

Single-cell Gene Regulatory Network (scGRN)

BIML2020

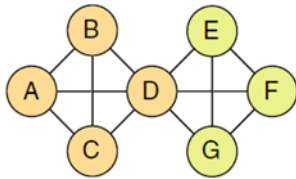
2020/02/01

Insuk Lee, Yonsei University

❖ Why single-cell network biology?

- Currently we can link only ~60% of the GWAS SNPs in regulatory elements to an eQTL effect. Many of disease-associated SNP may have **cell-type-specific** effects.
- Independent genetic risk factors can converge into key regulatory pathways. To understand pathway-level effect of genetic variants, we need **gene regulatory network (GRN)**.
- Therefore, a comprehensive understand of disease genetics needs **cell-type-specific GRN**.
- Furthermore, **personalized GRN** will facilitate implementing **precision medicine** in the future.

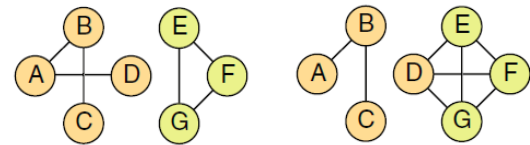
Integrated network for heterogeneous cell-types and population.



Specifying context

Single-cell data for each cell-type and individual.

Cell-type-specific and personalized GRNs



❖ Single-cell transcriptome data for modeling GRN

➤ Pros

- Larger numbers of data points (>1000's cells in general) yield **higher statistical power**.
- Regulatory relationship can be inferred by cell-to-cell variation within **cell-type** or **single person**.

➤ Cons

- High **noise** and **sparsity** (dropout): high false positive rates

❖ Types of single-cell GRN

- Single-cell Transcription Regulatory Network (**scTRN**)
- Single-cell Functional Gene Network (**scFGN**)

❖ Single-cell Transcription Regulatory Network (scTRN) from scRNA-seq data


- TRN inference from transcriptome data relies on the **assumption that regulatory information can be extracted from the expression pattern.**
- **TRN inference methods using pseudotime ordered cells**
 - Most TRN modeling methods developed for scRNA-seq data requires the cells to be ordered by pseudotime in the input data.
 - For example, LEAP (*Bioinformatics* 33:764, 2017) applies Pearson's correlation over temporal window of a fixed size with different time lags.
 - Others are SINCERITIES (*Bioinformatics* 34:258, 2018), SCODE (*Bioinformatics* 33:2314, 2017)
 - Then, network inferences will rely on the quality of pseudotime analysis of scRNA-seq data: "**robustness issue**".
- **TRN inference methods based on Boolean models**
 - Many TRN inference methods for scRNA-seq are based on those for bulk transcriptome data using models based on Boolean logic, correlations, regressions, information theory, and etc.
 - **Boolean models** focus on **logical combination of TFs** required to transit from one state to another in dynamic process, resulting in state-graph for key TFs involved in state changes.
 - However, it does not provide target information and computational demands increase rapidly with network size because of high-dimensional parameter spaces. (thus generally used for network with <100 genes): "**scalability issue**"
- ✓ We prefer TRN inference methods which are **robust** and **scalable** to any single-cell transcriptome data: **Partial correlation** (calculated by R package PPCOR), **PIDC**, **GENIE3** and **GRNBoost**

➤ Benchmarking TRN inferences from scRNA-seq data


Nature Methods (2020) e-published

- Since PIDC, GENIE3, GRNBoost2, and PPCOR do not require pseudotime-ordered cells, they are immune to any errors in pseudotime computation.
- MI (mutual information); RF (random forest); BT (boosting); Corr (correlation);
- In recent benchmarking based on scRNA-seq data from human and mouse, the TRN inference methods with no requirement for time-ordered cells were all top ranked in terms of accuracy.
- GENIE3 and PIDC also had better stability across multiple runs, whereas GRNBoost2 was less sensitive to the presence of dropouts.
- GENIE3 (RF) and GRNBoost2 (BT) infer **directional edges** (TF → target), whereas PPCOR (Corr) and PIDC (MI) infer **unidirectional edges**.
- Since GRNBoost2 and GENIE3 have multithreaded implementations now, they are as fast as PIDC.

		Properties			Accuracy			Stability			Scalability (genes)									
Category		Addl. inputs	Time ordered?	Directed?	Signed?	Synthetic	Curated	scRNA-seq	Datasets	Runs	Dropouts	Pseudotime	Time				Memory			
													100	500	1,000	2,000	100	500	1,000	2,000
PIDC	MI	-	X	X	X	High	High	High	Low	Low	Low	-	1 s	1 m	5 m	30 m	0.1 G	0.1 G	0.5 G	1 G
GENIE3	RF	-	X	✓	X	High	High	High	Low	Low	Low	-	5 m	1 h	3 h	12 h	1 G	2 G	2 G	2 G
GRNBOOST2	BT	-	X	✓	X	High	High	High	Low	Low	Low	-	1 m	10 m	30 m	1 h	0.1 G	0.1 G	0.5 G	1 G
PPCOR	Corr	-	X	X	✓	High	High	High	Low	Low	Low	-	1 s	1 s	1 s	1 s	1 M	0.1 G	0.1 G	0.1 G



Low/Poor High/Good



Low/Poor High/Good

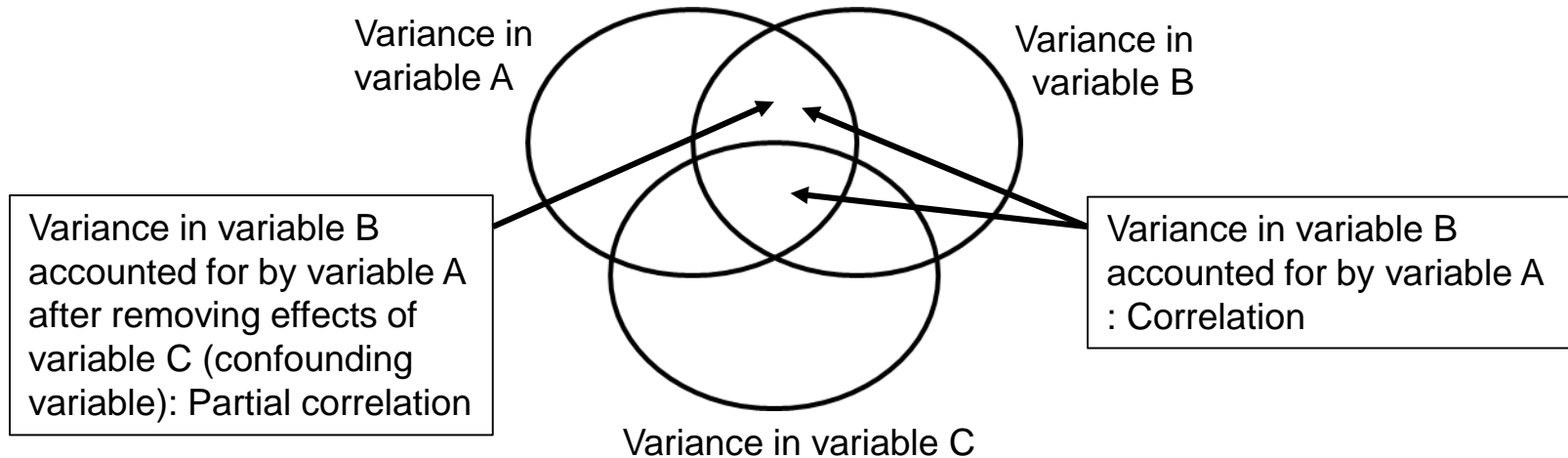
❖ Partial Correlation

- The principle underlying correlation networks is that if two genes have highly-correlated expression patterns (i.e. they are co-expressed), then they are assumed to participate together in a regulatory interaction.
- It is important to highlight that **co-expressed genes are indicative of an interaction but this is not a necessary and sufficient condition**. Partial correlation is a measure of the relationship between two variables while controlling for the effect of other variables.
- In complex system, processes often interdependent. For example, the abundance of clouds is often correlated with the amount of aerosol particles in the atmosphere.
- But both are also correlated with wind speed. Wind speed might be a “**mediating**” or “**confounding**” variable.
- Here we want to **test for an association two variables after controlling for the effect of one or more potentially confounding variables**.
- Correlation coefficient is adjusted for **correlations between each variable (A, B) and potential confounding variable C**.

$$r_{AB.C} = \frac{r_{AB} - r_{AC}r_{BC}}{\sqrt{1 - r_{AC}^2}\sqrt{1 - r_{BC}^2}}$$

- **Null hypothesis:** there is no association between the two variables after controlling for effects of confounding variable(s).
- Therefore the presence of an **edge between A and B** indicates that a **correlation exists between A and B regardless of which other nodes are being conditioned on**.

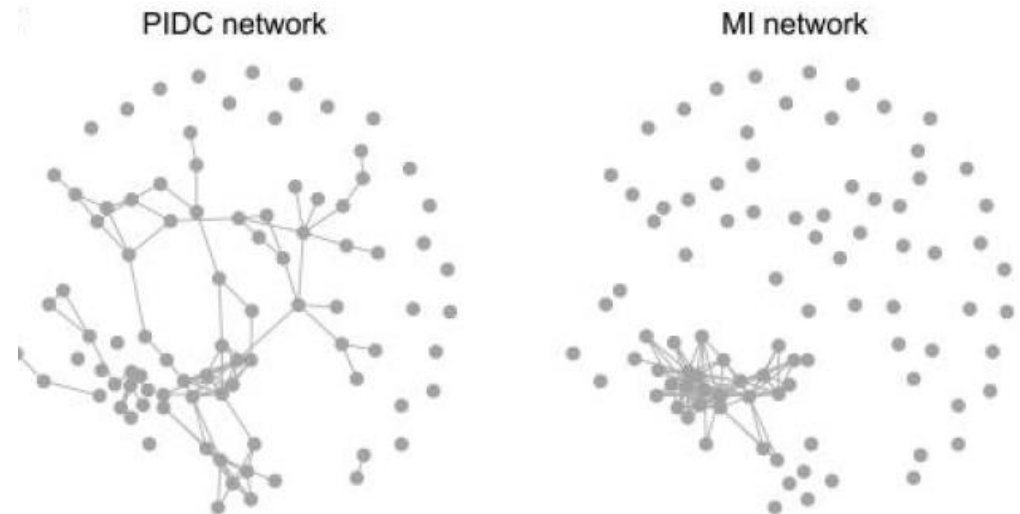
- **Venn Diagram explanation**



- Typically, gene expression profiles from single cell data follow multimodal distribution rather than a unimodal continuous shape. Therefore, Pearson correlation coefficients are less suited for single cell expression data because this metric measures a linear dependency between two variables.
- Given the non-linear nature of single cell gene expression data, nonparametric methods such as the **Spearman correlation** and **Kendall rank correlation** coefficients are more appropriate.
- It also computes a p -value for each correlation.
- Since these values are symmetric, this method yields an **undirected** regulatory network.
- We use the **sign of the correlation**, which is bounded between -1 and 1, to signify whether an interaction is **inhibitory (negative)** or **activating (positive)**.

❖ **PIDC (Partial Information Decomposition and Context)** *Cell Syst. 2017;5(3):251–67. e3*

- PIDC is a method **developed for scRNA-seq data** that uses multivariate information measures to identify potential regulatory relationships between genes.
- Partial information decomposition (PID) considers the information provided by **a set of source variables** (or genes), $S = \{X, Y\}$, about another **target variable, Z**, partitioned into redundant, synergistic, and unique information. **Redundant information** is the portion of information about Z that can be provided by either variable in S alone; the **unique information** from X (or Y) is the portion of information provided only by X (or only Y); and the **synergistic information** is the portion of information that is only provided by knowledge of both X and Y. Thus, the PID between the set S and the target variable Z is equal to the sum of the four partial information terms.
- The unique information terms can be calculated from the redundant information and the pairwise mutual information (MI), via the relationship, $I(X; Y) = \text{Unique}_Z(Y; X) + \text{Redundancy}(Y; X, Z)$
- PID computes the **ratio between the unique component and the MI**. The **sum of this ratio over all other genes z is the proportional unique contribution between x and y**. *The ratio of the unique information to the MI tends to be higher between pairs of connected genes.*
- PIDC outperforms pairwise MI-based algorithms.
- The resulting network is **undirected** since the proportional unique contribution is symmetric.



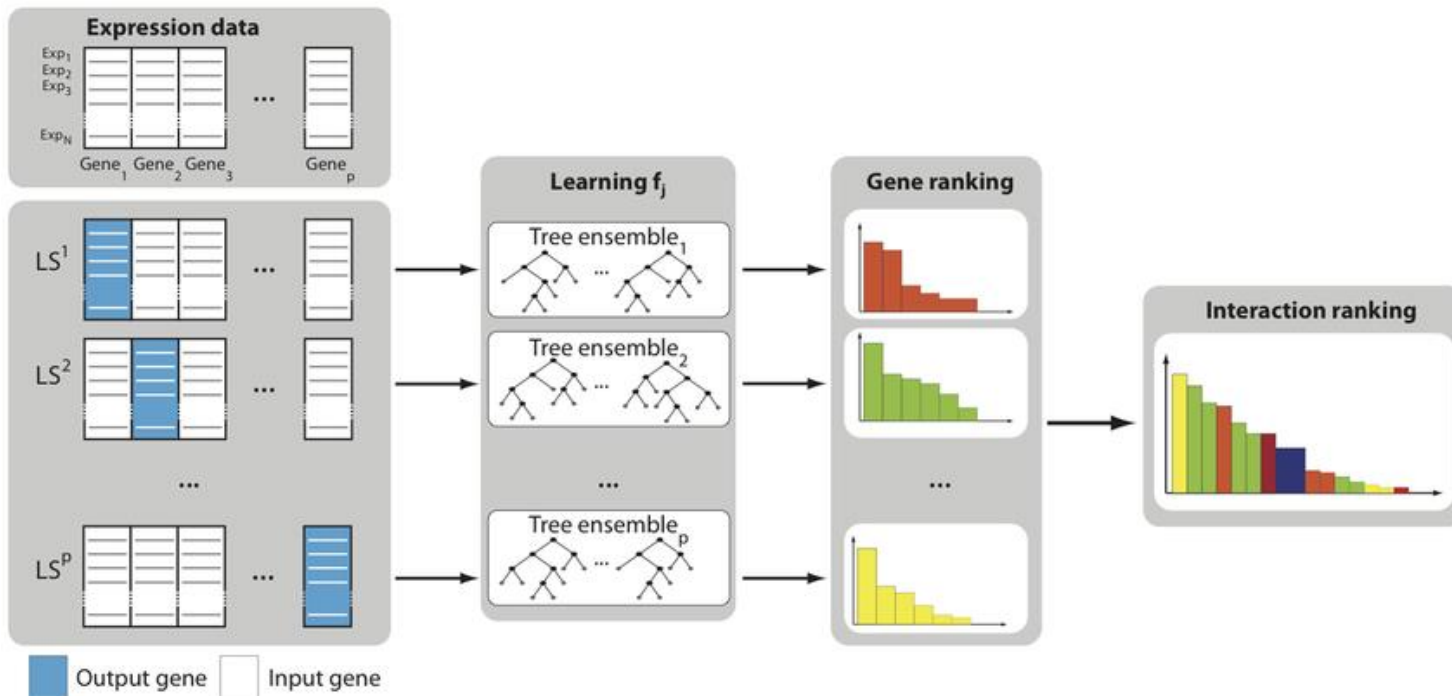
❖ **GENIE3** (GEne Network Inference with Ensemble of trees) *PLoS ONE 5(9): e12776 (2010)*

- GENIE3 is a TRN inference method based on variable selection with ensembles of regression trees. In each of the **regression problems**, the **expression pattern of one of the genes (target gene) is predicted from the expression patterns of all the other genes (input genes)**, using tree-based ensemble methods Random Forests (RF).
- The **importance of an input gene in the prediction of the target gene expression pattern** is taken as an **indication of a putative regulatory link**.
- Putative regulatory links are then aggregated over all genes to provide a ranking of interactions from which the whole network is reconstructed.
- Tree-based ensemble methods doesn't make any assumption about the nature of gene regulation, can potentially capture high-order conditional dependencies between expression patterns.
- Importantly, GENIE3 produces **directed** GRNs, and naturally allows for the presence of feedback loops in the network. It is also **fast** and **scalable**.
- A network inference algorithm was defined as a **procedure that exploits a set of gene expression vectors to assign weights to putative regulatory links from any gene i to any gene j** , with the aim of yielding large values for weights which correspond to actual regulatory interactions.
- Exploiting expression data, the identification of the regulatory genes for a given target gene is defined as **determining the subset of genes whose expression directly influences or is predictive of the expression of the target gene**.
- Therefore, here the **network inference** problem is equivalent to a **feature selection** problem.
- Importantly, **variable (i.e., gene) importance can be computed from a tree** that allows to rank the input features according to their relevance for predicting the output. GENIE3 uses a measure which at each test node computes the total reduction of the variance of the output variable due to the split.

- The **overall importance of one variable** is computed by summing the importance values of all tree nodes where this variable is used to split. Those attributes that are not selected at all obtain a zero value of their importance, and those that are selected close to the root node of the tree typically obtain high scores.
- Attribute importance measures can be easily extended to ensembles, simply by **averaging importance scores over all trees** in the ensemble.

▪ **GENIE3 procedure**

1. For each gene $j = 1, \dots, p$, a learning sample LS^j is generated with expression levels of j as output values and expression levels of all other genes as input values.
2. A function f_j is learned (with RF) from LS^j and a local ranking of all genes except j is computed.
3. The p local rankings are then aggregated to get a global ranking of all regulatory links.



❖ Ensemble Learning

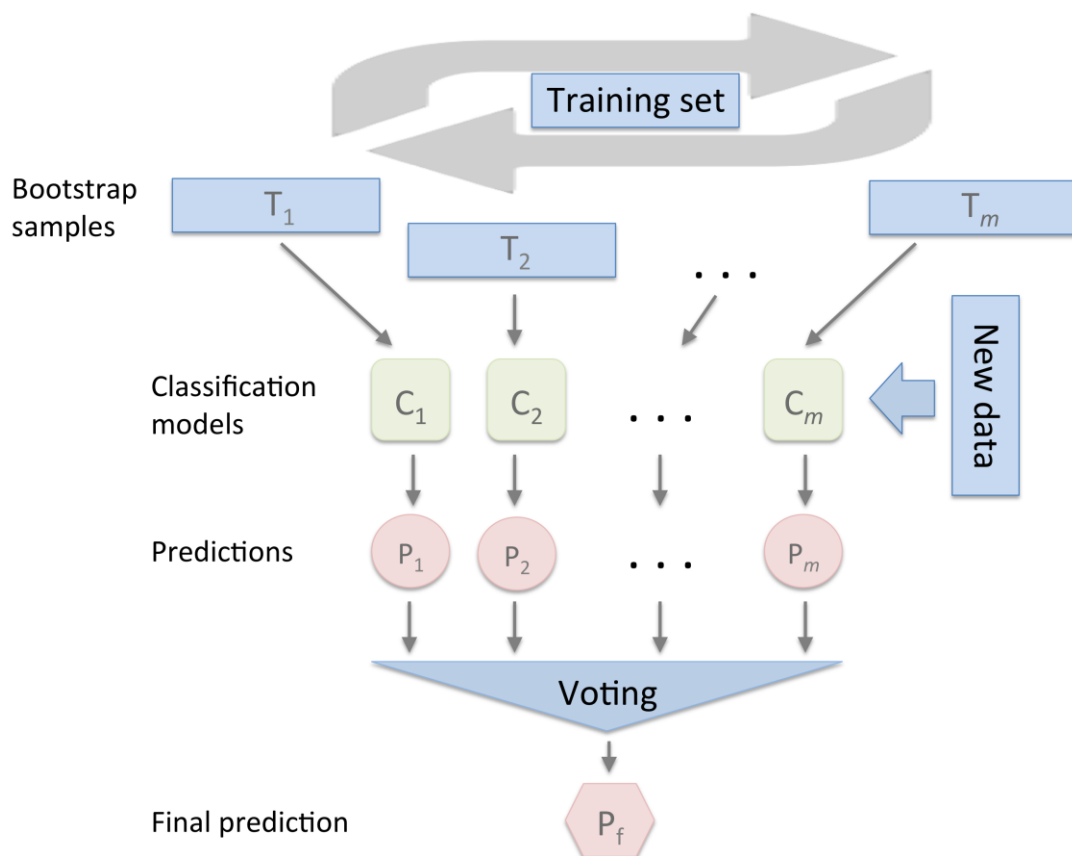
- Goal: to combine *weak* models (classifiers or regressions) into a final model that has a better generalization performance than the individual models.

▪ Ensemble method

- Ensemble methods use multiple learning algorithms (e.g., decision tree, logistic regression, etc) to obtain better predictive performance.
- **Two major types of ensemble learning** approaches: **Bagging** and **Boosting**

❖ Bagging (L. Breiman, 1994)

- Building multiple models (e.g., classifiers C_1, C_2, \dots, C_m) **on the same learner** using bootstrap samples of the original training sets (T_1, T_2, \dots, T_m) → Aggregating prediction results (e.g., majority voting in classification) for the final model

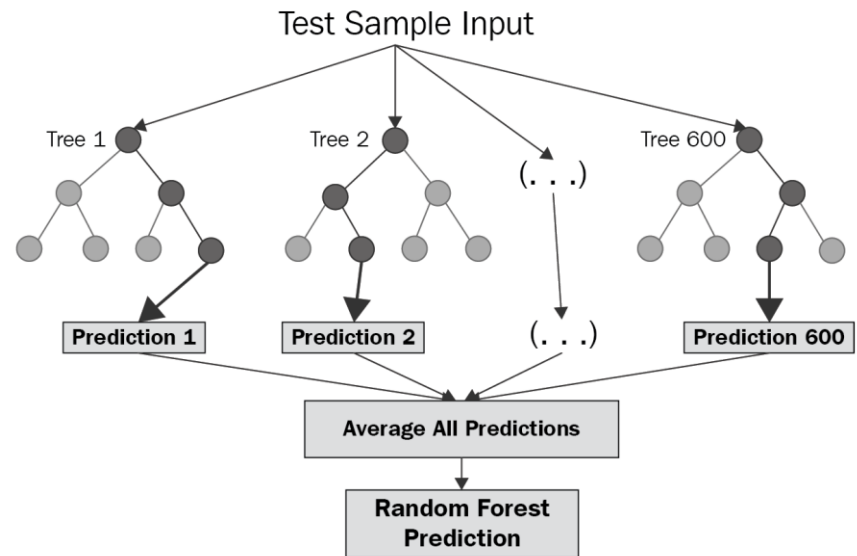


❖ Random Forests

- A popular ensemble learner with **bagging** approach
- Combining individual trees (weak learners) to build random forests (strong learner)

▪ Steps

- Draw a random **bootstrap** sample of size n (choose n random samples out of total n samples with replacement)
- Make **weak** decision trees from the bootstrap samples with two **hyperparameters**:
 - Maximum depth of the tree: d
 - The number of trees in the forest: k
- Split input data using the best feature to maximize the information gain.
- Repeat above steps for d features for k trees
- Aggregate the prediction of each tree by **majority voting (in classification)** or **averaging (variable weight scores in regression)**

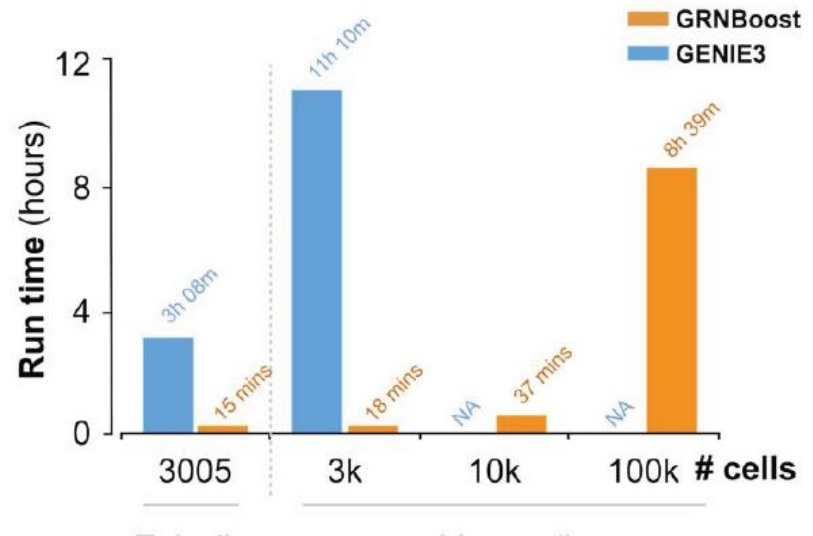
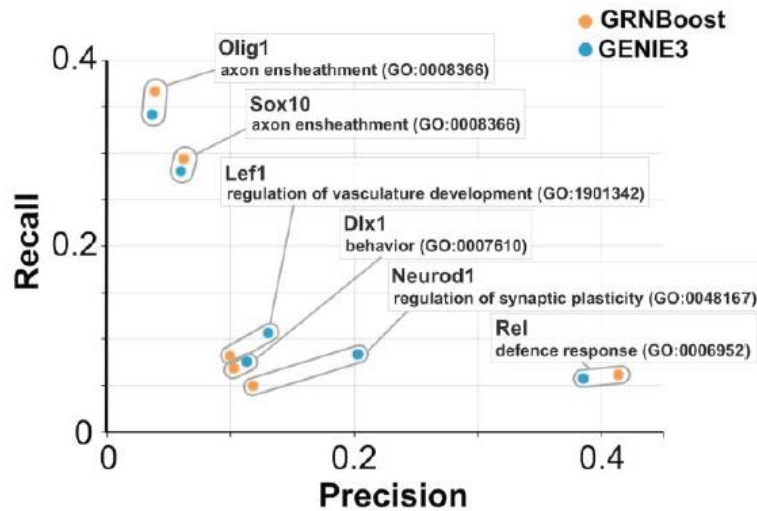


▪ Pros and Cons

- Don't need to prune the random forest in general, since the ensemble model is quite robust to the noise from individual decision trees
- The larger the number of trees k , the better the performance of the random forest
- Large computational cost for large k

❖ GRNBoost *Nature Methods 14:1083 (2017)*

- GRNBoost is based on the same concept as GENIE3 but using the **gradient-boosting machines (GBM)**. Boosting is an ensemble learning strategy.
- GRNBoost uses **stumps (regression trees of depth 1) as the base learner**.

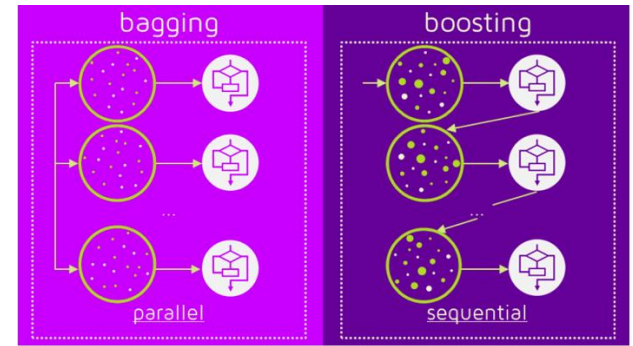


❖ GRNBoost2 *Bioinformatics 35:2159 (2019)*

- GRNBoost2 employs a regularized stochastic variation on GBMs. It equips GBM regressions with a **heuristic early-stopping regularization strategy** using out-of-bag improvement estimates.
- Each new decision tree is trained in function of a random subset of observations (90%, hence stochastic), whereas the remaining (10%, out-of-bag) observations are used to calculate an estimate of the loss function improvement entailed by adding that tree to the ensemble.
- When the average of the last n improvement values drops below 0, the early-stopping criterion is met and no more trees are added to the ensemble.
- Regressions that do not display net improvement early on are aborted and thus **prevented from causing useless computational workload**.

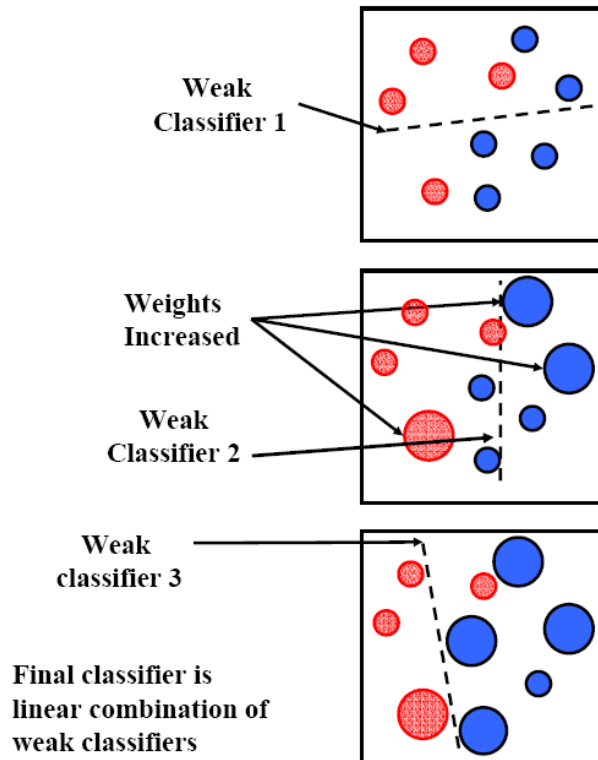
❖ Boosting

- Boosting is different from bagging. In boosting, we consider the mistakes of previous predictors and **train the new predictors on those mistakes** and then repeat the process till we get a better fit.



❖ AdaBoost (Adaptive Boosting) [Y. Freund & R. Shapire 1995]

- **Iteratively reweight your dataset**, placing **higher weights on the examples you are getting wrong**. At each iteration, refit and add the result to ensemble.

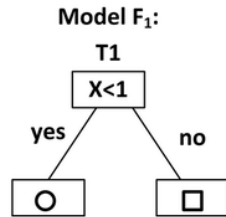
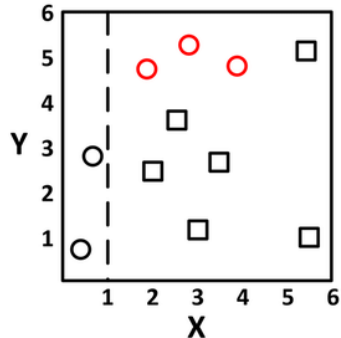


Algorithm

1. Start by applying some method to the learning data, where each observation is assigned an equal weight.
2. Compute the predicted classifications, and assign greater weight to those observations that were difficult to classify (where the misclassification rate was high), and lower weights to those that were easy to classify (where the misclassification rate was low).
3. Then apply the classifier again to the weighted data (or with different misclassification costs), and continue with the next iteration (application of the analysis method for classification to the re-weighted data).

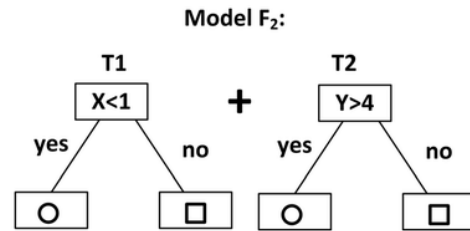
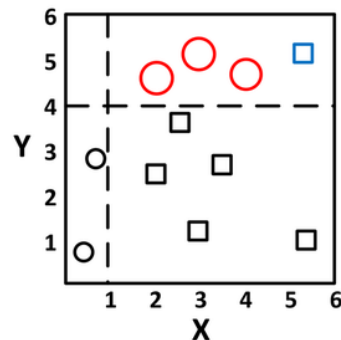
- A simple example of visualizing boosting with trees.

Iteration 1



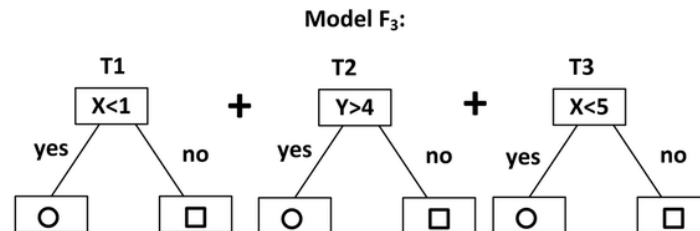
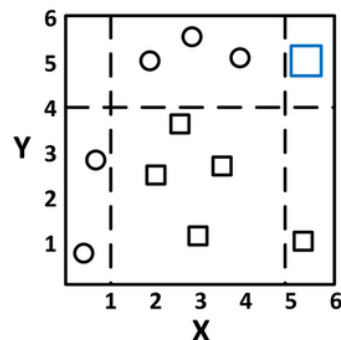
- Boosting is a framework that iteratively improves *any* weak learning model. **In practice** however, **boosted algorithms almost always use decision trees as the base-learner.**
- Whereas random forests build an ensemble of deep independent trees, **Boosting machines** build an **ensemble of shallow and weak successive trees** with **each tree learning and improving on the previous.**

Iteration 2



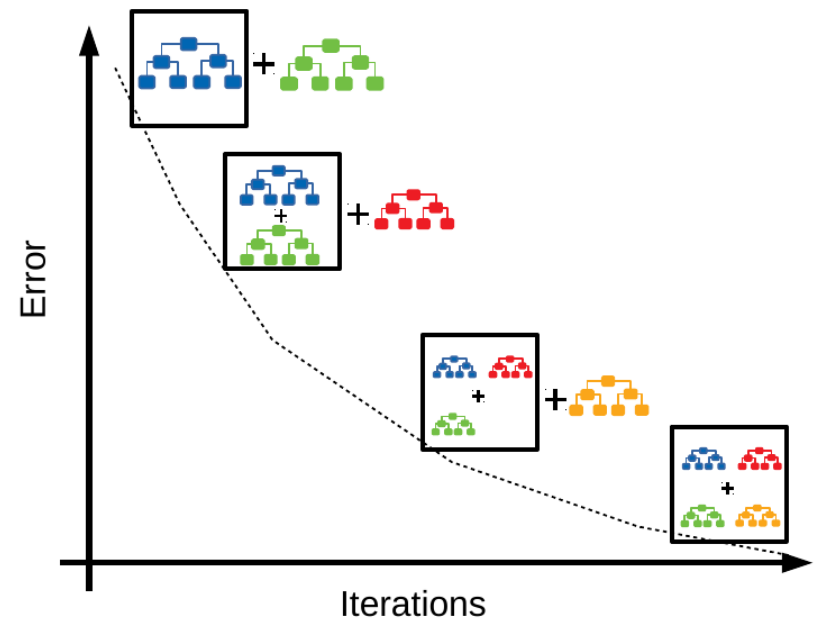
- When combined, these many weak successive trees produce a powerful “committee” that are often hard to beat with other algorithms.
- Fits consecutive trees where each solves for the net loss of the prior trees. Results of new trees are applied partially to the entire solution.
- Final model is the **linear combination of weak models** with **weighted votes for each of the base models.**

Iteration 3



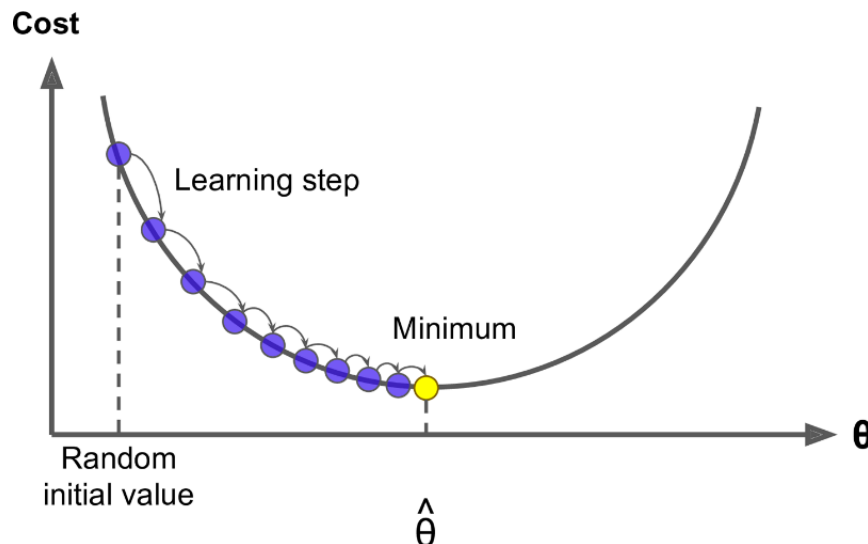
❖ Gradient Boosting Machines (GBM) [J. Friedman 1999]

- The basic principle is same for both AdaBoost and Gradient Boost. **The differences is how the new predictor learns from the old one.**
- **Adaboost learns the weights of weak predictors** during the learning process. It **keeps adding +ve and -ve weights to predictors** about certain data points till we have predictors that can combine to give a better result.
- **GBM generates predictors** during the learning process. Instead of adding any weights to predictors, wrong predicted data points are considered as a new training set and the new predictor tries to fit these data points making a new model. It **keeps fitting wrongly predicted data points with the new predictor till lesser predictions are wrong** and then use all predictors together to predict output by voting or averaging.
- GBM uses **gradient descent algorithm** which can **optimize any differential loss function**. Each tree in GBM is a successive gradient descent step.
- **GBM = Gradient Descent + Boosting**
- In GBM instead of reweighting used in AdaBoost, each tree is fit to the negative gradients of the previous tree.
- Basic elements of GBM: loss function, weak learner, additive model
- Improvement of basic GBM: tree constraints, shrinkage, random sampling, penalized learning (=regularization)



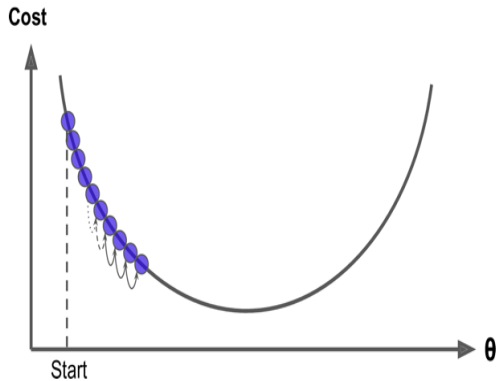
❖ Gradient Descent

- Many algorithms, including decision trees, focus on minimizing the residuals and, therefore, emphasize the mean squared error (MSE) loss function. **Gradient boosting machines** can be generalized to loss functions other than MSE.
- **Gradient descent** is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems. The general idea of gradient descent is to tweak parameters iteratively in order to minimize a loss function.
- Suppose you are a downhill skier racing your friend. A good strategy to beat your friend to the bottom is to take the path with the steepest slope. This is exactly what gradient descent does - it measures the local gradient of the loss function for a given set of parameters and takes steps in the direction of the descending gradient.
- Once the gradient is zero, we have reached the minimum.
- Gradient descent can be performed on **any loss function that is differentiable**. Consequently, this allows GBMs to optimize different loss functions as desired.

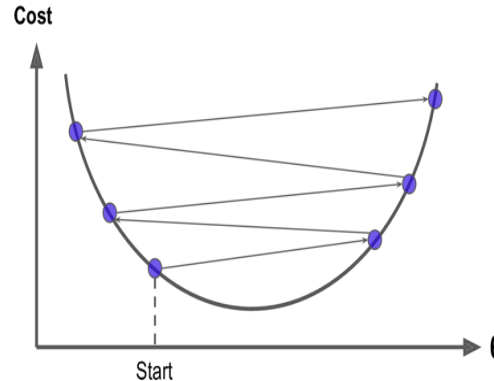


- An important parameter in gradient descent is the size of the steps which is determined by the **learning rate**.
- If the learning rate is too small, then the algorithm will take many iterations to find the minimum. On the other hand, if the learning rate is too high, you might jump cross the minimum and end up further away than when you started.
- Not all cost (loss) functions are convex (bowl shaped). There may be local minimas, plateaus, and other irregular terrain of the loss function that makes finding the global minimum difficult. **Stochastic gradient descent** can help us address this problem by sampling a fraction of the training observations (typically without replacement) and growing the next tree using that subsample.
- This makes the algorithm faster but the stochastic nature of random sampling also adds some random nature in descending the loss function gradient. Although this randomness does not allow the algorithm to find the absolute global minimum, it can actually help the algorithm jump out of local minima and off plateaus and get near the global minimum.

[Learning rate comparisons]

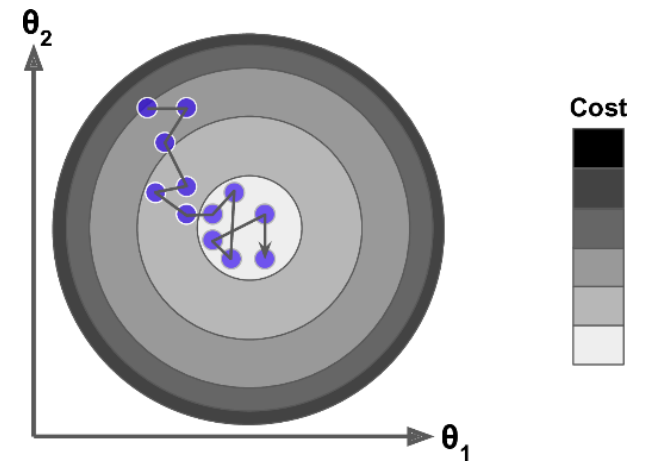


a) too small



a) too big

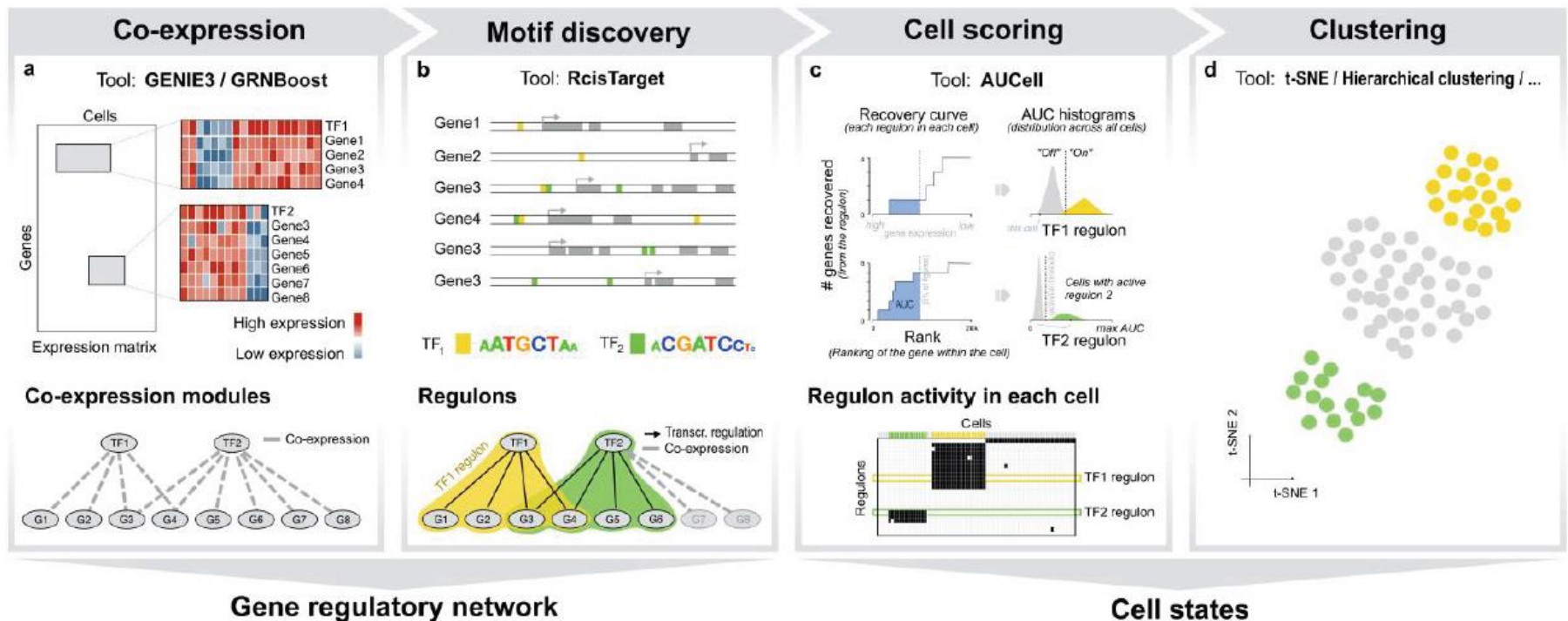
[Stochastic gradient descent]



❖ SCENIC (single-cell regulatory network inference and clustering)

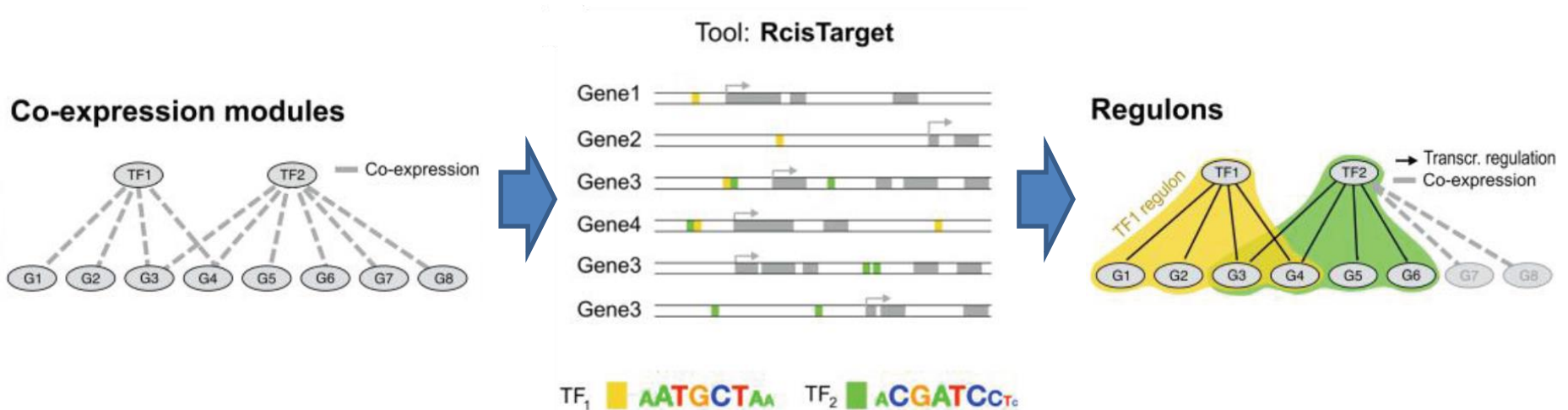
Nature Methods 14:1083 (2017)

- To overcome the high noise and sparsity of scRNA-seq data, SCENIC links cis-regulatory sequences to single-cell gene expression. SCENIC workflow consists of 3 steps:
 - Sets of genes that are coexpressed with TFs are identified using **GENIE3** or **GRNBoost**.
 - To identify putative direct-binding targets, each coexpression module is subjected to cis-regulatory motif analysis using **RcisTarget**. Only modules with significant motif enrichment of the correct upstream regulator are retained. → **Regulon**
 - AUCell** scores the **activity of each regulon in each cell**, thereby yielding a **binarized activity matrix with reduced dimensionality**, which can be useful for downstream analyses. For example, clustering based on this matrix identifies cell types and states based on the shared activity of a regulatory subnetwork. Since the regulon is scored as a whole, instead of using the expression of individual genes, this approach is robust against dropouts.



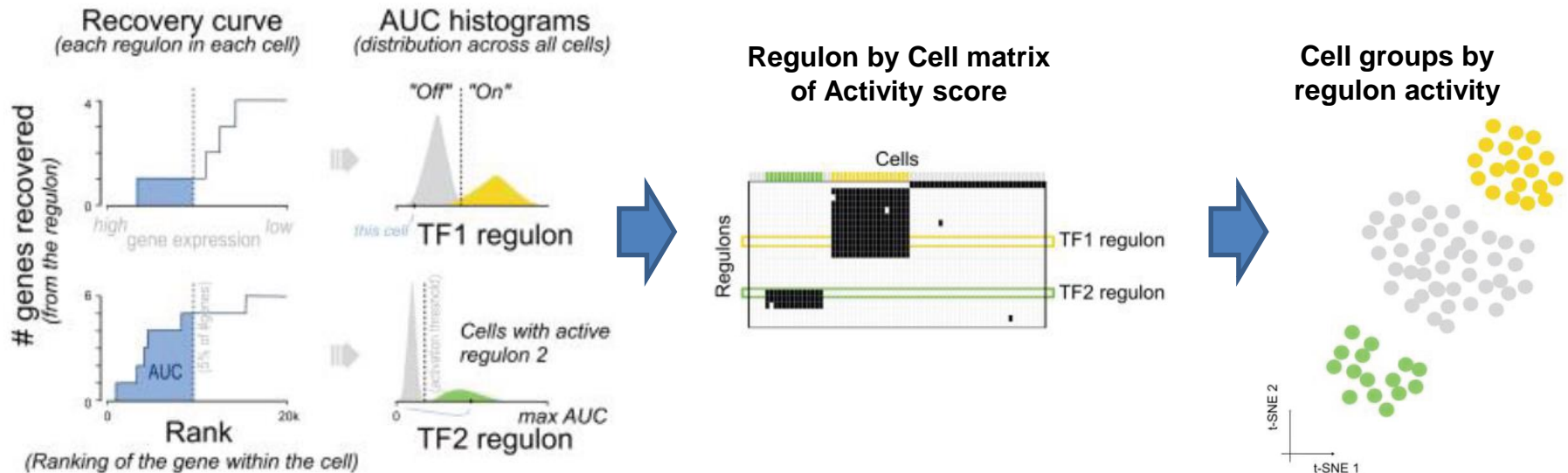
❖ RcisTarget

- RcisTarget is based on two steps.
- 1. **Identification of enriched TF-binding motif across the genes of Regulon.** For each TF, RcisTarget selects DNA motifs that are significantly over-represented in the surroundings of the transcription start site (TSS) of the target genes. This is achieved by applying a recovery-based method on a database that contains genome-wide cross-species rankings for each motif. The motifs that are annotated to the corresponding TF and obtain a normalized enrichment score (NES) > 3.0 are retained.
- 2. **Prediction of target genes by enriched motif.** (i.e., genes in the target gene set that have the enriched motif).
- **The final GRN = TF-target by expression patterns \cap TF-target by enriched motif**
- There could be negative-correlated TF modules. However, these modules are generally less numerous and showed very low motif enrichment. For this reason, we take only positive-correlated targets.



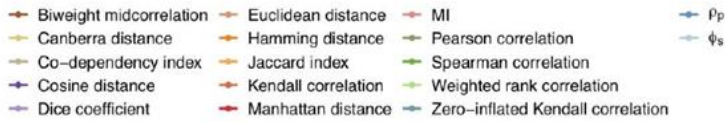
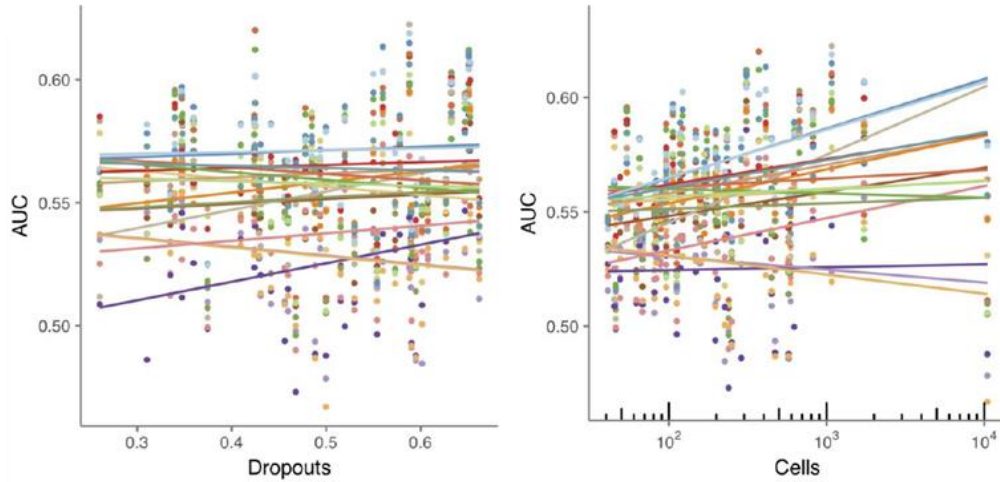
❖ AUCell

- AUCell can identify cells with **active regulons** in single-cell RNA-seq data.
- AUCell scoring method is based on a recovery analysis where the x-axis is the ranking of all genes based on expression level (genes with the same expression value, e.g., '0', are randomly sorted); and the y-axis is the number of genes recovered from the input set (regulon genes).
- AUCell then uses an area under the recovery curve (AUC) to calculate whether a critical subset of the input gene set is enriched at the top of the ranking for each cell.
- The output of this step is a **matrix with the AUC score for each regulon** (of each TF) in **each cell**. We use either the AUC scores (across regulons) directly as continuous values to cluster single cells, or we generate a binary matrix using a cutoff of the AUC score for each regulon.
- Clustering cells for regulon activity profiles can group cell types, suggesting that network activity score can **complement to expression data** in single-cell analysis.

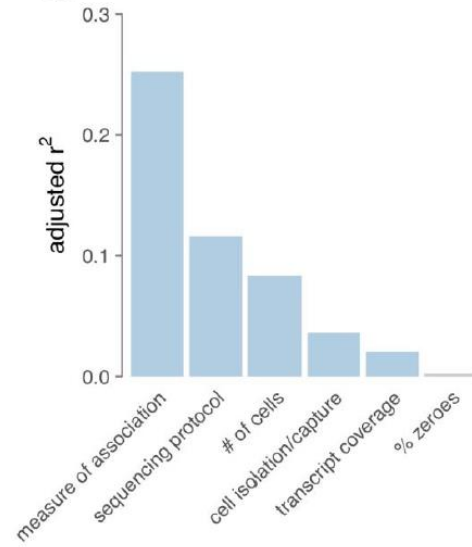
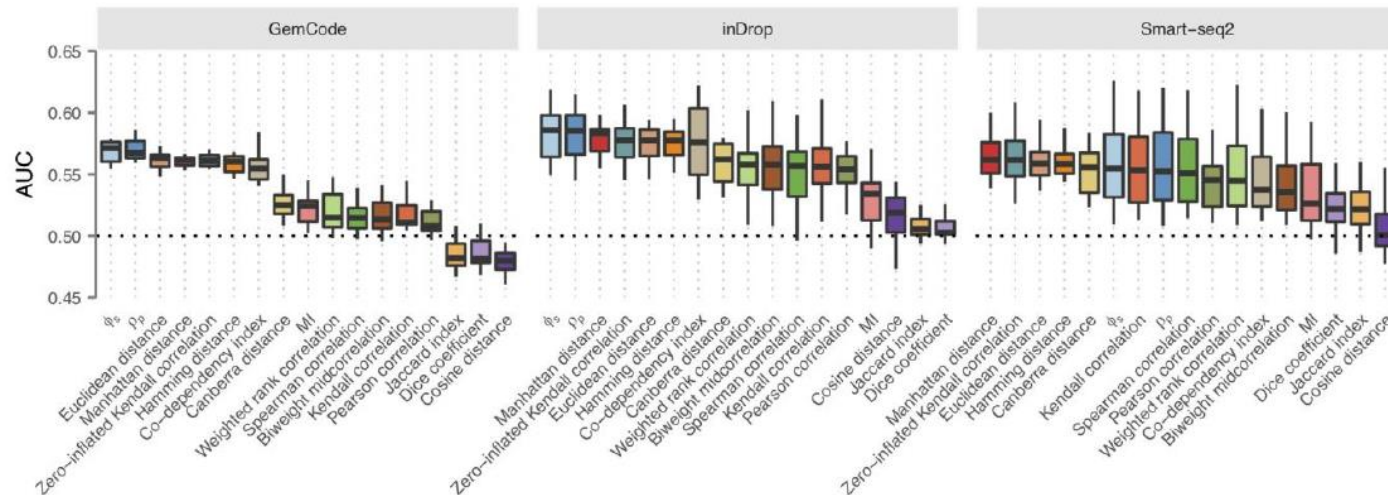
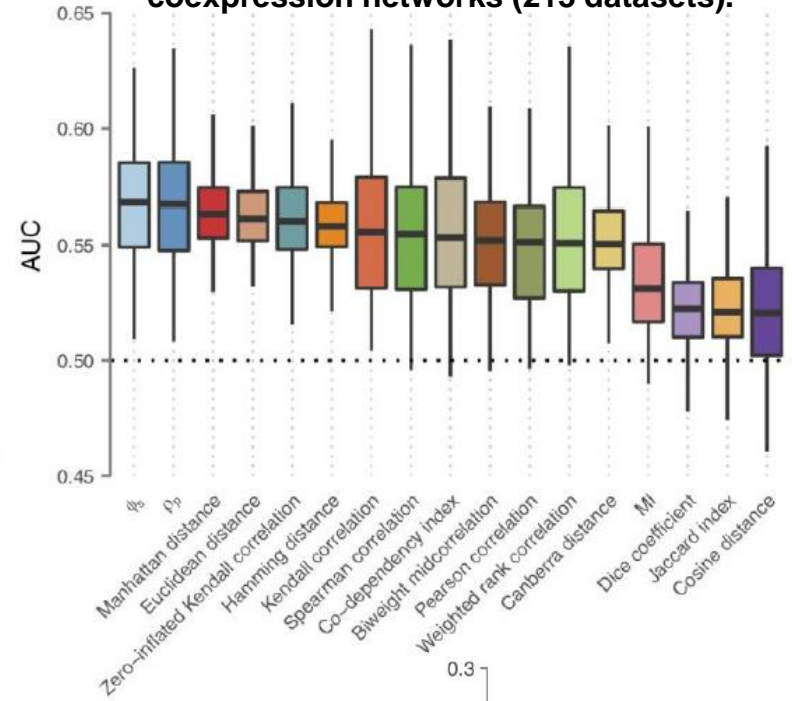


➤ **Benchmarking of Association metrics for FGN with scRNA-seq data** *Nat. Methods 16:381 (2019)*

- **All performed very poor**, although measures of *proportionality* between two variables worked best.



Functional coherence of scRNA-seq coexpression networks (213 datasets).



❖ bigScale method for scRNA-seq data transformation

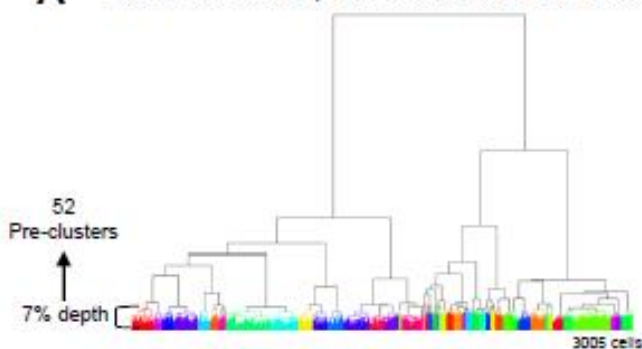
Genome Research 28:878 (2018)

Genome Biology 20:110 (2019)

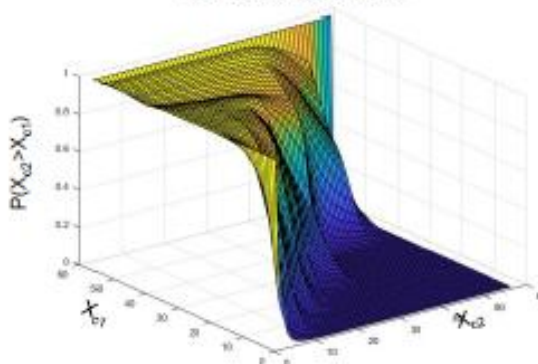
▪ Step 1: Preclustering and numerical modeling

- To handle the noise and sparsity of scRNA-seq data, bigScale method **uses large sample sizes to estimate an accurate numerical probabilistic model of noise.**
- **Preclustering cells into groups sharing highly similar expression profiles**, which are next **treated as biological replicates to allow evaluation of the noise.**
- Preclustering procedure: (1) read/count data normalization (2) $\log_{10}(X + 1)$ transformation (3) gene normalization to avoid severe effect of highly expressed genes on clustering (4) clustering cells with Pearson correlation and hierarchical clustering.
- Different cutting depths give different numerical models. It finds the **deepest possible cut** (10-20% of total tree height in general) in the tree to ensure that **only highly similar cells are grouped together.** → final clusters
- At this stage, **the cells within each group are treated as replicates**, assuming their **changes of expression to be solely due to noise and not to biological differences.**
- **All within-group pairwise comparisons between cells** are enumerated in order to **determine how rare/common (i.e., assigning a *P*-value) each combination of expression values is.** Specifically, if a cluster contains n cells, it produces $C(n,2) = n*(n-1)/2$ combinations of cells. Each of these combinations contain k couples of expression values (X_{cell_1}, X_{cell_2}) , where k is equal to the total number of genes and X_{cell_1}, X_{cell_2} is the expression of a gene in the two compared cells.
- **The numerical model is robust to the different tree cut.** Difference in numerical probabilistic models between default 7% cut and forced 4% cut or 20% cut is marginal.

A PRE-CLUSTERING (PEARSON, UNSUPERVISED, 7%)

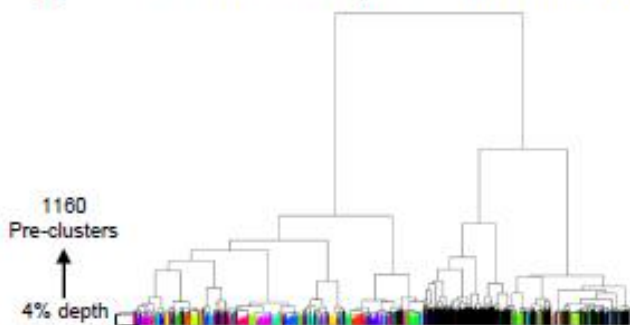


NUMERICAL MODEL

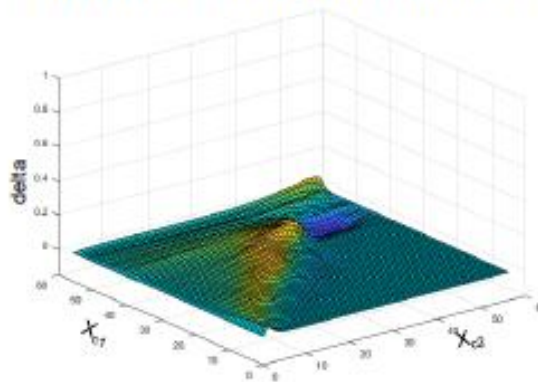


(A) (Left) Default, unsupervised heuristic sets a cut of 7% of the total dendrogram depth, which results in 52 pre-clusters. (Right) The numerical model calculated using the 52 pre-clusters. X_{c1} and X_{c2} represent the expression (in a binned UMIs grid) of a given gene X in two cells $c1$ and $c2$ belonging to the same pre-cluster. The cumulative distribution plot estimates the frequency, hence likelihood, of an expression change.

B PRE-CLUSTERING (PEARSON, FORCED TO 4%)

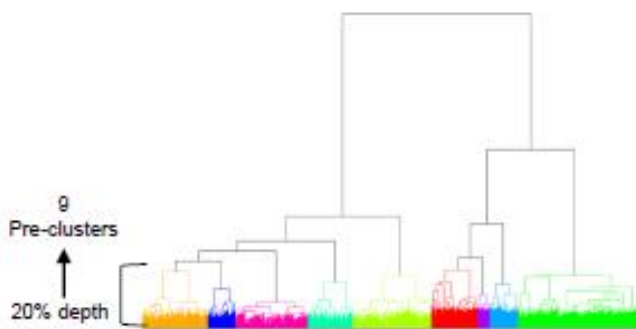


NUMERICAL MODEL: DELTA COMPARED TO 7%

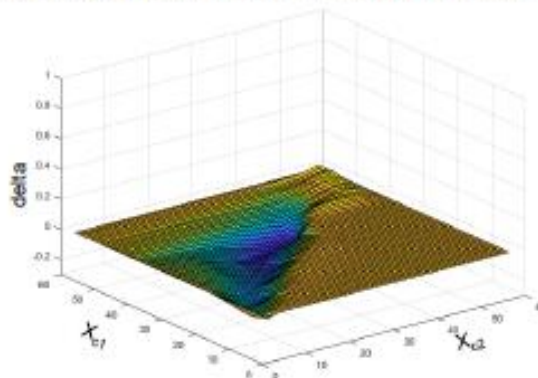


(B-C) The difference between the numerical model of 4% cut and 7% cut (B. Right) or 20% (C. Right) is marginal.

C PRE-CLUSTERING (PEARSON, FORCED TO 20%)



NUMERICAL MODEL: DELTA COMPARED TO 7%

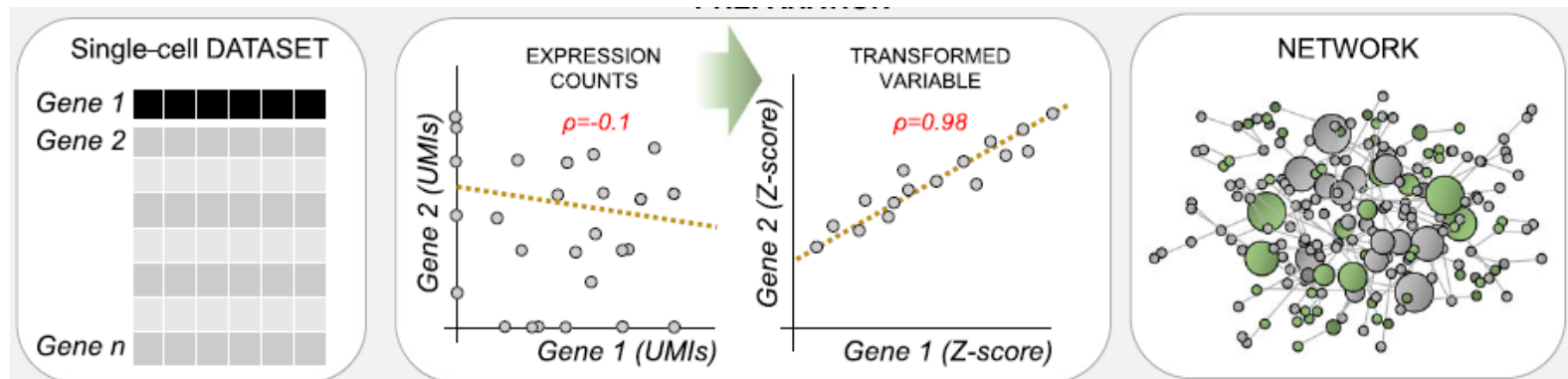


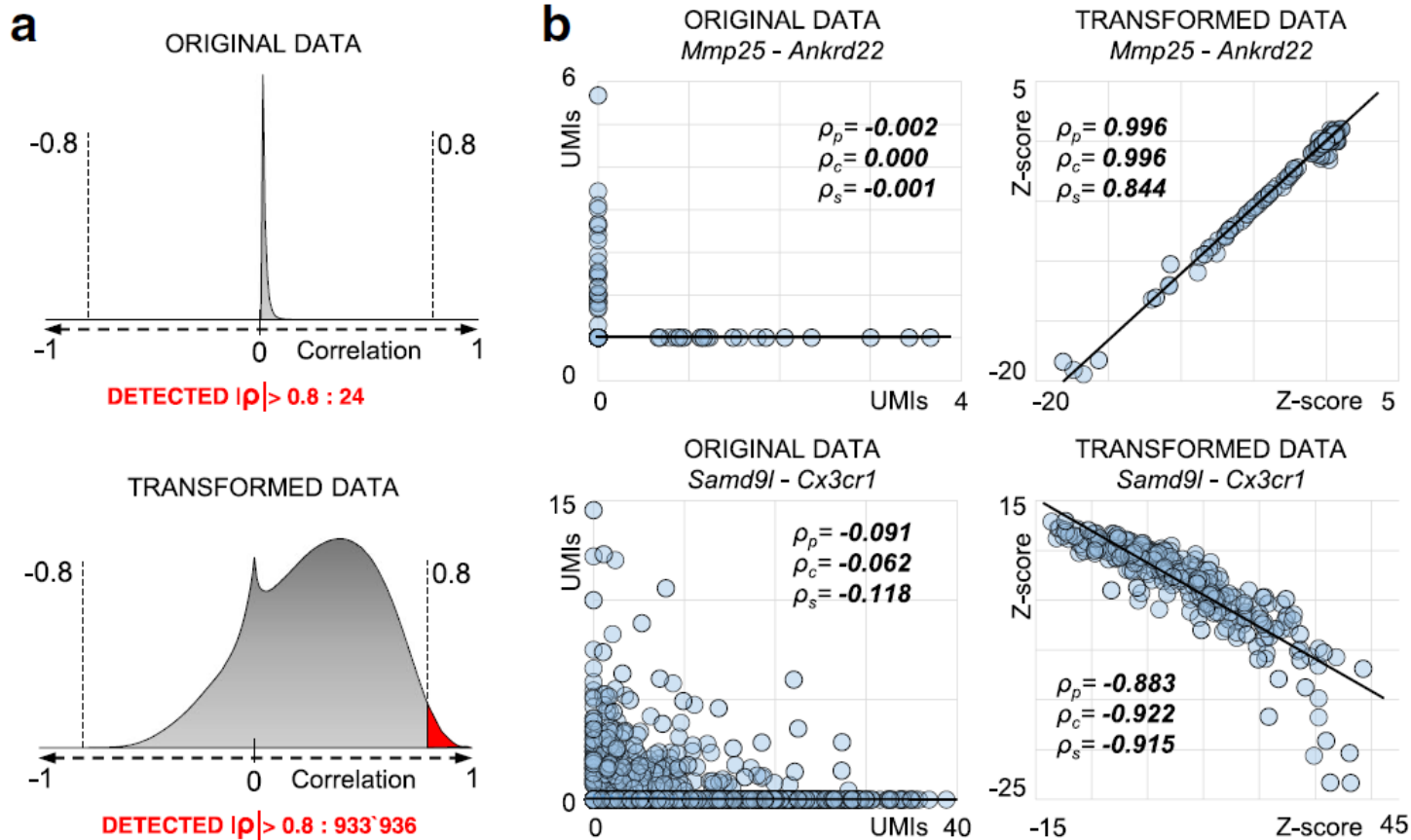
▪ Step 2: Differential expression analysis

- bigScale assigns a Z-score to each gene, representing the likelihood of an expression change between two groups of cells. The numerical model is used to identify differential expression (DE) between groups.
- After clustering the cells to the highest feasible granularity, we used *bigScale* to run an iterative DE analysis between all pairs of clusters. For x clusters, this results in a total of $x \times (x - 1)/2$ unique comparisons, each yielding a Z-score for each gene that indicates the likelihood of an expression change between two clusters.
- Thus, if we started with $(10 \text{ clusters}) \times (k \text{ genes})$, we end up with $[45 \times k]$ matrix of Z-scores.

▪ Step 2: Network inference using Z-score

- We then compute correlations between genes using Z-scores instead of expression values.
- Therefore, **linear correlations in the Z-score space can reflect non-linear correlations in the original expression space**. Hence, Pearson (or Spearman) correlation coefficient is recommended to measure association between genes.



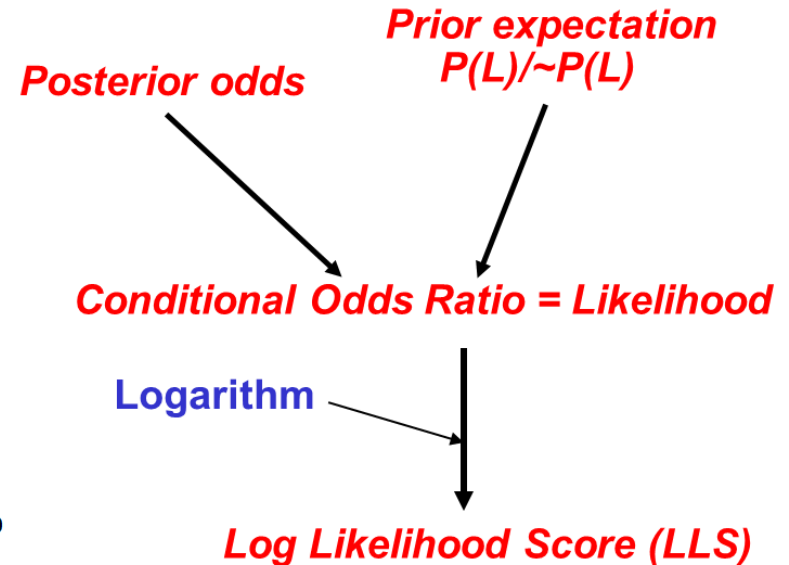
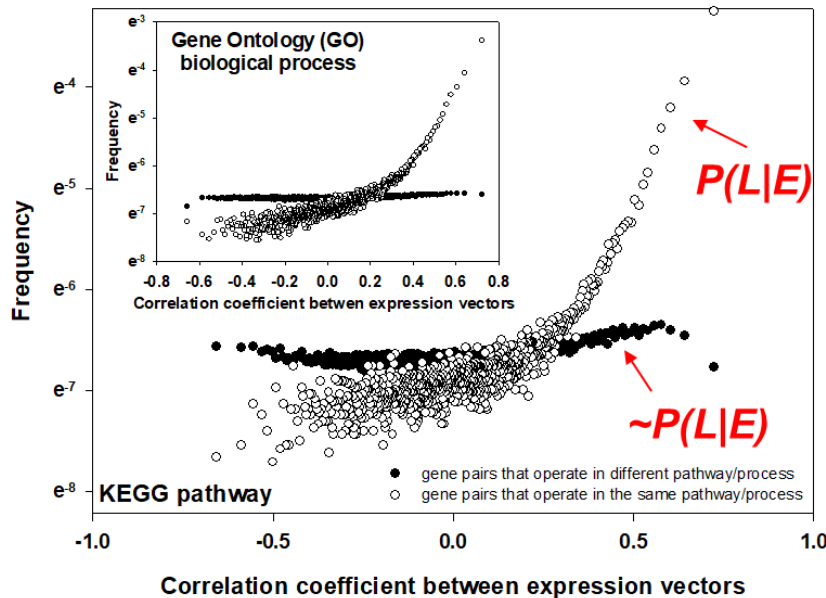


Transformed single-cell data allow detection of hidden correlations.

- Distribution of *Pearson* correlations ρ_p in normalized expression data (7697 microglia cells) or in the *Z*-score space. We detect only 24 correlations $|\rho_p| > 0.8$ in the first scenario, but almost one million $|\rho_p| > 0.8$ in the *Z*-score space.
- Examples of correlations using either expression values or *Z*-score-transformed data (ρ_p *Pearson*, ρ_c *Cosine*, ρ_s *Spearman*). Due to drop-out events and other artifacts, the positive correlation between *Mmp25* and *Ankrd22* is only exposed using *Z*-scores. Similarly for the negative correlation between *Samd9l* and *Cx3cr1*.

❖ Benchmarking co-functional association using Bayesian statistics

- We use Bayesian statistics to measure **Likelihood** of being associated.



Posterior Odds (belief after the observed data)

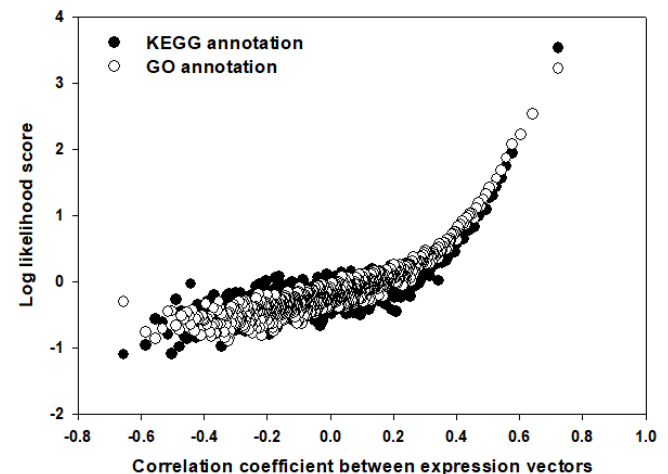
$$\text{Log Likelihood Scores (LLS)} = \ln \left(\frac{P(L | E) / \sim P(L | E)}{P(L) / \sim P(L)} \right)$$

Prior Expectation (Belief before the observed data)

L: linkage between two genes

E: an evidence by given data

If $LLS = 0$, the likelihood of two gene's association is no better than random chance

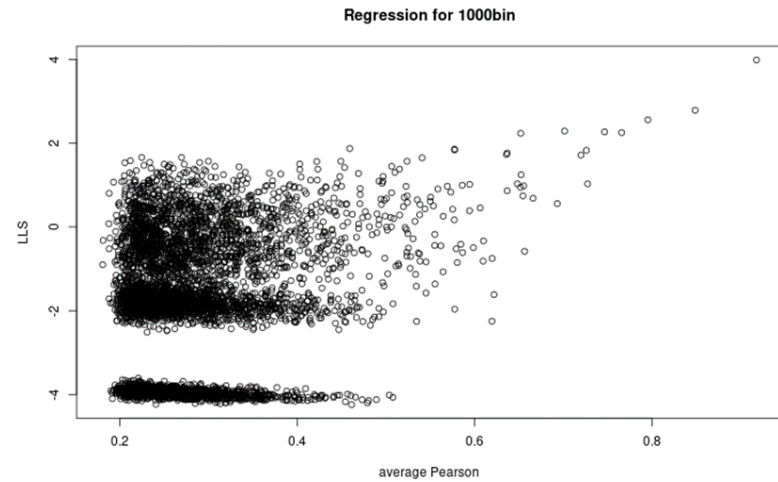
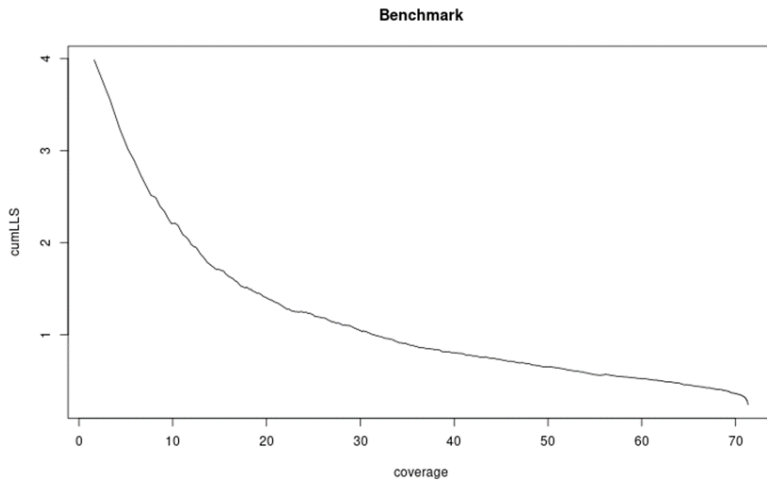


❖ How to make a benchmarking data set from pathway database

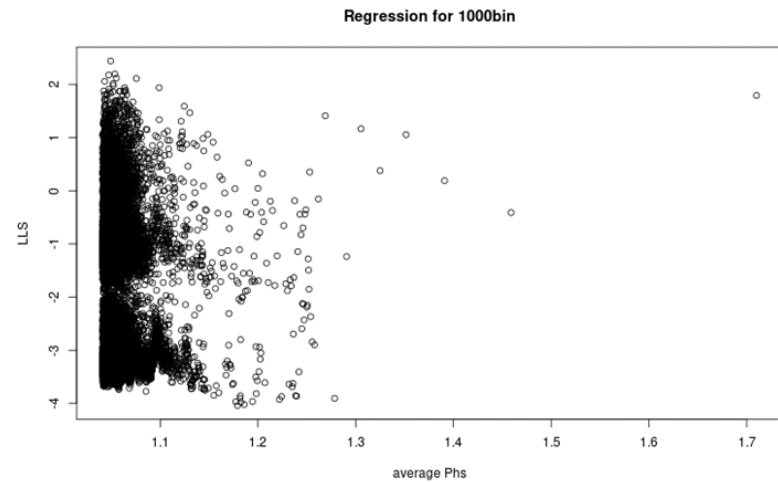
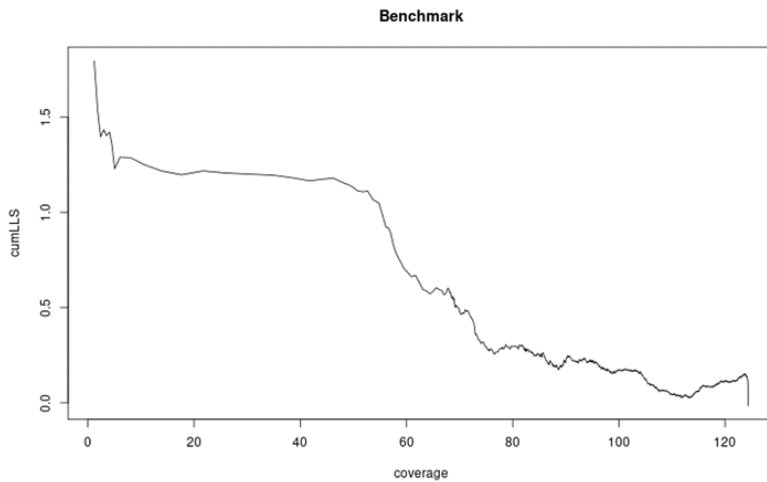
- Collect pairs of genes that belong to the same pathway.
- Use pathway annotation DBs (Gene Ontology biological process, KEGG pathway, MetaCyc, ...).
- What makes a good pathway annotation DB for network modeling?
 - Frequent update
 - comprehensive
 - Evidence codes
- From pathway annotation to pathway links for network training: for example, a pathway has 4 member genes (gene A, B, C, D). Then we can make the following training samples by the pathway
 - A – B
 - A – C
 - A – D
 - B – C
 - B – D
 - C – D

➤ **Big scale → FGN (PCC), compare with proportionality score GSE99254 (~12k T cells fax sorted from NSCLC patients)**

- **bigScale transformation → PCC**

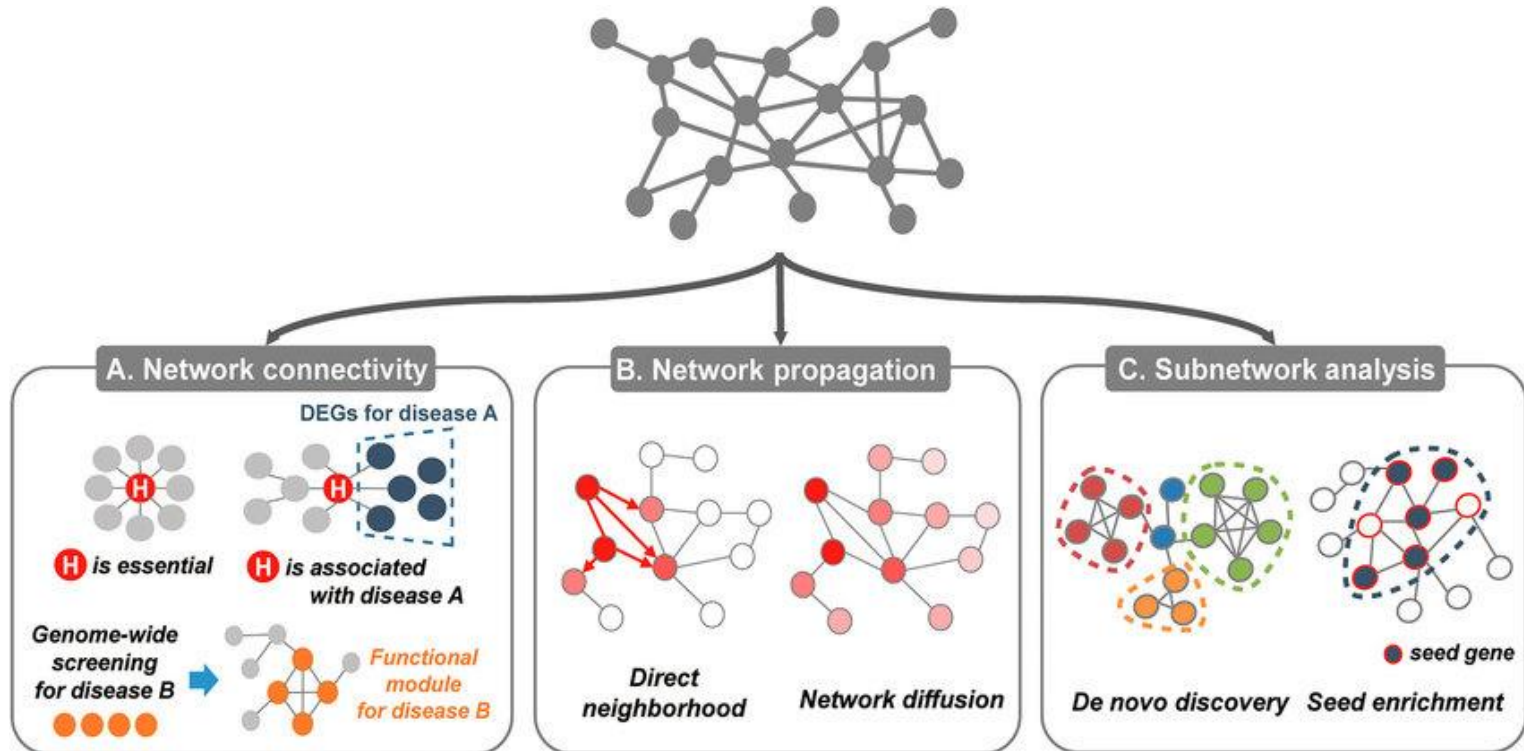


- **Standard preprocessed data → Proportionality**



❖ Hypothesis generation using gene/protein networks

1. **Network connectivity:** Hub genes tend to be functionally more important (e.g., essential genes)
2. **Network propagation:** Genes for the same phenotype (e.g., disease) tend to be connected in the interactome. Thus, novel disease genes can be inferred by propagated information from neighboring disease genes.
3. **Subnetwork analysis:** Functional or disease modules can be represented as subnetworks of tightly connected genes



❖ Approaches to network-assisted scRNA-seq data analysis

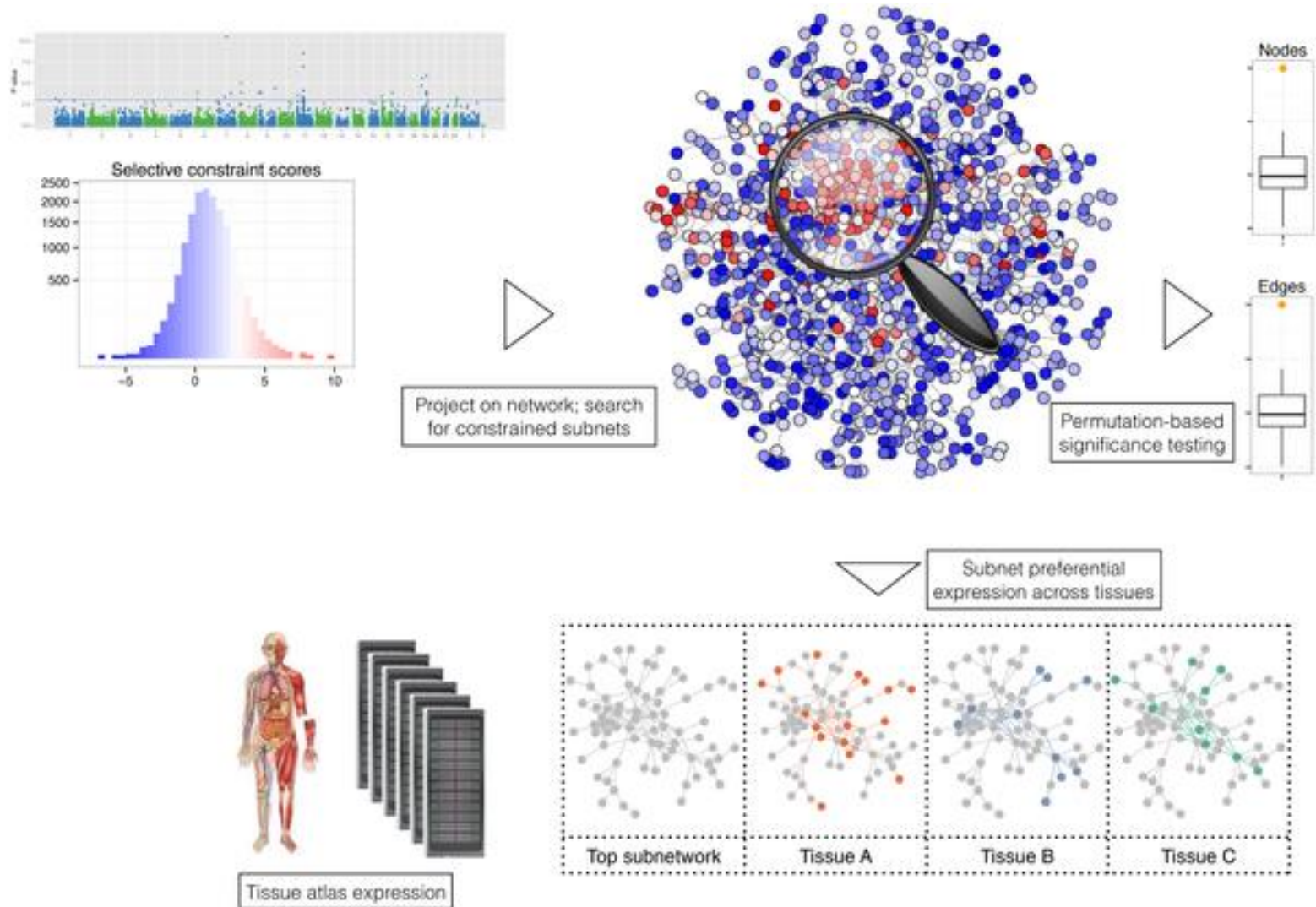
1. **Edge connectivity:** regulators tend to connect to dysregulated genes;
2. **Subnetwork detection:** Highly connected sub-network structures may reflect functional modules in which functionally related proteins are highly interconnected.
3. **Network dynamics:** Networks for different spatiotemporal context should be different. Genes for a function or disease would show different network connectivity across context-specific networks. For example, we may be able to infer association between a gene set and context (e.g., disease and specific tissue) by their interactions on a context-specific network.

Network dynamics

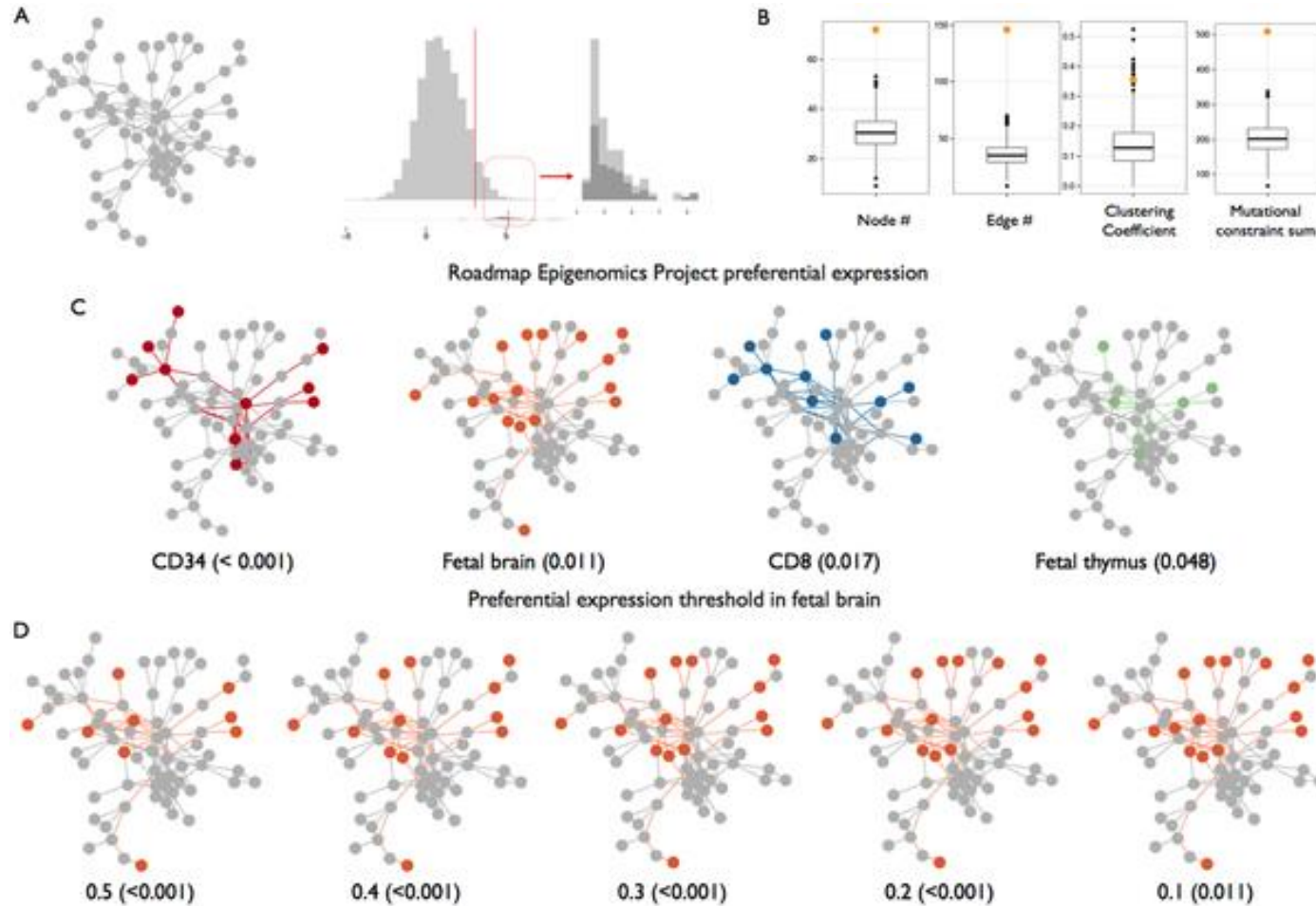
Subnetwork spatiotemporal specificity analysis

❖ Protein Interaction Network Tissue Search (PINTS) (*PLoS Genetics* 12:e1006121, 2016)

Measure significance of modularity of disease genes -> Detect subnetwork enriched for disease genes -> Measure significance of preferential expression of subnetwork for each spatiotemporal context



- A. 72/107 genes are densely connected in a subnetwork.
- B. 107 disease genes for the study are functionally aggregated.
- C. Disease subnetwork preferentially express in several tissues/cell types
- D. Association between the disease subnetwork and fetal brain is robust to the score threshold.



The results indicate that carefully orchestrated developmental processes are important in early brain development and perturbations caused by mutation have adverse effect.