5 Supplement for "Joint Structure Learning of Multiple Non-Exchangeable Networks", Chris. J. Oates and Sach Mukherjee, AISTATS 2014.

5.1 Belief propagation for SLTs

Following marginalisation of continuous parameters $\boldsymbol{\theta}$, inference for SLTs reduces to inference for a discrete Bayesian network whose nodes are themselves graphical models (SFig. 4). In this Section we describe the use of belief propagation (BP; Pearl (1982)) for inference in this setting and provide pseudocode for the 2-tier SLT model.

Denote by $\boldsymbol{X}$ a vector of random variables whose density factorizes according to

$$p_{\boldsymbol{X}}(\boldsymbol{x}) = \prod_{f \in \mathcal{F}} f(\boldsymbol{x}_f) \tag{7}$$

where $\boldsymbol{x}_f$ denotes the components of vector $\boldsymbol{x}$ upon which the factor f depends. The factor graph corresponding to the 2-tier SLT model is shown in SFig. 4c. We use $\mu_{v \to f}$ to denote a message passed from a variable v to a factor f, whereas $\overline{\mu}_{f \to v}$ will be used to denote a message passed from a factor f to a variable v. The message from a variable v to a factor f takes the following form:

$$\mu_{v \to f}(x_v) = \prod_{f^* \in N(v) \setminus \{f\}} \overline{\mu}_{f^* \to v}(x_v) \tag{8}$$

where $N(v)$ denotes the neighbours of variable v according to the factor graph. Similarly the message from a factor f to a variable v takes the form

$$\overline{\mu}_{f \to v}(x_v) = \sum_{\boldsymbol{x}' : x'_v = x_v} f(\boldsymbol{x}'_f) \prod_{v^* \in N(f) \setminus \{v\}} \mu_{v^* \to f}(x'_{v^*}) \tag{9}$$

where $N(f)$ denotes the neighbours of factor f according to the factor graph.

To simplify notation, we describe our algorithm using subscript notation as in the Main Text; e.g. $\overline{G_{1ij}}$ denotes the network that is the jth child of the ith child of the root network $\overline{G_1}$ in T. BP nominates one node in the factor graph as a "root"; of the remaining nodes, those with degree one are known as "leaves". For BP applied to SLT we nominate the network $\overline{G_1}$ as the root node. Messages are initiated at the leaves of the factor graph; specifically, in our 2-tier example, each variable node $\boldsymbol{Y}_{1ij}$ is initialised with an atomic distribution $\delta\{\boldsymbol{Y}_{1ij} = \boldsymbol{y}_{1ij}\}$ centered on the observed data $\boldsymbol{y}_{1ij}$. Messages are passed through to the root node before being returned to the leaves.

Once the message passing has been completed, it is possible to extract marginals of interest by taking products of messages from factors neighboring the random variable of interest:

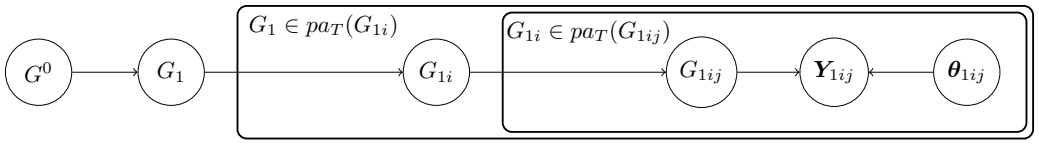
$$p_{X_v}(x_v) \propto \prod_{f \in N(v)} \overline{\mu}_{f \to v}(x_v) \tag{10}$$

Alg. 1 contains pseudocode for the BP algorithm in the context of 2-tier SLTs.

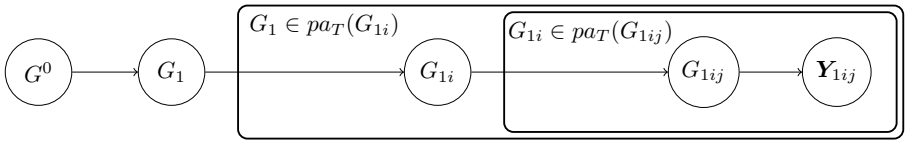
5.2 Simulation study

5.2.1 Data generation

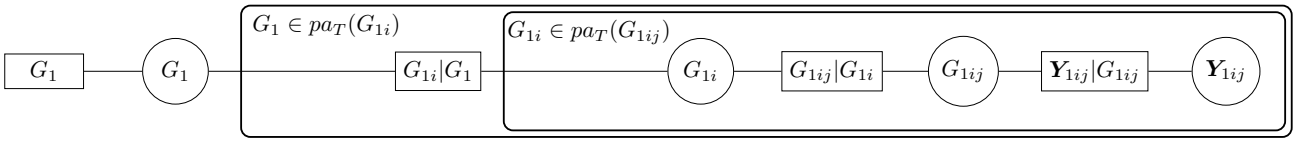
From each network $\overline{G_{1ij}}$ we generated time series data $\boldsymbol{y}_{1ij}$, each containing n time points, according to a linear VAR(1) process. For each time series one variable was selected uniformly at random to be the target of a perfect intervention (Spencer and Mukherjee, 2012). Dynamical parameters were assigned such that for each edge (i, j) we select a data-generating coefficient $\beta \in \{-1, +1\}$ uniformly at random. For all experiments we used a noise magnitude $\sigma = 1$. In each regime we generated data of varying sample size n and edge density ρ. Specifically, we considered both varying n for fixed $\rho = 0.5$ and varying ρ for fixed $n = 60$.



(a) Graphical model representation.



(b) The marginal SLT.



(c) Factor graph representation.

Figure 4: Structure Learning Trees (SLT); 2-tier example. (a) Graphical model representation. [G^0 = prior network, G_1 = root network, G_{1i} = tier-1 networks, G_{1ij} = tier-2 networks, $\boldsymbol{Y}_{1ij}$ = data available on network G_{1ij}, $\boldsymbol{\theta}_{1ij}$ = parameters describing the distribution of the data $\boldsymbol{Y}_{1ij}$. Bounding boxes are used to denote multiplicity of variables.] (b) The marginal SLT is obtained from (a) by integrating out continuous parameters $\boldsymbol{\theta}_{1ij}$. (c) Factor graph representation of the marginal SLT (b). [Circled nodes are random variables, rectangular nodes are factors. Dependence on the prior network is suppressed.]

5.2.2 Performance measures

Denote the true data-generating (binary) adjacency matrix by $\boldsymbol{A}^0$. In this work we considered the performance of two kinds of estimator; (i) the weighted adjacency matrices $\boldsymbol{A}$ produced by collecting together posterior marginal inclusion probabilities, and (ii) the binary adjacency matrices $\boldsymbol{A}(\tau)$ with (i,j)th entry $\mathbb{I}(A_{ij} > \tau)$, i.e. including edges if and only if the corresponding posterior marginal inclusion probabilities exceed a threshold τ. Write $\mathrm{TP}(\tau)$, $\mathrm{FP}(\tau)$, $\mathrm{TN}(\tau)$, $\mathrm{FN}(\tau)$ for, respectively, the true positive, false positive, true negative and false negative counts obtained by comparing $\boldsymbol{A}(\tau)$ to $\boldsymbol{A}^0$. Further write $\mathrm{TPR}(\tau) = \mathrm{TP}(\tau) \,/\, (\mathrm{TP}(\tau) + \mathrm{FN}(\tau))$, $\mathrm{FPR}(\tau) = \mathrm{FP}(\tau) \,/\, (\mathrm{TN}(\tau) + \mathrm{FP}(\tau))$, $\mathrm{PPV}(\tau) = \mathrm{TP}(\tau) \,/\, (\mathrm{TP}(\tau) + \mathrm{FP}(\tau))$.

For (i) we considered the following performance measures:

(1) L1 Error $= \sum_{ij} |A_{ij} - A^0_{ij}|$

(2) Relative Density $= \sum_{ij} |A_{ij}| / \sum_{ij} |A^0_{ij}|$

(3) AUROC $= \int \mathrm{TPR}(\tau) d\mathrm{FPR}(\tau)$

(4) AUPR $= \int \mathrm{PPV}(\tau) d\mathrm{TPR}(\tau)$

For (ii) special attention is afforded to the "median" estimator with $\tau = 0.5$. Specifically we considered the performance measures

(1) Matthews Correlation Coefficient $= (\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}) / \sqrt{((\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN}))}$

(2) Misclassification Rate = (FP + FN) / P^2

Algorithm 1 Belief propagation (BP) for the 2-tier SLT model. Here we list all steps of the BP algorithm in order; at each stage messages are passed for all relevant networks indexed by i and j, but we leave this implicit for clarity.

1: $\mu_{\boldsymbol{Y}_{1ij}\to\boldsymbol{Y}_{1ij}|G_{1ij}}(\boldsymbol{Y}_{1ij}) = \delta\{\boldsymbol{Y}_{1ij} = \boldsymbol{y}_{1ij}\}$

2: $\overline{\mu}_{\boldsymbol{Y}_{1ij}|G_{1ij}\to G_{1ij}}(G_{1ij}) = \int_{\boldsymbol{Y}_{1ij}} p(\boldsymbol{Y}_{1ij}|G_{1ij})\mu_{\boldsymbol{Y}_{1ij}\to\boldsymbol{Y}_{1ij}|G_{1ij}}(\boldsymbol{Y}_{1ij})d\boldsymbol{Y}_{1ij}$

3: $\mu_{G_{1ij}\to G_{1ij}|G_{1i}}(G_{1ij}) = \overline{\mu}_{\boldsymbol{Y}_{1ij}|G_{1ij}\to G_{1ij}}(G_{1ij})$

4: $\overline{\mu}_{G_{1ij}|G_{1i}\to G_{1i}}(G_{1i}) = \sum_{G_{1ij}} p(G_{1ij}|G_{1i})\mu_{G_{1ij}\to G_{1ij}|G_{1i}}(G_{1ij})$

5: $\mu_{G_{1i}\to G_{1i}|G_1}(G_{1i}) = \prod_j \overline{\mu}_{G_{1ij}|G_{1i}\to G_{1i}}(G_{1i})$

6: $\overline{\mu}_{G_{1i}|G_1\to G_1}(G_1) = \sum_{G_{1i}} p(G_{1i}|G_1)\mu_{G_{1i}\to G_{1i}|G_1}(G_{1i})$

7: $\overline{\mu}_{G_1\to G_1}(G_1) = p(G_1)$

8: $\mu_{G_1\to G_{1i}|G_1}(G_1) = \overline{\mu}_{G_1\to G_1}(G_1)\prod_{i'\neq i}\overline{\mu}_{G_{1i'}|G_1\to G_1}(G_1)$

9: $\overline{\mu}_{G_{1i}|G_1\to G_{1i}}(G_{1i}) = \sum_{G_1} p(G_{1i}|G_1)\mu_{G_1\to G_{1i}|G_1}(G_1)$

10: $\mu_{G_{1i}\to G_{1ij}|G_{1i}}(G_{1i}) = \overline{\mu}_{G_{1i}|G_1\to G_{1i}}(G_{1i})$

11: $\overline{\mu}_{G_{1ij}|G_{1i}\to G_{1ij}}(G_{1ij}) = \sum_{G_{1i}} p(G_{1ij}|G_{1i})\mu_{G_{1i}\to G_{1ij}|G_{1i}}(G_{1i})$

12: $p(G_1|\boldsymbol{y}) = \overline{\mu}_{G_1\to G_1}(G_1)\prod_i \overline{\mu}_{G_{1i}|G_1\to G_1}(G_1)$

13: $p(G_{1i}|\boldsymbol{y}) = \overline{\mu}_{G_{1i}|G_1\to G_{1i}}(G_{1i})\prod_j \overline{\mu}_{G_{1ij}|G_{1i}\to G_{1i}}(G_{1i})$

14: $p(G_{1ij}|\boldsymbol{y}) = \overline{\mu}_{G_{1ij}|G_{1i}\to G_{1ij}}(G_{1ij})\overline{\mu}_{\boldsymbol{Y}_{1ij}|G_{1ij}\to G_{1ij}}(G_{1ij})$

(3) Misclassification Rate (top k edges) = As for the misclassification rate, but with τ chosen such that $\boldsymbol{A}(\tau)$ contains exactly k non-zero entries, where k is the number of edges in the true data-generating network.

(4) Precision = TP / (TP + FP).

5.2.3 Additional results

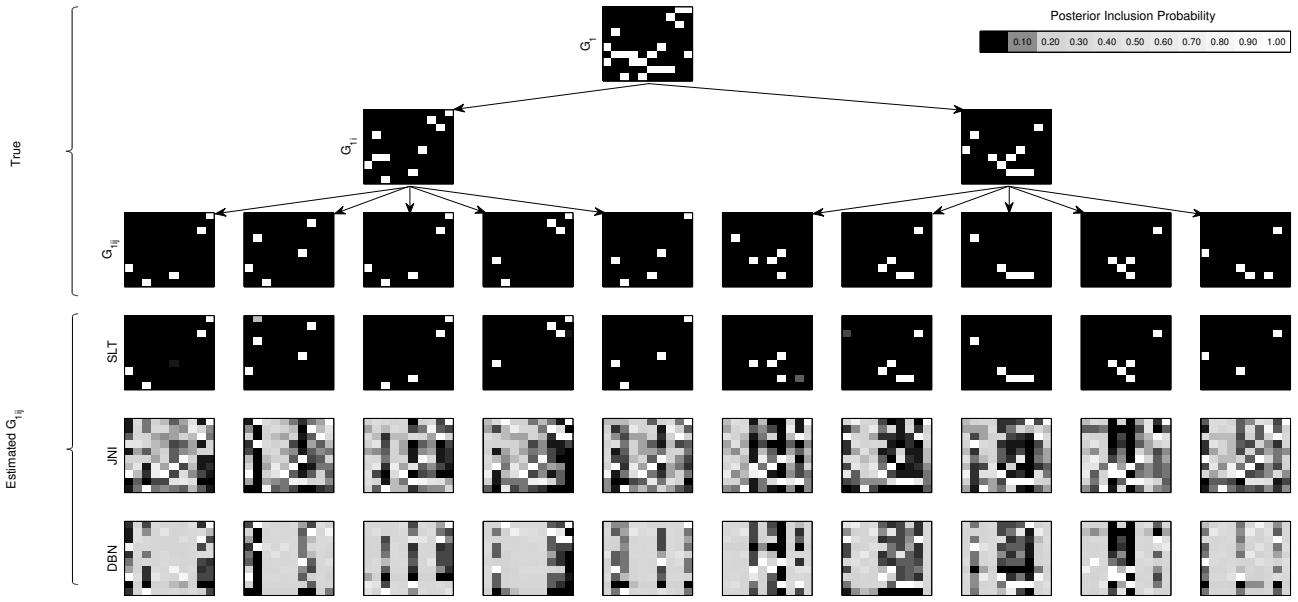
SFigs. 5a, 5b, 6a, 6b display typical simulation examples for regimes 2-5 respectively, and SFigs. 7-11 display full simulation results for each of the the 5 regimes described in the Main Text.

5.3 Experimental protocol

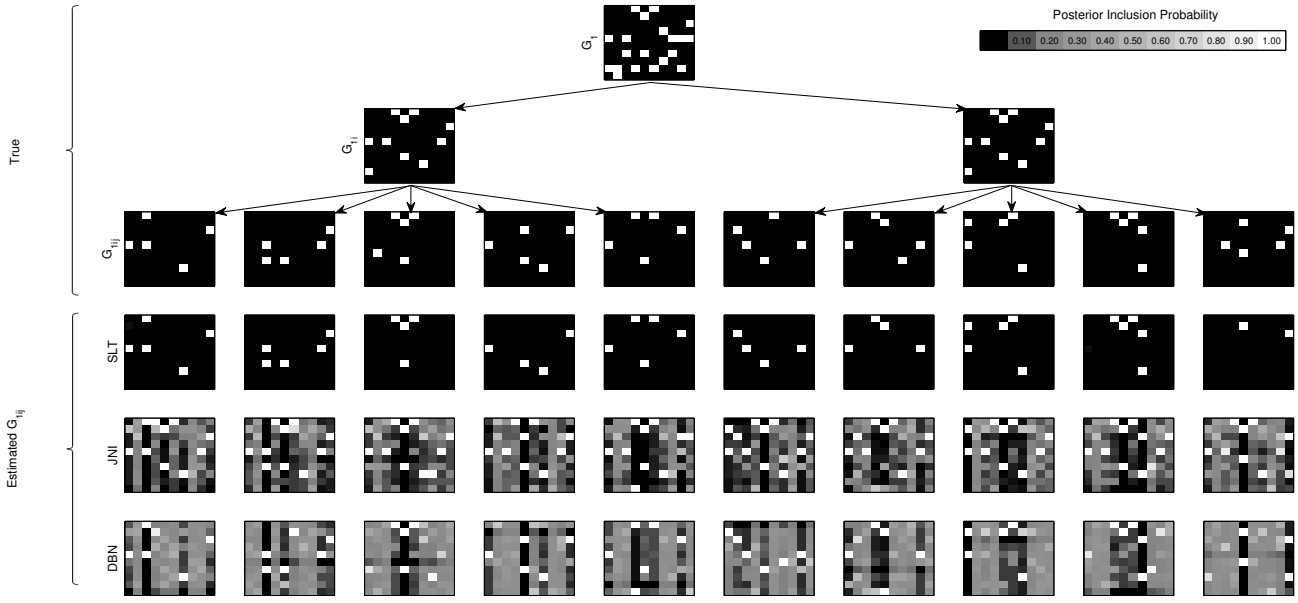
Cells were plated into 10 cm^2 dishes at a density of $1 - 2 \times 10^6$ cells. After 24 hours, cells were treated with 250 nM lapatinib or 250 nM AKTi (GSK690693). DMSO served as a control. Cells were grown in 10% FBS and harvested in RPPA lysis buffer at 30 min, 1h, 2h, 4h, 8h, 24h, 48h, and 72h post-treatment. Cell lysates were quantitated, diluted, arrayed, and probed following Tibes *et al.* (2006). Imaging and quantitation of signal intensity was done following Tibes *et al.* (2006). The particular protein species analysed were 4EBP1(pT37), AKT(pS473), BAD(pS112), c-Myc(pT58), EGFR(pY1173), ELK1(pS383), ER, FOXO3a(pS318), GSK3ab(pS21), HER2, IRS1(pS307), MAPK(pT202), MEK1/2(pS217), p38(pT180), p53, PR and S6(pS240).

References

Tibes *et al.* (2006) Reverse phase protein array: validation of a novel proteomic technology and utility for analysis of primary leukemia specimens and hematopoietic stem cells. *Mol. Cancer Ther.* **5**(10):2512-2521.

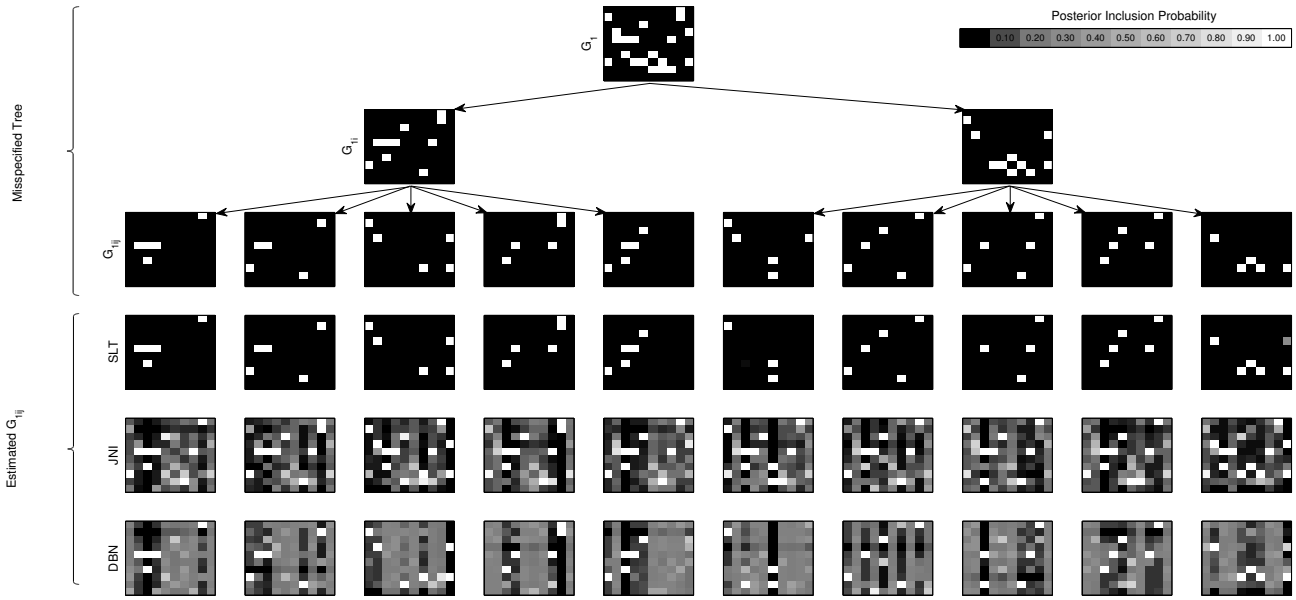


(a) Weakly exchangeable

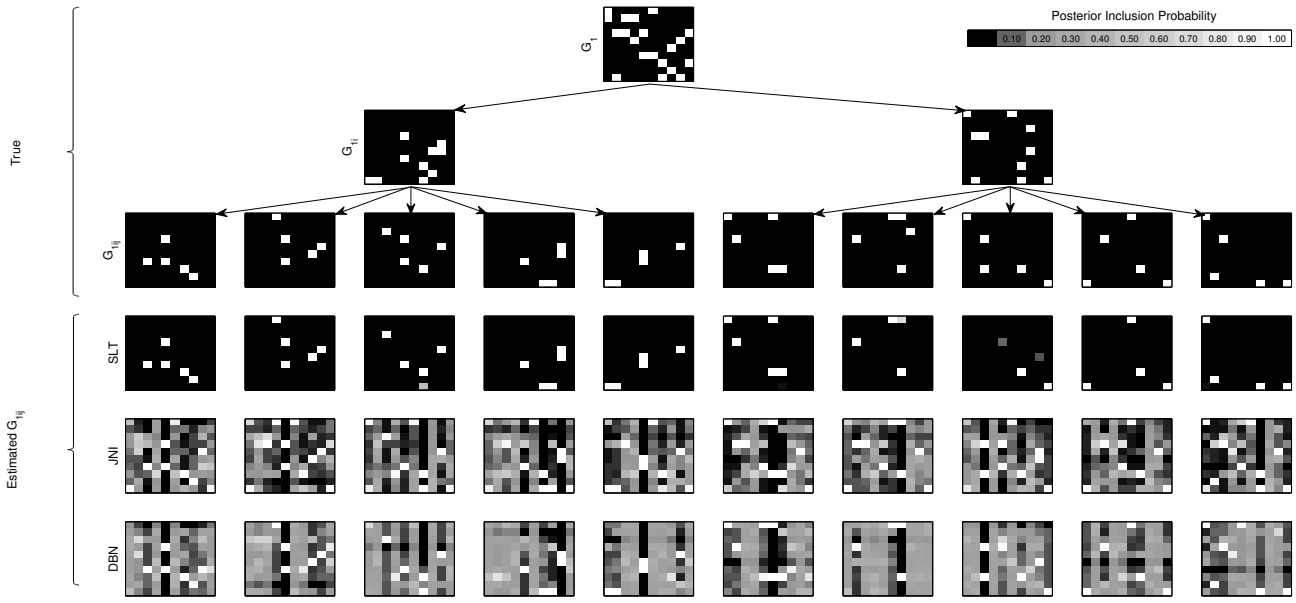


(b) Fully exchangeable

Figure 5: Results on simulated data generated from 2-tier SLTs; (a) a weakly exchangeable population, where $\overline{G_{11}}$, $\overline{G_{12}}$ are likely to share edges, and (b) a fully exchangeable population. [Inference methods: "SLT" = structure learning trees, "JNI" = joint network inference (Oates *et al.*, 2013), "DBN" = independent network inference (this corresponds to structure learning under the same local likelihood as SLT and JNI but applied separately to the data-generating networks located at the leaves of the tree).]



(a) Misspecified Tree



(b) Subset violation

Figure 6: Results on simulated data generated from 2-tier SLTs; (a) a misspecified tree structure T, and (b) a weakly exchangeable population which violates the subset assumptions encoded in the joint structural prior used by SLT. [Inference methods: "SLT" = structure learning trees, "JNI" = joint network inference (Oates *et al.*, 2013), "DBN" = independent network inference (this corresponds to structure learning under the same local likelihood as SLT and JNI but applied separately to the data-generating networks located at the leaves of the tree).]

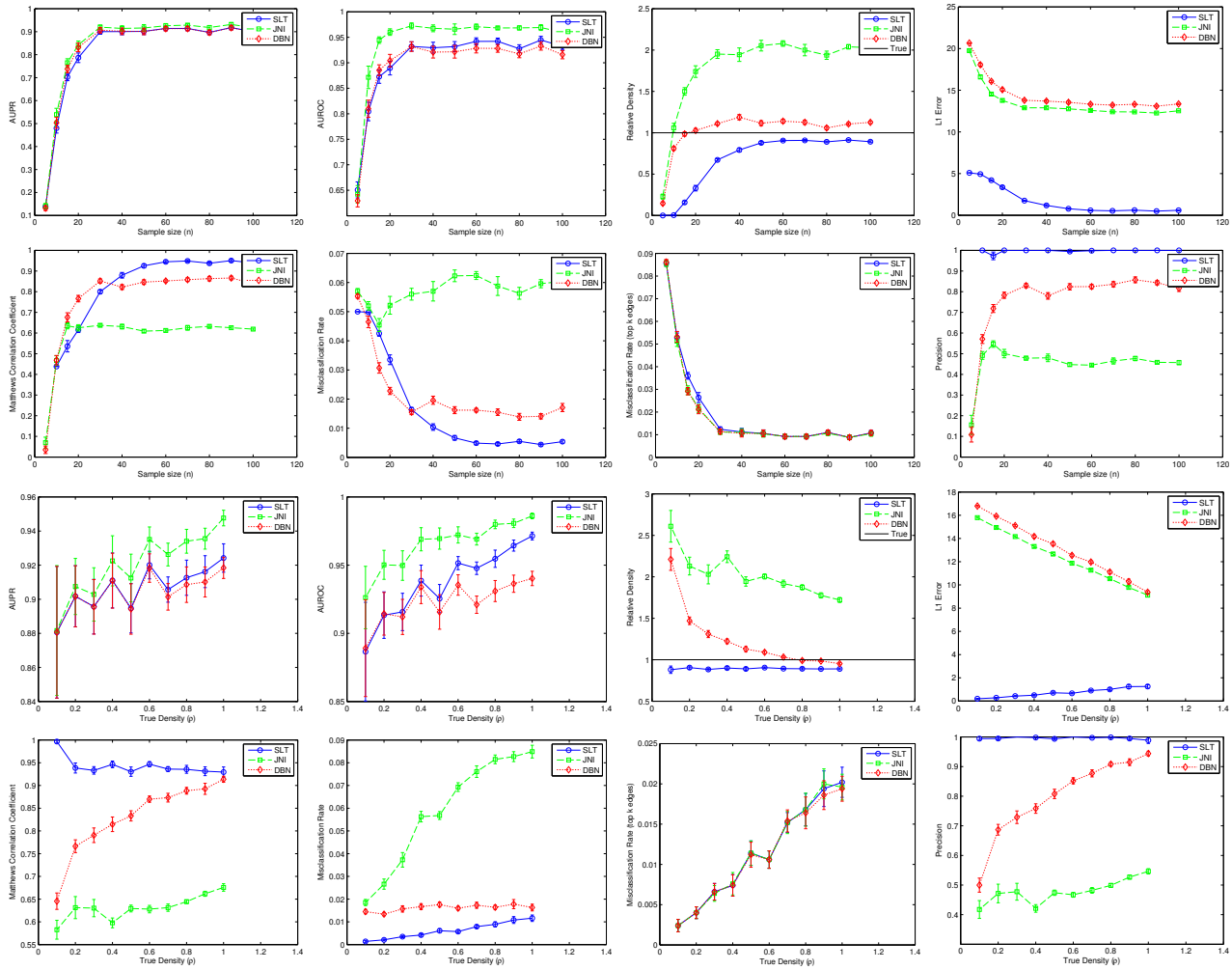


Figure 7: Results on simulated data generated from a 2-tier SLT with disjoint sub-group structure. [Network estimators: "SLT" = structure learning trees; "JNI" = joint network inference; "DBN" = inference for each network independently. For each estimator we considered both thresholded and un-thresholded adjacency matrices. Performance scores: "AUROC" = area under the receiver operating characteristic curve; "AUPR" = area under the precision-recall curve; "L1 Error" = ℓ_1 distance from the true adjacency matrices to the inferred weighted adjacency matrices; "top k edges" = the ρP most probable edges. Performance scores were averaged over all 10 data-generating networks and all 10 datasets; error bars denote standard errors of mean performance over datasets. We considered both varying n for fixed $\rho = 0.5$ and varying ρ for fixed $n = 60$.]

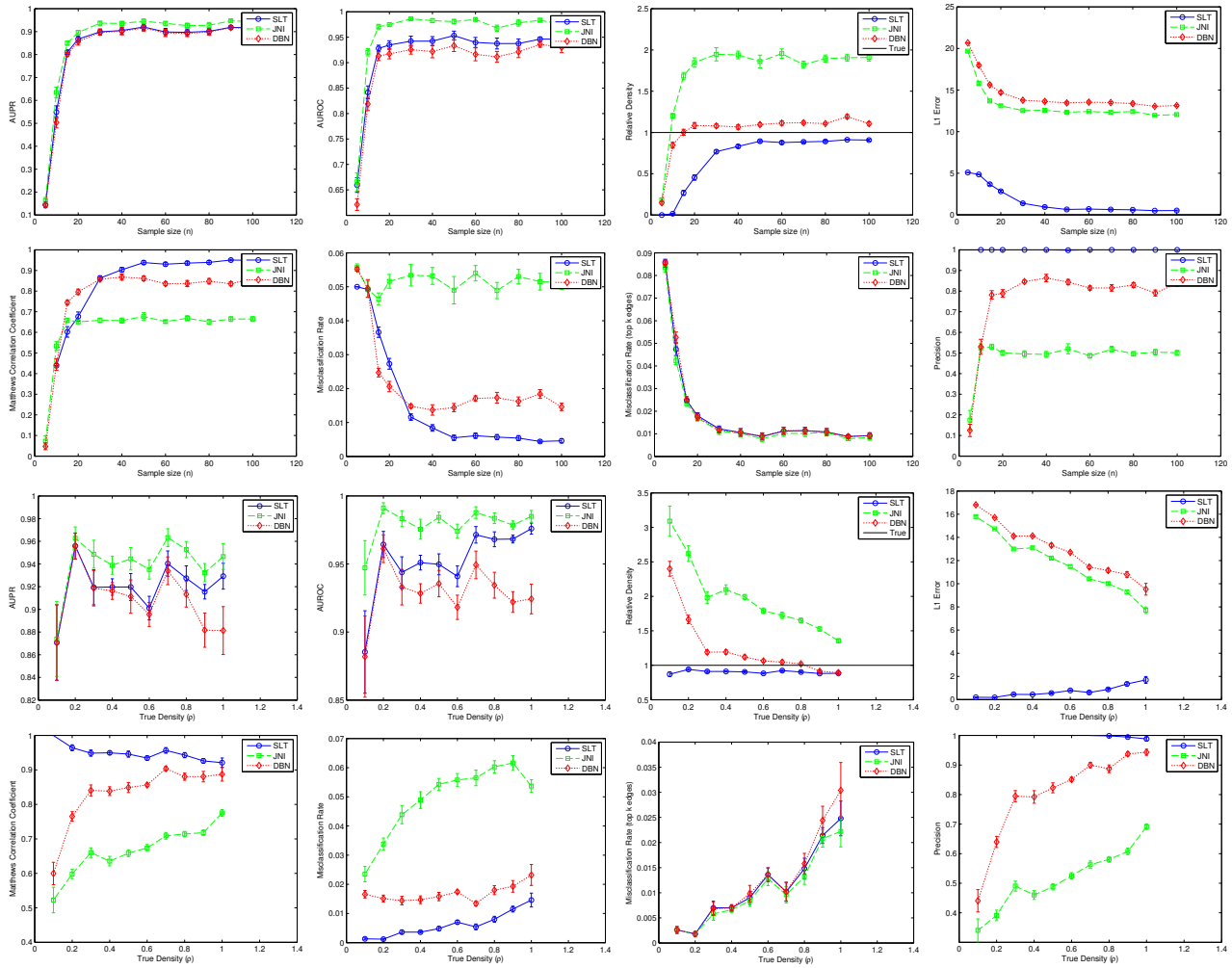


Figure 8: Results on simulated data generated from a 2-tier SLT with weakly exchangeable structure [Network estimators: "SLT" = structure learning trees; "JNI" = joint network inference; "DBN" = inference for each network independently. For each estimator we considered both thresholded and un-thresholded adjacency matrices. Performance scores: "AUROC" = area under the receiver operating characteristic curve; "AUPR" = area under the precision-recall curve; "L1 Error" = ℓ_1 distance from the true adjacency matrices to the inferred weighted adjacency matrices; "top k edges" = the ρP most probable edges. Performance scores were averaged over all 10 data-generating networks and all 10 datasets; error bars denote standard errors of mean performance over datasets. We considered both varying n for fixed $\rho = 0.5$ and varying ρ for fixed $n = 60$.]

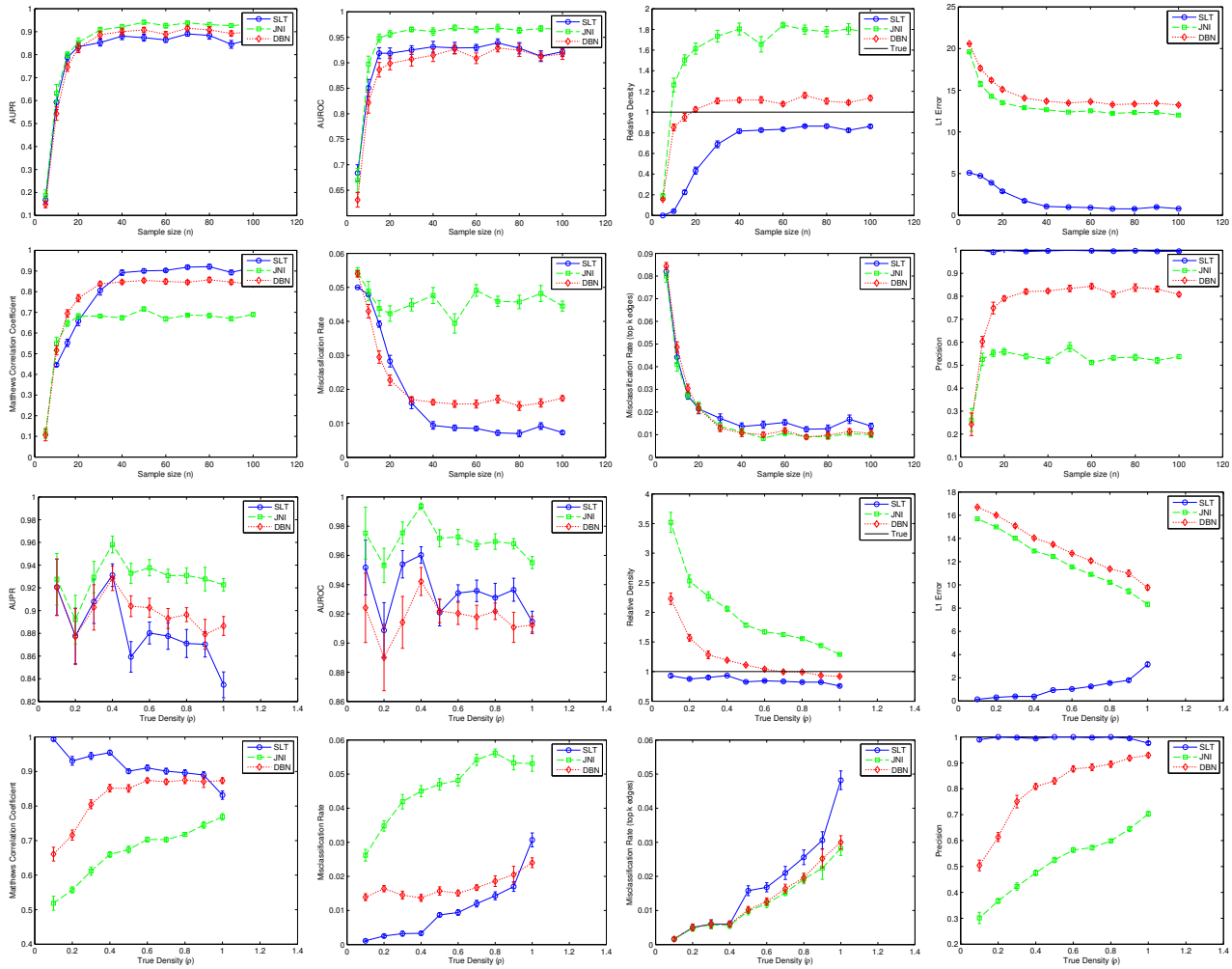


Figure 9: Results on simulated data generated from a fully exchangeable SLT. [Network estimators: "SLT" = structure learning trees (based on 2 tiers); "JNI" = joint network inference; "DBN" = inference for each network independently. For each estimator we considered both thresholded and un-thresholded adjacency matrices. Performance scores: "AUROC" = area under the receiver operating characteristic curve; "AUPR" = area under the precision-recall curve; "L1 Error" = ℓ_1 distance from the true adjacency matrices to the inferred weighted adjacency matrices; "top k edges" = the ρP most probable edges. Performance scores were averaged over all 10 data-generating networks and all 10 datasets; error bars denote standard errors of mean performance over datasets. We considered both varying n for fixed $\rho = 0.5$ and varying ρ for fixed $n = 60$.]

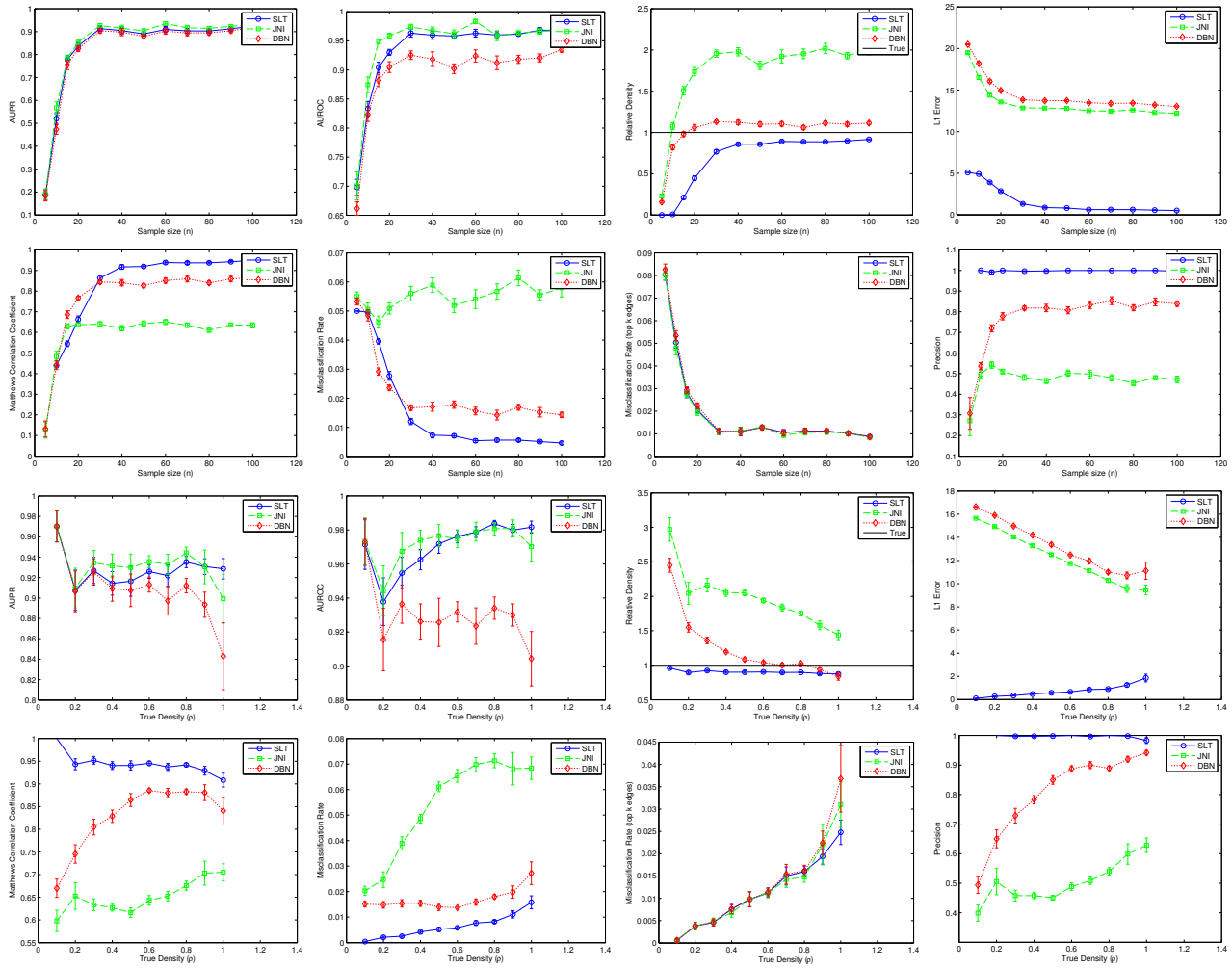


Figure 10: Results on simulated data generated from a 2-tier SLT with misspecified tree structure. [Network estimators: "SLT" = structure learning trees; "JNI" = joint network inference; "DBN" = inference for each network independently. For each estimator we considered both thresholded and un-thresholded adjacency matrices. Performance scores: "AUROC" = area under the receiver operating characteristic curve; "AUPR" = area under the precision-recall curve; "L1 Error" = ℓ_1 distance from the true adjacency matrices to the inferred weighted adjacency matrices; "top k edges" = the ρP most probable edges. Performance scores were averaged over all 10 data-generating networks and all 10 datasets; error bars denote standard errors of mean performance over datasets. We considered both varying n for fixed $\rho = 0.5$ and varying ρ for fixed $n = 60$.]

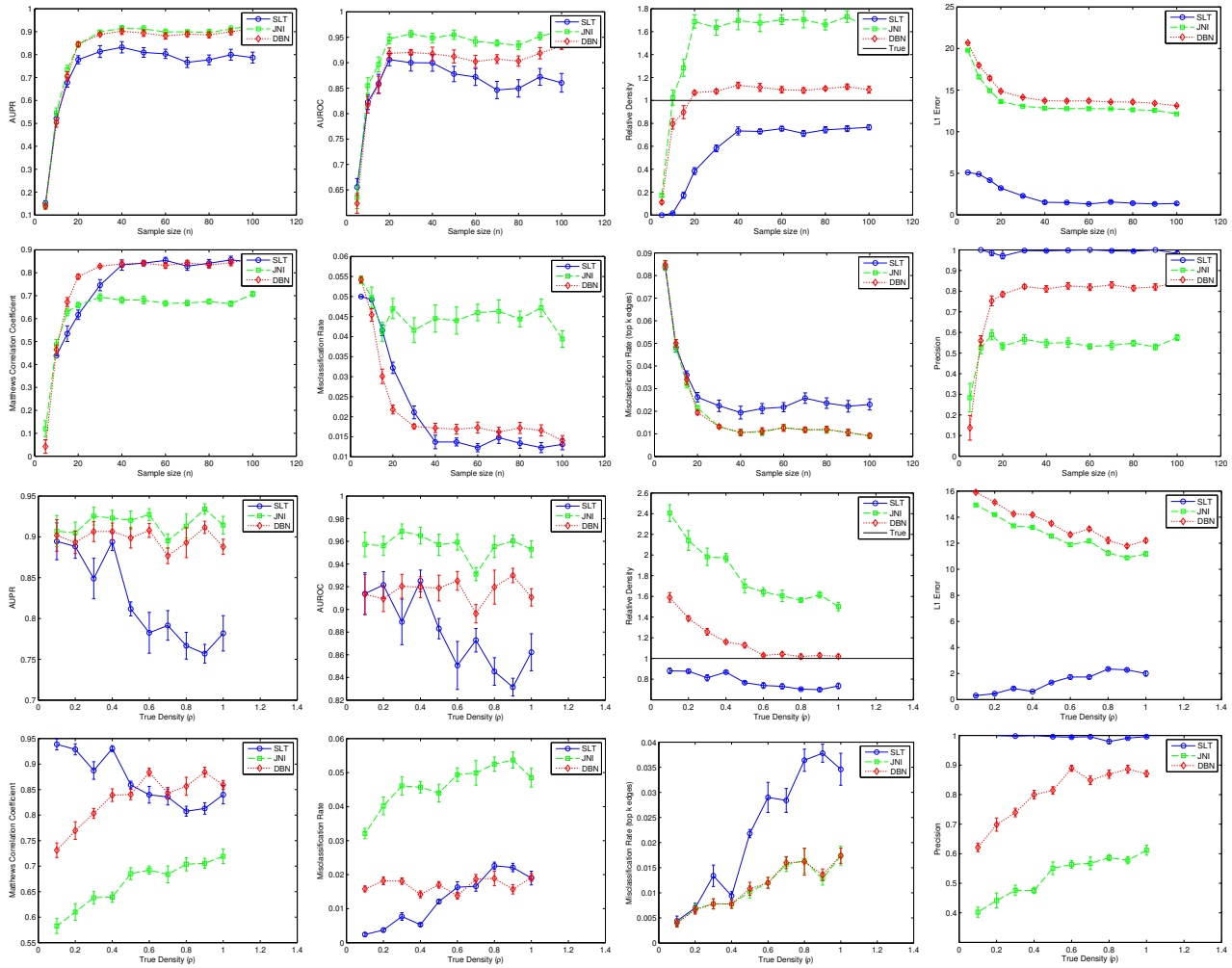


Figure 11: Results on simulated data generated from a 2-tier SLT with structure which violates the subset assumption. [Network estimators: "SLT" = structure learning trees; "JNI" = joint network inference; "DBN" = inference for each tier-3 network independently. For each estimator we considered both thresholded and un-thresholded adjacency matrices. Performance scores: "AUROC" = area under the receiver operating characteristic curve; "AUPR" = area under the precision-recall curve; "L1 Error" = ℓ_1 distance from the true adjacency matrices to the inferred weighted adjacency matrices; "top k edges" = the ρP most probable edges. Performance scores were averaged over all 10 data-generating networks and all 10 datasets; error bars denote standard errors of mean performance over datasets. We considered both varying n for fixed $\rho = 0.5$ and varying ρ for fixed $n = 60$.]