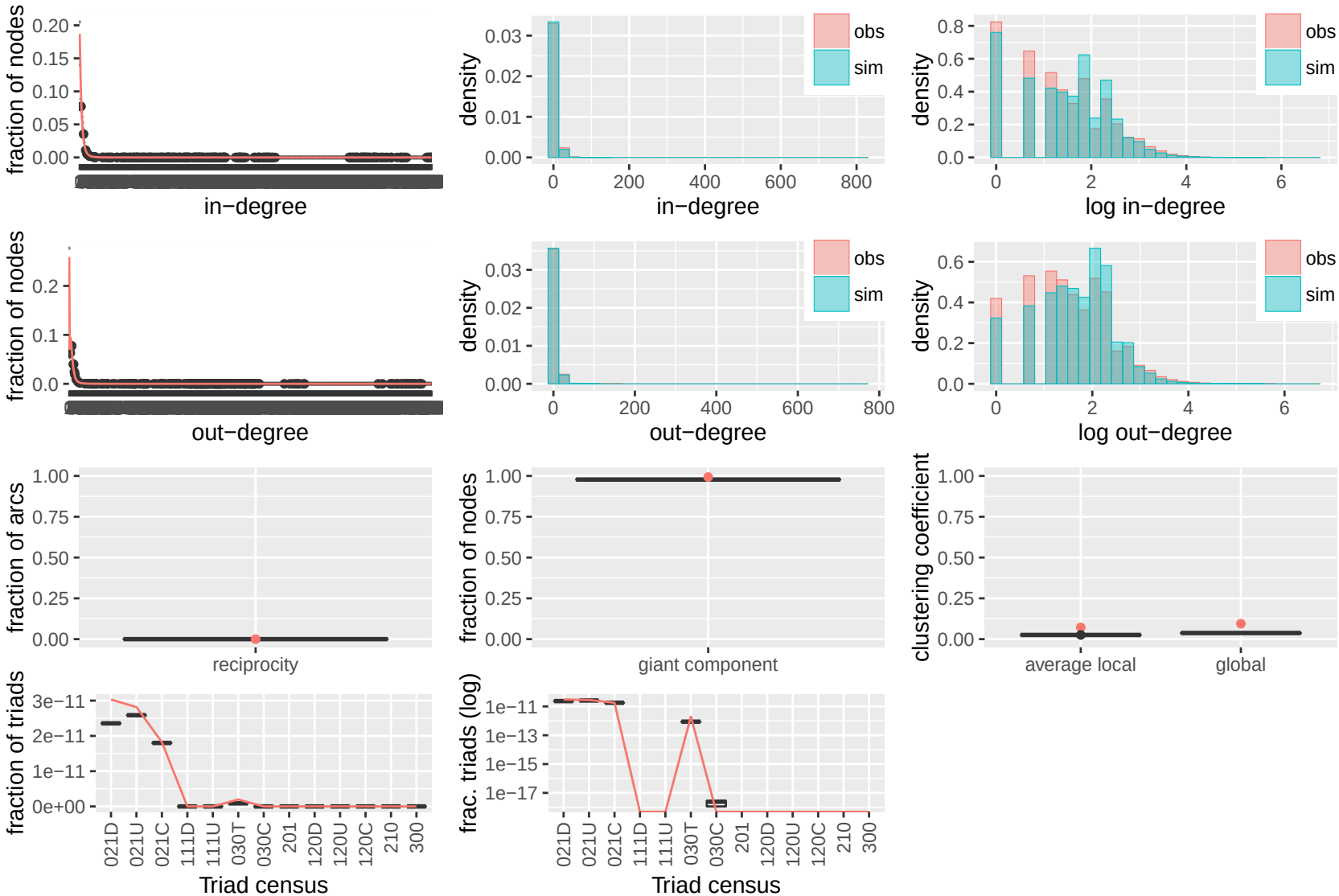
Patent convergence trace (subgraph)



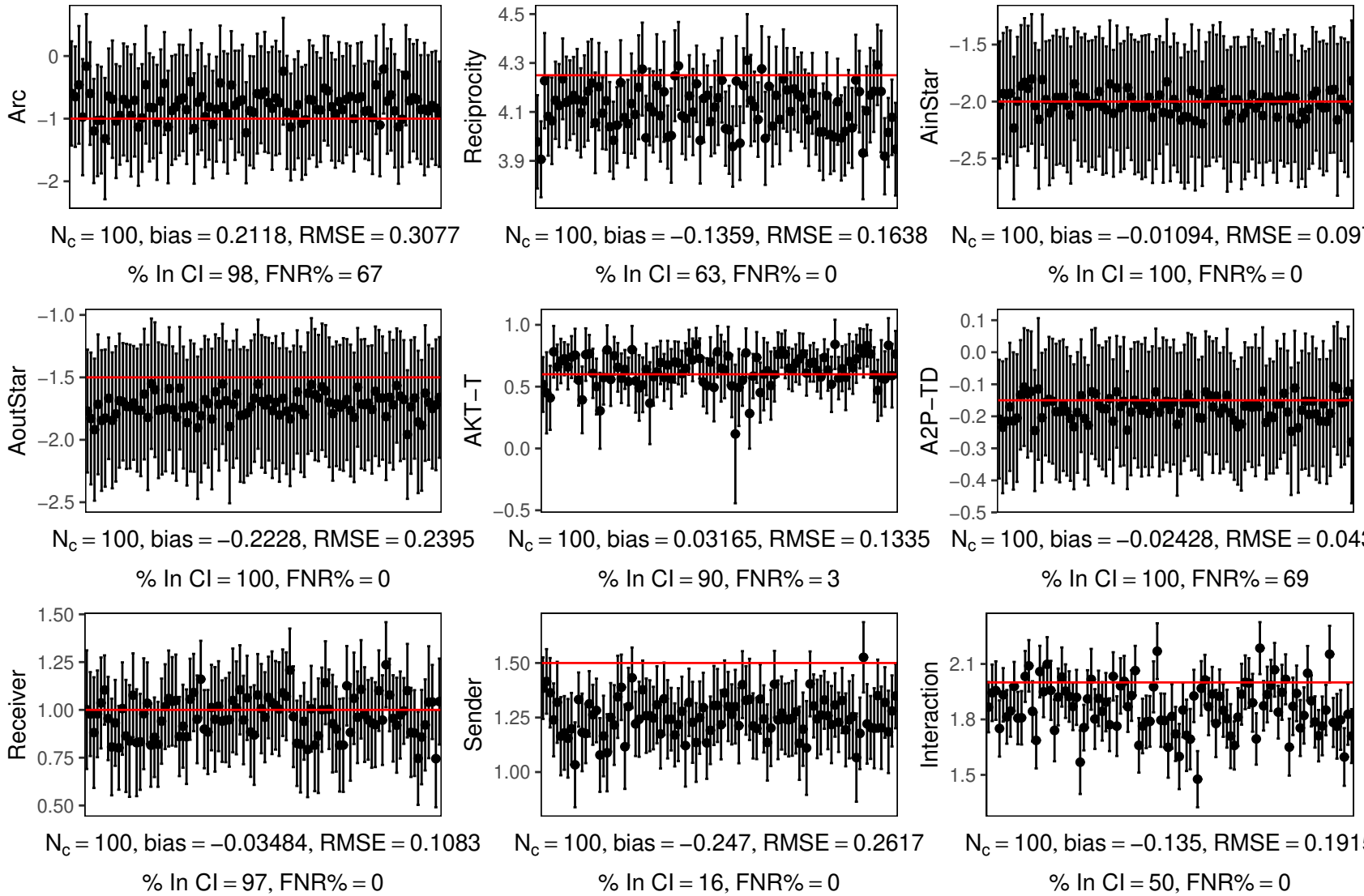
Patent citation pseudo-GoF



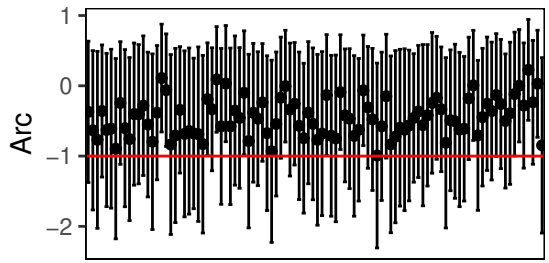
Patent citation (NA excluded subgraph) pseudo-GoF



Validation on simulated network $N = 2000$ binary attributes

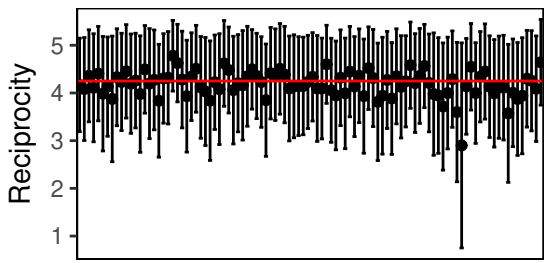


Validation on simulated network $N = 2000$ categorical attributes



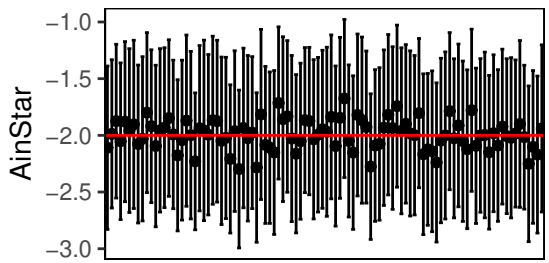
$N_c = 100$, bias = 0.5514, RMSE = 0.612

% In CI = 85, FNR% = 100



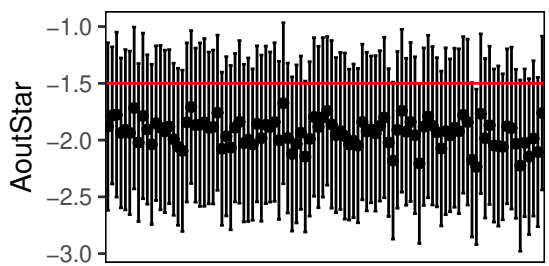
$N_c = 100$, bias = −0.06365, RMSE = 0.2745

% In CI = 100, FNR% = 0



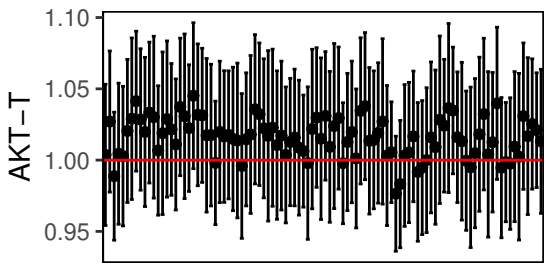
$N_c = 100$, bias = 0.006004, RMSE = 0.12

% In CI = 100, FNR% = 0



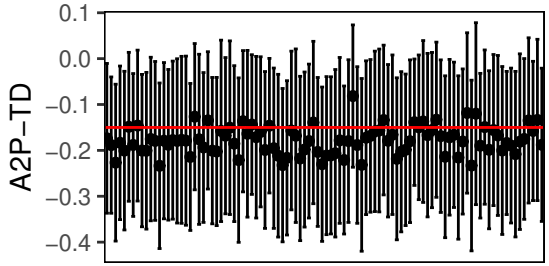
$N_c = 100$, bias = −0.4346, RMSE = 0.4506

% In CI = 98, FNR% = 0



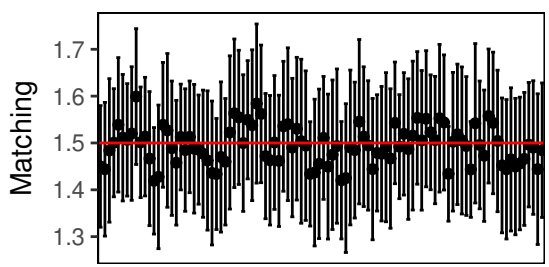
$N_c = 100$, bias = 0.01574, RMSE = 0.02084

% In CI = 100, FNR% = 0



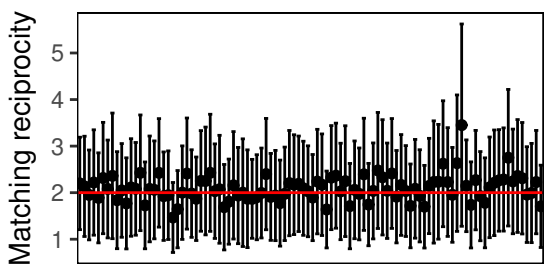
$N_c = 100$, bias = −0.02848, RMSE = 0.04

% In CI = 100, FNR% = 40



$N_c = 100$, bias = −0.005768, RMSE = 0.03965

% In CI = 100, FNR% = 0



$N_c = 100$, bias = 0.09297, RMSE = 0.2901

% In CI = 100, FNR% = 0

Type I error rate validation $N = 2000$ node simulated networks

N	Attr.	Effect	Bias	RMSE	False positive rate (%)			in C.I. (%)	Total networks converged	Mea run converge
					Estim.	95% C.I. lower	95% C.I. upper			
2000	Cat.	A2P-TD	-0.0217	0.0657	1	0	5	99	100	31.9
2000	Cat.	AinS	-0.0017	0.0648	1	0	5	99	100	32.0
2000	Cat.	AKT-T	-0.0154	0.0837	0	0	4	100	100	32.0
2000	Cat.	AoutS	-0.0129	0.0706	1	0	5	99	100	32.0
2000	Cat.	Match.Recip.	0.1246	0.1981	9	5	16	91	100	32.0
2000	Cat.	Reciprocity	0.4809	0.5493	2	1	7	98	100	30.8
2000	Bin.	A2P-TD	-0.0143	0.0198	2	1	7	98	100	32.0
2000	Bin.	AinS	-0.1234	0.1830	1	0	5	99	100	32.0
2000	Bin.	AKT-T	-0.2473	0.5563	1	0	5	99	100	29.3
2000	Bin.	AoutS	-0.0011	0.0954	0	0	4	100	100	32.0
2000	Bin.	Interaction	-0.7966	3.0590	4	1	15	96	46	7.0
2000	Bin.	Receiver	0.0313	0.1577	5	2	11	95	100	31.3
2000	Bin.	Reciprocity	-0.3127	1.2360	0	0	14	100	24	6.9
2000	Bin.	Sender	0.0244	0.1252	2	1	7	98	100	30.7

Institute of Computational Science (ICS) cluster details

USI ICS cluster `https://intranet.ics.usi.ch/HPC`

operating system CentOS 7.5 x86_64, OpenMPI

slim (22 nodes) 2 x Intel Xeon E5-2650 v3 @ 2.30GHz, 20 (2 x 10) cores, 64 GB RAM

fat (7 nodes) 2 x Intel Xeon E5-2650 v3 @ 2.30GHz, 20 (2 x 10) cores, 128 GB RAM

bigMem (2 nodes) 2 x Intel Xeon E5-2650 v3 @ 2.30GHz, 20 (2 x 10) cores, 512 GB RAM