Table 1: Performance statistics on the GNW dataset

	AROC	AUPR	F1
GENIE3	0.719	0.322	0.397
xgboost.Gxgb	0.754	0.283	0.377
bnlearn.bde.bootstrap	0.751	0.282	0.380
minet.mrnet	0.746	0.268	0.386
naive.spearman	0.662	0.159	0.278
GeneNet	0.604	0.139	0.195

Table 2: Comparision of one-shot algorithms that output binary weights

TP	FP	TN	FN	PR	RC	SP	NPN	F1	method
8	14	705	53	0.364	0.131	0.981	0.930	0.193	bnlearn.pc
11	25	694	50	0.306	0.180	0.965	0.933	0.227	bnlearn.bde

Contents

0.1 Comparing DREAM5 competitors 1
0.2 One-shot algorithms 2
0.3 A homemade regressional predictor with xgboost([2]) 2
0.4 Comparison of algorithms 2
0.5 Assessing the performance of algorithms with bootstrapped subnetworks 3


```
## All packages requirements are met: "doSNOW","doParallel","doMPI"
```

0.1 Comparing DREAM5 competitors

0.1.1 Method

0.1.1.1 Data

GNW network: I used the pre-simulated data "medium_network.rda" that included the expression levels of 40 genes along with the underlying true network to test the performance of various algorithms. This dataset is generated by GeneNetWeaver ([1]) hence called GNW dataset.

DREAM5 dataset: The E. coli network in the Dream5 competition has proved to be fittable and hence used here to evaluate the consistence of algorithms.

0.1.1.2 Formalism

I assumed the interaction matrix A_{ij} is symmetric and binary, and $A_{ij} = 1$ if there is interaction between gene i and gene j, otherwise $A_{ij} = 0$. The problem then reduce to a binary classification (existence of an undirected edge) and standard confusion matrix may be constructed to evaluate the performance. Specifically, area-under-precision-recall-curve (AUPR) and area-under-the-ROC (AROC) will be plotted for comparision, along with a list of F1 score.

```
## TableGrob (2 x 1) "arrange": 2 grobs
##   z     cells    name             grob
## 1 1 (1-1,1-1) arrange   gtable[arrange]
## 2 2 (2-2,1-1) arrange gtable[guide-box]
```

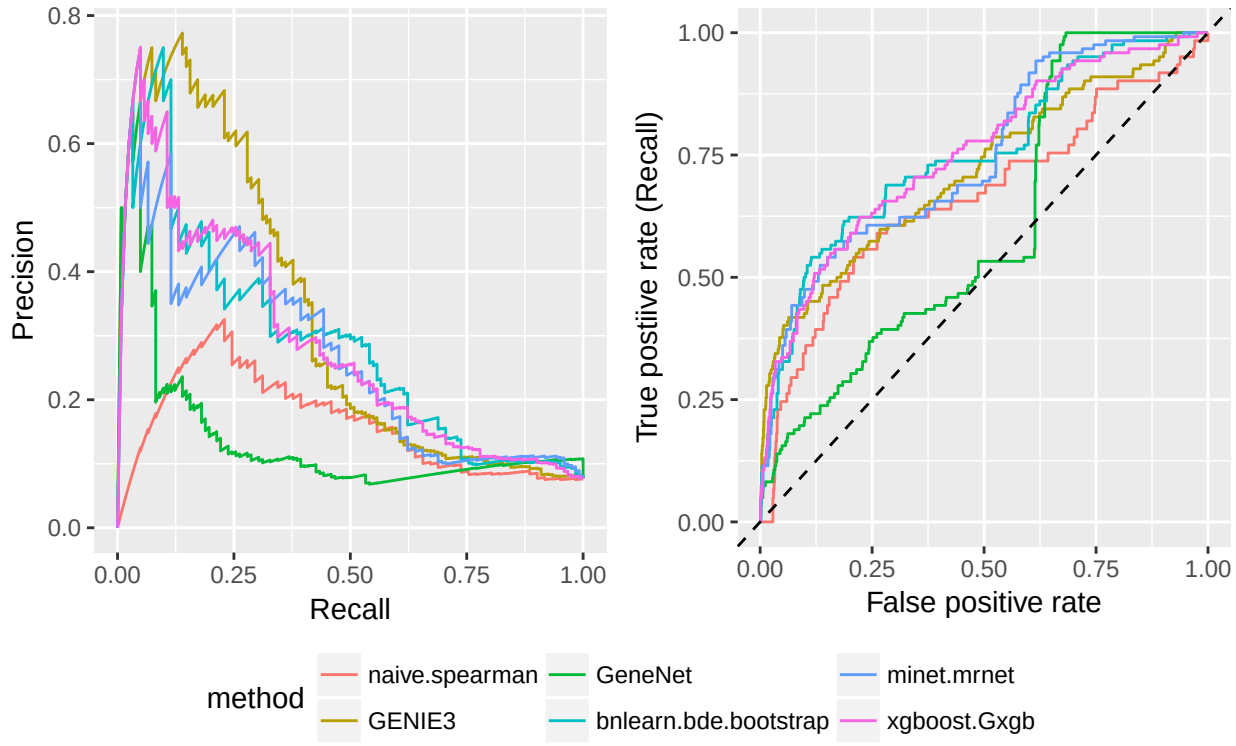

Figure 1: Comparing performances of different methods on the provided GNW sample

0.2 One-shot algorithms

pc.stable and hc(score='bde') was run on the GNW dataset but the precision was not so exciting (table 2). However, notice pc.stable is giving a higher precision, potentially warranting a bootstrap checking in the future.

0.3 A homemade regressional predictor with xgboost([2])

GENIE3 ([3]) uses random forest to fit the decision tree and then extract the importance. But here I will be fitting the trees with "xgboost" instead because of its simple interface and famous efficacy ([2]). Specifically, I fitted a decision tree to the function $x_j = f(\vec{x}_{-j})$, where x_j is expression of the chosen gene and x_{-j} is that of the rest genes. I used the Least sqaure loss ("reg:linear") for the fitting.

By the magic of xgboost, an importance vector is obatined (p_j^i indicates relative importance of gene i in predicting gene j, with $\sum_i p_j^i = 1$). I intuitively intepret it as the ratio of variance explained by splitting on gene i (though without firm matheamtical checking). Hence the importance vector is then scaled by the variance of the predictee ($P_{ij} = p_j^i \cdot Var(p_j)$) and then pooled together as the final confidence score for the connection between gene i and gene j, with a higher score indicating a higher likelihood of interaction.

0.4 Comparison of algorithms

A preliminary comparision is done between GeneNet, naive-spearman-predictor, bootstraped discrete bayesian net (bnlearn::boot.strength), GENIE3, homemade-xgboost (Gxgb) and MRnet. (figure 1, table 1). It is conceivable that GENIE3 gave the best AUPR (~0.377), but discrete bayesian net also gave a very robust prediction.

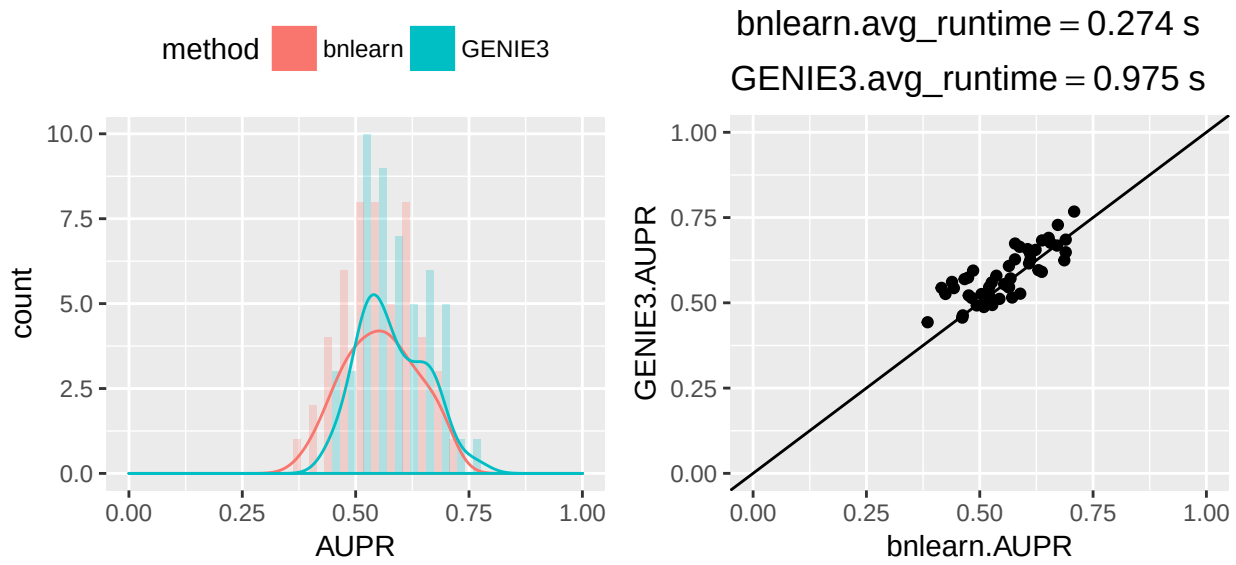

Figure 2: Comparing "GENIE3" and "bnlearn" on the bootstrapped subnets of the provided GeneNetWeaver simulation

0.5 Assessing the performance of algorithms with bootstrapped subnetworks

Note all predictions are very far from perfect (AUPR=1.00), which leads to the evaluation part. In short, each algorithm has pros and cons. Regressional predictors usually has very poor intepretability, and there is no mathematical framework that would guarantee its convergence to the true network. With the discrete bayesian networks, the discretisation is a somewhat arbitrary process that might introduce artefact to the result, as well as choosing a suitable imaginary sample size (ISS) for the network (here I will stick to ISS=1 for simplicity).

Furthermore, a big headache for evaluating any algorithm is its performance will depend on the dataset. A most trivial example consider comparing the prediction result on the 40-gene dataset and then on a random sub-network generated by performing a random walk on the original network in a post-hoc fashion (so that the sub-network has non-zero adjacency matrix due to connection between neighbors). The AUPR is chosen as the main statistics and recorded for each sub-network to produce a bootstrapped distribution (figure 2). This procedure is performed for GENIE3 and bnlearn because they gave the best performance in preliminary benchmarking (table 1). Gxgb was not included due to its slowness (further optimisation is required).

The same bootstrapping was performed on the original DREAM5 training data for network 3 (the E. coli network, 4511 genes, figure 3). It can be seen that bootstrapped bnlearn performs similarly to the GENIE3, while taking a much shorter time to run (both measured using 16 cores). Although this may be due to the small size of the sub-network, (10 nodes), it is possible that GENIE3 is performing redundant computation than actual required by the inference. It is also observed that both algorithms performs consistently poorer on the DREAM5 datasets.

[1] Schaffter T, Marbach D, Floreano D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics (Oxford, England) 2011;27:2263–70. doi:10.1093/bioinformatics/btr373.

[2] Chen T, Guestrin C. XGBoost: A scalable tree boosting system. CoRR 2016;abs/1603.02754.

[3] Schaffter T, Marbach D, Floreano D. GeneNetWeaver: in silico benchmark generation and performance profiling of network inference methods. Bioinformatics (Oxford, England) 2011;27:2263–70. doi:10.1093/bioinformatics/btr373.

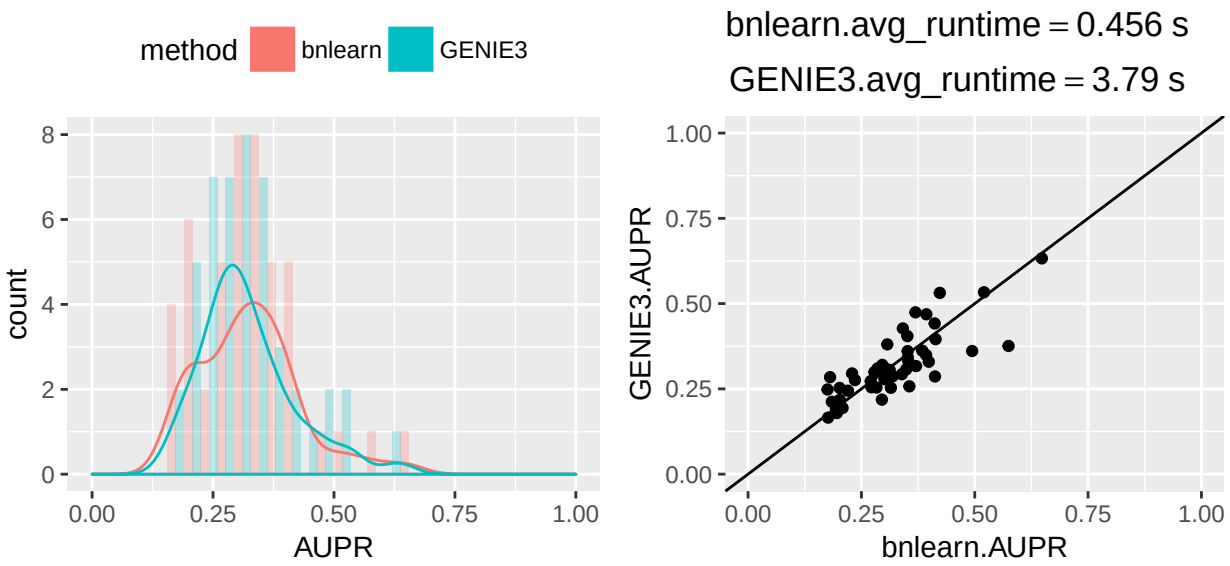

Figure 3: Comparing "GENIE3" and "bnlearn" on the bootstrapped subnets of DREAM5 network 3 (E. coli)