Supplementary Materials Scalable Bayesian Structure Learning for Gaussian Graphical Models Using Marginal Pseudo-likelihood

Reza Mohammadi *, Marit Schoonhoven* Lucas Vogels * and . lker Birbil*

The supplementary material provides further detailed results and is organized as follows:

- Section 1 presents supplementary results for the simulations discussed in the manuscript.
- Section 2 contains further results for the real-world application to human Gene expression covered in the manuscript.
- Section 3 provides additional results for the real-world application to Gene expression in immune cells described in the manuscript.

1 Additional Materials for Simulation Study

Here we present additional simulation results: the edge density of the simulated graphs in Table 1, four additional graph precision recovery metrics, and the graph precision recovery metrics over time in Figure 1.

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) [1] evaluates a classifier's ability to distinguish between true edges and non-edges in the graph. The ROC curve plots the True Positive Rate against the False Positive Rate at various threshold settings. The AUC-ROC measures the area under the curve, ranging from 0 to 1, with higher values indicating better performance. Table 3 presents the AUC-ROC scores, and Table 2 reports the computational time required for AUC-ROC convergence. Here, AUC-ROC convergence is defined as the time at which the AUC-ROC value reaches a value within 0.01 of its final iteration value.

The $F1$ Score [2] is the harmonic mean of Precision and Recall, providing a single metric that balances both. It is defined as:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{1}$$

To report the $F1$ values, we first obtain the estimated graph $\hat{G} = (V, \hat{E})$, where $\hat{E} = \left\{ e = (i,j) \mid \hat{P}_e \geq 0.5 \right\}$. In Bayesian graphical learning, the $F1$ Score ranges from 0 to 1,

arXiv: 2307.00127v3

* Department of Business Analytics, Faculty of Economics and Business, University of Amsterdam , a.mohammadi@uva.nl

p	10		100		1000	
Density	Sparse	Dense	Sparse	Dense	Sparse	Dense
Random	11.1%	44.4%	1.0%	5.0%	0.5%	5.0%
Cluster	11.1%	44.4%	1.0%	5.0%	0.5%	5.0%
Scale-free	20.0%		2.0%		0.2%	

Table 1: *The edge density of the graphs is defined as the proportion of the number of edges to the total number of possible edges in the graphs.*

with higher values indicating better overall performance in detecting true edges while minimizing false positives and false negatives. Table 4 presents the $F1$ scores.

Pr^+ and Pr^- represent the average inclusion probability for all edges and non-edges, respectively, in the true graph $G = (V, E)$ [3]. They are calculated as:

$$Pr^+ = \frac{1}{|E|} \sum_{e \in E} \hat{P}_e \tag{2}$$

and

$$Pr^- = \frac{1}{|\bar{E}|} \sum_{e \in \bar{E}} \hat{P}_e, \tag{3}$$

where $\hat{P}_e$ are the estimated edge-inclusion probabilities of the manuscript. These probabilities serve as measures of calibration accuracy. Ideally, algorithms should achieve a high Pr^+ to enhance edge detection accuracy and a low Pr^- to effectively reject edges not present in the true graph $G = (V, E)$. We report the Pr^+ values in Table 5 and Pr^- in Table 6.

In summary, for AUC-ROC and $F1$ metrics, the RJ-MPL and BD-MPL methods perform as well as or better than other algorithms. For Pr^+ at $p = 100$ and $p = 1000$ (Table 5), B-CON occasionally shows higher values, likely due to its higher Pr^- values. Generally, our MPL approaches (RJ-MPL and BD-MPL) perform well in terms of AUC-PR, F1, and Pr^-, but not as well for Pr^+. This tendency is likely because our methods tend to select sparser graphs compared to other approaches. Ideally, we aim for a high Pr^+ to improve edge detection accuracy while maintaining a low Pr^- to effectively reject non-edges.

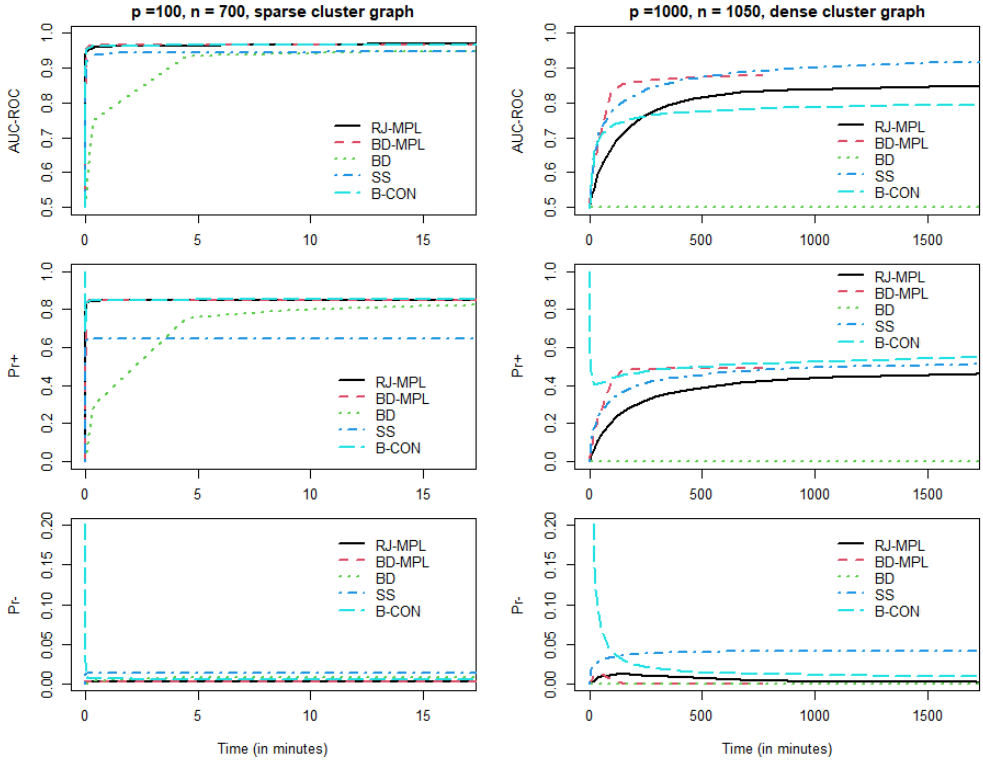

Figure 1: *The convergence of AUC-ROC (top row), Pr+ (middle row), and Pr− (bottom row) scores over running time for all algorithms (RJ-MPL, BD-MPL, BD, SS, B-CON). The plots on the left represent the instance with* $p = 100$, $n = 700$ *for the sparse* Cluster *graph. The plots on the right represent the instance with* $p = 1000$, $n = 1050$ *for the dense* Cluster *graph.*

2 Additional Materials for Application to Human Gene Expression

Here, we present additional results from Application to Human Gene Expression, comparing the output of five algorithms using two metrics: the average absolute differences in edge inclusion probabilities for all unique edges, shown in Table 8, and the percentage of edges identified by method A that are also detected by method B, using a threshold of 0.9 for edge inclusion probability, presented in Table 9. The average absolute differences in edge inclusion probabilities in Table 8 are relatively low but are influenced by the presence of many edges with inclusion probabilities close to zero.

Table 9 demonstrates that, with a 0.9 threshold for edge inclusion probabilities, B-CON identifies the highest number of edges (87), followed by RJ-MPL and BD-MPL (75 and 73, respectively), BD (68), and SS (35). Starting with SS, which identifies the

p	Graph	Density	n	RJ-MPL	BD-MPL	BD	SS	B-CON
1000	Random	Sparse	400	1287	**45**	-	321	664
		Sparse	1050	851	**79**	-	328	338
		Dense	400	2581	**499**	-	2097	3528
		Dense	1050	2125	**901**	-	2122	1752
	Cluster	Sparse	400	1374	**38**	-	301	633
		Sparse	1050	874	**71**	-	318	240
		Dense	400	2429	**480**	-	2111	1626
		Dense	1050	2160	**402**	-	1791	916
	Scale-free	Sparse	400	2143	**19**	-	387	595
		Sparse	1050	1173	**38**	-	1086	67
100	Random	Sparse	40	1	2	63	**0**	3
		Sparse	700	1	**0**	31	**0**	**0**
		Dense	40	4	1	89	1	4
		Dense	700	3	**0**	50	1	1
	Cluster	Sparse	40	2	1	60	**0**	3
		Sparse	700	1	1	49	5	**0**
		Dense	40	4	1	107	**0**	3
		Dense	700	2	**1**	54	**1**	**1**
	Scale-free	Sparse	40	5	4	85	**0**	3
		Sparse	700	3	**0**	54	3	**0**

Table 2: *Computational cost (T) in minutes until AUC-ROC convergence for various instances. T represents the average time until AUC-ROC convergence, based on 16 replications for $p \in \{10, 100\}$ and 8 replications for $p = 1000$. The table excludes the $p = 10$ case since the computational time for all algorithms was less than one minute. A - indicates that an algorithm did not converge within five days. For each setting, the best-performing algorithm is highlighted in* ***bold****.*

fewest edges, Table 9 shows that nearly all edges identified by SS are also identified by the other algorithms. For BD, 68% to 79% of its identified edges overlap with those identified by other algorithms, such as B-CON, RJ-MPL, and BD-MPL, which have a higher number of identified edges. Notably, B-CON, BD-MPL, and RJ-MPL exhibit substantial overlap: approximately 71% to 75% of B-CON's edges are also identified by BD-MPL and RJ-MPL, respectively, while 97% of BD-MPL's edges overlap with those identified by RJ-MPL.

3 Additional Materials for Application to Gene Expression in Immune Cells

Here, we report Tables 10 and 11 for evaluation of the results for Application to Gene Expression in Immune Cells.

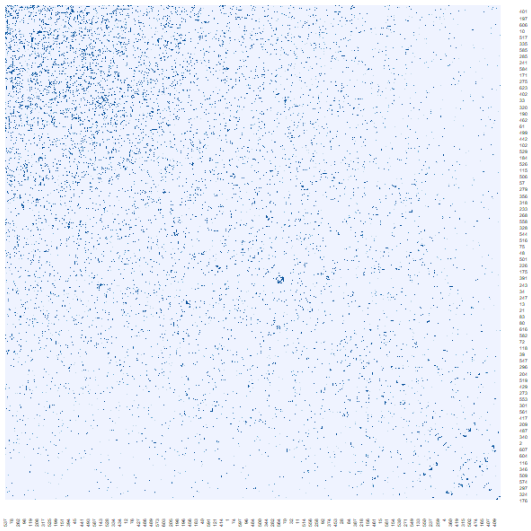

Figure 2: *A Heatmap of the edge inclusion probabilities of the BD-MPL algorithm on the mice gene dataset ($p = 623$). The probabilities range from* 0 *(gray) to* 1 *(dark blue).*

References

[1] Hanley, J. and Mcneil, B. (1982). "The Meaning and Use of the Area Under a Receiver Operating Characteristic (ROC) Curve." *Radiology*, 143(1): 29–36. 1

[2] Powers, D. M. (2020). "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." *arXiv preprint arXiv:2010.16061*. 1

[3] Vogels, L., Mohammadi, R., Schoonhoven, M., and Birbil, Ş. İ. (2024). "Bayesian structure learning in undirected Gaussian graphical models: Literature review with empirical comparison." *Journal of the American Statistical Association*, 119(548): 3164–3182. 2

p	Graph	Density	n	RJ-MPL	BD-MPL	BD	SS	B-CON
1000	Random	Sparse	400	0.87	0.89	0.50	**0.90**	0.89
		Sparse	1050	0.91	0.92	0.50	0.92	**0.94**
		Dense	400	0.70	0.74	0.50	**0.76**	0.70
		Dense	1050	0.77	0.80	0.50	**0.83**	0.78
	Cluster	Sparse	400	0.88	**0.90**	0.50	**0.90**	**0.90**
		Sparse	1050	0.92	0.93	0.50	0.92	**0.94**
		Dense	400	0.78	0.84	0.50	**0.89**	0.72
		Dense	1050	0.86	0.88	0.50	**0.93**	0.80
	Scale-free	Sparse	400	0.89	0.90	0.50	**0.92**	0.91
		Sparse	1050	0.93	0.93	0.50	0.93	**0.95**
100	Random	Sparse	40	0.85	0.86	0.86	**0.87**	0.86
		Sparse	700	**0.97**	**0.97**	**0.97**	0.96	**0.97**
		Dense	40	0.75	0.75	0.75	**0.77**	0.76
		Dense	700	**0.94**	**0.94**	**0.94**	0.92	**0.94**
	Cluster	Sparse	40	**0.85**	**0.85**	0.84	**0.85**	**0.85**
		Sparse	700	**0.97**	**0.97**	0.96	0.95	**0.97**
		Dense	40	0.77	0.77	0.77	**0.79**	0.77
		Dense	700	0.94	**0.95**	**0.95**	0.92	0.94
	Scale-free	Sparse	40	0.81	0.80	0.82	**0.84**	0.81
		Sparse	700	0.95	0.95	**0.96**	0.95	0.95
10	Random	Sparse	20	**0.80**	**0.80**	**0.80**	0.78	0.75
		Sparse	350	0.95	**0.96**	**0.96**	0.92	0.94
		Dense	20	0.68	0.68	**0.69**	0.68	0.68
		Dense	350	**0.92**	**0.92**	**0.92**	0.91	0.90
	Cluster	Sparse	20	0.73	**0.75**	**0.75**	**0.75**	0.74
		Sparse	350	0.90	0.91	0.91	0.90	**0.92**
		Dense	20	0.81	0.81	**0.82**	0.79	0.75
		Dense	350	0.94	0.94	**0.95**	0.92	0.92
	Scale-free	Sparse	20	0.76	0.76	0.78	**0.79**	0.76
		Sparse	350	0.92	**0.94**	**0.94**	0.92	0.92

Table 3: *$AUC-ROC$ scores of the algorithms for different instances. The $AUC-PR$ reaches its best score at* 1 *and its worst at* 0*. The values are averages over 16 replications for $p \in \{10, 100\}$ and over 8 replications for $p = 1000$. For each setting, the best-performing algorithm is highlighted in* ***bold****.*

p	Graph	Density	n	RJ-MPL	BD-MPL	BD	SS	B-CON
1000	Random	Sparse	400	**0.73**	**0.73**	0.00	0.68	0.65
		Sparse	1050	**0.84**	**0.84**	0.00	0.75	0.72
		Dense	400	**0.40**	**0.40**	0.00	0.39	**0.40**
		Dense	1050	0.59	**0.60**	0.00	0.55	0.56
	Cluster	Sparse	400	**0.75**	**0.75**	0.00	0.69	0.70
		Sparse	1050	**0.85**	**0.85**	0.00	0.76	0.75
		Dense	400	0.47	0.47	0.00	0.47	**0.55**
		Dense	1050	0.65	0.65	0.00	0.63	**0.67**
	Scale-free	Sparse	400	**0.62**	**0.62**	0.00	0.63	0.49
		Sparse	1050	0.75	0.75	0.00	**0.79**	0.53
100	Random	Sparse	40	0.41	0.41	0.54	**0.57**	0.52
		Sparse	700	0.85	0.84	**0.89**	0.75	0.79
		Dense	40	0.38	0.38	**0.42**	0.37	0.39
		Dense	700	0.85	0.85	**0.86**	0.65	0.79
	Cluster	Sparse	40	0.44	0.44	0.52	**0.54**	0.51
		Sparse	700	0.83	0.83	**0.87**	0.75	0.78
		Dense	40	0.40	0.39	**0.42**	0.39	0.41
		Dense	700	0.85	0.85	**0.86**	0.69	0.81
	Scale-free	Sparse	40	0.41	0.41	**0.50**	0.48	0.46
		Sparse	700	0.86	0.86	**0.89**	0.70	0.73
10	Random	Sparse	20	**0.40**	**0.40**	0.36	0.27	0.33
		Sparse	350	**0.9**	**0.9**	0.89	0.55	**0.90**
		Dense	20	**0.37**	**0.37**	0.33	0.24	0.33
		Dense	350	0.84	0.84	0.83	0.6	0.84
	Cluster	Sparse	20	**0.43**	**0.43**	0.35	0.19	0.38
		Sparse	350	0.83	**0.84**	0.82	0.60	0.82
		Dense	20	0.44	0.44	**0.41**	0.28	0.37
		Dense	350	0.81	0.81	0.81	0.69	**0.84**
	Scale-free	Sparse	20	0.47	0.47	**0.49**	0.33	0.41
		Sparse	350	**0.88**	**0.88**	0.88	0.66	0.83

Table 4: *$F1$ scores (at a threshold of* 0.5*) of the algorithms for different instances. The $F1$ score reaches its best score at* 1 *and its worst at* 0*. The values are averages over 16 replications for $p \in 10, 100$ and over 8 replications for $p = 1000$. For each setting, the best-performing algorithm is highlighted in* ***bold****.*

p	Graph	Density	n	RJ-MPL	BD-MPL	BD	SS	B-CON
1000	Random	Sparse	400	0.63	0.65	0.00	0.62	**0.67**
		Sparse	1050	0.74	0.76	0.00	0.64	**0.78**
		Dense	400	0.26	0.26	0.00	0.31	**0.34**
		Dense	1050	0.41	0.44	0.00	0.44	**0.51**
	Cluster	Sparse	400	0.64	0.66	0.00	0.64	**0.69**
		Sparse	1050	0.75	0.78	0.00	0.66	**0.8**
		Dense	400	0.31	0.32	0.00	0.38	**0.42**
		Dense	1050	0.48	0.49	0.00	0.53	**0.56**
	Scale-free	Sparse	400	**0.70**	**0.70**	0.00	0.68	**0.70**
		Sparse	1050	0.80	**0.81**	0.00	0.68	**0.81**
100	Random	Sparse	40	**0.54**	**0.54**	0.53	0.50	0.52
		Sparse	700	**0.87**	**0.87**	0.86	0.65	**0.87**
		Dense	40	0.30	0.30	**0.33**	0.28	0.29
		Dense	700	0.75	0.75	0.77	0.52	**0.78**
	Cluster	Sparse	40	**0.54**	**0.54**	0.52	0.49	0.52
		Sparse	700	**0.85**	**0.85**	0.84	0.64	**0.85**
		Dense	40	0.31	0.31	**0.34**	0.30	0.30
		Dense	700	0.75	0.75	**0.78**	0.55	**0.78**
	Scale-free	Sparse	40	0.40	0.41	**0.43**	0.38	0.38
		Sparse	700	0.82	0.82	**0.83**	0.56	0.81
10	Random	Sparse	20	**0.36**	**0.36**	0.31	0.24	0.25
		Sparse	350	**0.84**	**0.84**	0.82	0.44	0.83
		Dense	20	**0.28**	**0.28**	0.26	0.19	0.21
		Dense	350	0.73	0.73	0.73	0.49	**0.75**
	Cluster	Sparse	20	**0.36**	**0.36**	0.30	0.21	0.26
		Sparse	350	0.74	**0.75**	0.73	0.46	0.74
		Dense	20	**0.32**	**0.32**	0.3	0.23	0.25
		Dense	350	0.69	0.69	0.69	0.55	**0.73**
	Scale-free	Sparse	20	**0.37**	**0.37**	0.36	0.26	0.29
		Sparse	350	**0.79**	**0.79**	**0.79**	0.53	0.78

Table 5: *Pr^+ scores of the algorithms for different instances. The Pr^+ reaches its best score at 1 and its worst at 0. The values are averages over 16 replications for $p \in \{10, 100\}$ and over 8 replications for $p = 1000$. For each setting, the best-performing algorithm is highlighted in **bold**.*

p	Graph	Density	n	RJ-MPL	BD-MPL	BD	SS	B-CON
1000	Random	Sparse	400	**0.00**	**0.00**	**0.00**	0.03	0.01
		Sparse	1050	**0.00**	**0.00**	**0.00**	0.02	0.01
		Dense	400	**0.00**	**0.00**	**0.00**	0.06	0.02
		Dense	1050	**0.00**	**0.00**	**0.00**	0.05	0.02
	Cluster	Sparse	400	**0.00**	**0.00**	**0.00**	0.03	0.01
		Sparse	1050	**0.00**	**0.00**	**0.00**	0.02	**0.00**
		Dense	400	**0.00**	**0.00**	**0.00**	0.05	0.01
		Dense	1050	**0.00**	**0.00**	**0.00**	0.04	0.01
	Scale-free	Sparse	400	**0.00**	**0.00**	**0.00**	0.03	0.01
		Sparse	1050	**0.00**	**0.00**	**0.00**	0.02	0.01
100	Random	Sparse	40	0.02	0.02	0.04	0.04	**0.01**
		Sparse	700	**0.00**	**0.00**	0.01	0.01	0.01
		Dense	40	0.02	0.02	0.05	0.05	**0.01**
		Dense	700	**0.00**	**0.00**	0.01	0.02	0.01
	Cluster	Sparse	40	0.02	0.02	0.03	0.04	**0.01**
		Sparse	700	**0.00**	**0.00**	0.01	0.01	0.01
		Dense	40	0.02	0.02	0.05	0.05	**0.01**
		Dense	700	**0.00**	**0.00**	0.01	0.02	0.01
	Scale-free	Sparse	40	0.02	0.02	0.04	0.04	**0.01**
		Sparse	700	**0.00**	**0.00**	0.01	0.01	0.01
10	Random	Sparse	20	0.04	0.04	0.05	0.04	**0.01**
		Sparse	350	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
		Dense	20	0.07	0.07	0.09	0.07	**0.05**
		Dense	350	**0.01**	**0.01**	0.03	0.02	0.03
	Cluster	Sparse	20	0.05	0.05	0.05	0.04	**0.02**
		Sparse	350	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**
		Dense	20	0.04	0.04	0.05	0.04	**0.00**
		Dense	350	**0.00**	**0.00**	0.01	0.02	0.01
	Scale-free	Sparse	20	0.05	0.05	0.06	0.05	**0.02**
		Sparse	350	**0.01**	**0.01**	0.02	**0.01**	0.03

Table 6: *Pr^- scores of the algorithms for different instances. The Pr^- reaches its best score at* 0 *and its worst at* 1*. The values are averages over 16 replications for $p \in \{10, 100\}$ and over 8 replications for $p = 1000$. For each setting, the best-performing algorithm is highlighted in* ***bold****.*

p	Graph	Density	n	RJ-MPL	BD-MPL	BD	SS	B-CON
1000	Random	Sparse	400	16000K	300K	10	600	40K
		Sparse	1050	16000K	200K	10	400	40K
		Dense	400	30000K	1500K	10	1500	250K
		Dense	1050	10000K	500K	10	1500	80K
	Cluster	Sparse	400	16000K	300K	10	600	40K
		Sparse	1050	16000K	200K	10	400	40K
		Dense	400	30000K	1500K	10	1500	250K
		Dense	1050	30000K	500K	10	1500	80K
	Scale-free	Sparse	400	30000K	200K	10	600	50K
		Sparse	1050	30000K	200K	10	200	50K
100	Random	Sparse	40	125000K	2500K	30K	45K	400K
		Sparse	700	125000K	2500K	30K	45K	400K
		Dense	40	125000K	2500K	30K	45K	400K
		Dense	700	125000K	2500K	30K	45K	400K
	Cluster	Sparse	40	125000K	2500K	30K	45K	400K
		Sparse	700	125000K	2500K	30K	45K	400K
		Dense	40	125000K	2500K	30K	45K	400K
		Dense	700	125000K	2500K	30K	45K	400K
	Scale-free	Sparse	40	125000K	2500K	30K	45K	400K
		Sparse	700	125000K	2500K	30K	45K	400K
10	Random	Sparse	20	100K	30K	30K	3K	10K
		Sparse	350	100K	30K	30K	3K	10K
		Dense	20	100K	30K	30K	3K	10K
		Dense	350	100K	30K	30K	3K	10K
	Cluster	Sparse	20	100K	30K	30K	3K	10K
		Sparse	350	100K	30K	30K	3K	10K
		Dense	20	100K	30K	30K	3K	10K
		Dense	350	100K	30K	30K	3K	10K
	Scale-free	Sparse	20	100K	30K	30K	3K	10K
		Sparse	350	100K	30K	30K	3K	10K

Table 7: *Number of MCMC iterations until AUC-PR convergence for different instances. The time limit was set to five days, which is why the number of iterations for the BD algorithm for cases with* $p = 1000$ *is only 10.*

	BD-MPL	RJ-MPL	BD	SS	B-CON
BD-MPL	-	0.005	0.067	0.077	0.023
RJ-MPL	-	-	0.067	0.077	0.023
BD	-	-	-	0.045	0.074
SS	-	-	-	-	0.085
B-CON	-	-	-	-	-

Table 8: *Average absolute difference in edge inclusion probabilities between algorithms on the human gene data set.*

	BD-MPL	RJ-MPL	BD	SS	B-CON
BD-MPL (73)	-	0.97	0.63	0.45	0.85
RJ-MPL (75)	0.95	-	0.61	0.45	0.87
BD (68)	0.68	0.68	-	0.49	0.79
SS (35)	0.94	0.97	0.94	-	1.00
B-CON (87)	0.71	0.75	0.62	0.40	-

Table 9: *Proportion of edges identified by the row algorithm that are also found by the column algorithm on the human gene data set, using an edge inclusion probability threshold of 0.9. The numbers in brackets indicate the count of edges with an edge inclusion probability greater than 0.9.*

	BD-MPL	RJ-MPL	SS	B-CON
BD-MPL	-	0.019	0.026	0.071
RJ-MPL	-	-	0.026	0.072
SS	-	-	-	0.078
B-CON	-	-	-	-

Table 10: *Average absolute difference in edge inclusion probabilities across algorithms for the mice gene data set ($p = 623$).*

	BD-MPL	RJ-MPL	SS	B-CON
BD-MPL (3,965)	-	0.70	0.15	0.80
RJ-MPL (4,282)	0.65	-	0.14	0.78
SS (656)	0.92	0.91	-	0.96
B-CON (14,258)	0.22	0.23	0.04	-

Table 11: *Proportion of edges identified by the row algorithm that are also found by the column algorithm on the mice data set, using an edge inclusion probability threshold of 0.9. Between brackets is the number of edges with an edge inclusion probability higher than* 0.9*.*