SUPPLEMENTARY INFORMATION

for the manuscript "Robust reconstruction of causal graphical models based on conditional 2-point and 3-point information"

Séverine Affeldt, Hervé Isambert
Institut Curie, Research Center, UMR168, 26 rue d'Ulm, 75005, Paris France;
and Université Pierre et Marie Curie, 4 Place Jussieu, 75005, Paris, France
herve.isambert@curie.fr

SUPPLEMENTARY METHODS

Complexity of graphical models

The complexity $k_{\mathcal{G},\mathcal{D}}$ of a graphical model is related to the normalization constant $Z(\mathcal{G},\mathcal{D})$ of its maximum likelihood as $k_{\mathcal{G},\mathcal{D}} = \log Z(\mathcal{G},\mathcal{D})$,

$$\mathcal{L}_{\mathcal{G}} = \frac{e^{-NH(\mathcal{G},\mathcal{D})}}{Z(\mathcal{G},\mathcal{D})} = e^{-NH(\mathcal{G},\mathcal{D})-k_{\mathcal{G},\mathcal{D}}} \tag{1}$$

For Bayesian networks with decomposable entropy, *i.e.* $H(\mathcal{G},\mathcal{D}) = \sum_i H(X_i|\{\mathrm{Pa}_{X_i}\})$, it is convenient to use decomposable complexities, $k_{\mathcal{G},\mathcal{D}} = \sum_i k_{X_i|\{\mathrm{Pa}_{X_i}\}}$,

$$\mathcal{L}_{\mathcal{G}} = e^{-N\sum_i H(X_i|\{\mathrm{Pa}_{X_i}\})-\sum_i k_{X_i|\{\mathrm{Pa}_{X_i}\}}} \tag{2}$$

such that the comparison between alternative models $\mathcal{G}$ and $\mathcal{G}_{\setminus X\to Y}$ (*i.e.* $\mathcal{G}$ with one missing edge $X \to Y$) leads to a simple local increment of the score,

$$\frac{\mathcal{L}_{\mathcal{G}_{\setminus X\to Y}}}{\mathcal{L}_{\mathcal{G}}} = e^{-NI(X;Y|\{\mathrm{Pa}_Y\}_{\setminus X})+\Delta k_{Y|\{\mathrm{Pa}_Y\}_{\setminus X}}} \tag{3}$$

$$I(X;Y|\{\mathrm{Pa}_Y\}_{\setminus X}) = H(Y|\{\mathrm{Pa}_Y\}_{\setminus X}) - H(Y|\{\mathrm{Pa}_Y\}) \geqslant 0$$
$$\Delta k_{Y|\{\mathrm{Pa}_Y\}_{\setminus X}} = k_{Y|\{\mathrm{Pa}_Y\}} - k_{Y|\{\mathrm{Pa}_Y\}_{\setminus X}} \geqslant 0$$

A common complexity criteria in model selection is the Bayesian Information Criteria (BIC) or Minimal Description Length (MDL) criteria (Rissanen, 1978; Hansen and Yu, 2001),

$$k^{\mathsf{MDL}}_{Y|\{\mathrm{Pa}_Y\}} = \frac{1}{2}(r_y-1)\prod_j^{\mathrm{Pa}_Y} r_j \;\log N \tag{4}$$

$$\Delta k^{\mathsf{MDL}}_{Y|\{\mathrm{Pa}_Y\}_{\setminus X}} = \frac{1}{2}(r_x-1)(r_y-1)\prod_j^{\mathrm{Pa}_{y_{\setminus X}}} r_j \;\log N \tag{5}$$

where r_x, r_y and r_j are the number of levels of each variable, x, y and j. The MDL complexity, Eq. 4, is simply related to the normalisation constant reached in the asymptotic limit of a large dataset $N \to \infty$ (Laplace approximation). The MDL complexity can also be derived from the Stirling approximation on the Bayesian measure (Schwarz, 1978; Bouckaert, 1993). Yet, in practice, this limit distribution is only reached for very large datasets, as some of the least-likely $(r_y-1)\prod_j r_j$ combinations of states of variables are in fact rarely (if ever) sampled in typical finite datasets. As a result, the MDL complexity criteria tends to underestimate the relevance of edges connecting variables with many levels, r_i, leading to the removal of false negative edges.

To avoid such biases with finite datasets, the normalisation of the maximum likelihood can be done over all possible datasets with the same number N of data points. This corresponds to the (universal) Normalized Maximum Likelihood (NML) criteria (Shtarkov, 1987; Rissanen and Tabus, 2005; Kontkanen and Myllymäki, 2007; Roos et al., 2008),

$$\mathcal{L}_{\mathcal{G}} = \frac{e^{-NH(\mathcal{G},\mathcal{D})}}{\sum_{|\mathcal{D}'|=N} e^{-NH(\mathcal{G},\mathcal{D}')}} = e^{-NH(\mathcal{G},\mathcal{D})-k^{\mathsf{NML}}_{\mathcal{G},\mathcal{D}}} \tag{6}$$

We introduce here the factorized version of the NML criteria (Kontkanen and Myllymäki, 2007; Roos et al., 2008) which corresponds to a decomposable NML score, $k^{\mathsf{NML}}_{\mathcal{G},\mathcal{D}} = \sum_{X_i} k^{\mathsf{NML}}_{X_i|\{\mathrm{Pa}_{X_i}\}}$, defined as,

$$k^{\mathsf{NML}}_{Y|\{\mathrm{Pa}_Y\}} = \sum_j^{q_y} \log \mathcal{C}^{r_y}_{N_{yj}} \tag{7}$$

$$\Delta k^{\mathsf{NML}}_{Y|\{\mathrm{Pa}_Y\}_{\setminus X}} = \sum_j^{q_y} \log \mathcal{C}^{r_y}_{N_{yj}} - \sum_{j'}^{q_y/r_x} \log \mathcal{C}^{r_y}_{N_{yj'}} \tag{8}$$

where N_{yj} is the number of data points corresponding to the jth state of the parents of Y, $\{\mathrm{Pa}_Y\}$, and $N_{yj'}$ the number of data points corresponding to the j'th state of the parents of Y, excluding X, $\{\mathrm{Pa}_Y\}_{\setminus X}$. Hence, the factorized NML score for each node X_i corresponds to a separate normalisation for each state $j = 1, ..., q_i$ of its parents and involving exactly N_{ij} data points of the finite dataset,

$$\mathcal{L}_{\mathcal{G}} = e^{-N\sum_i H(X_i|\{\mathrm{Pa}_{X_i}\})-\sum_i\sum_j^{q_i}\log\mathcal{C}_{N_{ij}}^{r_i}} \tag{9}$$

$$= e^{N\sum_i\sum_j^{q_i}\sum_k^{r_i}\frac{N_{ijk}}{N}\log\left(\frac{N_{ijk}}{N_{ij}}\right)-\sum_i\sum_j^{q_i}\log\mathcal{C}_{N_{ij}}^{r_i}} \tag{10}$$

$$= \prod_i\prod_j^{q_i}\frac{\prod_k^{r_i}\left(\frac{N_{ijk}}{N_{ij}}\right)^{N_{ijk}}}{\mathcal{C}_{N_{ij}}^{r_i}} \tag{11}$$

where N_{ijk} corresponds to the number of data points for which the ith node is in its kth state and its parents in their jth state, with $N_{ij} = \sum_k^{r_i} N_{ijk}$. The universal normalization constant $\mathcal{C}_n^r$ is then obtained by averaging over all possible partitions of the n data points into a maximum of r subsets, $\ell_1 + \ell_2 + \cdots + \ell_r = n$ with $\ell_k \geqslant 0$,

$$\mathcal{C}_n^r = \sum_{\ell_1+\ell_2+\cdots+\ell_r=n}\frac{n!}{\ell_1!\ell_2!\cdots\ell_r!}\prod_{k=1}^r\left(\frac{\ell_k}{n}\right)^{\ell_k} \tag{12}$$

which can in fact be computed in linear-time using the following recursion (Kontkanen and Myllymäki, 2007),

$$\mathcal{C}_n^r = \mathcal{C}_n^{r-1} + \frac{n}{r-2}\mathcal{C}_n^{r-2} \tag{13}$$

with $\mathcal{C}_0^r = 1$ for all r, $\mathcal{C}_n^1 = 1$ for all n and applying the general formula Eq. 12 for $r = 2$,

$$\mathcal{C}_n^2 = \sum_{h=0}^n\binom{n}{h}\left(\frac{h}{n}\right)^h\left(\frac{n-h}{n}\right)^{n-h} \tag{14}$$

or its Szpankowski approximation for large n (needed for $n > 1000$ in practice) (Szpankowski, 2001; Kontkanen et al., 2003; Kontkanen, 2009),

$$\mathcal{C}_n^2 = \sqrt{\frac{n\pi}{2}}\left(1 + \frac{2}{3}\sqrt{\frac{2}{n\pi}} + \frac{1}{12n} + \mathcal{O}\left(\frac{1}{n^{3/2}}\right)\right) \tag{15}$$

$$\simeq \sqrt{\frac{n\pi}{2}}\exp\left(\sqrt{\frac{8}{9n\pi}} + \frac{3\pi-16}{36n\pi}\right) \tag{16}$$

Then, following the rationale of constraint-based approaches, we can reformulate the likelihood ratio of Eq. 3 by replacing the parent nodes $\{\mathrm{Pa}_Y\}_{\backslash X}$ in the conditional mutual information, $I(X;Y|\{\mathrm{Pa}_Y\}_{\backslash X})$, with an unknown separation set $\{U_i\}$ to be learnt simultaneously with the missing edge candidate XY,

$$\frac{\mathcal{L}_{\mathcal{G}_{\backslash XY|\{U_i\}}}}{\mathcal{L}_{\mathcal{G}}} = e^{-NI(X;Y|\{U_i\})+k_{X;Y|\{U_i\}}} \tag{17}$$

where we have also transformed the asymmetric parent-dependent complexity difference, $\Delta k_{Y|\{\mathrm{Pa}_Y\}_{\backslash X}}$, into a $\{U_i\}$-dependent complexity term, $k_{X;Y|\{U_i\}}$, with the same XY-symmetry as $I(X;Y|\{U_i\})$,

$$k_{X;Y|\{U_i\}}^{\mathsf{MDL}} = \frac{1}{2}(r_x-1)(r_y-1)\prod_i r_{u_i}\log N \tag{18}$$

$$k_{X;Y|\{U_i\}}^{\mathsf{NML}} = \frac{1}{2}\sum_{j'}^{\{U_i\}}\Big(\sum_{k_x}^{r_x}\log\mathcal{C}_{N_{k_xj'}}^{r_y} - \log\mathcal{C}_{N_{j'}}^{r_y}$$

$$+ \sum_{k_y}^{r_y}\log\mathcal{C}_{N_{k_yj'}}^{r_x} - \log\mathcal{C}_{N_{j'}}^{r_x}\Big) \tag{19}$$

Note, in particular, that the MDL complexity term in Eq. 18 is readily obtained from Eq. 5 due to the Markov equivalence of the MDL score, corresponding to its XY-symmetry whenever $\{\mathrm{Pa}_Y\}_{\backslash X} = \{\mathrm{Pa}_X\}_{\backslash Y}$. By contrast, the factorized NML score, Eq. 7, is not a Markov-equivalent score (although its non-factorized version, Eq. 6, is Markov equivalent by definition). To circumvent this non-equivalence of factorized NML score, we propose to recover the expected XY-symmetry of $k_{X;Y|\{U_i\}}^{\mathsf{NML}}$ through the simple XY-symmetrization of Eq. 8, leading to Eq. 19.

References

Bouckaert, R. R. 1993. Probabilistic network construction using the minimum description length principle. *in Symbolic and Quantitative Approaches to Reasoning and Uncertainty (Clarke M, Kruse R, Moral S, eds)* 747:41–48.

Colombo, D., and Maathuis, M. H. 2014. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research* 15:3741–3782.

Hansen, M. H., and Yu, B. 2001. Model selection and the principle of minimum description length. *Journal of the American Statistical Association* 96:746–774.

Kontkanen, P., and Myllymäki, P. 2007. A linear-time algorithm for computing the multinomial stochastic complexity. *Inf. Process. Lett.* 103(6):227–233.

Kontkanen, P.; Buntine, W.; Myllymäki, P.; Rissanen, J.; and Tirri, H. 2003. Efficient computation of stochastic complexity. *in: C. Bishop, B. Frey (Eds.) Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics, Society for Artificial Intelligence and Statistics* 103:233–238.

Kontkanen, P. 2009. *Computationally Efficient Methods for MDL-Optimal Density Estimation and Data Clustering*. Ph.D. Dissertation.

Rissanen, J., and Tabus, I. 2005. Kolmogorovs structure function in mdl theory and lossy data compression. In *Adv. Min. Descrip. Length Theory Appl.* MIT Press. Chap. 10.

Rissanen, J. 1978. Modeling by shortest data description. *Automatica* vol. 14:465–471.

Roos, T.; Silander, T.; Kontkanen, P.; and Myllymäki, P. 2008. Bayesian network structure learning using factorized nml universal models. In *Proc. 2008 Information Theory and Applications Workshop (ITA-2008)*. IEEE Press. invited paper.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics* 6:461–464.

Shtarkov, Y. M. 1987. Universal sequential coding of single messages. *Problems of Information Transmission (Translated from)* 23(3):3–17.

Szpankowski, W. 2001. *Average case analysis of algorithms on sequences*. John Wiley & Sons.

Tsamardinos, I.; Brown, L. E.; and Aliferis, C. F. 2006. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning* 65(1):31–78.

nodes	edges	$\langle k\rangle$	Model	$\langle k_{\mathrm{max}}\rangle$	$\langle k^{in}_{\mathrm{max}}\rangle$	$\langle k^{out}_{\mathrm{max}}\rangle$	N	Replicates
50	**20**	0.8	*1*	4	2	3	$[50-50,000]$	20
			2	4	2	2	$[50-50,000]$	20
			3	3	3	2	$[50-50,000]$	20
			4	3	3	2	$[50-50,000]$	20
			5	3	2	2	$[50-50,000]$	20
			Avg.	**3.4**	**2.4**	**2.2**		
50	**40**	1.6	*1*	5	3	5	$[50-50,000]$	20
			2	6	3	3	$[50-50,000]$	20
			3	5	3	3	$[50-50,000]$	20
			4	4	4	4	$[50-50,000]$	20
			5	5	3	3	$[50-50,000]$	20
			Avg.	**5**	**3.2**	**3.6**		
50	**60**	2.4	*1*	7	5	3	$[50-50,000]$	20
			2	6	6	3	$[50-50,000]$	20
			3	6	4	4	$[50-50,000]$	20
			4	6	5	3	$[50-50,000]$	20
			5	7	3	5	$[50-50,000]$	20
			Avg.	**6.4**	**4.6**	**3.6**		
50	**80**	3.2	*1*	7	5	7	$[50-50,000]$	20
			2	7	5	5	$[50-50,000]$	20
			3	6	5	5	$[50-50,000]$	20
			4	6	5	6	$[50-50,000]$	20
			5	6	4	5	$[50-50,000]$	20
			Avg.	**6.4**	**4.8**	**5.6**		
50	**120**	4.8	*1*	10	10	7	$[50-50,000]$	20
			2	13	10	7	$[50-50,000]$	20
			3	9	6	8	$[50-50,000]$	20
			4	13	9	7	$[50-50,000]$	20
			5	12	9	7	$[50-50,000]$	20
			Avg.	**11.4**	**8.8**	**7.2**		
50	**160**	6.4	*1*	12	10	9	$[50-50,000]$	20
			2	13	9	9	$[50-50,000]$	20
			3	14	7	9	$[50-50,000]$	20
			4	11	7	8	$[50-50,000]$	20
			5	11	10	8	$[50-50,000]$	20
			Avg.	**12.2**	**8.6**	**8.6**		

Table S1: **Description summary of the 30 benchmark networks used to evaluate the reconstruction methods.** The 30 benchmark networks of 50 nodes, and 20 to 160 edges, have been instantiated with the causal modeling tool Tetrad IV (http://www.phil.cmu.edu/tetrad/). For each model, 20 dataset replicates of size ranging between 50 and 50,000 were generated with Tetrad IV.

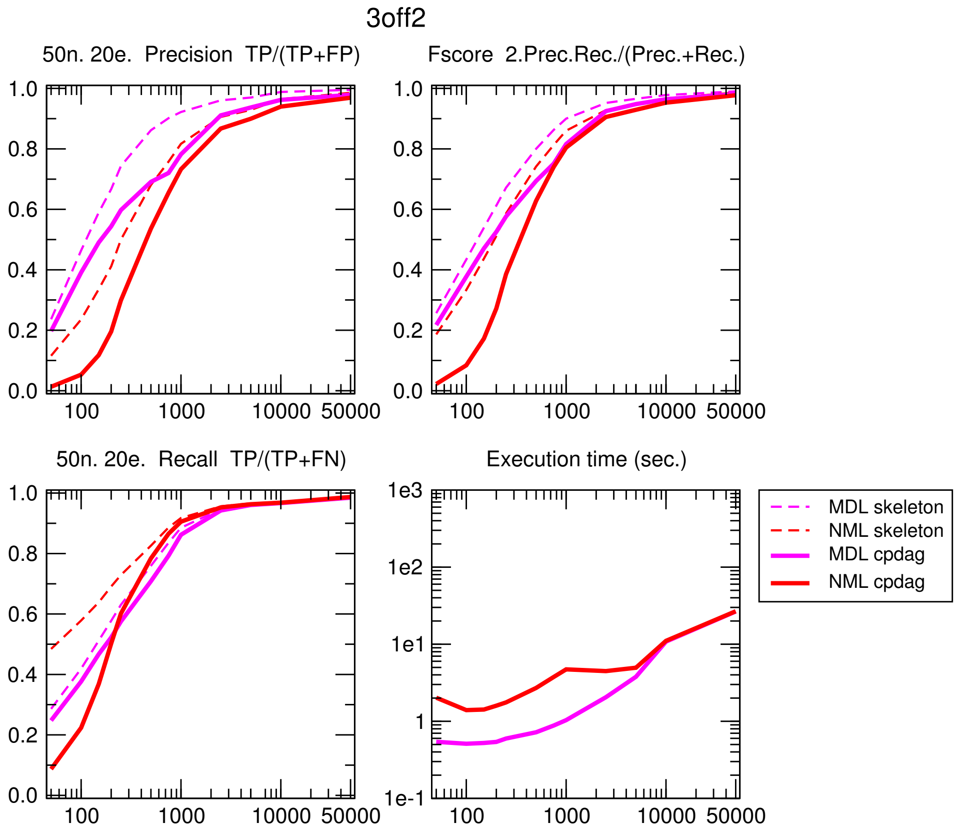

Figure S1: 3off2 **reconstruction, effect of complexity MDL and NML.** 50 node, 20 edge benchmark networks generated using Tetrad. $\langle k \rangle = 0.8$, $\langle k^{in}_{\max} \rangle = 2.4$ and $\langle k^{out}_{\max} \rangle = 2.2$. The change of slope in execution time at sample size N=1000 for NML corresponds to the use of the Szpankowski approximation (see Supplementary Methods).

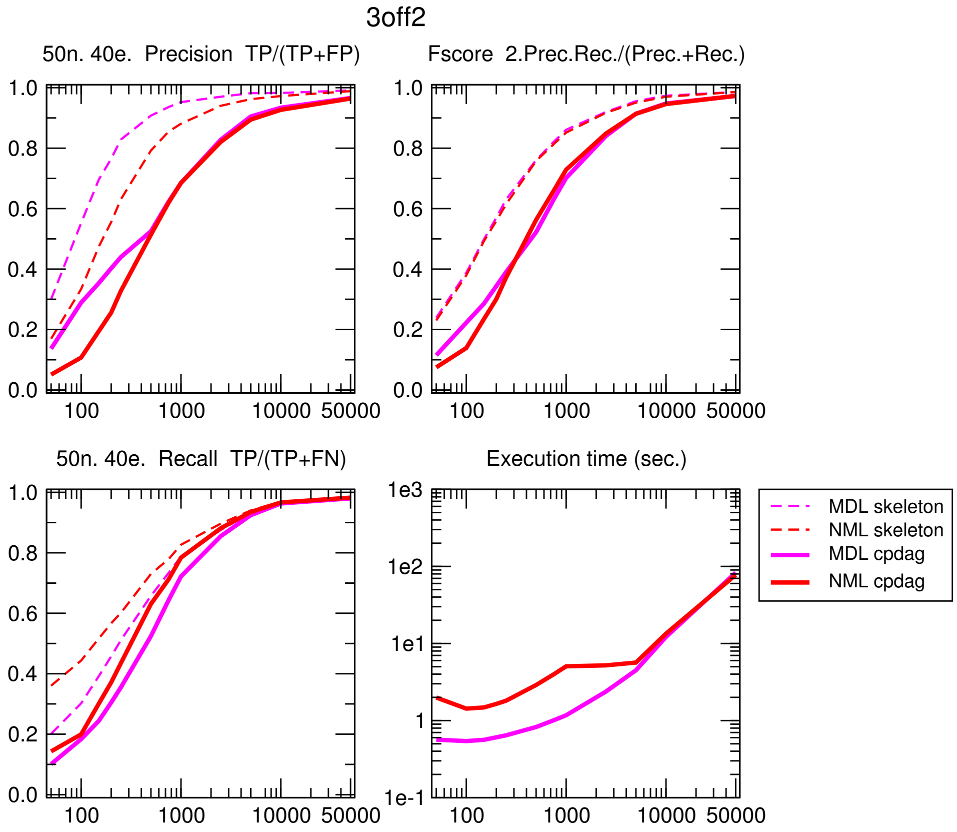

Figure S2: 3off2 **reconstruction, effect of complexity MDL and NML.** 50 node, 40 edge benchmark networks generated using Tetrad. $\langle k \rangle = 1.6$, $\langle k^{in}_{\max} \rangle = 3.2$ and $\langle k^{out}_{\max} \rangle = 3.6$. The change of slope in execution time at sample size N=1000 for NML corresponds to the use of the Szpankowski approximation (see Supplementary Methods).

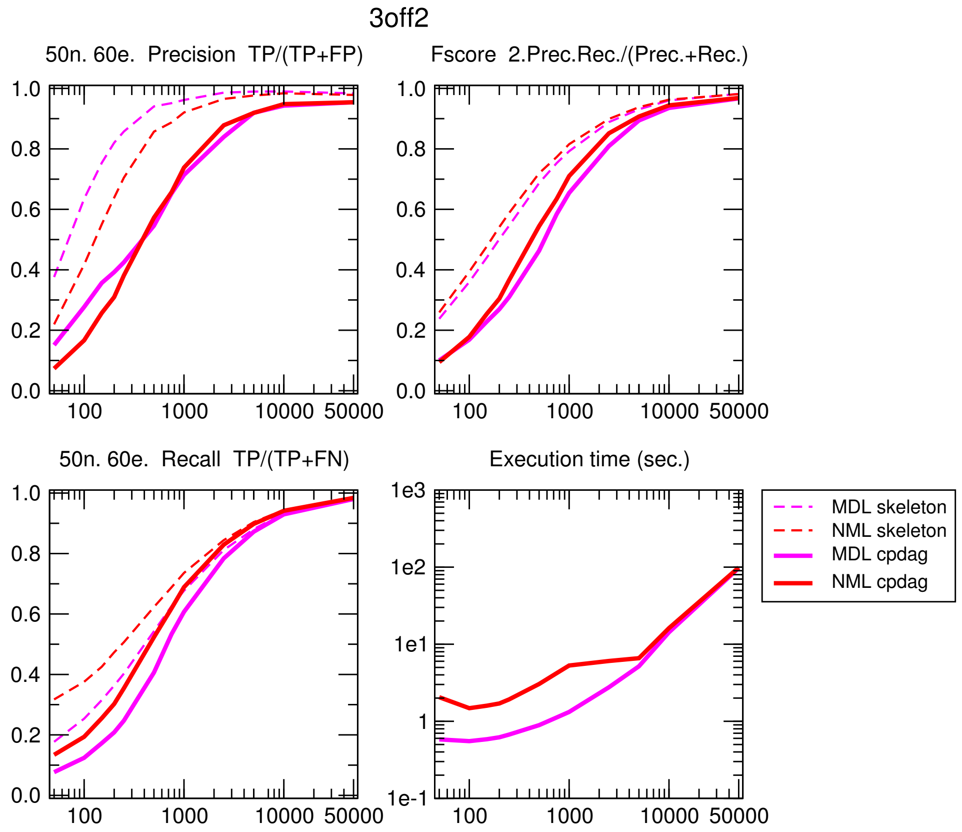

Figure S3: 3off2 **reconstruction, effect of complexity MDL and NML.** 50 node, 60 edge benchmark networks generated using Tetrad. $\langle k \rangle = 2.4$, $\langle k_{\max}^{in} \rangle = 4.6$ and $\langle k_{\max}^{out} \rangle = 3.6$. The change of slope in execution time at sample size N=1000 for NML corresponds to the use of the Szpankowski approximation (see Supplementary Methods).

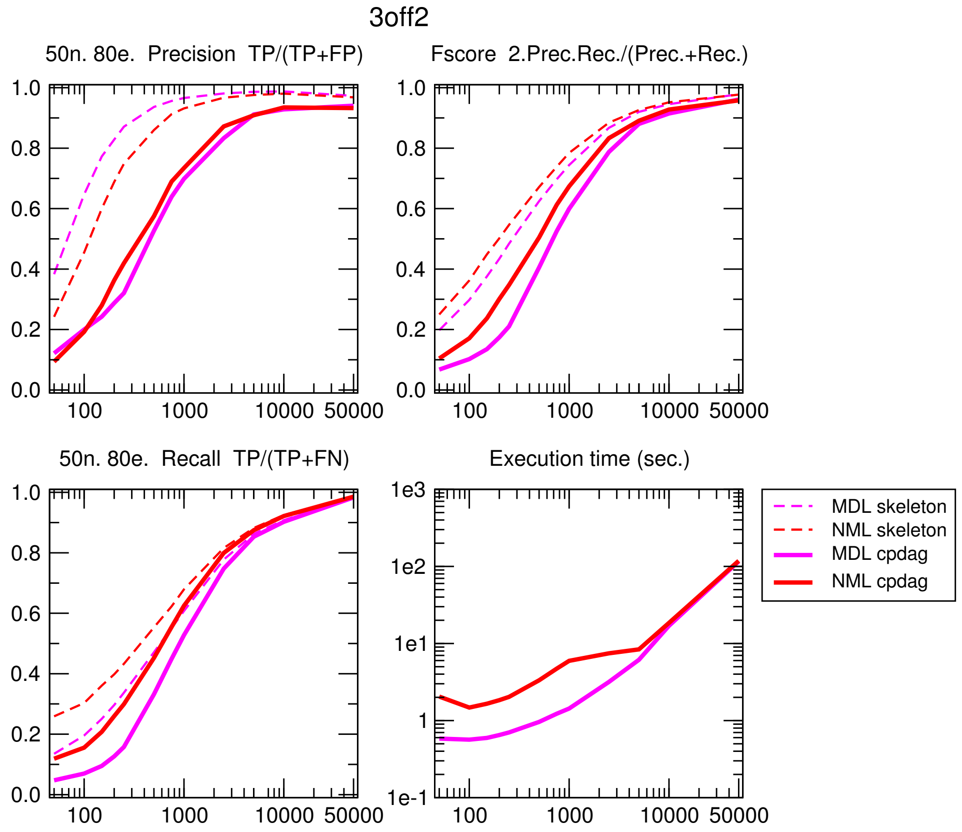

Figure S4: 3off2 **reconstruction, effect of complexity MDL and NML.** 50 node, 80 edge benchmark networks generated using Tetrad. $\langle k \rangle = 3.2$, $\langle k_{\max}^{in} \rangle = 4.8$ and $\langle k_{\max}^{out} \rangle = 5.6$. The change of slope in execution time at sample size N=1000 for NML corresponds to the use of the Szpankowski approximation (see Supplementary Methods).

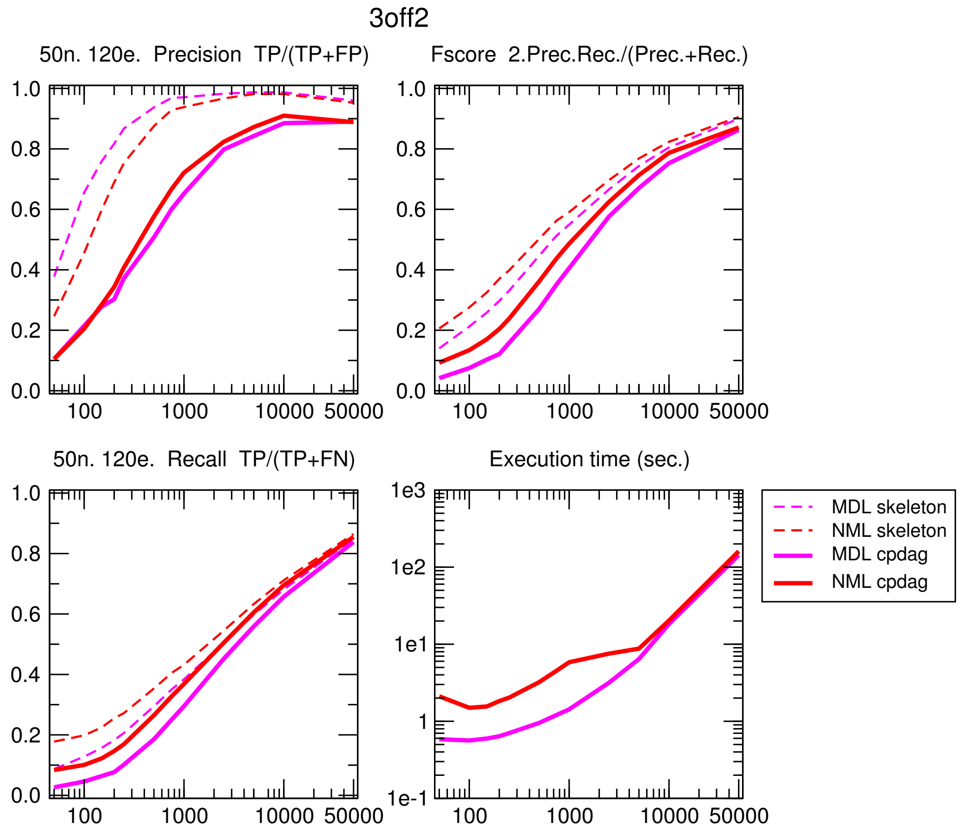

Figure S5: 3off2 **reconstruction, effect of complexity MDL and NML.** 50 node, 120 edge benchmark networks generated using Tetrad. $\langle k \rangle = 4.8$, $\langle k^{in}_{\max} \rangle = 8.8$ and $\langle k^{out}_{\max} \rangle = 7.2$. The change of slope in execution time at sample size N=1000 for NML corresponds to the use of the Szpankowski approximation (see Supplementary Methods).

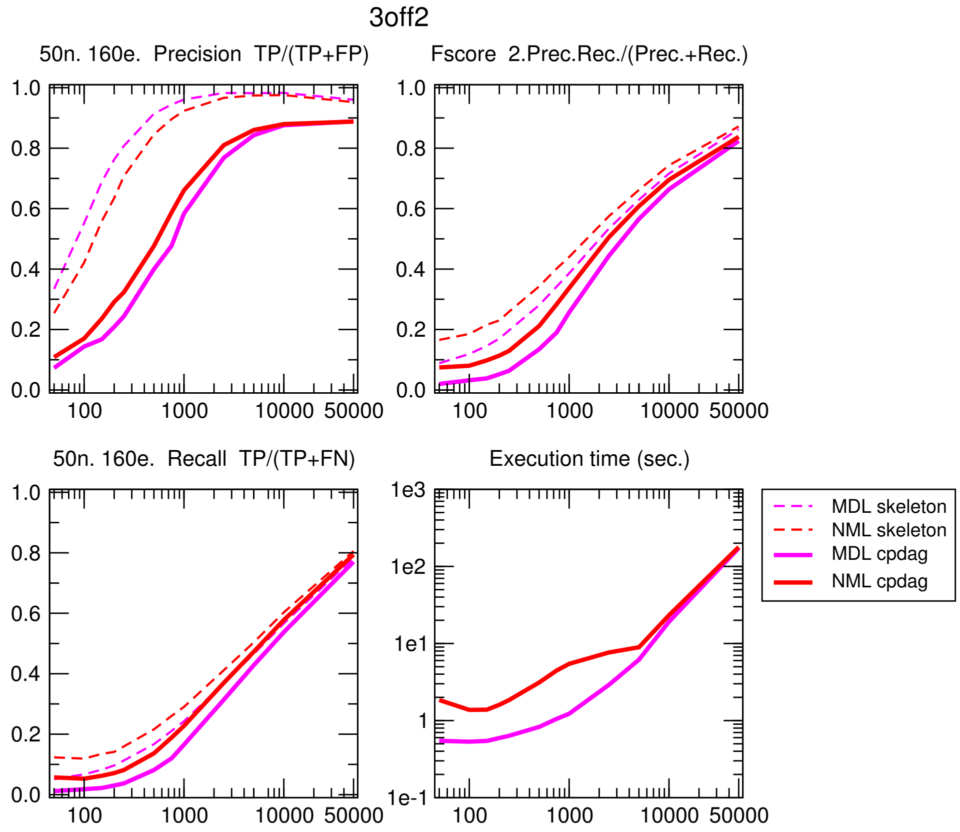

Figure S6: 3off2 **reconstruction, effect of complexity MDL and NML.** 50 node, 160 edge benchmark networks generated using Tetrad. $\langle k \rangle = 6.4$, $\langle k^{in}_{\max} \rangle = 8.6$ and $\langle k^{out}_{\max} \rangle = 8.6$. The change of slope in execution time at sample size N=1000 for NML corresponds to the use of the Szpankowski approximation (see Supplementary Methods).

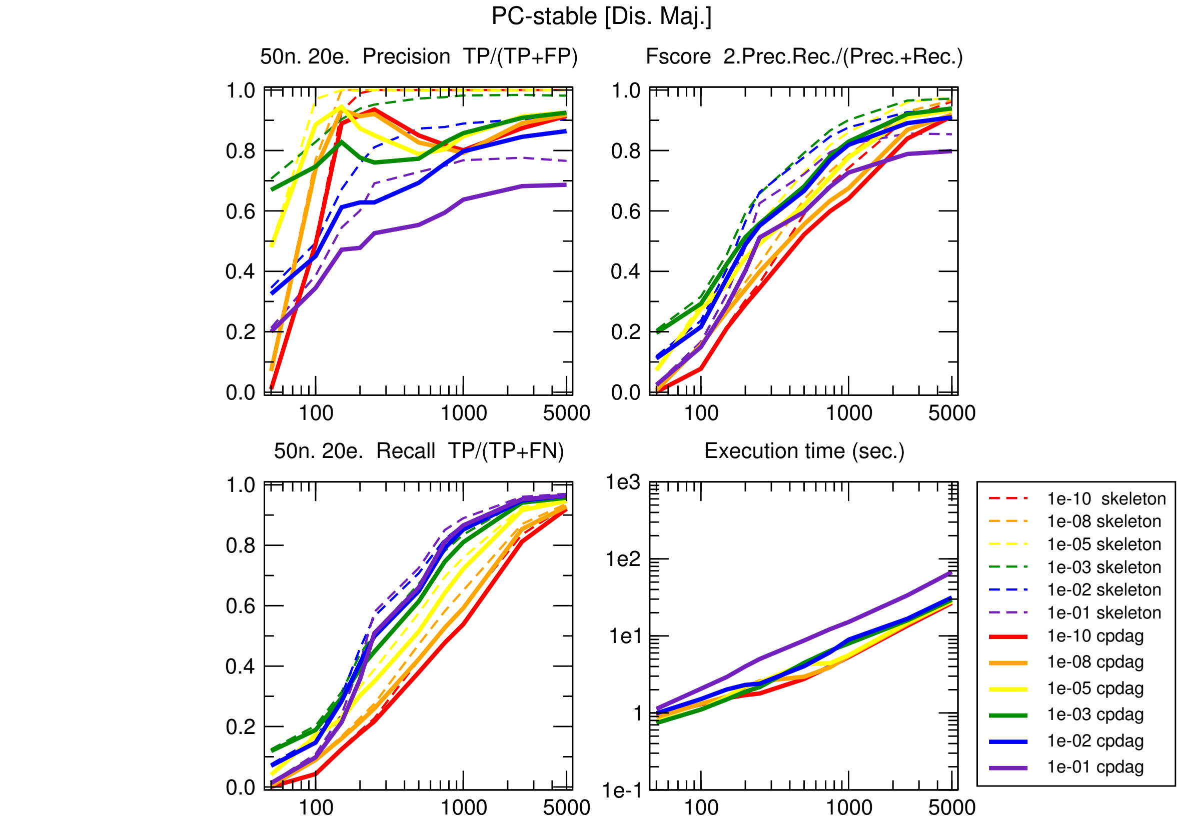

Figure S7: **PC, effect of independence test parameter** α**.** 50 node, 20 edge benchmark networks generated using Tetrad. $\langle k \rangle = 0.8$, $\langle k_{\max}^{in} \rangle = 2.4$ and $\langle k_{\max}^{out} \rangle = 2.2$. G^2 independence test; PC-stable, majority rule (Colombo and Maathuis, 2014).

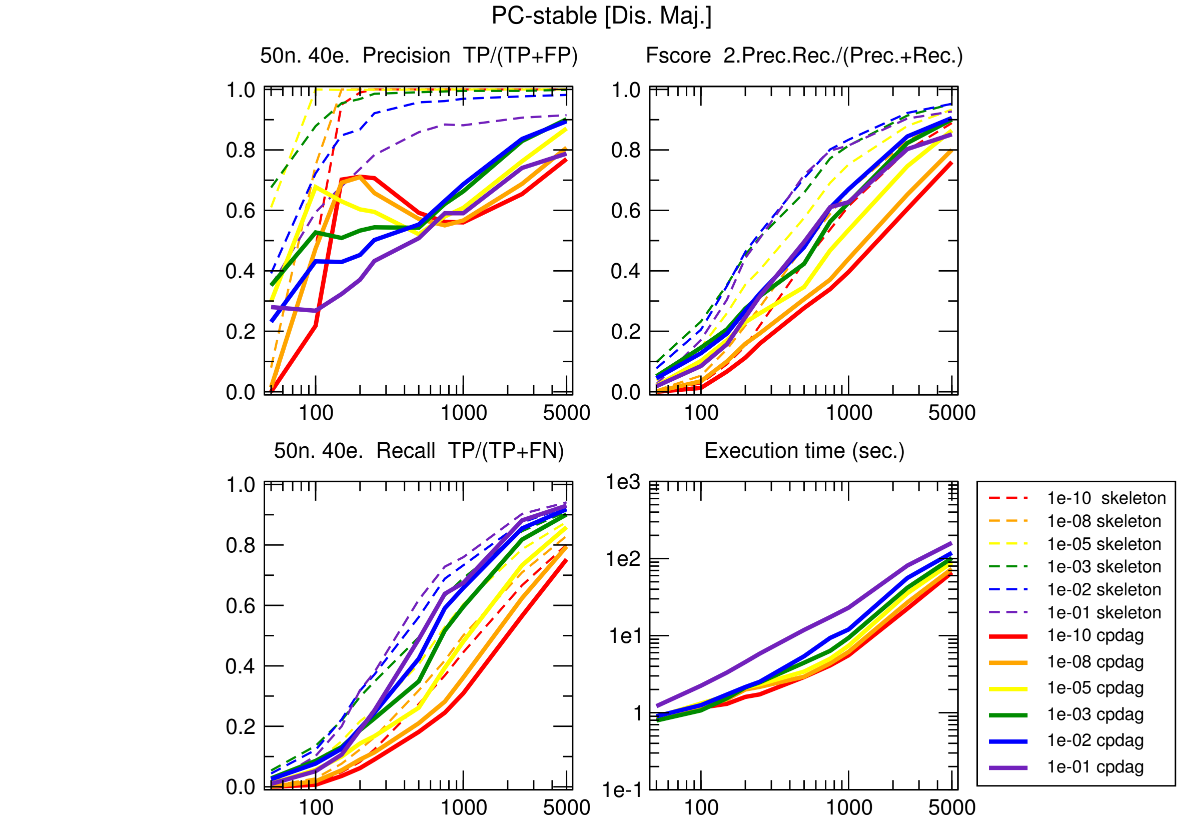

Figure S8: **PC, effect of independence test parameter** α**.** 50 node, 40 edge benchmark networks generated using Tetrad. $\langle k \rangle = 1.6$, $\langle k_{\max}^{in} \rangle = 3.2$ and $\langle k_{\max}^{out} \rangle = 3.6$. G^2 independence test; PC-stable, majority rule (Colombo and Maathuis, 2014).

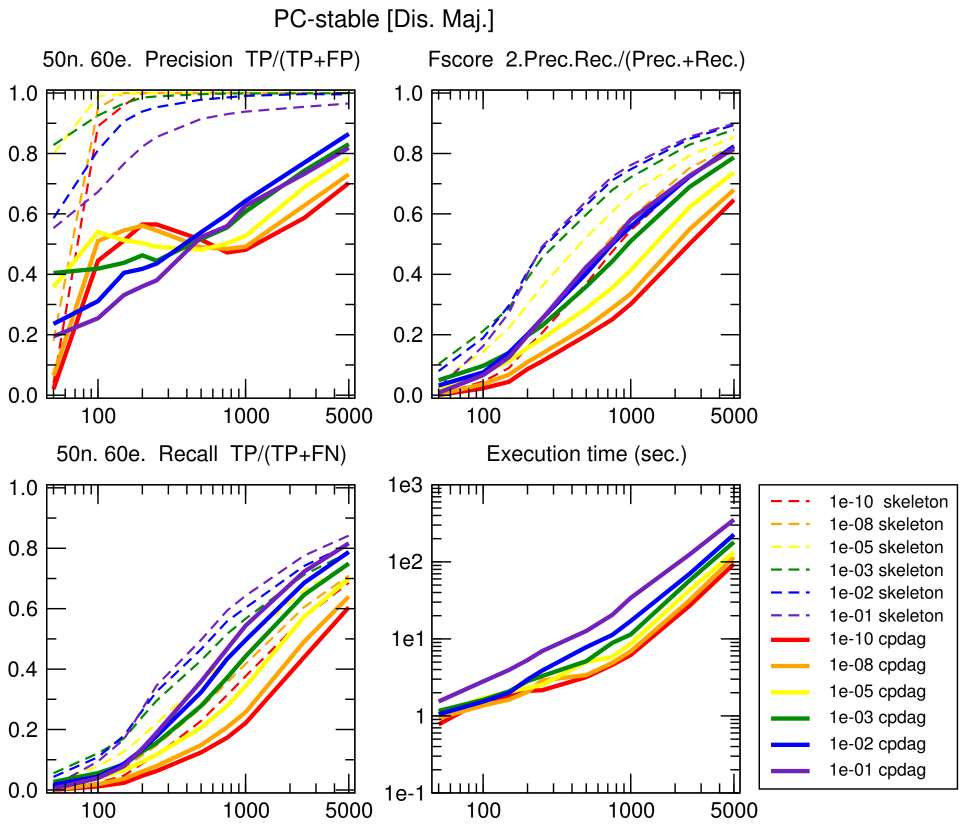

Figure S9: **PC, effect of independence test parameter** α**.** 50 node, 60 edge benchmark networks generated using Tetrad. $\langle k \rangle = 2.4$, $\langle k_{\max}^{in} \rangle = 4.6$ and $\langle k_{\max}^{out} \rangle = 3.6$. G^2 independence test; PC-stable, majority rule (Colombo and Maathuis, 2014).

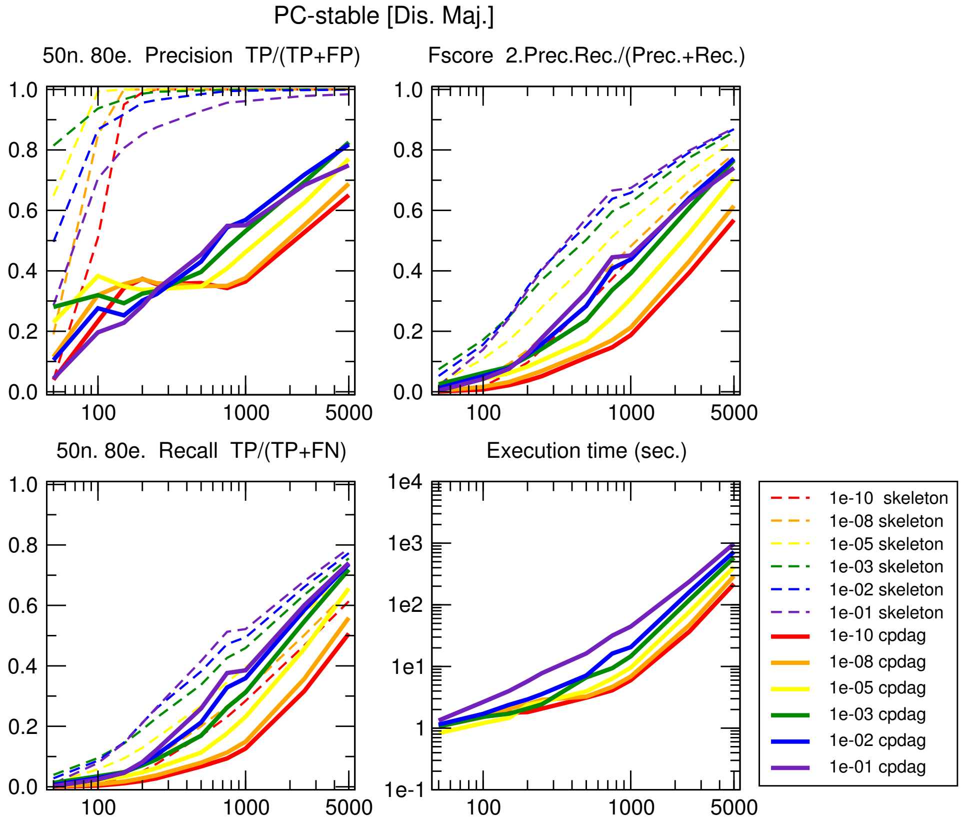

Figure S10: **PC, effect of independence test parameter** α**.** 50 node, 80 edge benchmark networks generated using Tetrad. $\langle k \rangle = 3.2$, $\langle k_{\max}^{in} \rangle = 4.8$ and $\langle k_{\max}^{out} \rangle = 5.6$. G^2 independence test; PC-stable, majority rule (Colombo and Maathuis, 2014).

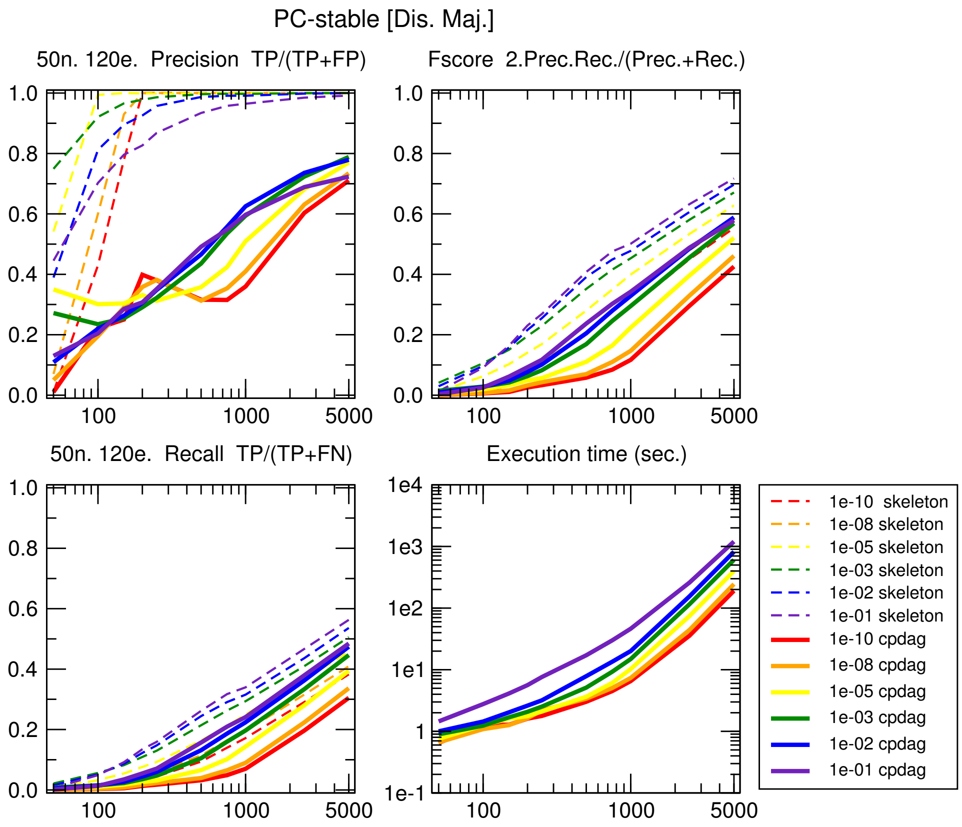

Figure S11: **PC, effect of independence test parameter** α**.** 50 node, 120 edge benchmark networks generated using Tetrad. $\langle k \rangle = 4.8$, $\langle k_{\max}^{in} \rangle = 8.8$ and $\langle k_{\max}^{out} \rangle = 7.2$. G^2 independence test; PC-stable, majority rule (Colombo and Maathuis, 2014).

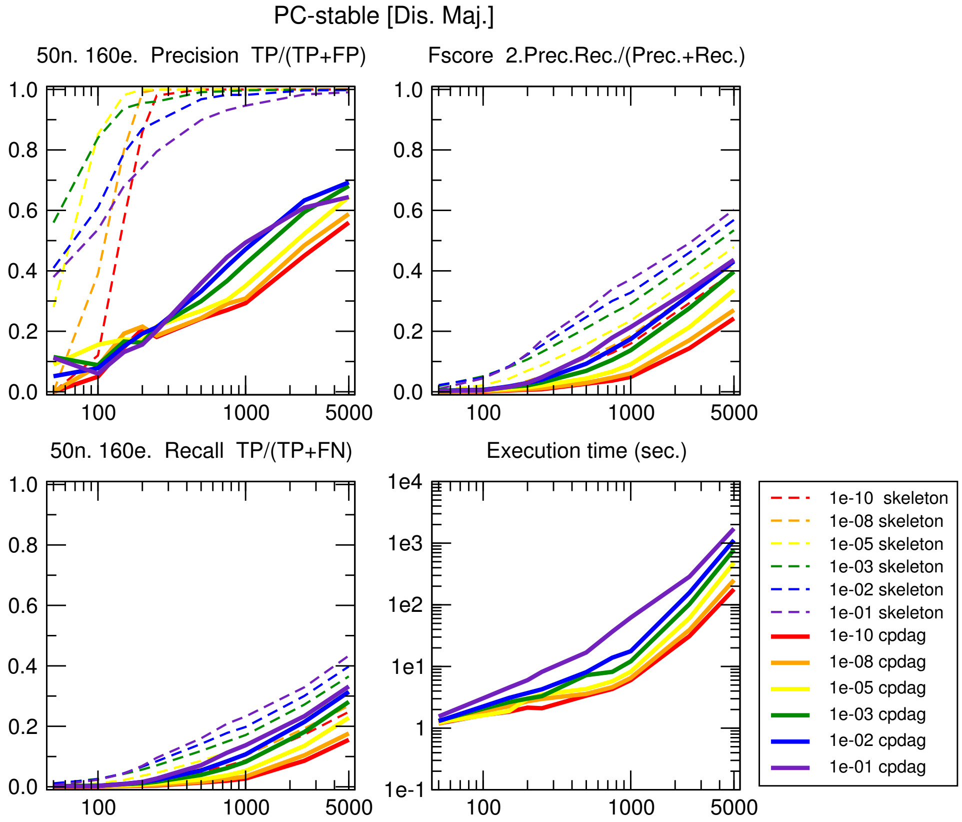

Figure S12: **PC, effect of independence test parameter** α**.** 50 node, 160 edge benchmark networks generated using Tetrad. $\langle k \rangle = 6.4$, $\langle k_{\max}^{in} \rangle = 8.6$ and $\langle k_{\max}^{out} \rangle = 8.6$. G^2 independence test; PC-stable, majority rule (Colombo and Maathuis, 2014).

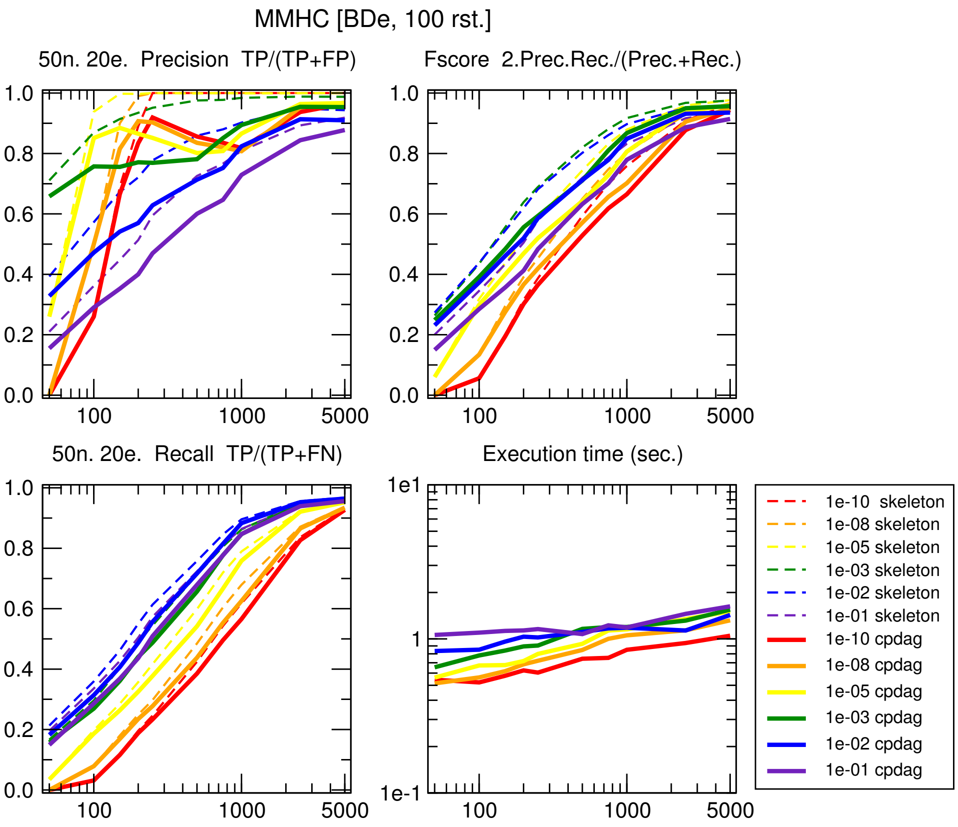

Figure S13: **MMHC, effect of independence test parameter** α**.** 50 node, 20 edge benchmark networks generated using Tetrad. $\langle k \rangle = 0.8$, $\langle k^{in}_{\max} \rangle = 2.4$ and $\langle k^{out}_{\max} \rangle = 2.2$. G^2 independence test; MMHC, BDe score (Tsamardinos, Brown, and Aliferis, 2006).

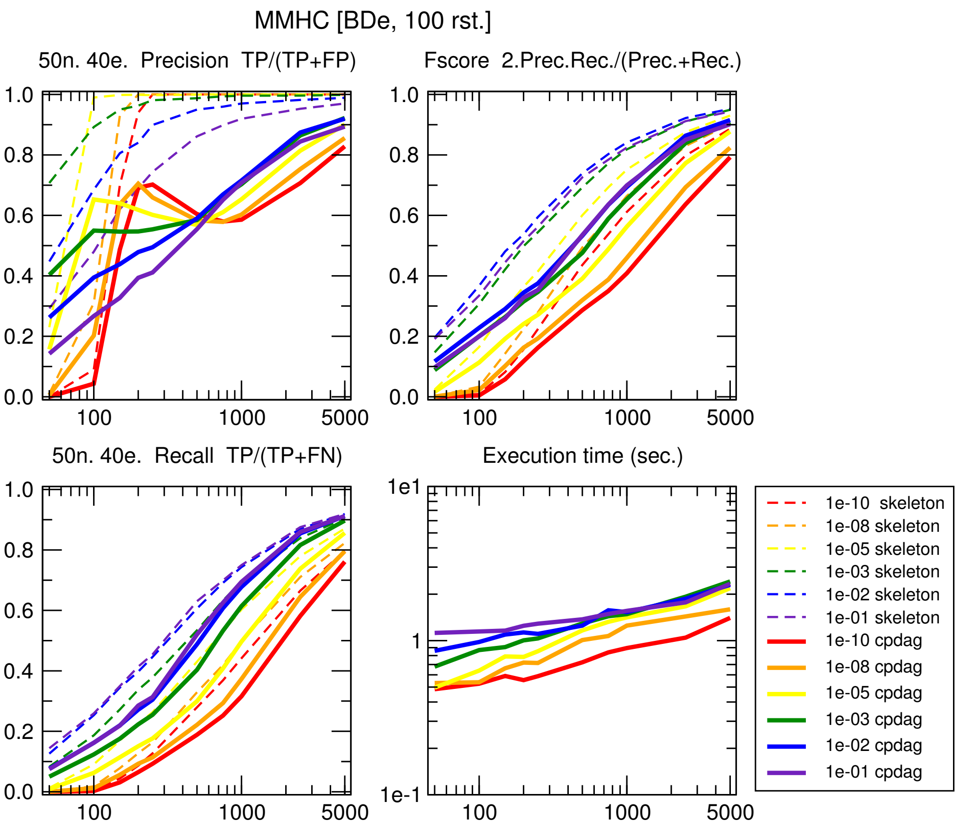

Figure S14: **MMHC, effect of independence test parameter** α**.** 50 node, 40 edge benchmark networks generated using Tetrad. $\langle k \rangle = 1.6$, $\langle k^{in}_{\max} \rangle = 3.2$ and $\langle k^{out}_{\max} \rangle = 3.6$. G^2 independence test; MMHC, BDe score (Tsamardinos, Brown, and Aliferis, 2006).

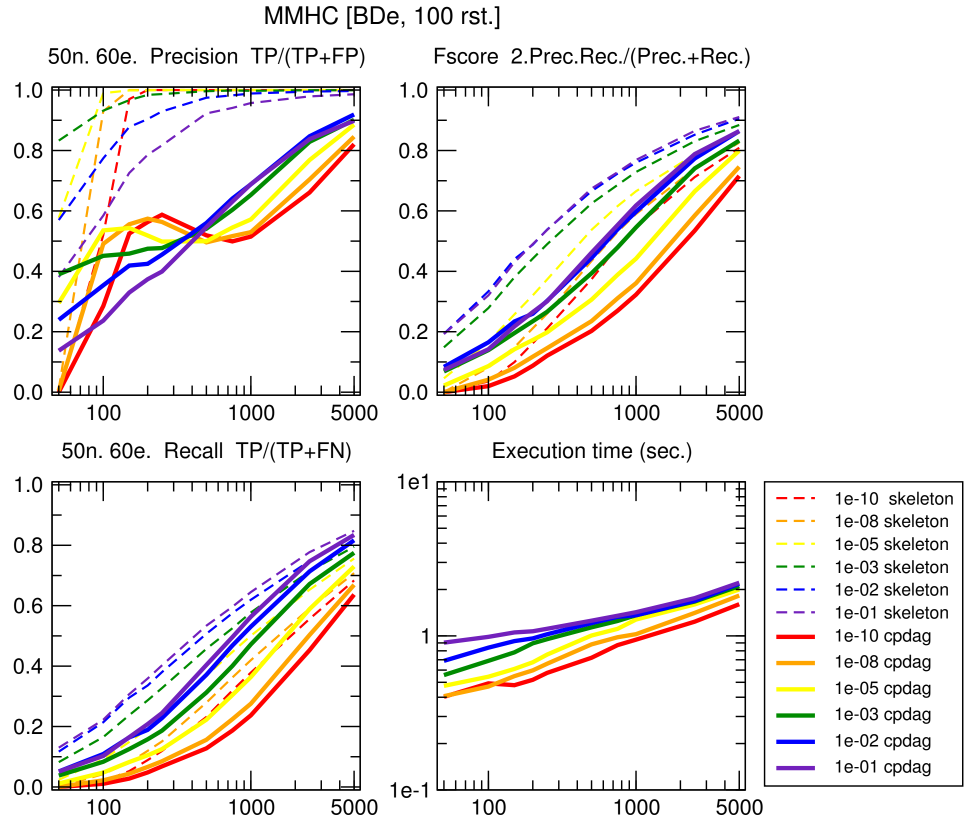

Figure S15: **MMHC, effect of independence test parameter** α**.** 50 node, 60 edge benchmark networks generated using Tetrad. $\langle k \rangle = 2.4$, $\langle k^{in}_{\max} \rangle = 4.6$ and $\langle k^{out}_{\max} \rangle = 3.6$. G^2 independence test; MMHC, BDe score (Tsamardinos, Brown, and Aliferis, 2006).

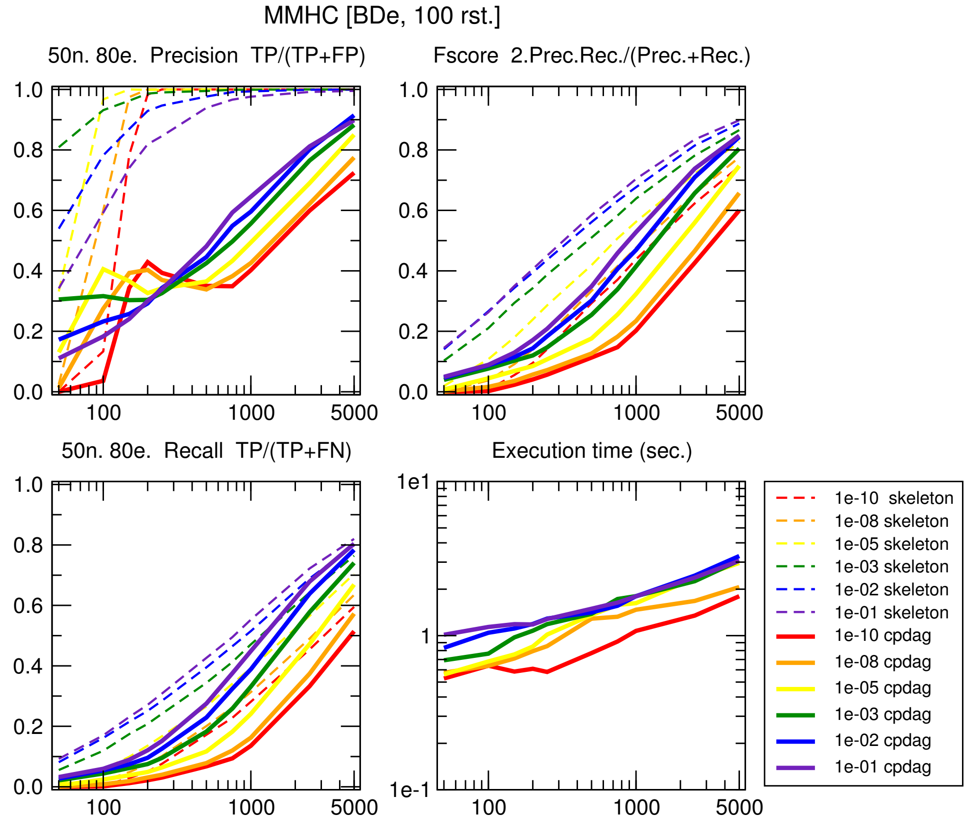

Figure S16: **MMHC, effect of independence test parameter** α**.** 50 node, 80 edge benchmark networks generated using Tetrad. $\langle k \rangle = 3.2$, $\langle k^{in}_{\max} \rangle = 4.8$ and $\langle k^{out}_{\max} \rangle = 5.6$. G^2 independence test; MMHC, BDe score (Tsamardinos, Brown, and Aliferis, 2006).

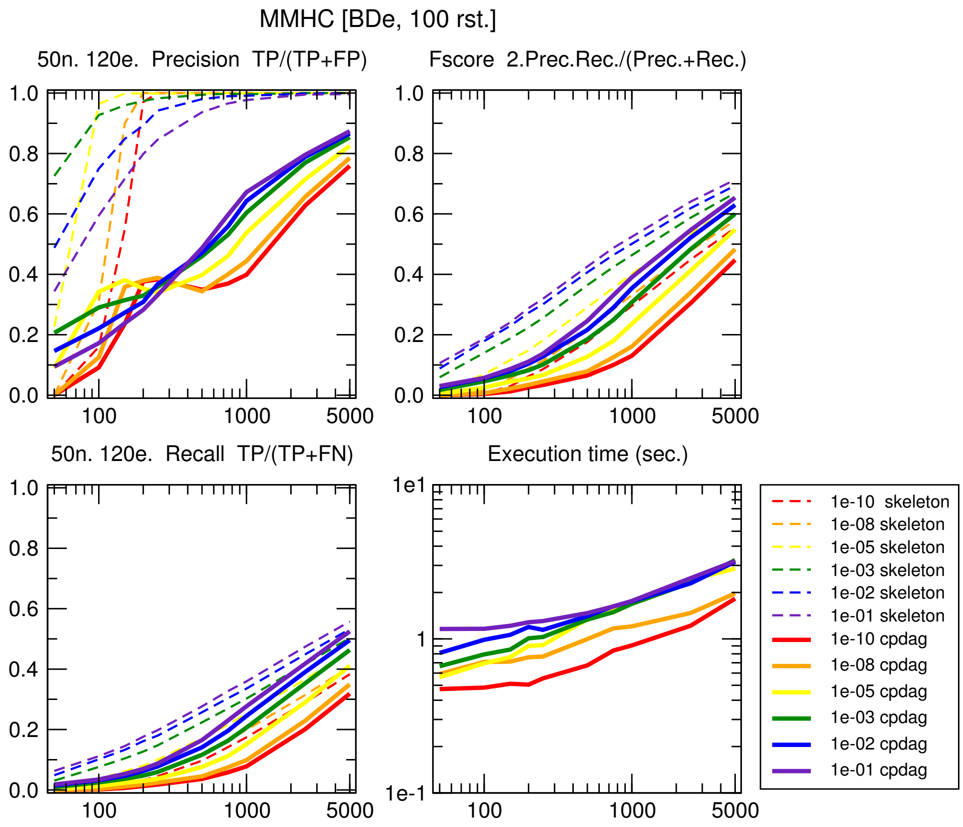

Figure S17: **MMHC, effect of independence test parameter** α**.** 50 node, 120 edge benchmark networks generated using Tetrad. $\langle k \rangle = 4.8$, $\langle k_{\max}^{in} \rangle = 8.8$ and $\langle k_{\max}^{out} \rangle = 7.2$. G^2 independence test; MMHC, BDe score (Tsamardinos, Brown, and Aliferis, 2006).

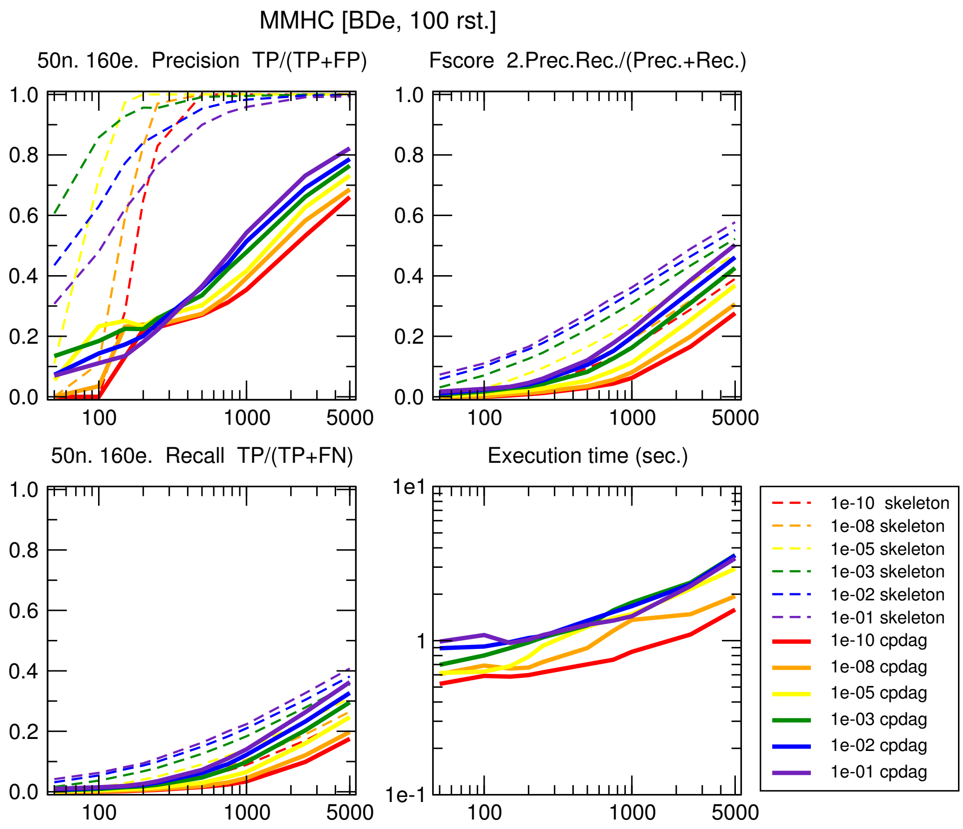

Figure S18: **MMHC, effect of independence test parameter** α**.** 50 node, 160 edge benchmark networks generated using Tetrad. $\langle k \rangle = 6.4$, $\langle k_{\max}^{in} \rangle = 8.6$ and $\langle k_{\max}^{out} \rangle = 8.6$. G^2 independence test; MMHC, BDe score (Tsamardinos, Brown, and Aliferis, 2006).

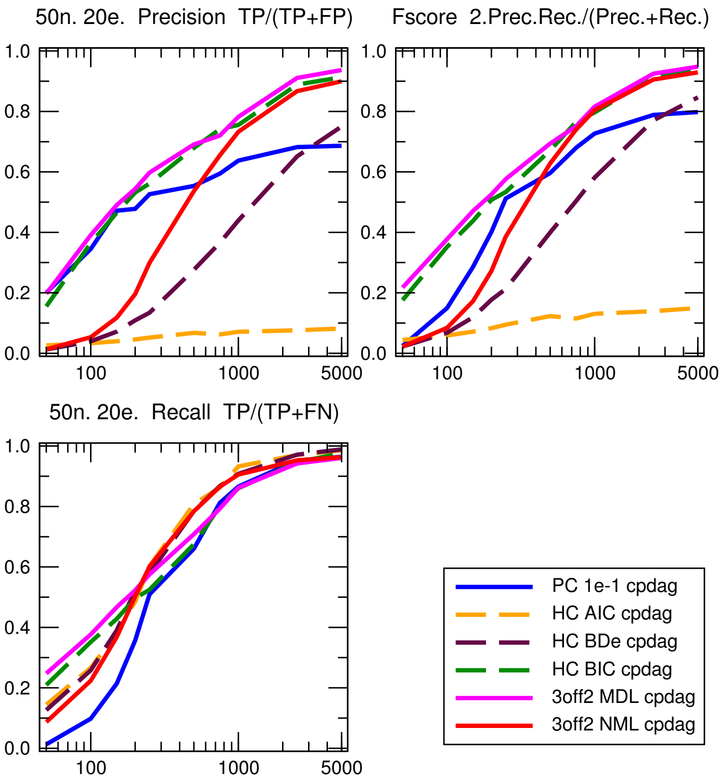

Figure S19: **CPDAG comparison between 3off2, PC and Bayesian hill climbing.** 50 node, 20 edge benchmark networks generated using Tetrad. $\langle k \rangle = 0.8$, $\langle k_{\max}^{in} \rangle = 2.4$ and $\langle k_{\max}^{out} \rangle = 2.2$. Bayesian scores: AIC, BDe and BIC.

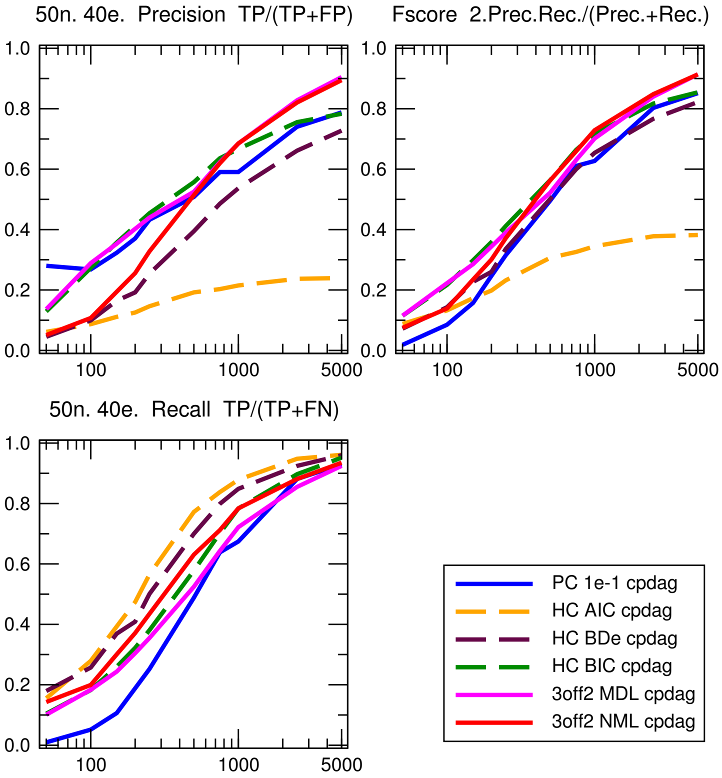

Figure S20: **CPDAG comparison between 3off2, PC and Bayesian hill climbing.** 50 node, 40 edge benchmark networks generated using Tetrad. $\langle k \rangle = 1.6$, $\langle k_{\max}^{in} \rangle = 3.2$ and $\langle k_{\max}^{out} \rangle = 3.6$. Bayesian scores: AIC, BDe and BIC.

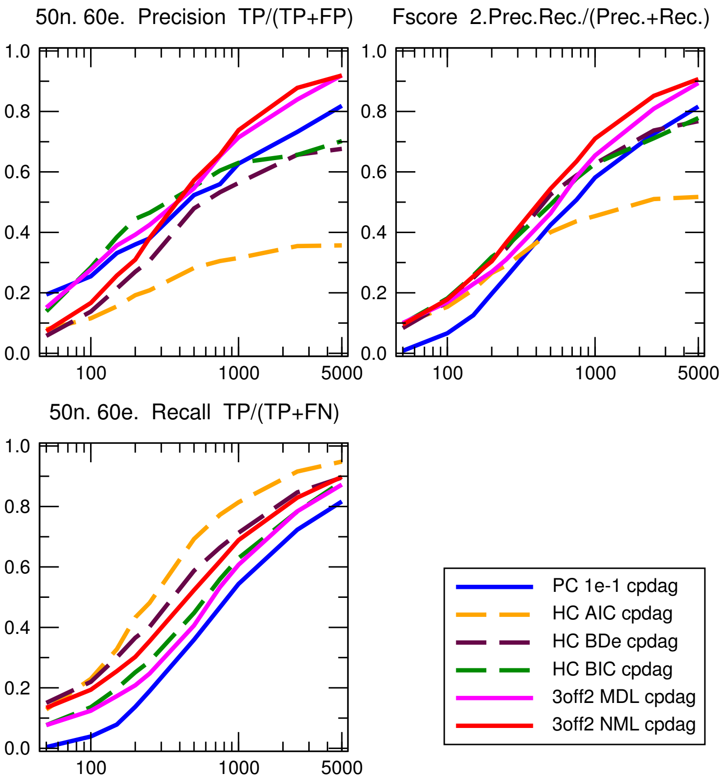

Figure S21: **CPDAG comparison between 3off2, PC and Bayesian hill climbing.** 50 node, 60 edge benchmark networks generated using Tetrad. $\langle k \rangle = 2.4$, $\langle k_{\max}^{in} \rangle = 4.6$ and $\langle k_{\max}^{out} \rangle = 3.6$. Bayesian scores: AIC, BDe and BIC.

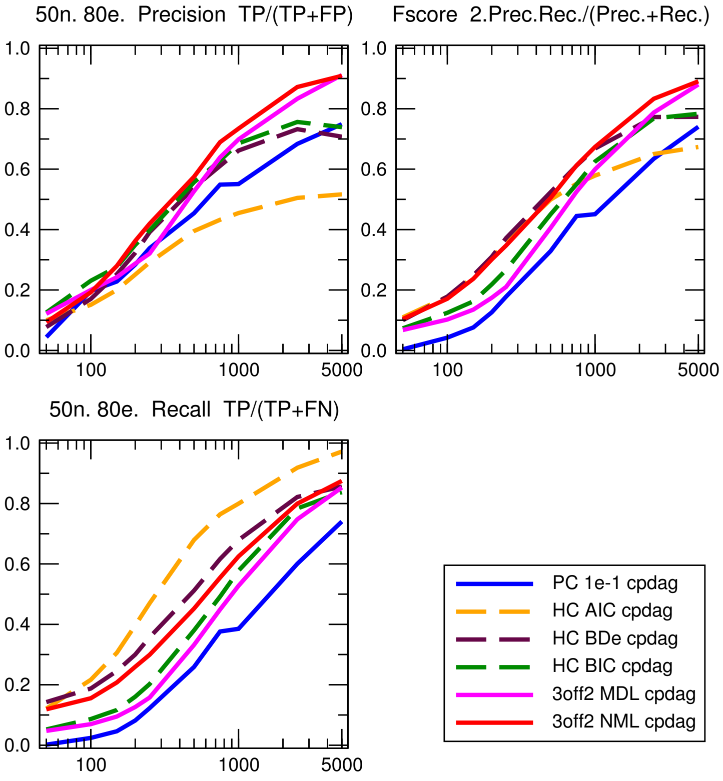

Figure S22: **CPDAG comparison between 3off2, PC and Bayesian hill climbing.** 50 node, 80 edge benchmark networks generated using Tetrad. $\langle k \rangle = 3.2$, $\langle k_{\max}^{in} \rangle = 4.8$ and $\langle k_{\max}^{out} \rangle = 5.6$. Bayesian scores: AIC, BDe and BIC.

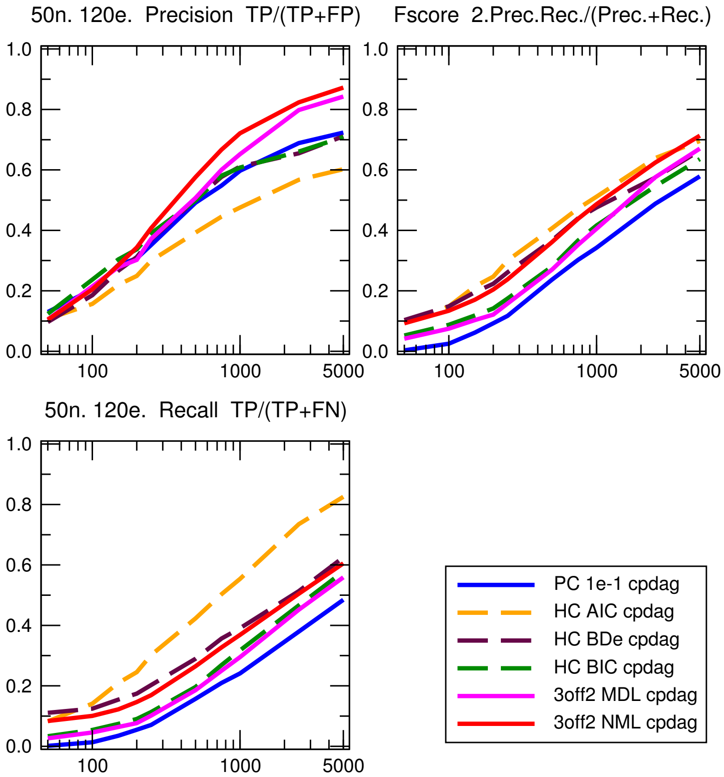

Figure S23: **CPDAG comparison between 3off2, PC and Bayesian hill climbing.** 50 node, 120 edge benchmark networks generated using Tetrad. $\langle k \rangle = 4.8$, $\langle k_{\max}^{in} \rangle = 8.8$ and $\langle k_{\max}^{out} \rangle = 7.2$. Bayesian scores: AIC, BDe and BIC.

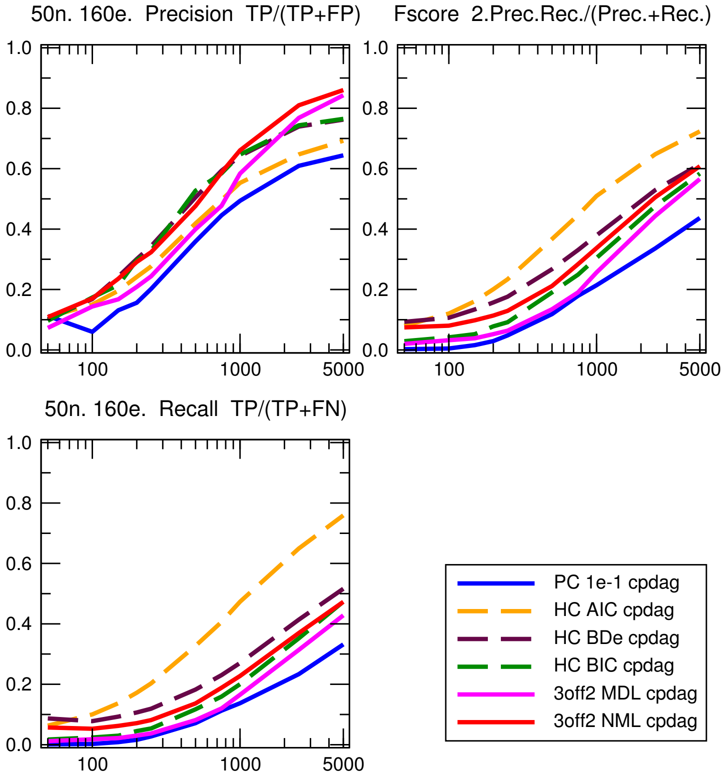

Figure S24: **CPDAG comparison between 3off2, PC and Bayesian hill climbing.** 50 node, 160 edge benchmark networks generated using Tetrad. $\langle k \rangle = 6.4$, $\langle k_{\max}^{in} \rangle = 8.6$ and $\langle k_{\max}^{out} \rangle = 8.6$. Bayesian scores: AIC, BDe and BIC.

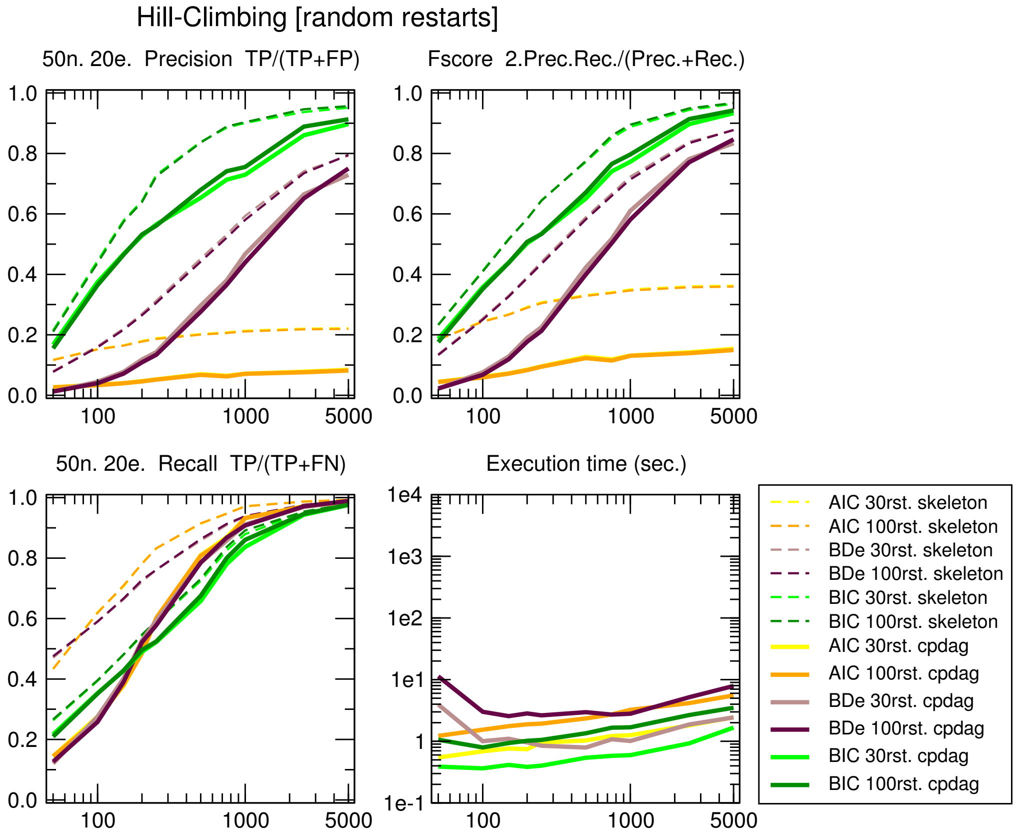

Figure S25: **Bayesian Hill-Climbing, effect of Bayesian score AIC, BDe and BIC.** 50 node, 20 edge benchmark networks generated using Tetrad. $\langle k \rangle = 0.8$, $\langle k_{\max}^{in} \rangle = 2.4$ and $\langle k_{\max}^{out} \rangle = 2.2$.

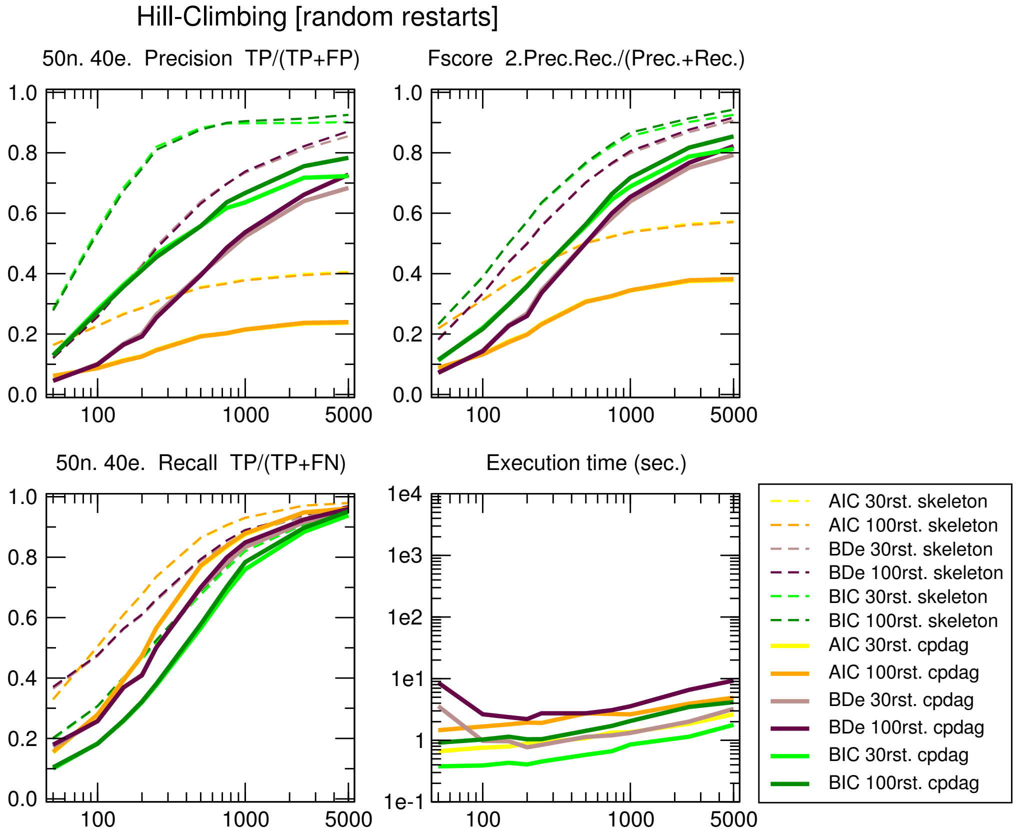

Figure S26: **Bayesian Hill-Climbing, effect of Bayesian score AIC, BDe and BIC.** 50 node, 40 edge benchmark networks generated using Tetrad. $\langle k \rangle = 1.6$, $\langle k_{\max}^{in} \rangle = 3.2$ and $\langle k_{\max}^{out} \rangle = 3.6$.

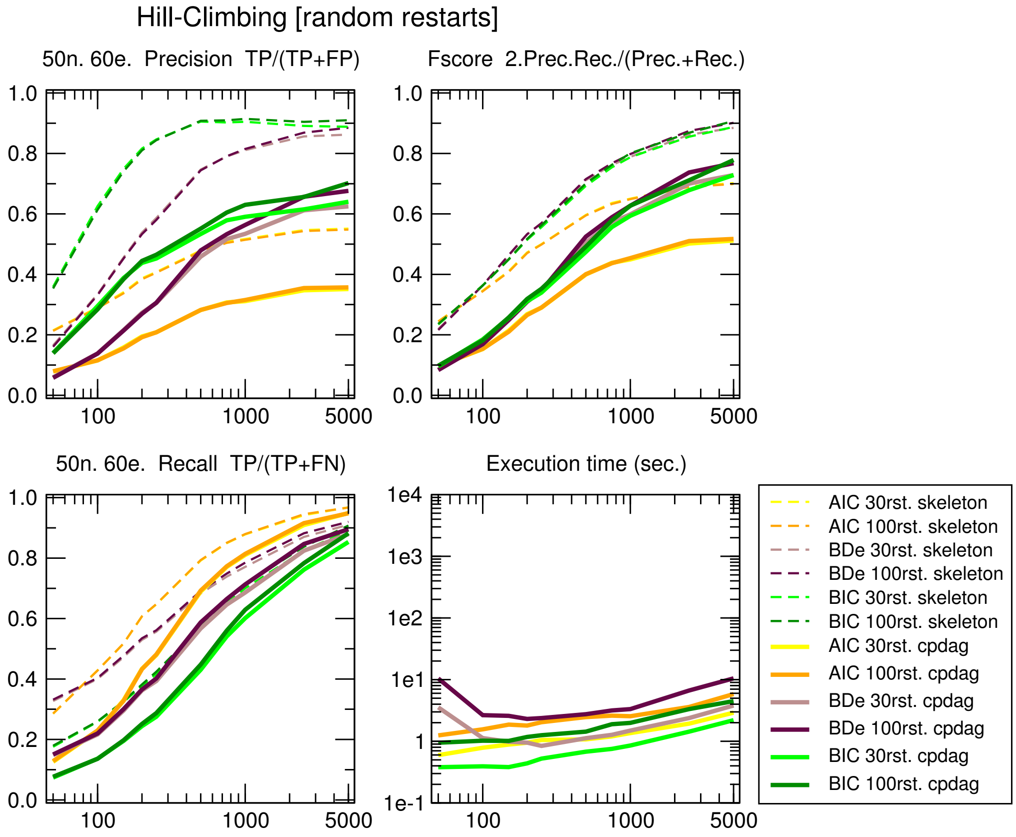

Figure S27: **Bayesian Hill-Climbing, effect of Bayesian score AIC, BDe and BIC.** 50 node, 60 edge benchmark networks generated using Tetrad. $\langle k\rangle = 2.4$, $\langle k^{in}_{\max}\rangle = 4.6$ and $\langle k^{out}_{\max}\rangle = 3.6$.

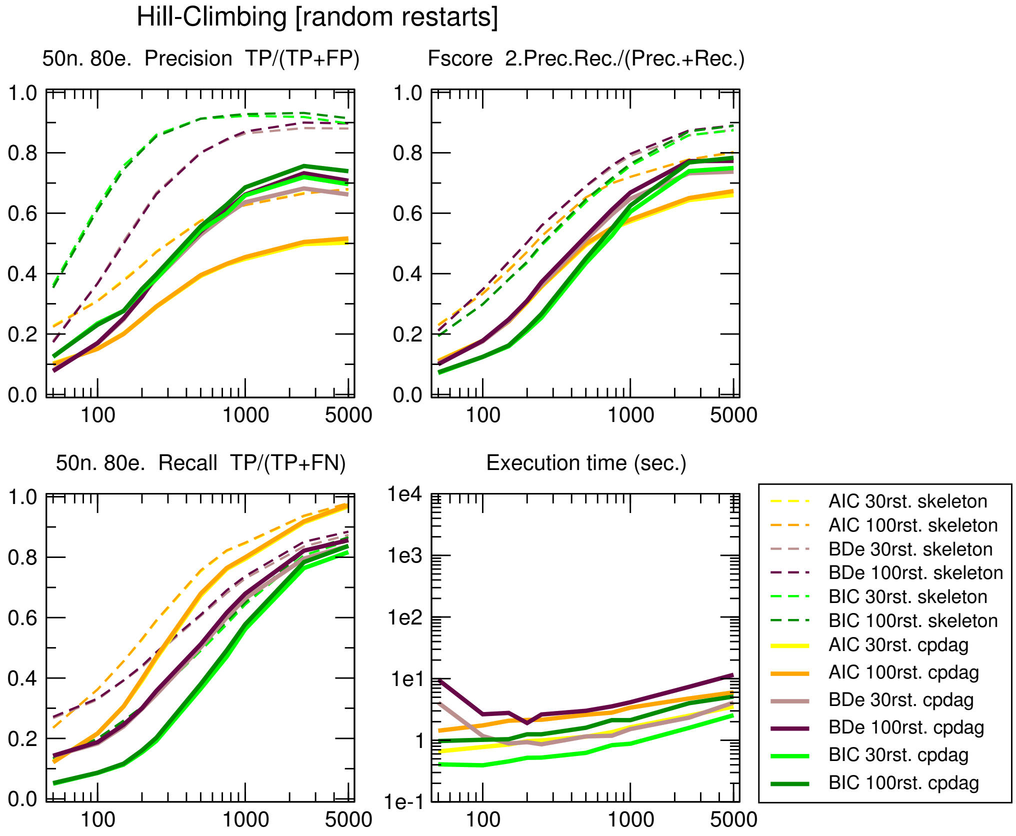

Figure S28: **Bayesian Hill-Climbing, effect of Bayesian score AIC, BDe and BIC.** 50 node, 80 edge benchmark networks generated using Tetrad. $\langle k\rangle = 3.2$, $\langle k^{in}_{\max}\rangle = 4.8$ and $\langle k^{out}_{\max}\rangle = 5.6$.

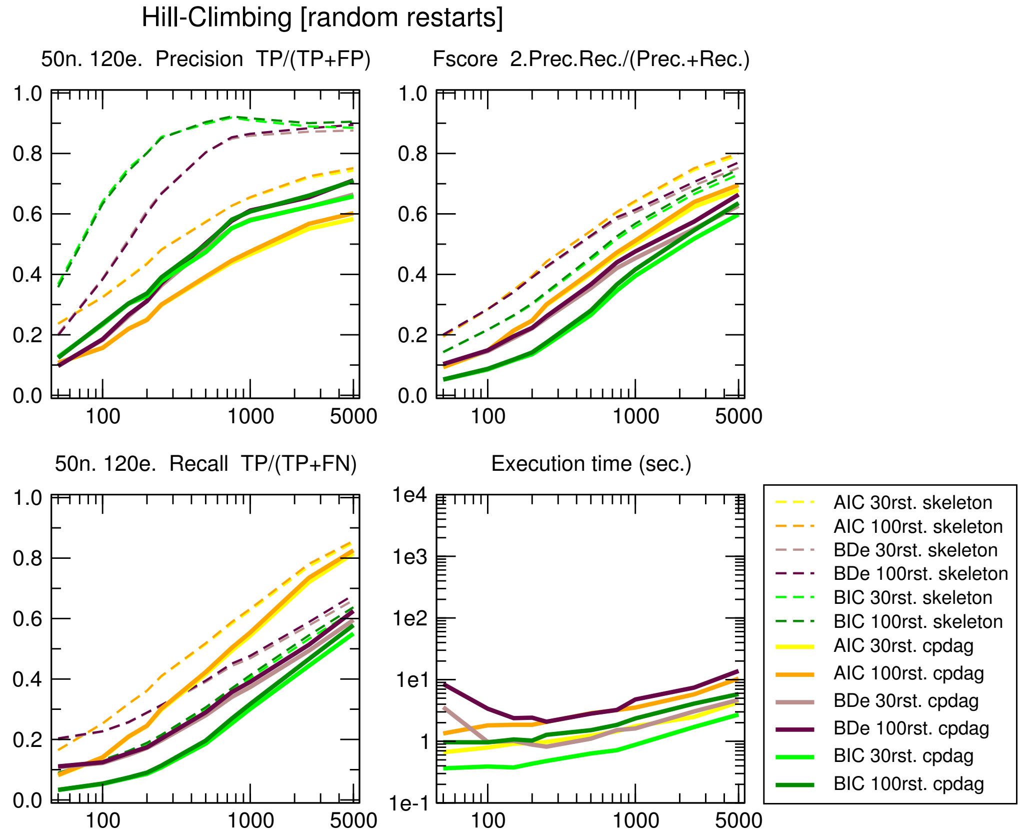

Figure S29: **Bayesian Hill-Climbing, effect of Bayesian score AIC, BDe and BIC.** 50 node, 120 edge benchmark networks generated using Tetrad. $\langle k \rangle = 4.8$, $\langle k^{in}_{\max} \rangle = 8.8$ and $\langle k^{out}_{\max} \rangle = 7.2$.

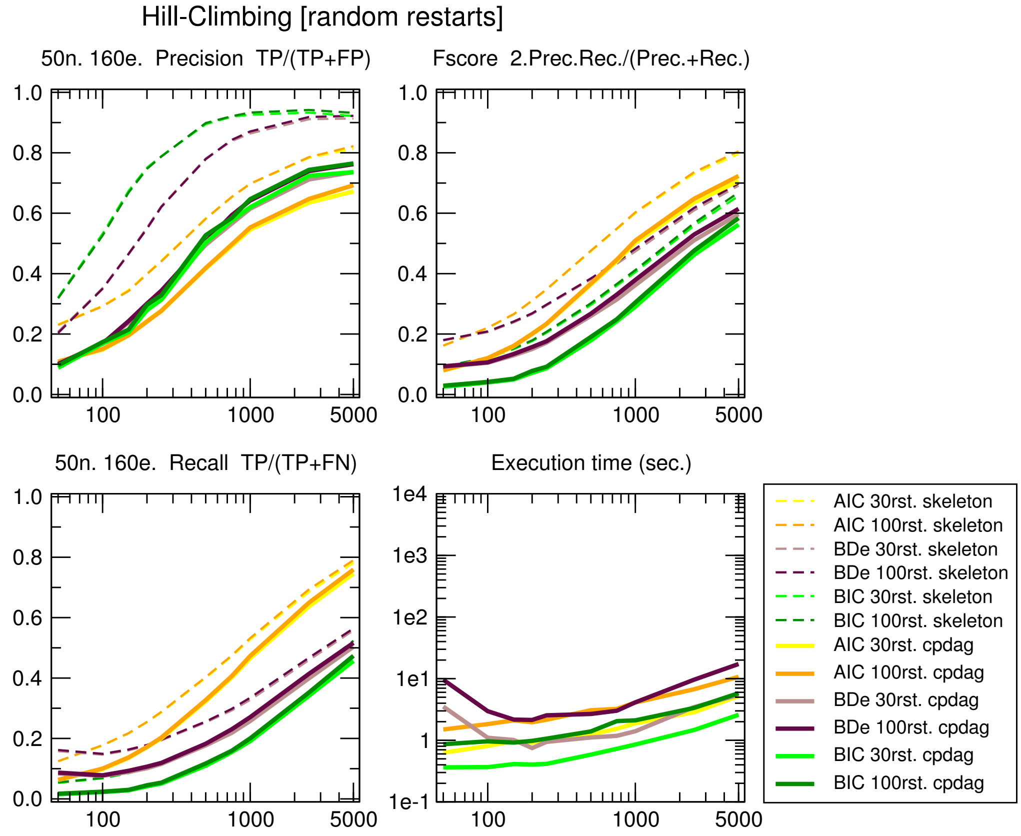

Figure S30: **Bayesian Hill-Climbing, effect of Bayesian score AIC, BDe and BIC.** 50 node, 160 edge benchmark networks generated using Tetrad. $\langle k \rangle = 6.4$, $\langle k^{in}_{\max} \rangle = 8.6$ and $\langle k^{out}_{\max} \rangle = 8.6$.