National Institute of Allergy and Infectious Diseases

Finding Master Genes Through Gene Regulatory Network Analysis

March 30, 2020

Yunhua Zhu

NIAID

Gene Regulatory Networks

Yunhua Zhu,
PhD in stem cell biology

Computational Genomics Specialist – Transcriptomics

Recent projects

- Single cell analysis on stem cell transformation in human liver
- Single cell analysis of immune responses in lung tumor module
- Bulk RNA-seq on irradiation response in primates
- Bulk RNA-seq of blood samples of HIV patients

NIH National Institute of Allergy and Infectious Diseases

- Contact our team via email:

  bioinformatics@niaid.nih.gov

  Instructor: zhuy16@nih.gov

  Or leave your question at this Googledoc:

  https://tinyurl.com/GRN-best

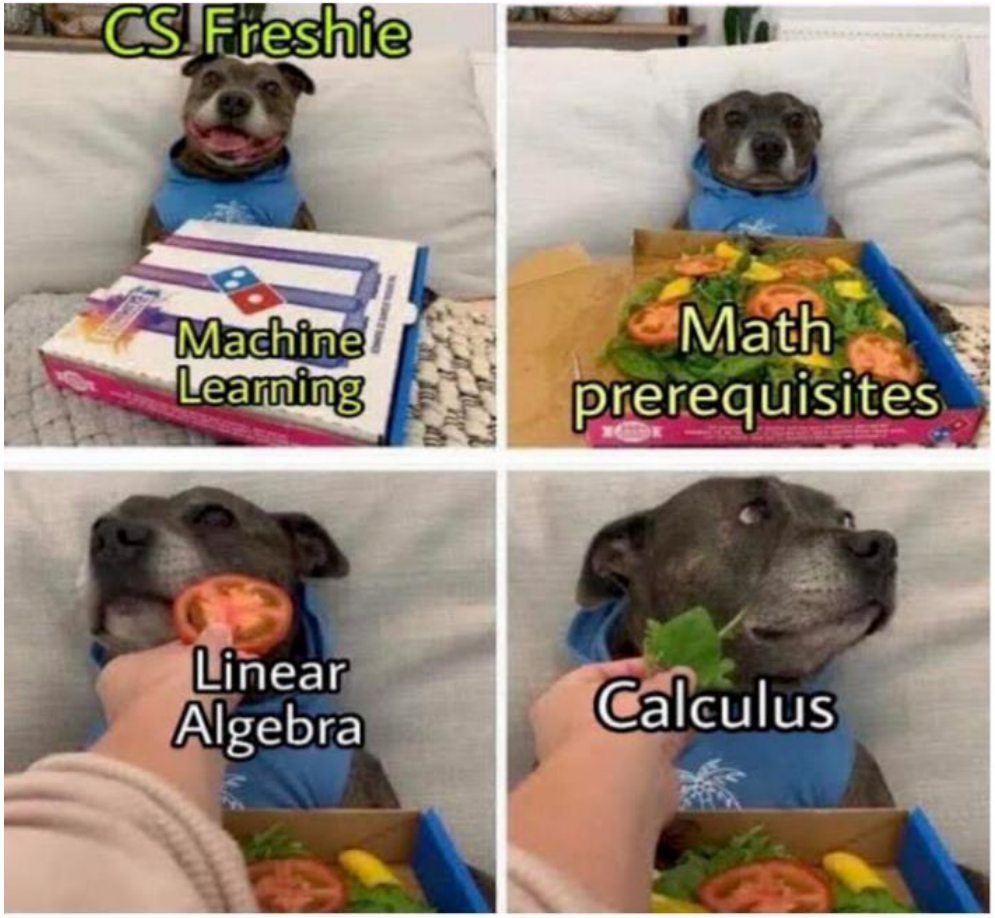
- Practical materials and PDF version of PPTs are here:

  git clone **https://github.com/niaid/Gene_Regulatory_Networks/**

Also a bit of personal learning tips:
https://github.com/zhuy16/learning_notes

NIAID

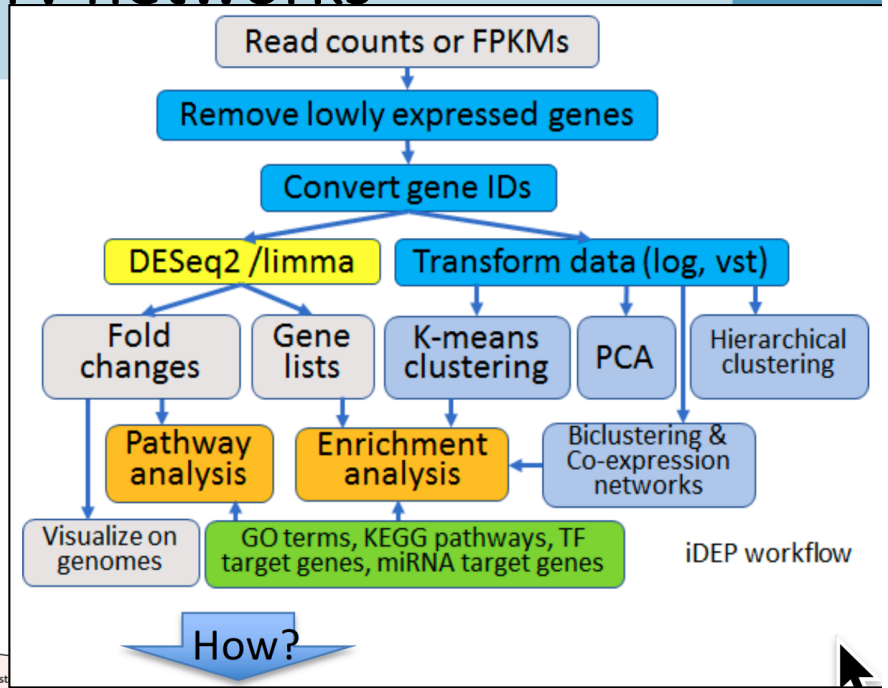
Learning Bioinformatics …

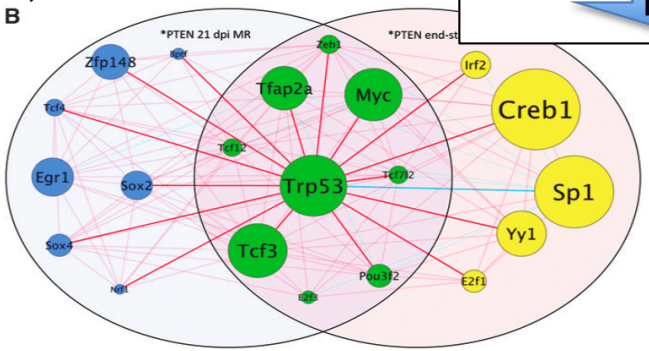



Biology in the modern era …
Microarray
RNA sequencing
Epigenomes
Single cell transcriptomics and Single cell Epigenomics
Microbiomes
Hard to digest
……
… Where is my key factor to study?
YouTube
Google
Wikipedia
stackoverflow
Statistics, Mathematics, Programming! All good stuff !!!

Why study gene regulatory networks

- Why
  - High throughput are screening procedures
  - Gene lists do not inform relationship between genes
  - Regulatory information is hidden in the statistical relationships
  - Down-stream validation is costly, tediously and risky
- Aims
  - To find regulatory network, to inform finding of key candidate genes/master regulators for validation
- Challenges
  - Data are massive, and mostly static that do not reflect dynamic regulation
  - Relationships can take various forms, linear or non-linear





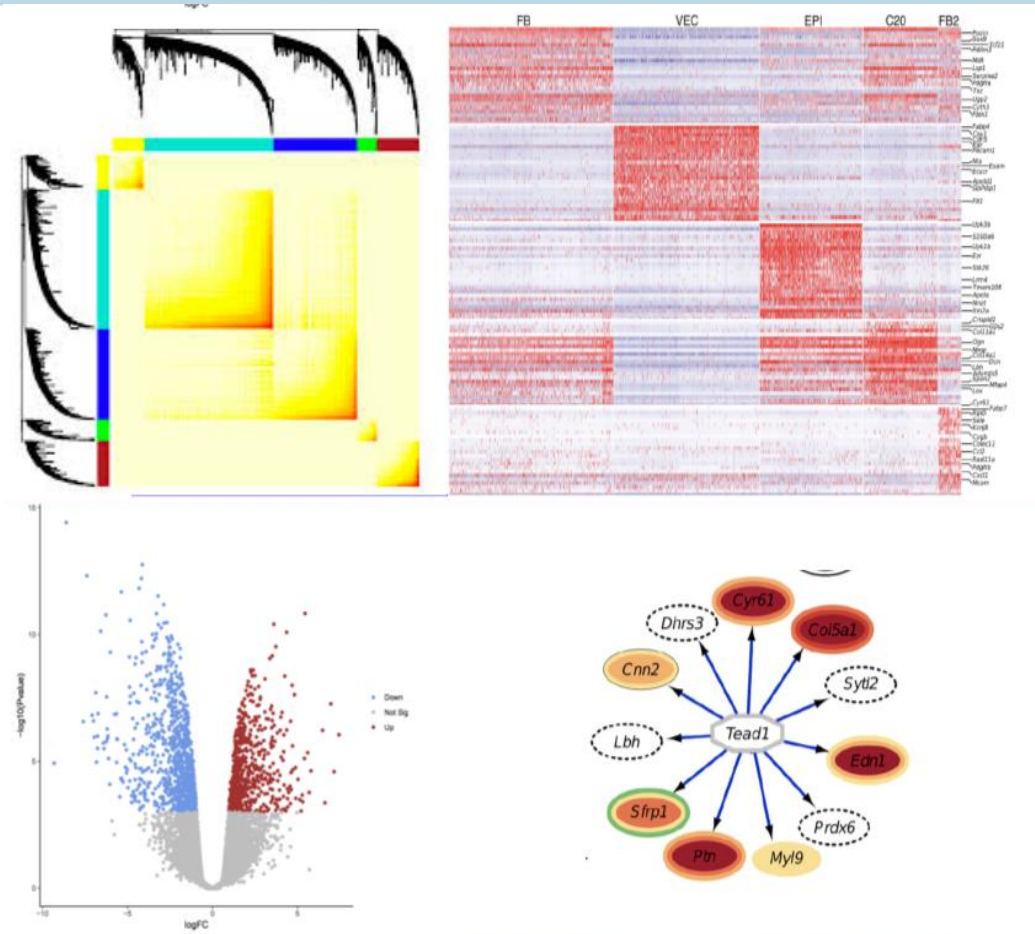
NIH National Institute of Allergy and Infectious Diseases

Objectives: achieve a general overview of gene regulatory network (GRN) analysis

- Biological rational
- Theoretical bases of theories, intuitions and tools.
- Simple practical demonstration
- Links to further resources

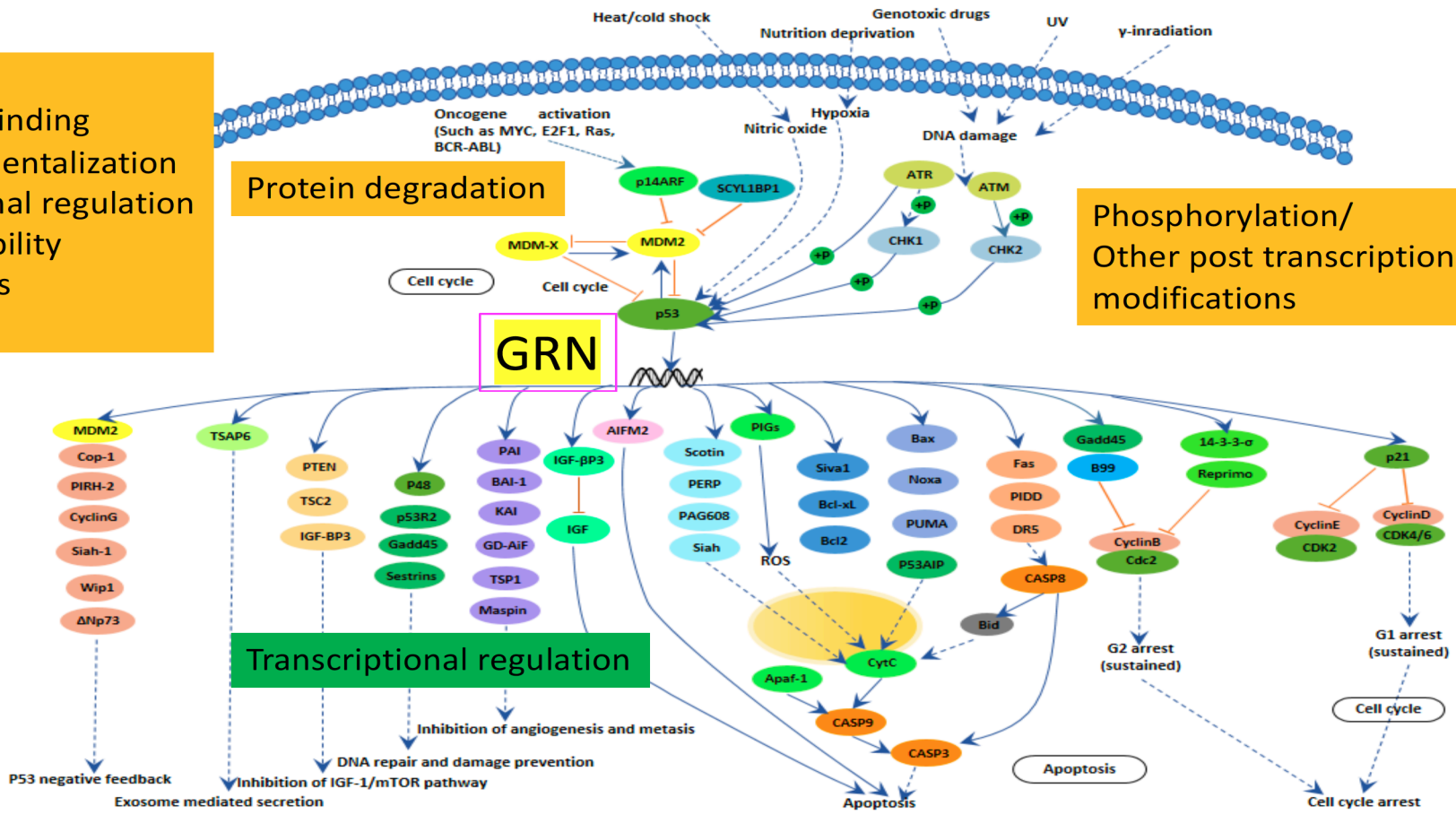
Prerequisite: identify a group of genes

- How to get a set of interesting genes?
  - Top variable genes
  - Differential expression analysis
  - Unsupervised Clustering
  - WGCNA: Weighted Gene Co-expression Network Analysis



A side note: GRN is only a subset of cellular networks

Others
Cofactor binding
Compartmentalization
Translational regulation
mRNA stability
Epigenetics
...



NIAID

A schematic illustration on the difference

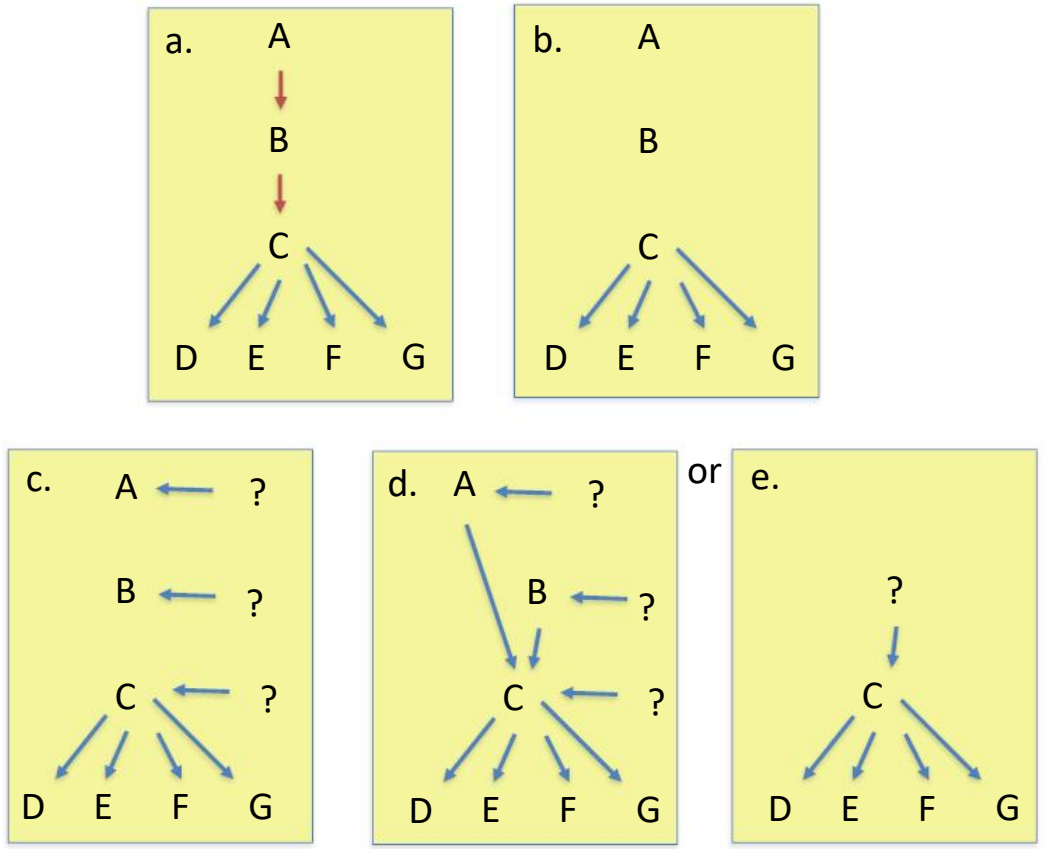
(a) conventional structure of intracellular signaling

(b) Biochemical link will be invisible in GRN

(c) Other regulators kick in

(d) Upstream factors may be independent from each other in the transcriptional GRN

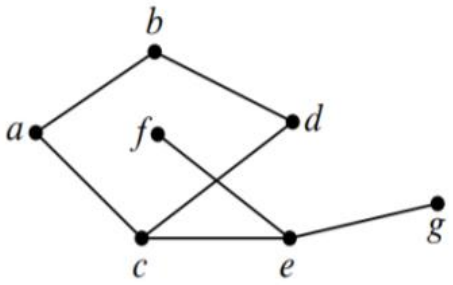
(e) Upstream factors may be totally invisible in the analysis



Notations in Network/Graph theory

- Graph (network) theory
  - Vertex/Vertices (nodes)
    - Genes
  - Edges (statistical interactions)
    - Undirected graph
      - ARACNE, PIDC
    - Directed Acyclic Graph (DAG)
      - Bayes net
    - Edge weights

Basic graph theory



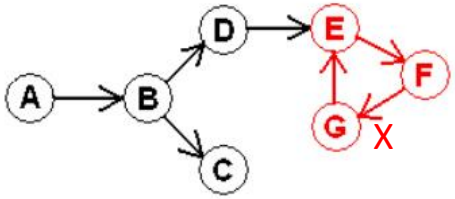
A undirected graph

The graph in Figure 5.1 is expressed mathematically as $G = (V, E)$, where:

$V = \{a, b, c, d, e, f, g, h, i\}$ Vertex/node

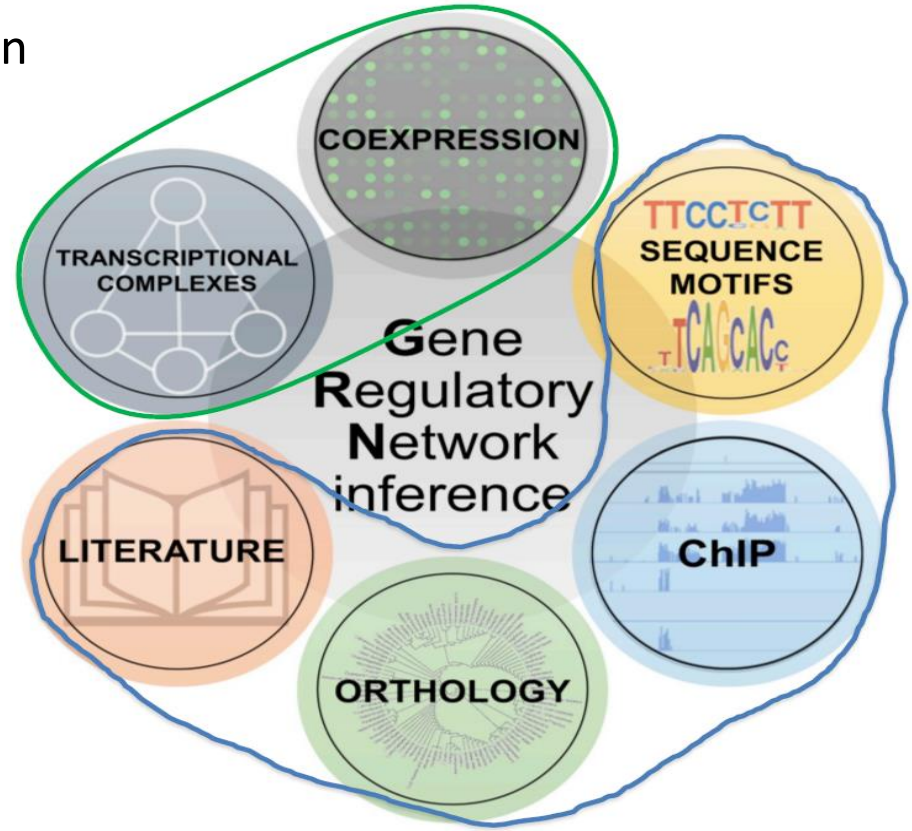
$E = \{\{a, b\}, \{a, c\}, \{b, d\}, \{c, d\}, \{c, e\}, \{e, f\}, \{e, g\}, \{h, i\}\}.$

Directed Acyclic Graph (DAG)



Computational Strategies

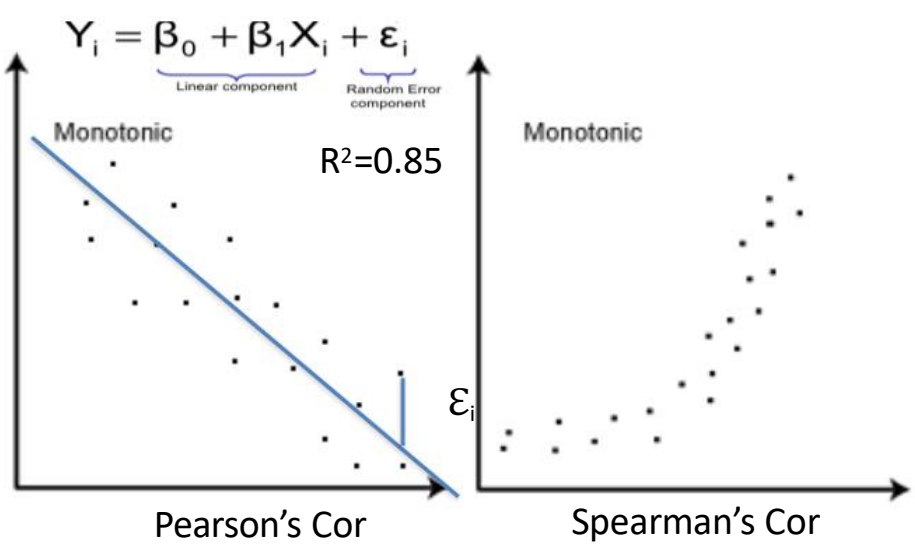
- Infer network through expression pattern based on statistical relationship between genes.
  - Linear relationships
  - Mutual information—linear/non-linear statistical relationships
  - Bayesian network /Bayes Net
  - Regression tree + random forest
- Predicting master regulators through promoter sequences
  - cis-Target and iRegulon
- Combinatory approach
  - SCENIC (GENIE3 and cis-Target)



Mercatelli et al., 2019

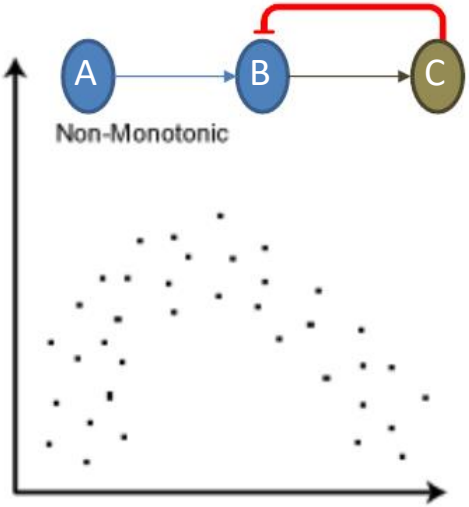
Types of statistical relationship

- Pearson's correlation, low computational complexity, and their capability of inferring genome-wide networks even when a relatively low number of samples (*n* = 50) is available
- Spearman's correlation, rank based, no assumption on normal distribution
- Mutual information, any statistical relationship. But requires N to be sufficiently large for a good approximation

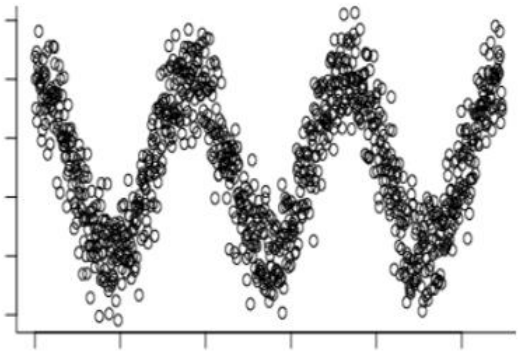


Pearson's Cor

Spearman's Cor

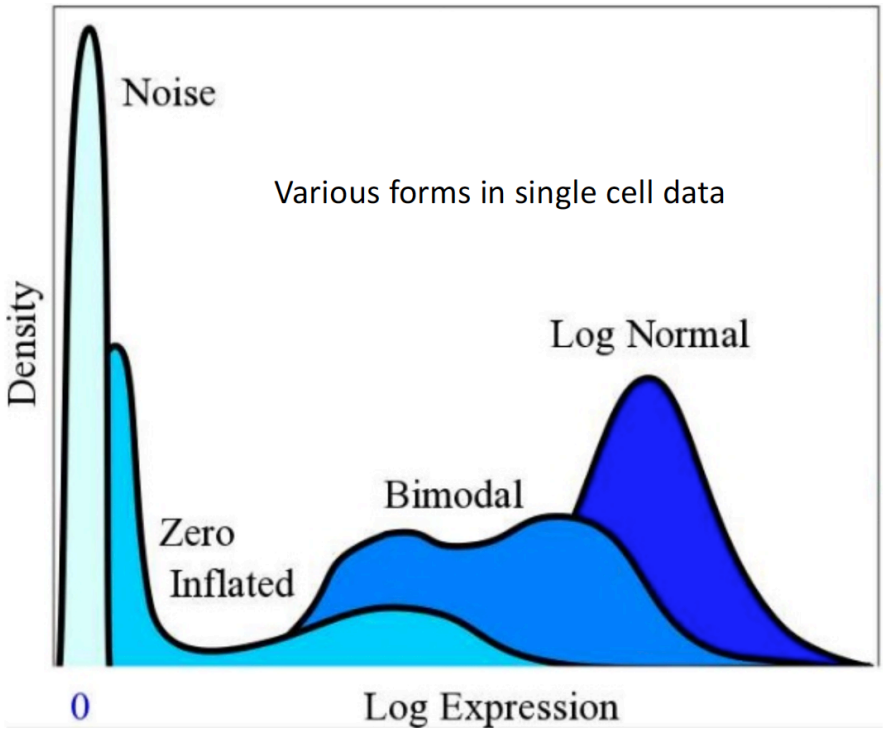


Even with cyclic information



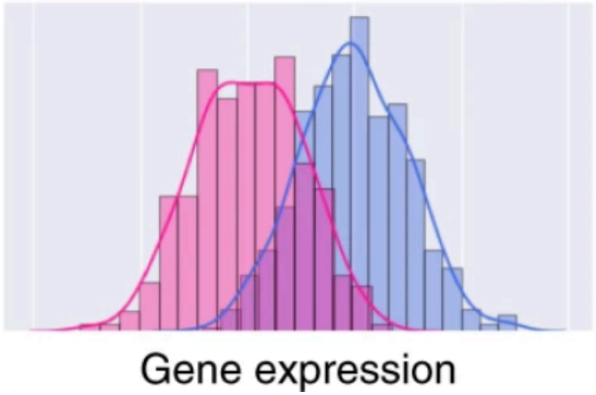
Mutual information

Various types of distributions of gene expression



https://www.slideshare.net/TimothyTickle/introduction-to-singlecell-rnaseq

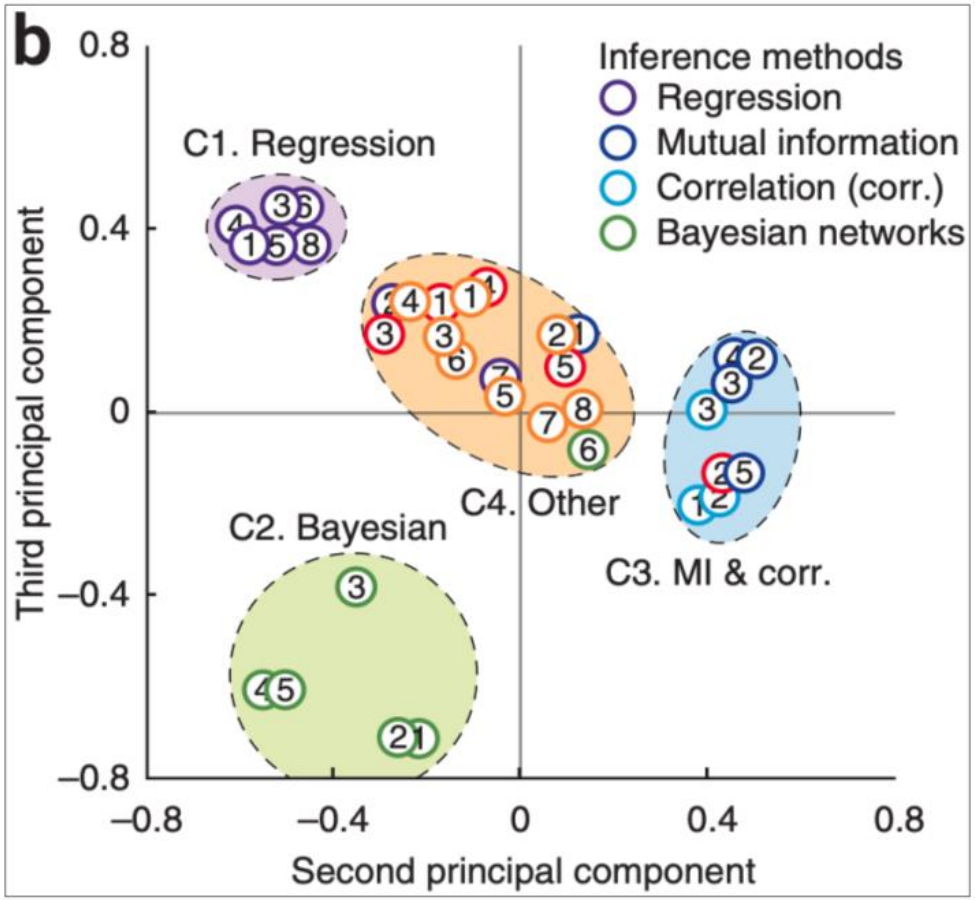
Two groups of cells
Log normal distribution



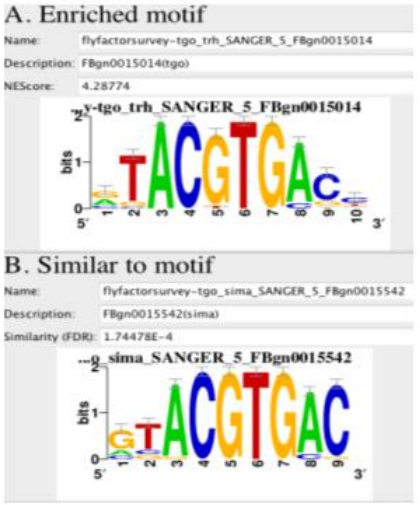
(Expression values of a given gene in different replicates /cells)

DREAM5– Network Inference Challenge (2010)

Dialogue for **R**everse **E**ngineering **A**ssessments and **M**ethods (Open science challenges)



iRegulon/cis-Target



Using database to predict upstream transcriptional regulators

National Institute Allergy and Infectious Diseas

NIAID

Overview of Tools in GRN inference

Software	ARACNE	NetworkInference/ PIDC	bnlearn	GENIE3	iRegulon	SCENIC
semantics	Mutual information	Partial Information Decomposition	Bayes theory	Random Forest, Regression tree	Promoter and TF binding sequence, database	Combination of regression and promoter sequence
years published	2006	2017	2009	2010	2014	2017
No. of cited	2179	82	894	658	337	265
FullName/ explanation	Algorithm for the Reconstruction of Accurate Cellular Networks	Using proportional unique contribution (PUC) to a target gene	Bayes net structure and parameter learning, causality	GEne Network Inference with Ensemble of trees	reverse-engineer the transcriptional regulatory network with regulatory sequence analysis	single-cell regulatory network inference and clustering
Implementation	GUI (geWorkbench)	Julia	R	R	GUI (Cytoscape)	R, Python
type of experiment	Microarray, bulk RNA-seq	Single cell data	General	single cell data	a list of gene names	single cell data
input format	csv	csv	csv	csv	a list	csv/loom file
output	network file	network file	directed network file	network file	network file/binding sequences	network file/heatmap

NIAID

Information theory

$$= \sum_x P(x) \cdot \log\left(\frac{1}{P(x)}\right)$$

The prob. of event X

WHAT IS THIS?

Surprise/ info of event X

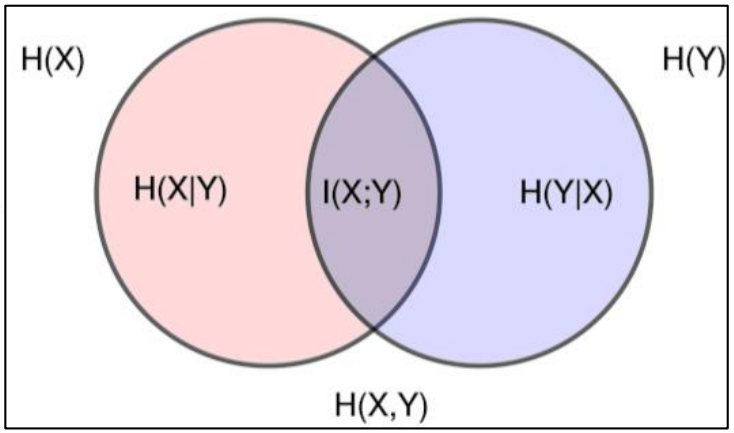
- Entropy
  - Information content is coded within the statistical distribution of a the variables and can be calculated as **Shannon's entropy**
  - Statistical interaction ("correlation") between two random variables can be calculated as their **mutual information**

$$entropy(flip\ a\ coin\ once) =$$

$$-\left[\left(\frac{1}{2}\right)log_2\left(\frac{1}{2}\right)\right] + -\left[\left(\frac{1}{2}\right)log_2\left(\frac{1}{2}\right)\right] = 1(bit)$$

Mutual Information (MI)

- H(X), H(Y) are the information contents for X and Y
- I(X;Y) is the mutual information of X and Y
- (MI is >=0 with no upper limit)



All the information you have got = entropy

Observed joint probability

Mutual information

$$MI_{ij} = \sum_{x_i} \sum_{x_j} p(x_i, x_j) \log_2 \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$$

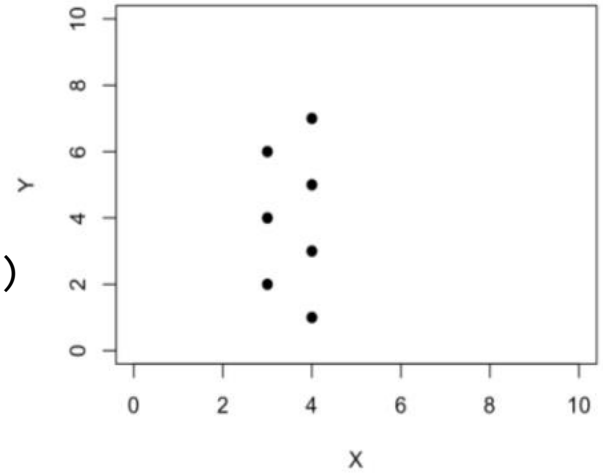
Equal to joint probability assuming x_i, x_j co-occur purely by chance.

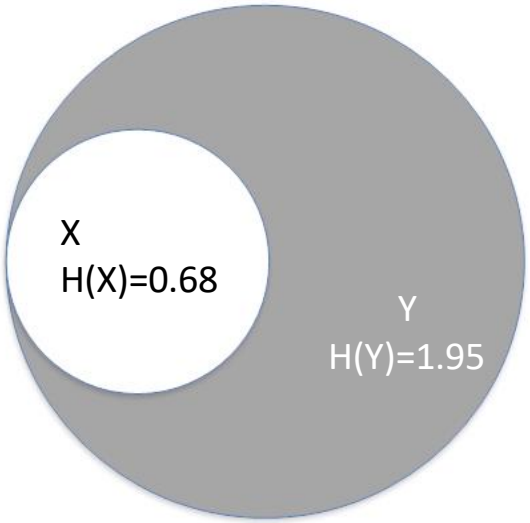
NIAID

Mutual information captures non-linear relationships

In R:

```
X=c(4,3,4,3,4,3,4)
Y=c(1,2,3,4,5,6,7)
plot(X,Y)
```



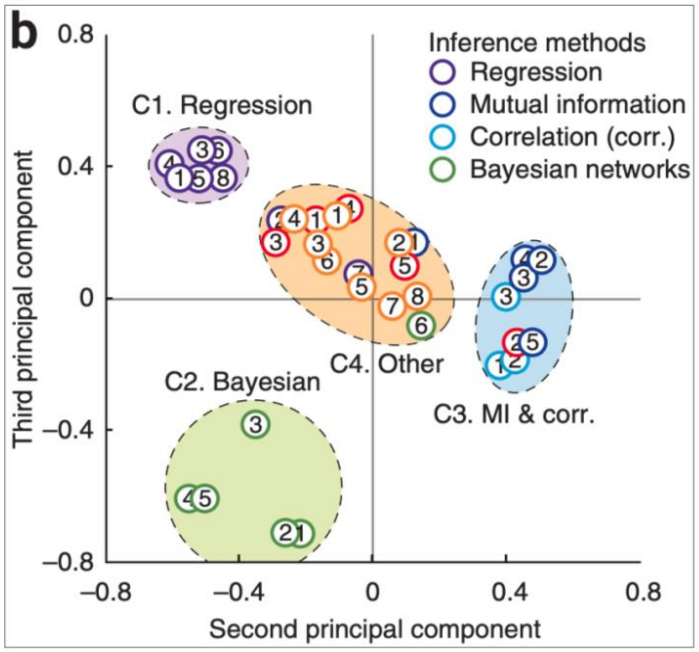



```
#install.packages("infotheo")
library(infotheo)
entropy(X)
entropy(Y)
mutinformation(X,Y)
#install.packages("eulerr")
library(eulerr)
fit <- euler(c(A = entropy(X), B = entropy(Y), "A&B" =
mutinformation(X,Y)))
plot(fit)
```

NIAID

Method 1: ARACNE (first in the class, 2006)

Algorithm for the Reconstruction of Accurate Cellular Networks



BMC Bioinformatics

BioMed Central

Proceedings

Open Access

ARACNE: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context

Adam A Margolin[1,2], Ilya Nemenman[2], Katia Basso[3], Chris Wiggins[2,4], Gustavo Stolovitzky[5], Riccardo Dalla Favera[3] and Andrea Califano*[1,2]

Address: [1]Department of Biomedical Informatics, Columbia University, New York, NY 10032, [2]Joint Centers for Systems Biology, Columbia University, New York, NY 10032, [3]Institute for Cancer Genetics, Columbia University, New York, NY 10032, [4]Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10032 and [5]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

Email: Adam A Margolin - adam@dbmi.columbia.edu; Ilya Nemenman - ilya.nemenman@columbia.edu; Katia Basso - kb451@columbia.edu; Chris Wiggins - chw2@columbia.edu; Gustavo Stolovitzky - gustavo@us.ibm.com; Riccardo Dalla Favera - rd10@columbia.edu; Andrea Califano* - califano@c2b2.columbia.edu

* Corresponding author

from NIPS workshop on New Problems and Methods in Computational Biology
Whistler, Canada. 18 December 2004

Published: 20 March 2006

BMC Bioinformatics 2006, **7**(Suppl 1):S7 doi:10.1186/1471-2105-7-S1-S7

Abstract

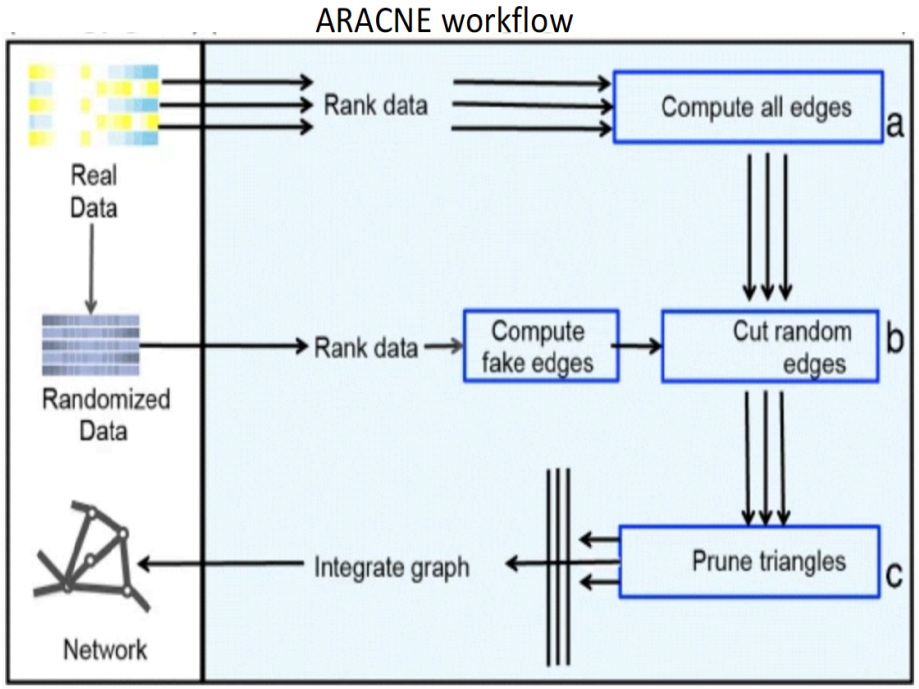
Background: Elucidating gene regulatory networks is crucial for understanding normal cell physiology and complex pathologic phenotypes. Existing computational methods for the genome-wide "reverse engineering" of such networks have been successful only for lower eukaryotes with simple genomes. Here we present *ARACNE*, a novel algorithm, using microarray expression profiles, specifically designed to scale up to the complexity of regulatory networks in mammalian cells, yet general enough to address a wider range of network deconvolution problems. This method uses an information theoretic approach to eliminate the majority of indirect interactions inferred by co-expression methods.

Results: We prove that ARACNE reconstructs the network exactly (asymptotically) if the effect of loops in the network topology is negligible, and we show that the algorithm works well in practice, even in the presence of numerous loops and complex topologies. We assess ARACNE's ability to reconstruct transcriptional regulatory networks using both a realistic synthetic dataset and a microarray dataset from human B cells. On synthetic datasets ARACNE achieves very low error rates and outperforms established methods, such as Relevance Networks and Bayesian Networks. Application to the deconvolution of genetic networks in human B cells demonstrates ARACNE's ability to infer validated transcriptional targets of the cMYC proto-oncogene. We also study the effects of misestimation of mutual information on network reconstruction, and show that algorithms based on mutual information ranking are more resilient to estimation errors.

NIH National Institute of Allergy and Infectious Diseases

NIAID

Determining interactions through MI



Data processing inequality (DPI), to remove indirect interactions:

$$x_i \rightarrow x_j \rightarrow x_k,$$

then,

$$I(x_i, x_k) \leq \min\left\{I(x_i, x_j), I(x_j, x_k)\right\}$$

If *i* and *k* are not directly connected, the correlation of them will not be high as a direct connection.

National Institute of Allergy and Infectious Diseases

Illustration of how mutual information is calculated with 3 variables
https://www.youtube.com/watch?v=3iplfAfGzzI

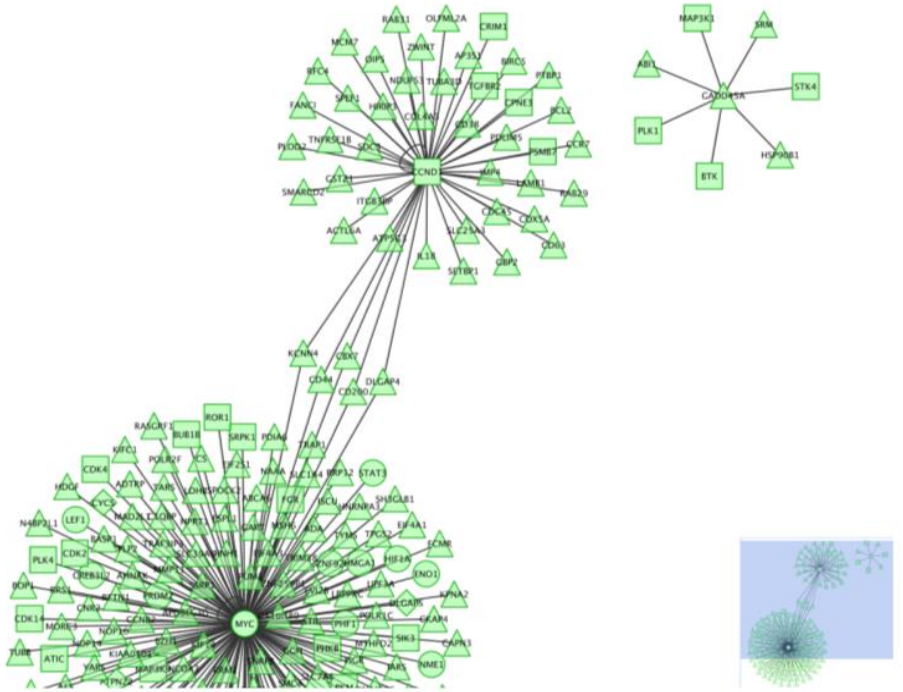
NIAID

No demo but with an instruction online: Notes on running ARACNE

- **geWorkbench** will have ARACNE installed by default
  - http://wiki.c2b2.columbia.edu/workbench/index.php/Download_and_Installation
- What data to use for ARACNE
  - The **Bcell-100.exp** as an expression file
  - http://wiki.c2b2.columbia.edu/workbench/index.php/Tutorial_-_Data#Tutorial_data_files
  - Use **HG_U95Av2.na36.annot.csv** - most updated version
  - To download file need registration on Thermofisher website!!! Already downloaded in github.
- Video Tutorial: **https://www.youtube.com/watch?v=CU4nukYswt0**
- The layout of the geWorkbench is different from the video
  1. The project file cannot be created.
  2. The window looks different as well; no sub-field of windows at the bottom.

Demo ARACNE

- Using a microarray data
  - The interface of geWorkbench
  - setting parameters
  - Run ARACNE
  - Export from geWorkbench
  - Import to Cytoscape
  - Visualization



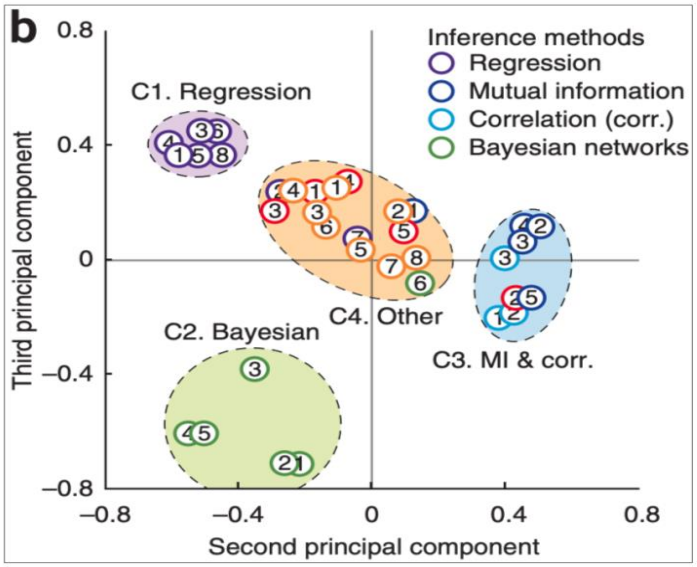
Discussion on ARACNE

- Straightforward concept of mutual information
- Processing inequality to break indirect linkages
- Most widely used GRN software
- Good tool to find potential target genes with known TF
- Implementation is very slow; newer ones may be more useful

GUI interface software (in geWorkbench) is available

And the tutorial materials in the repo

Method 2: Partial Information Decomposition (NetworkInference for single-cell data)



Article

Cell Systems

Gene Regulatory Network Inference from Single-Cell Data Using Multivariate Information Measures

Highlights

- PIDC infers gene regulatory networks from single-cell transcriptomic data
- Multivariate information measures and context in PIDC improve network inference
- Heterogeneity in single-cell data carries information about gene-gene interactions
- Fast, efficient, open-source software is made freely available

Authors

Thalia E. Chan, Michael P.H. Stumpf, Ann C. Babtie

Correspondence

m.stumpf@imperial.ac.uk (M.P.H.S.), a.babtie@imperial.ac.uk (A.C.B.)

In Brief

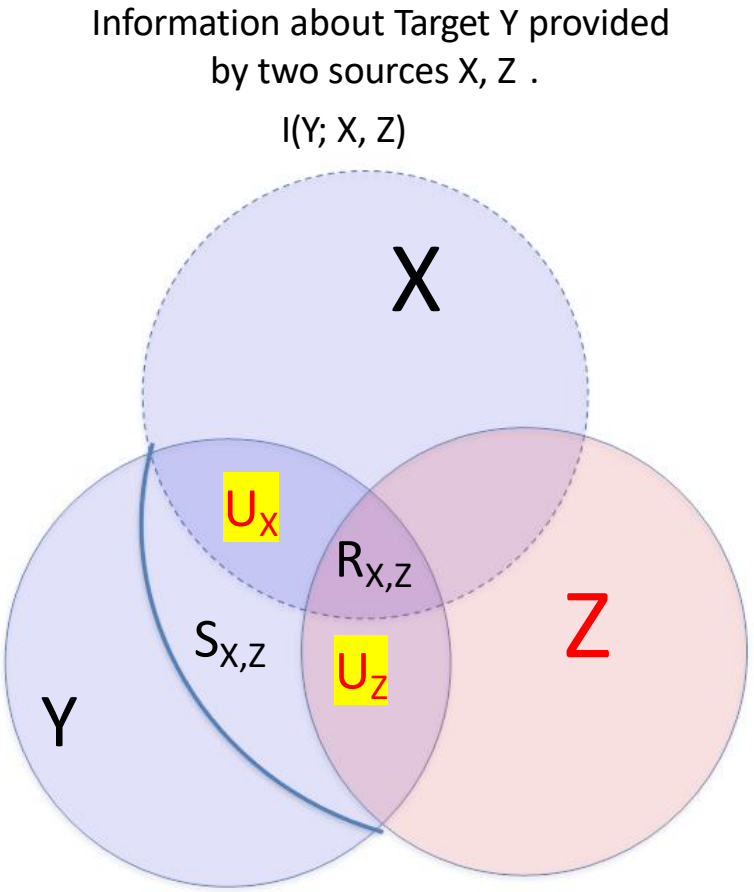
Chan et al. develop PIDC, a fast, efficient algorithm that makes use of multivariate information theory, to reliably infer gene-gene interactions in heterogeneous, single-cell gene expression data and build gene regulatory networks.

NIH National Institute of Allergy and Infectious Diseases

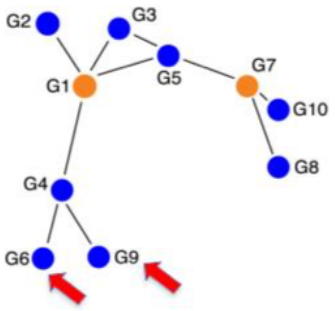
NIAID

Partial Information Decomposition

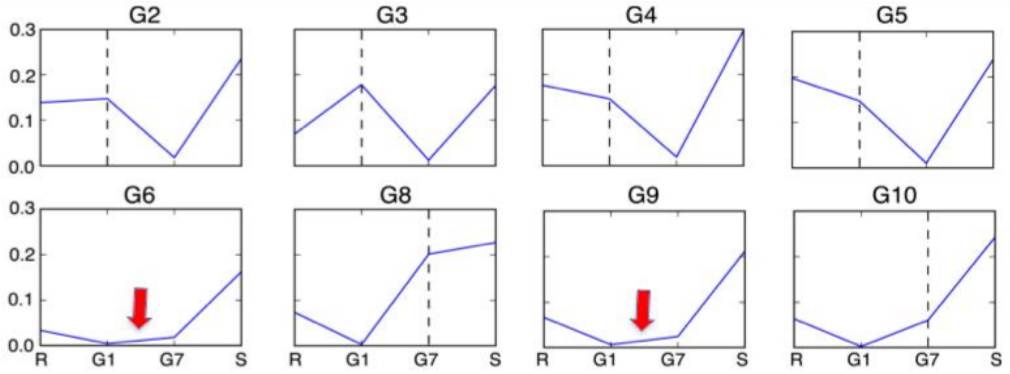
- Consider three genes X , Y and Z
- Consider that Y is statistically dependent on X, Z
- The information from X and Z that predict Y can be decomposed to 4 components (U_X, U_Z, $Redundancy_{X,Z}$ and $Synergy_{X,Z}$)



Are any of these components an indicator of a network connection?

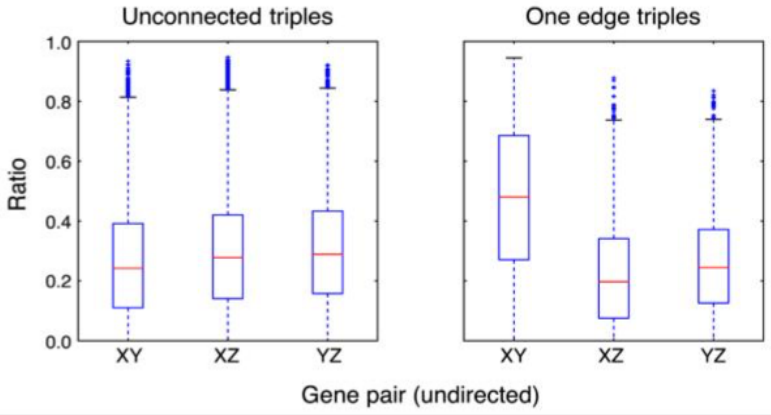


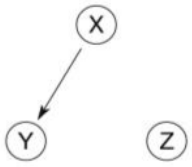
A synthetic model gene network

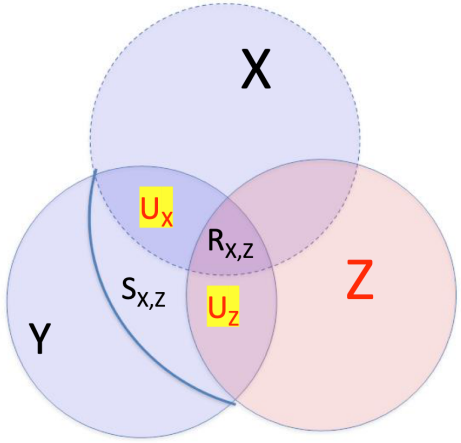


$R_{X,Z}$ U_X U_Z $S_{X,Z}$

Yes, Unique information is most sensitive to a direct connection



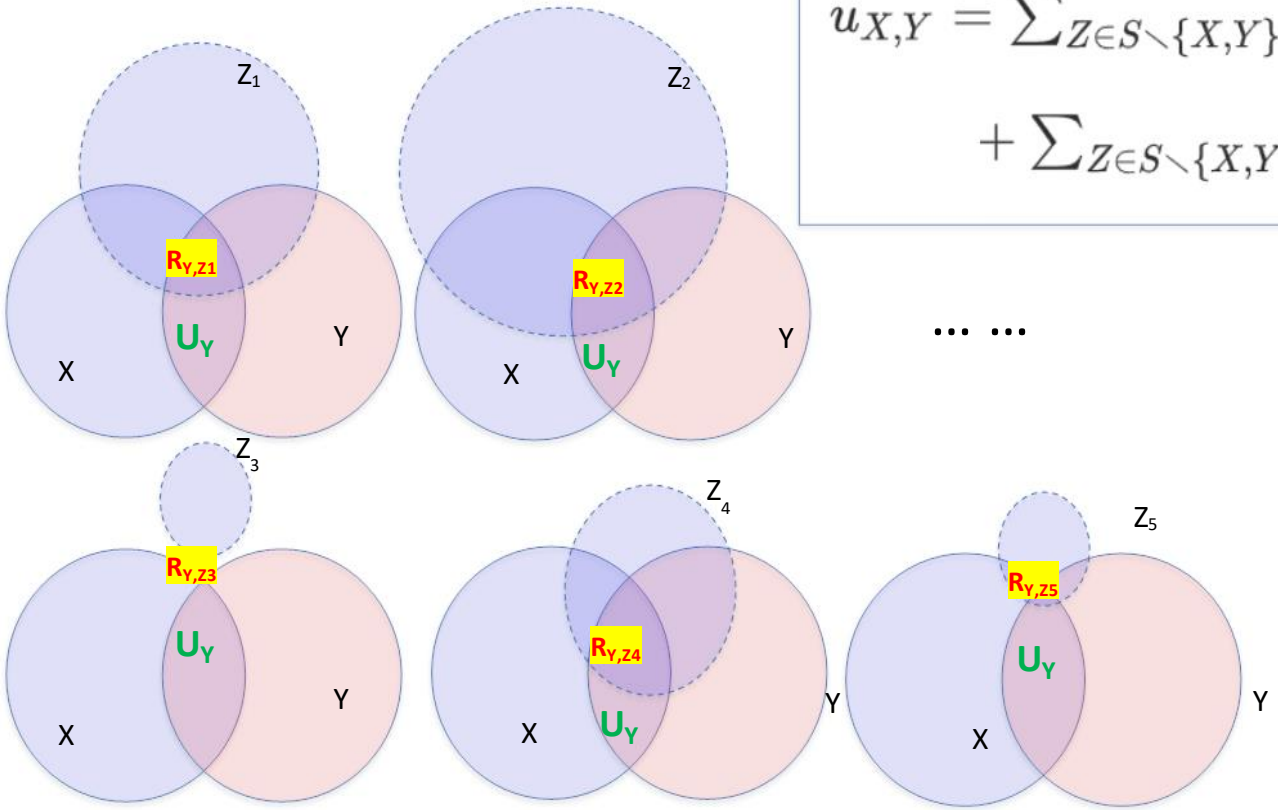


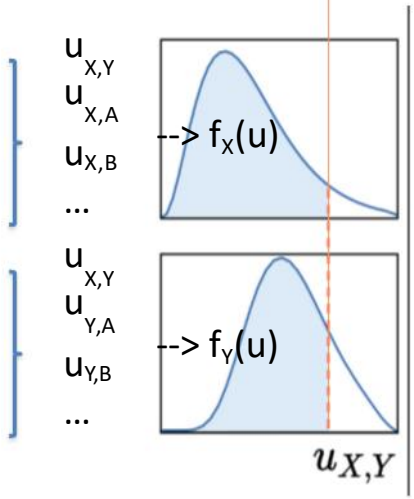


NIAID

Proportional Unique Contribution

$$u_{X,Y} = \sum_{Z \in S \setminus \{X,Y\}} \frac{\text{Unique}_Z(X;Y)}{I(X;Y)} + \sum_{Z \in S \setminus \{X,Y\}} \frac{\text{Unique}_Z(Y;X)}{I(X;Y)}$$





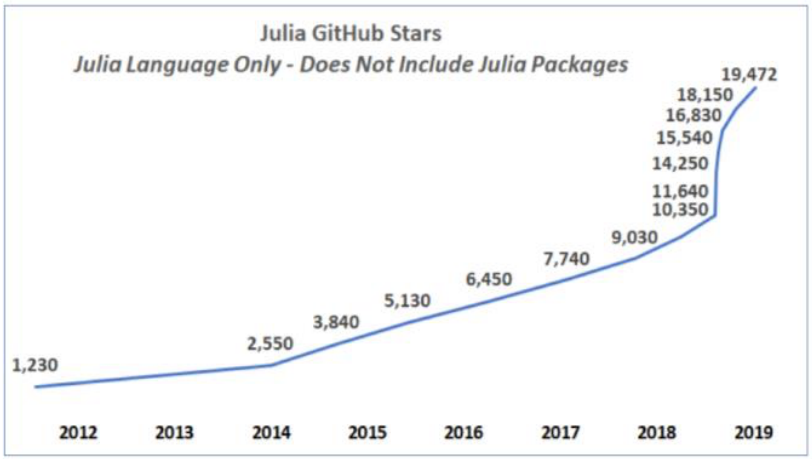
NIAID

Discussion on NetworkInference

- Good at discovery of single-edge connection, may systematically loss two-edge, 3-edge connections
- Relatively new, not sufficiently validated
- Undirected, and no attempt is made to find the directionality of a regulation
- Easy to use with good speed in Julia

Julia: A fresh approach to numerical computing

- A young language developed by researchers in MIT on 2012
- Advantages: speed of C and intuitive as python
- Can use all the python libraries
- Disadvantage, changes a lot as versions upgrade – not very stable

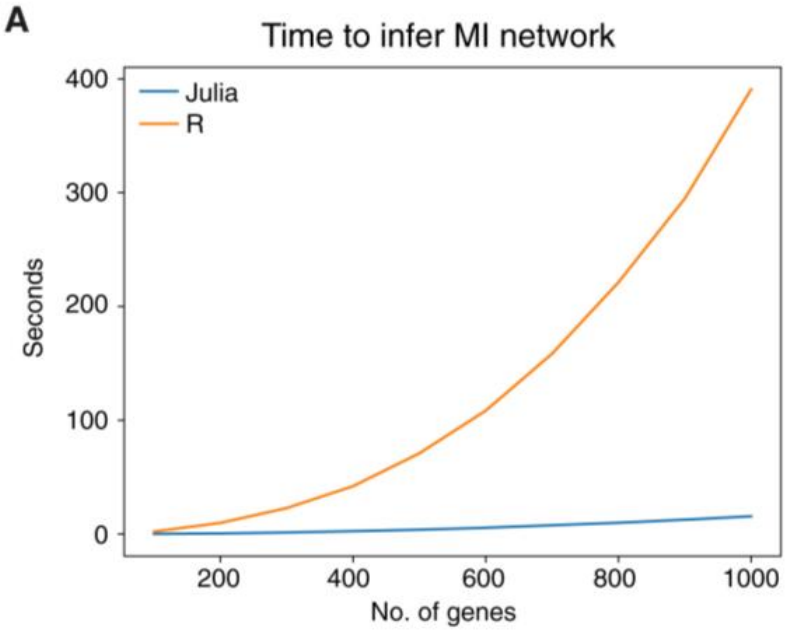


NIH National Institute of Allergy and Infectious Diseases

NIAID

Implementation in Julia

- Using NetworkInference in Julia
  - Install Julia
    - https://julialang.org/downloads/, download the binary file for Generic Linux 64-bit
  - Install NetworkInference, simple
    - using Pkg
    - Pkg.add("NetworkInference")
    - use NetworkInference

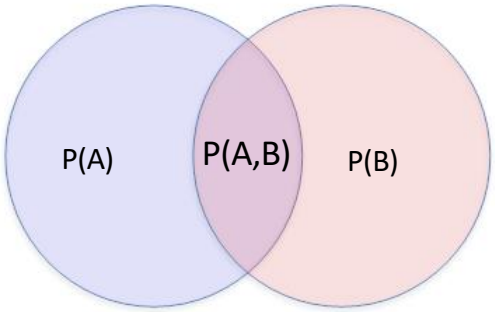


git clone https://github.com/niaid/BESTversion/Gene_Regulatory_Networks

Demonstration on NetworkInference in Julia

- Import data
  - Run NetworkInference
  - Parameters
  - Export to CSV files
  - Import to Cytoscape for visulization

Method 3:
Bayesian Network – conditional dependence



$$P(A,B)=P(A|B)*P(B)=P(B|A)*P(A)$$

Bayes rule:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

National Institute of Allergy and Infectious Diseases

Journal of Statistical Software

July 2010, Volume 35, Issue 3. *http://www.jstatsoft.org/*

Learning Bayesian Networks with the bnlearn R Package

Marco Scutari
University of Padova

Abstract

bnlearn is an R package (R Development Core Team 2010) which includes several algorithms for learning the structure of Bayesian networks with either discrete or continuous variables. Both constraint-based and score-based algorithms are implemented, and can use the functionality provided by the **snow** package (Tierney *et al.* 2008) to improve their performance via parallel computing. Several network scores and conditional independence algorithms are available for both the learning algorithms and independent use. Advanced plotting options are provided by the **Rgraphviz** package (Gentry *et al.* 2010).

Keywords: bayesian networks, R, structure learning algorithms, constraint-based algorithms, score-based algorithms, conditional independence tests.

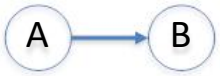
NIAID

Bayesian Approach – representation of network as a chain of conditional probabilities

If A and B are independent
A= winter (yes 25%, no 75%)
B= happy (yes 50%, no 50%)

P(A) = 25%
P(B) = 50%
P(winter, happy) =
P(A,B)=25%*50%=12.5%
Correct

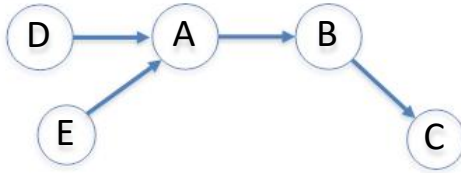
How about if B dependent on A:
A= winter (yes 25% or no 75%)
B= cold weather (overall yes 25%, no 100%)



P(A)=25%
P(B|A)=100% on winter and 0% on other seasons.
Overall 25%
The conditional probability of winter & cold is
P(A=winter, B=cold)=P(A)P(B)=25%*25%=6.25%
wrong!

P(winter, cold) =
P(A=winter;B=cold)=P(A)**P(B|A)**=25%*100%=25%

A= winter (yes or no 50% each)
B= cold weather (yes or no 100%)
C= wear sweater (yes or no, 50%)
D= month of a year
E= location on the earth



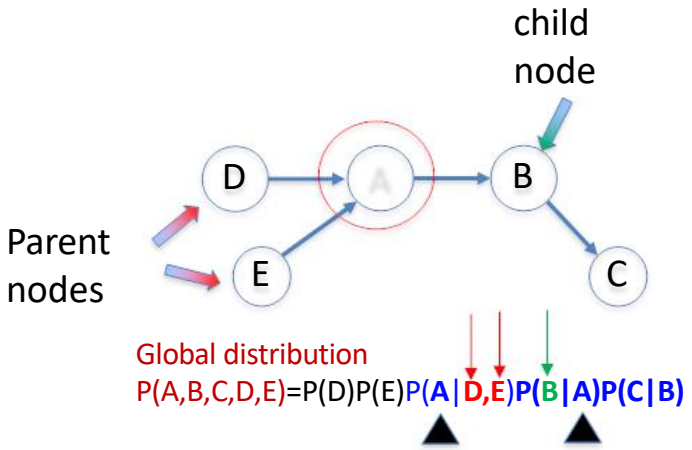
Global distribution
P(A,B,C,D,E)=P(D)P(E)P(A|D,E)**P(B|A)P(C|B)**

NIH National Institute of Allergy and Infectious Diseases

NIAID

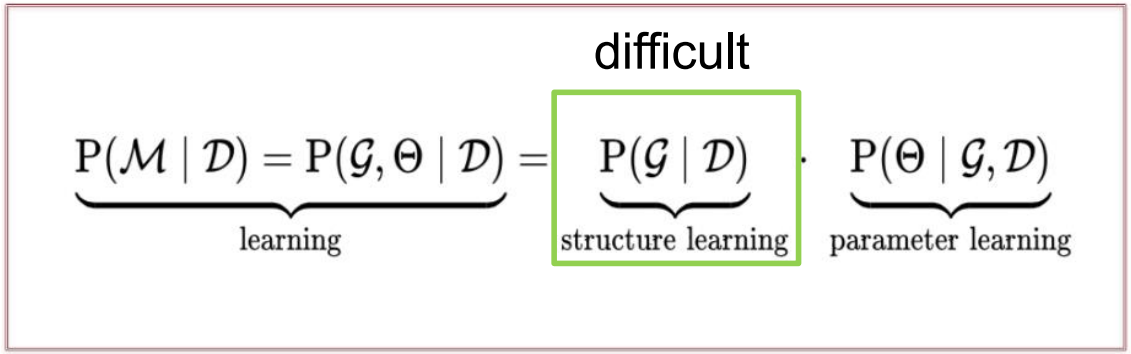
Parent nodes and child node

- To a given node
  - Conditional nodes (reasons or contributing factors) are called parents.
  - The consequent nodes are child nodes.
- 2 components in Bayes Net,
  - 1. Graph structure: edges (arrows, can be defined by a list of nodes and parents of each node).
  - 2. Parameters (θ): P(D), P(E), P(A|D,E), P(B|A), P(C|B)



Learn the network structure and parameters

Divide the Bayes net into two components:



Notations:

- Θ: the conditional probabilities
- G: the structure
  - Can be decomposed to each node with parents (Xi and Πi).

Learn the structure:

1. Use MI to find the undirected network.
2. Use Causality theory to assign directness

NIH National Institute of Allergy and Infectious Diseases

NIAID

Assign directions: Induction of Causality algorithm

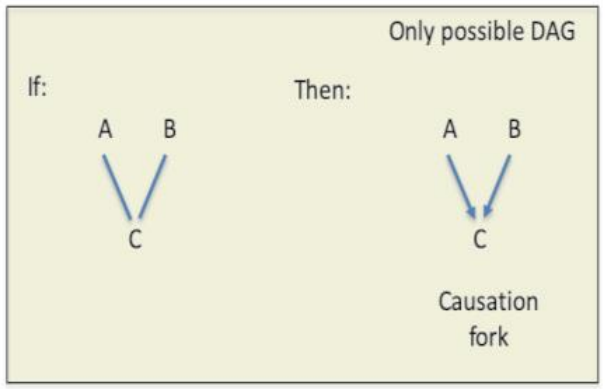
The Induction of Causality Algorithm:

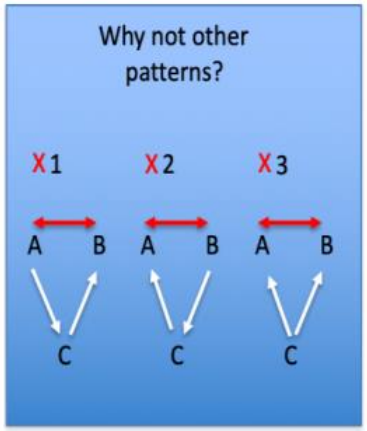
If,

1. A, B are not connected through any other connection.
2. A and B have a common connection to C

Then:

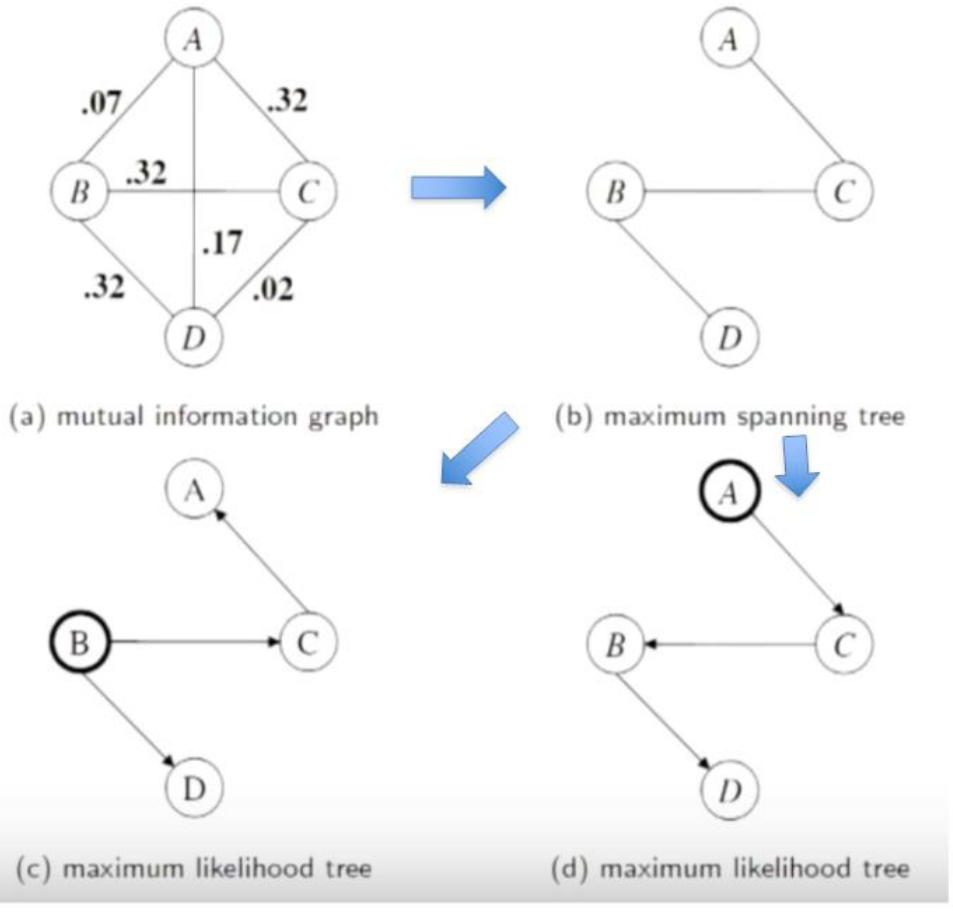
arrows point from A and B to C





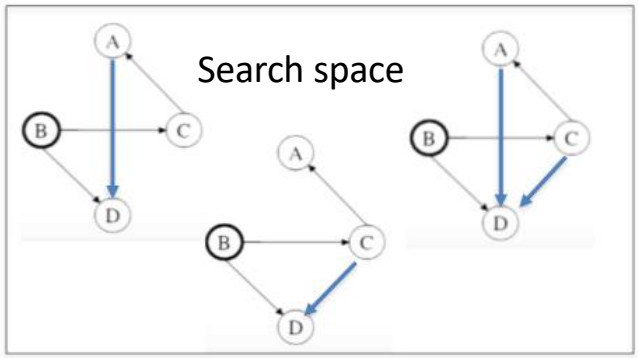
But direction cannot be fully solved by Causality reasoning complications---

- Constructing Maximum Spanning Tree.
- Add direction through starting from a random node.



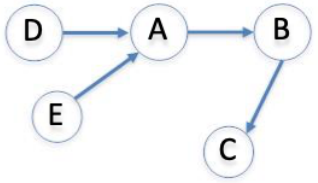
BIC score = Bayesian Information Criterion

$$\text{Score}(G|\mathcal{D}) \stackrel{def}{=} \text{LL}(G|\mathcal{D}) - \psi(N) \cdot ||G||$$



Curse of the dimensionality

- Complexity increases exponentially with respect to:
  - Number of bins each node can take.
  - Number of parent nodes that a node can have.
- Solutions
  - Breaking a huge network into conditionally independent units



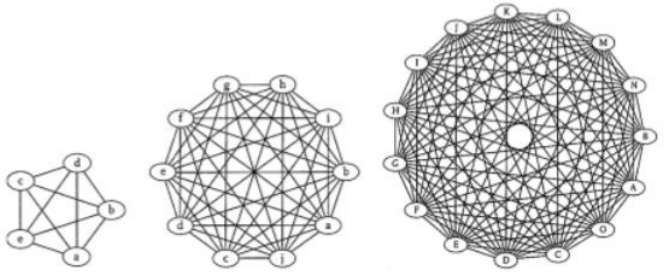
Global distribution

$$P(A,B,C,D,E) = P(D)P(E)P(A|D,E)P(B|A)P(C|B)$$

If each sample has 5 bins

Values to fully enumerate the distribution table: 5^5=1875

Enumerate the Bayesian network: 5 +5 +125 +25 +25=185



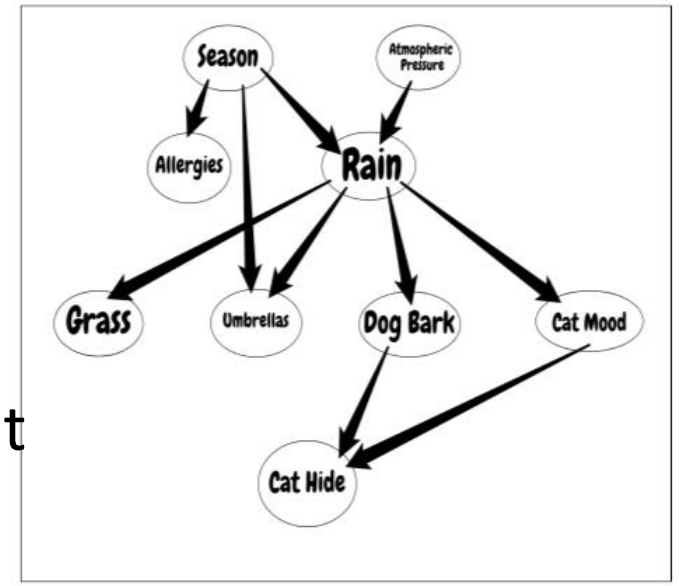
Discussion on Bayesian Network

Pros

- Easy interpretation

Cons

- Requires large computation
- Decomposition of Bayesian seems not efficient



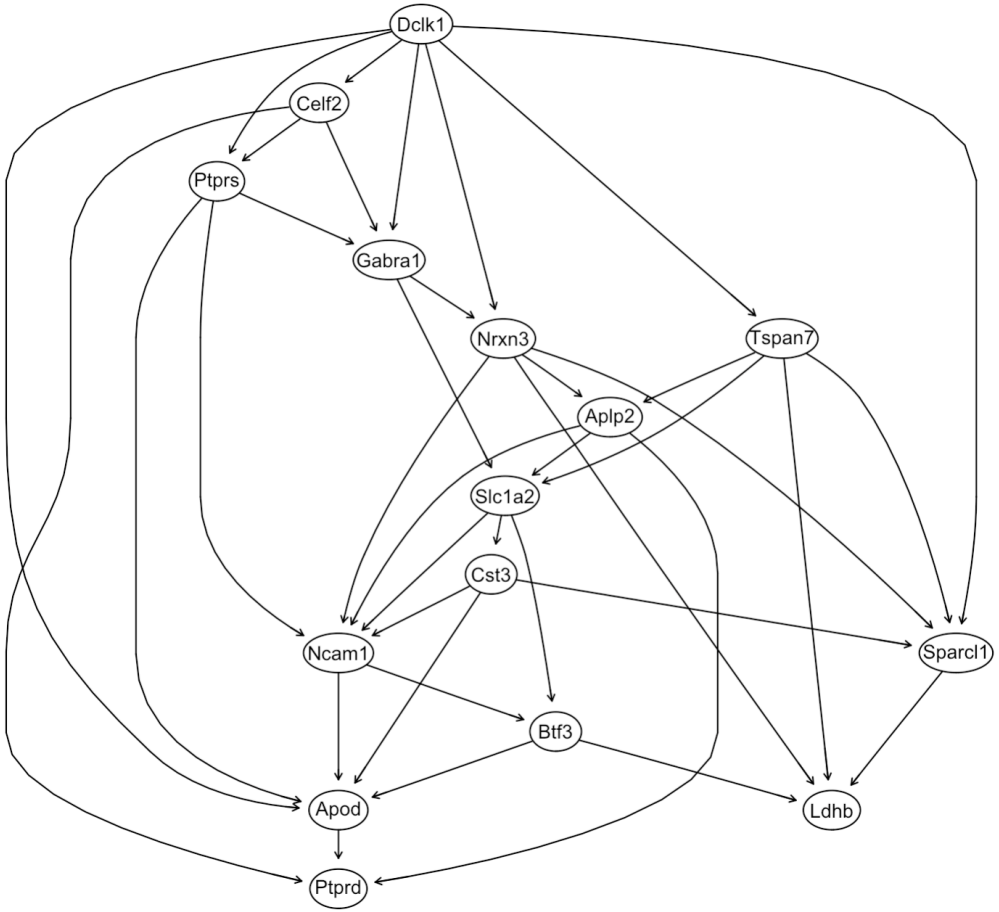
$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Demonstration on bnlearn in R

- bnlearn in R
  - Demonstrate the import of data
  - Data structure
  - Output structure
  - Demonstrate how the graph are calculated
  - Import in to Cytoscape for visualization

git clone
https://github.com/niaid/BESTversion/Gene_Regulatory_Networks

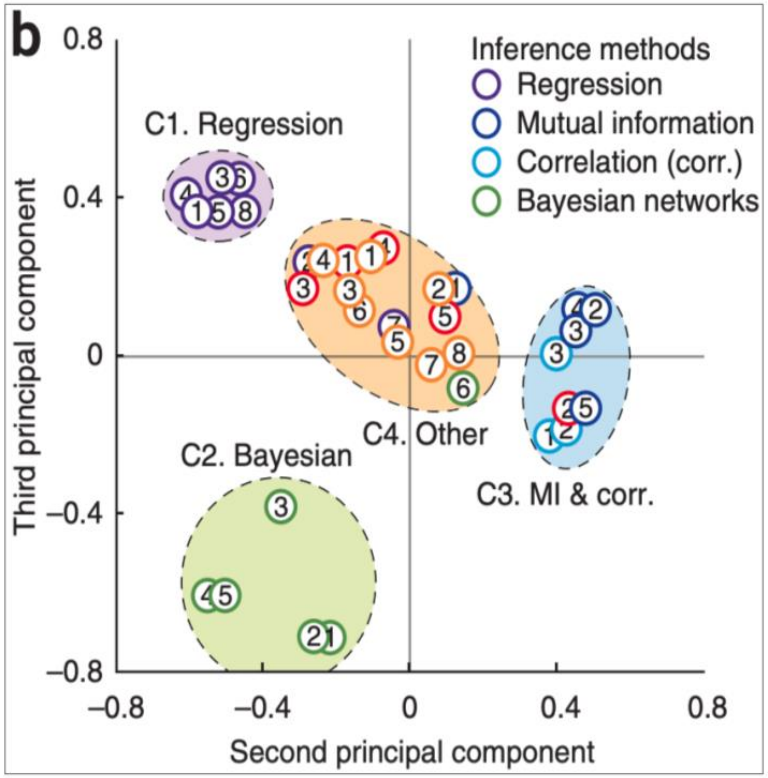
R out put



National Institute of Allergy and Infectious Diseases

NIAID

GENIE3—
GEne Network Inference with Ensemble of trees



NIH National Institute of Allergy and Infectious Diseases

OPEN ACCESS Freely available online

PLoS one

Inferring Regulatory Networks from Expression Data Using Tree-Based Methods

Vân Anh Huynh-Thu[1,2]*, Alexandre Irrthum[1,2], Louis Wehenkel[1,2], Pierre Geurts[1,2]

1 Department of Electrical Engineering and Computer Science, Systems and Modeling, University of Liège, Liège, Belgium, **2** GIGA-Research, Bioinformatics and Modeling, University of Liège, Liège, Belgium

Abstract

One of the pressing open problems of computational systems biology is the elucidation of the topology of genetic regulatory networks (GRNs) using high throughput genomic data, in particular microarray gene expression data. The Dialogue for Reverse Engineering Assessments and Methods (DREAM) challenge aims to evaluate the success of GRN inference algorithms on benchmarks of simulated data. In this article, we present GENIE3, a new algorithm for the inference of GRNs that was best performer in the DREAM4 *In Silico Multifactorial* challenge. GENIE3 decomposes the prediction of a regulatory network between p genes into p different regression problems. 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 or Extra-Trees. 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. In addition to performing well on the DREAM4 *In Silico Multifactorial* challenge simulated data, we show that GENIE3 compares favorably with existing algorithms to decipher the genetic regulatory network of *Escherichia coli*. It doesn't make any assumption about the nature of gene regulation, can deal with combinatorial and non-linear interactions, produces directed GRNs, and is fast and scalable. In conclusion, we propose a new algorithm for GRN inference that performs well on both synthetic and real gene expression data. The algorithm, based on feature selection with tree-based ensemble methods, is simple and generic, making it adaptable to other types of genomic data and interactions.

Citation: Huynh-Thu VA, Irrthum A, Wehenkel L, Geurts P (2010) Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. PLoS ONE 5(9): e12776. doi:10.1371/journal.pone.0012776

Editor: Mark Isalan, Center for Genomic Regulation, Spain

Received May 5, 2010; **Accepted** August 9, 2010; **Published** September 28, 2010

Funding: This work was partially funded by the Interuniversity Attraction Poles Programme (IAP P6/25 BIOMAGNET), initiated by the Belgian State, Science Policy Office, by the French Community of Belgium (ARC Biomod), and by the European Network of Excellence PASCAL2. VAHT is recipient of a Fonds pour la formation a la Recherche dans l'Industrie et dans l'Agriculture (F.R.I.A.) fellowship and PG is Research Associate of the Fonds de la Recherche Scientifique - Fonds National de la Recherche Scientifique (F.R.S.-FNRS), Belgium. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

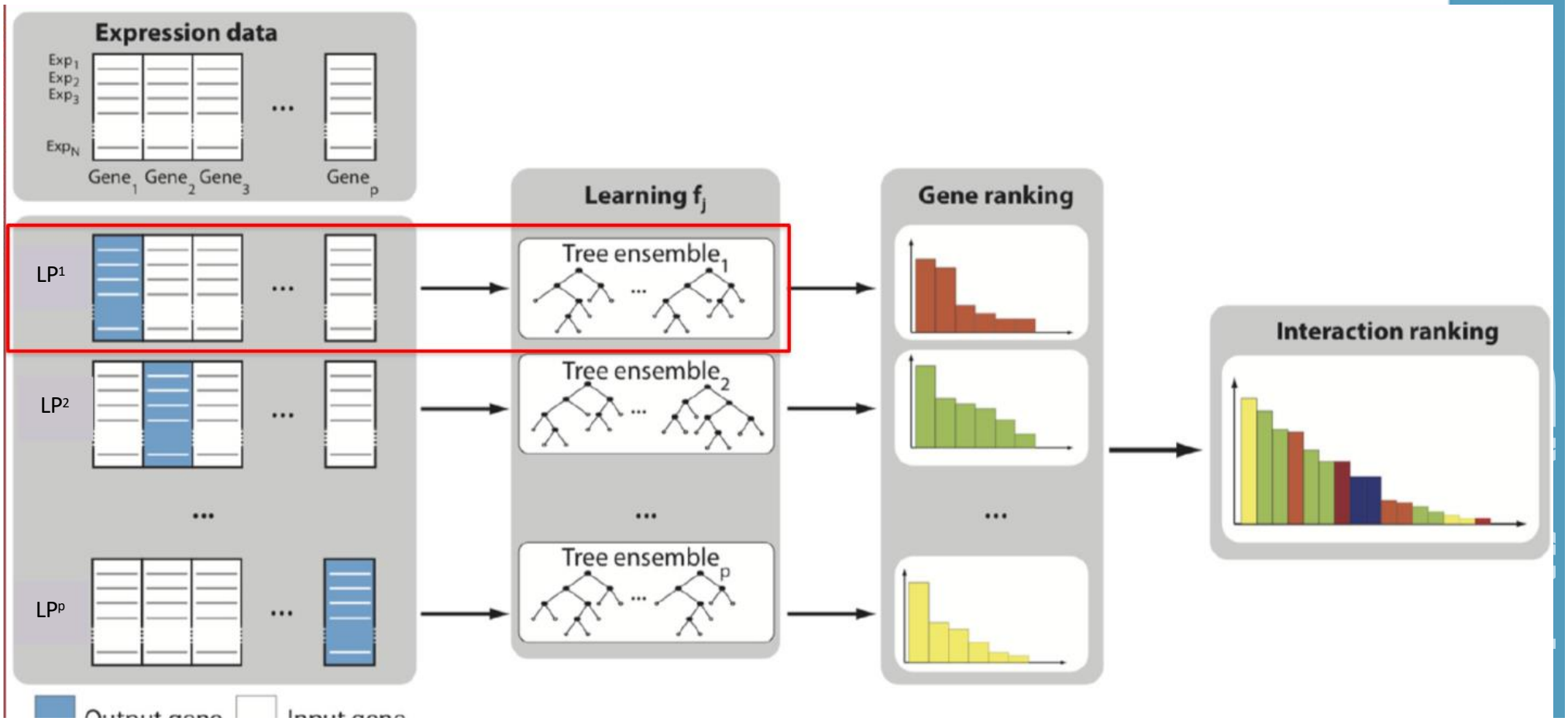
* E-mail: vahuynh@ulg.ac.be

NIAID

GENIE3-- GEne Network Inference with Ensemble of trees

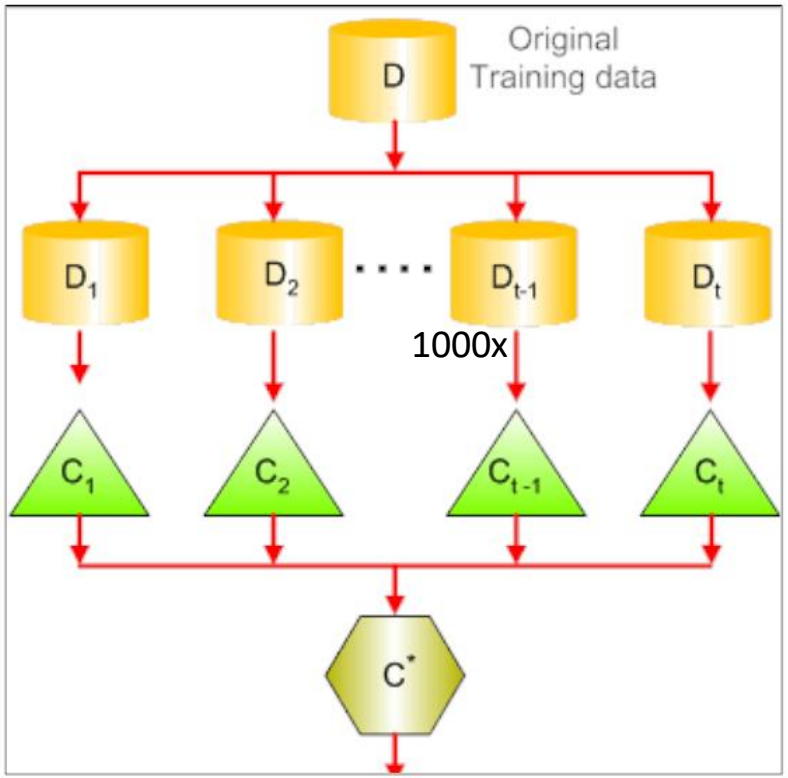
- Treat one gene a time as the target for regression.
- **Using bootstrapping/bagging to generate learning samples.**
- **Within each learning sample (LS), grow many regression trees, each provides an estimate of feature weight.**
- Average weights of each feature within the ensemble of trees to have a combined rank of feature importance of that gene.
- Combine all the feature-target interactions together. To have a rank of all gene—gene interaction throughout all genes in the list.

Bootstrapping/ ensemble of trees

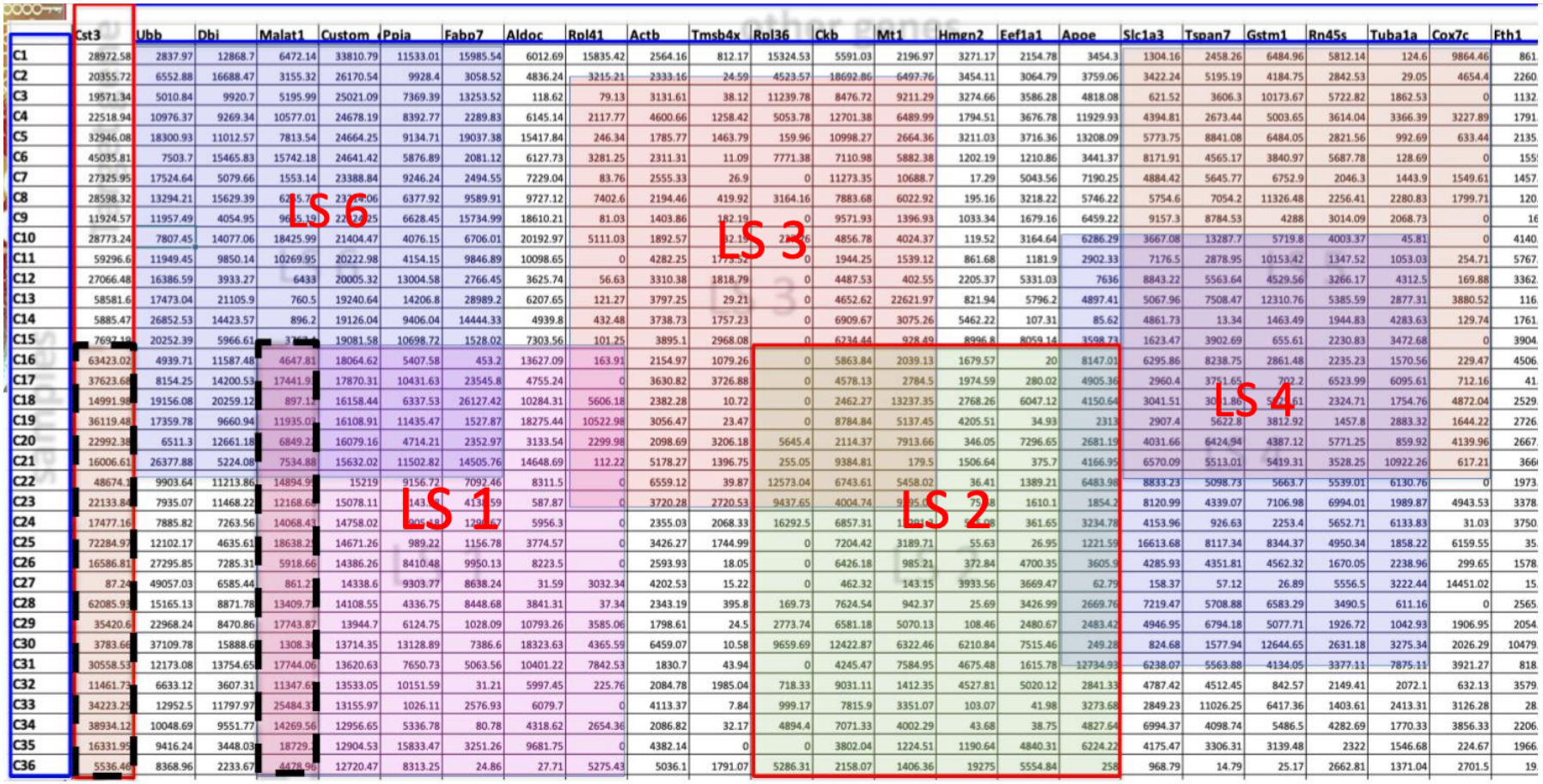


Building many small trees by bootstrapping/Bagging

- Question: how important is each other gene to the expression of target gene j
- From training data D, Derive multiple data sets/Learning sample (LS)
- From each data sets, build a a regression tree. To determine the importance of each other gene in determining gene j
- Combine all the feature importance to have a final one.



bootstrapping/Bagging and build/grow a regression tree with each learning sample (LS)



	Cst3	Ubb	Dbi	Malat1	Custom	Ppia	Fabp7	Aldoc	Rpl41	Actb	Tmsb4x	Rpl36	Ckb	Mt1	Hmgn2	Eef1a1	Apoe	Slc1a3	Tspan7	Gstm1	Rn45s	Tuba1a	Cox7c	Fth1
C1	28972.58	2837.97	12868.7	6472.14	33810.79	11533.01	15985.54	6012.69	15835.42	2564.16	812.17	15324.53	5591.03	2196.97	3271.17	2154.78	3454.3	1304.16	2458.26	6484.96	5812.14	124.6	9864.46	861
C2	20355.72	6552.88	16688.47	3155.32	26170.54	9928.4	3058.52	4836.24	3215.21	2333.16	24.59	4523.57	18692.86	6497.76	3454.11	3064.79	3759.06	3422.24	5195.19	4184.75	2842.53	29.05	4654.4	2260
C3	19571.34	5010.84	9920.7	5195.99	25021.09	7369.39	13253.52	118.62	79.13	3131.61	38.12	11239.78	8476.72	9211.29	3274.66	3586.28	4818.08	621.52	3606.3	10173.67	5722.82	1862.53	0	1132
C4	22518.94	10976.37	9269.34	10577.01	24678.19	8392.77	2289.83	6145.14	2117.77	4600.66	1258.42	5053.78	12701.38	6489.99	1794.51	3676.78	11929.93	4394.81	2673.44	5003.65	3614.04	3366.39	3227.89	1791
C5	32946.08	18300.93	11012.57	7813.54	24664.25	9134.71	19037.38	15417.84	246.34	1785.77	1463.79	159.96	10998.27	2664.36	3211.03	3716.36	13208.09	5773.75	8841.08	6484.05	2821.56	992.69	633.44	2135
C6	45035.81	7503.7	15465.83	15742.18	24641.42	5876.89	2081.12	6127.73	3281.25	2311.31	11.09	7771.38	7110.98	5882.38	1202.19	1210.86	3441.37	8171.91	4565.17	3840.97	5687.78	128.69	0	155
C7	27325.95	17524.64	5079.66	1553.14	23388.84	9246.24	2494.55	7229.04	83.76	2555.33	26.9	0	11273.35	10688.7	17.29	5043.56	7190.25	4884.42	5645.77	6752.9	2046.3	1443.9	1549.61	1457
C8	28598.32	13294.21	15629.39	6215.7	23744.06	6377.92	9589.91	9727.12	7402.6	2194.46	419.92	3164.16	7883.68	6022.92	195.16	3218.22	5746.22	5754.6	7054.2	11326.48	2256.41	2280.83	1799.71	120
C9	11924.57	11957.49	4054.95	9665.19	22272.25	6628.45	15734.99	18610.21	81.03	1403.86	182.19	0	9571.93	1396.93	1033.34	1679.16	6459.22	9157.3	8784.53	4288	3014.09	2068.73	0	16
C10	28773.24	7807.45	14077.06	18425.99	21404.47	4076.15	6706.01	20192.97	5111.03	1892.57	32.15	2[illegible]	4856.78	4024.37	119.52	3164.64	6286.29	3667.08	13287.7	5719.8	4003.37	45.81	0	4140
C11	59296.6	11949.45	9850.14	10269.95	20222.98	4154.15	9846.89	10098.65	0	4282.25	1773.52	0	1944.25	1539.12	861.68	1181.9	2902.33	7176.5	2878.95	10153.42	1347.52	1053.03	254.71	5767
C12	27066.48	16386.59	3933.27	6433	20005.32	13004.58	2766.45	3625.74	56.63	3310.38	1818.79	0	4487.53	402.55	2205.37	5331.03	7636	8843.22	5563.64	4529.56	3266.17	4312.5	169.88	3362
C13	58581.6	17473.04	21105.9	760.5	19240.64	14206.8	28989.2	6207.65	121.27	3797.25	29.21	0	4652.62	22621.97	821.94	5796.2	4897.41	5067.96	7508.47	12310.76	5385.59	2877.31	3880.52	116
C14	5885.47	26852.53	14423.57	896.2	19126.04	9406.04	14444.33	4939.8	432.48	3738.73	1757.23	0	6909.67	3075.26	5462.22	107.31	85.62	4861.73	13.34	1463.49	1944.83	4283.63	129.74	1761
C15	7692.19	20252.39	5966.61	3[illegible]	19081.58	10698.72	1528.02	7303.56	101.25	3895.1	2968.08	0	6234.44	928.49	8996.8	8059.14	3598.73	1623.47	3902.69	655.61	2230.83	3472.68	0	3904
C16	63423.02	4939.71	11587.48	4647.81	18064.62	5407.58	453.2	13627.09	163.91	2154.97	1079.26	0	5863.84	2039.13	1679.57	20	8147.01	6295.86	8238.75	2861.48	2235.23	1570.56	229.47	4506
C17	37623.68	8154.25	14200.53	17441.9[illegible]	17870.31	10431.63	23545.8	4755.24	0	3630.82	3726.88	0	4578.13	2784.5	1974.59	280.02	4905.36	2960.4	3751.65	702.2	6523.99	6095.61	712.16	41
C18	14991.98	19156.08	20259.12	897.12	16158.44	6337.53	26127.42	10284.31	5606.18	2382.28	10.72	0	2462.27	13237.35	2768.26	6047.12	4150.64	3041.51	3011.86	5[illegible]61	2324.71	1754.76	4872.04	2529
C19	36119.48	17359.78	9660.94	11935.0[illegible]	16108.91	11435.47	1527.87	18275.44	10522.98	3056.47	23.47	0	8784.84	5137.45	4205.51	34.93	2313	2907.4	5622.8	3812.92	1457.8	2883.32	1644.22	2726
C20	22992.38	6511.3	12661.18	6849.2[illegible]	16079.16	4714.21	2352.97	3133.54	2299.98	2098.69	3206.18	5645.4	2114.37	7913.66	346.05	7296.65	2681.19	4031.66	6424.94	4387.12	5771.25	859.92	4139.96	2667
C21	16006.61	26377.88	5224.08	7534.88	15632.02	11502.82	14505.76	14648.69	112.22	5178.27	1396.75	255.05	9384.81	179.5	1506.64	375.7	4166.95	6570.09	5513.01	5419.31	3528.25	10922.26	617.21	366
C22	48674.1[illegible]	9903.64	11213.86	14894.9[illegible]	15219	9156.72	7092.46	8311.5	0	6559.12	39.87	12573.04	6743.61	5458.02	36.41	1389.21	6483.98	8833.23	5098.73	5663.7	5539.01	6130.76	0	1973
C23	22133.84	7935.07	11468.22	12168.6[illegible]	15078.11	143.[illegible]	413[illegible]59	587.87	0	3720.28	2720.53	9437.65	4004.74	9[illegible]35.[illegible]	75[illegible]8	1610.1	1854.2	8120.99	4339.07	7106.98	6994.01	1989.87	4943.53	3378
C24	17477.16	7885.82	7263.56	14068.41	14758.02	8[illegible]5[illegible]	12[illegible]67	5956.3	0	2355.03	2068.33	16292.5	6857.31	1[illegible]91.3	9[illegible]08	361.65	3234.78	4153.96	926.63	2253.4	5652.71	6133.83	31.03	3750
C25	72284.97	12102.17	4635.61	18638.2[illegible]	14671.26	989.22	1156.78	3774.57	0	3426.27	1744.99	0	7204.42	3189.71	55.63	26.95	1221.59	16613.68	8117.34	8344.37	4950.34	1858.22	6159.55	35
C26	16586.81	27295.85	7285.31	5918.66	14386.26	8410.48	9950.13	8223.5	0	2593.93	18.05	0	6426.18	985.21	372.84	4700.35	3605.9	4285.93	4351.81	4562.32	1670.05	2238.96	299.65	1578
C27	87.24	49057.03	6585.44	861.2	14338.6	9303.77	8638.24	31.59	3032.34	4202.53	15.22	0	462.32	143.15	3933.56	3669.47	62.79	158.37	57.12	26.89	5556.5	3222.44	14451.02	15
C28	62085.93	15165.13	8871.78	13409.7[illegible]	14108.55	4336.75	8448.68	3841.31	37.34	2343.19	395.8	169.73	7624.54	942.37	25.69	3426.99	2669.76	7219.47	5708.88	6583.29	3490.5	611.16	0	2565
C29	35420.6	22968.24	8470.86	17743.87	13944.7	6124.75	1028.09	10793.26	3585.06	1798.61	24.5	2773.74	6581.18	5070.13	108.46	2480.67	2483.42	4946.95	6794.18	5077.71	1926.72	1042.93	1906.95	2054
C30	3783.66	37109.78	15888.6[illegible]	1308.3[illegible]	13714.35	13128.89	7386.6	18323.63	4365.59	6459.07	10.58	9659.69	12422.87	6322.46	6210.84	7515.46	249.28	824.68	1577.94	12644.65	2631.18	3275.34	2026.29	10479
C31	30558.53	12173.08	13754.65	17744.06	13620.63	7650.73	5063.56	10401.22	7842.53	1830.7	43.94	0	4245.47	7584.95	4675.48	1615.78	12734.93	6238.07	5563.88	4134.05	3377.11	7875.11	3921.27	818
C32	11461.73	6633.12	3607.31	11347.6[illegible]	13533.05	10151.59	31.21	5997.45	225.76	2084.78	1985.04	718.33	9031.11	1412.35	4527.81	5020.12	2841.33	4787.42	4512.45	842.57	2149.41	2072.1	632.13	3579
C33	34223.25	12952.5	11797.97	25484.3[illegible]	13155.97	1026.11	2576.93	6079.7	0	4113.37	7.84	999.17	7815.9	3351.07	103.07	41.98	3273.68	2849.23	11026.25	6417.36	1403.61	2413.31	3126.28	28
C34	38934.12	10048.69	9551.77	14269.56	12956.65	5336.78	80.78	4318.62	2654.36	2086.82	32.17	4894.4	7071.33	4002.29	43.68	38.75	4827.64	6994.37	4098.74	5486.5	4282.69	1770.33	3856.33	2206
C35	16331.95	9416.24	3448.03	18729.[illegible]	12904.53	15833.47	3251.26	9681.75	0	4382.14	0	0	3802.04	1224.51	1190.64	4840.31	6224.22	4175.47	3306.31	3139.48	2322	1546.68	224.67	1966
C36	5536.46	8368.96	2233.67	4478.96	12720.47	8313.25	24.86	27.71	5275.43	5036.1	1791.07	5286.31	2158.07	1406.36	19275	5554.84	258	968.79	14.79	25.17	2662.81	1371.04	2701.5	19

LS 6

LS 3

LS 1

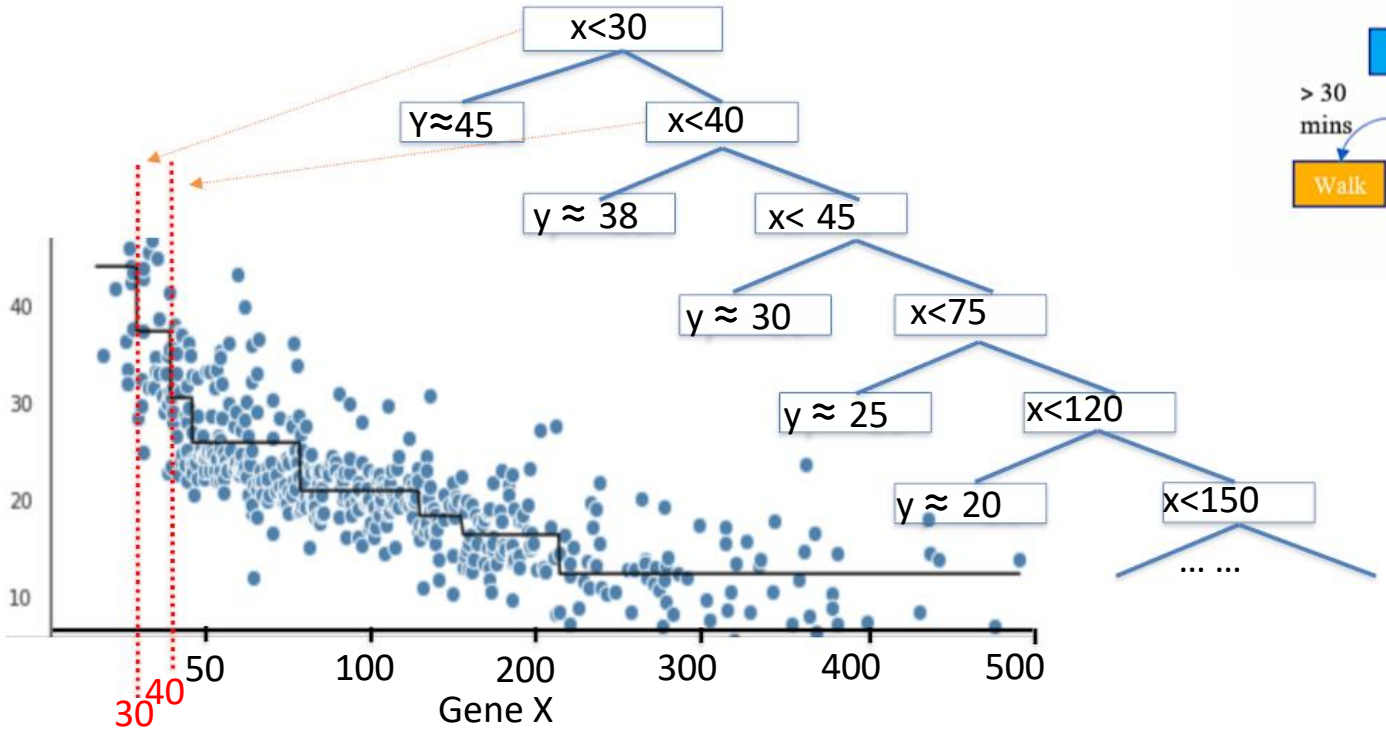
LS 2

LS 4

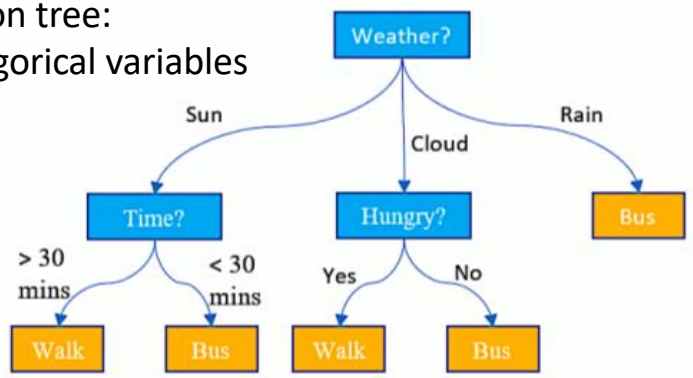
NIAD

Within each learning sample, grow a regression tree

A regression tree:
For continuous variables

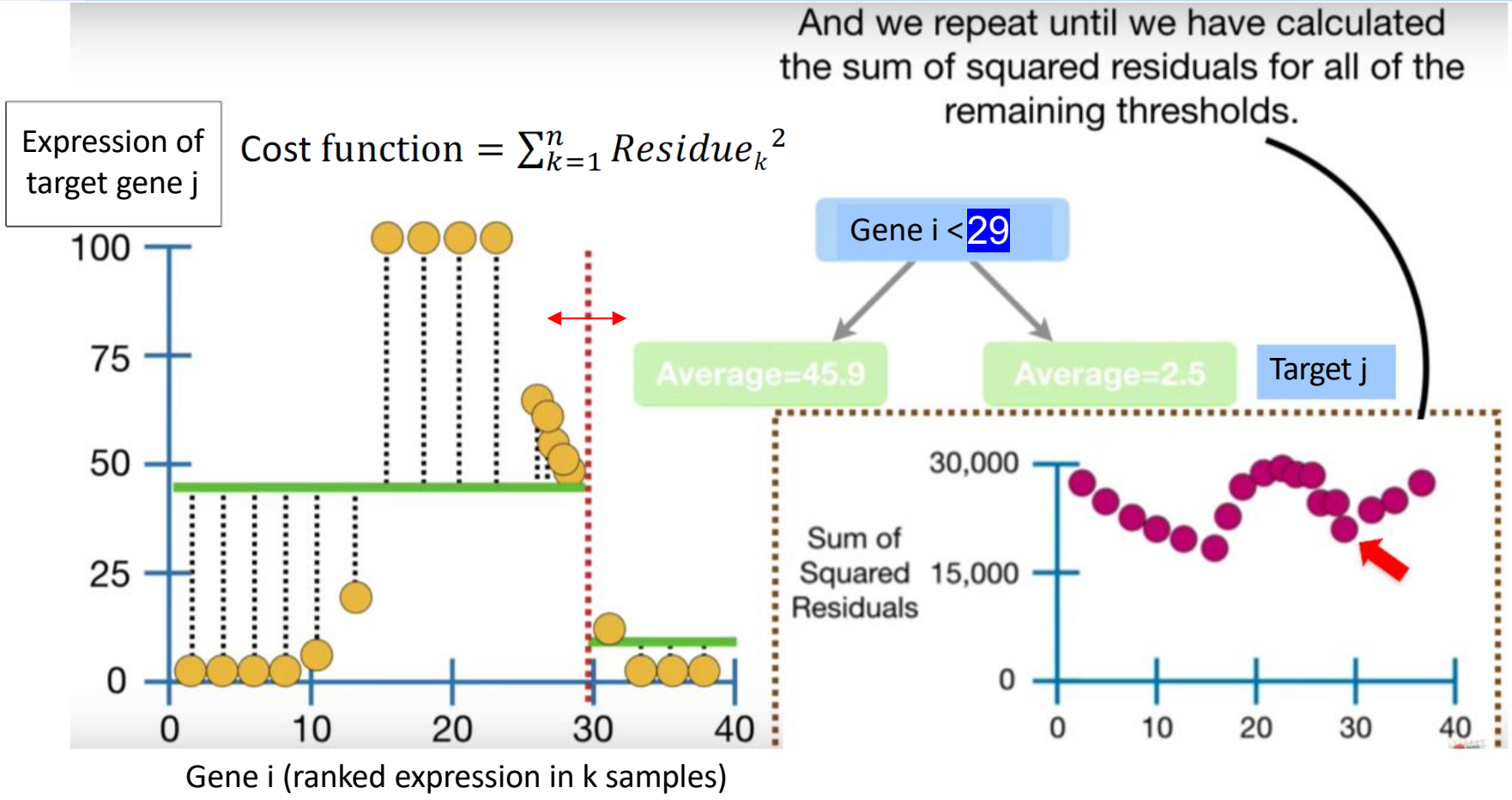


A decision tree:
For categorical variables

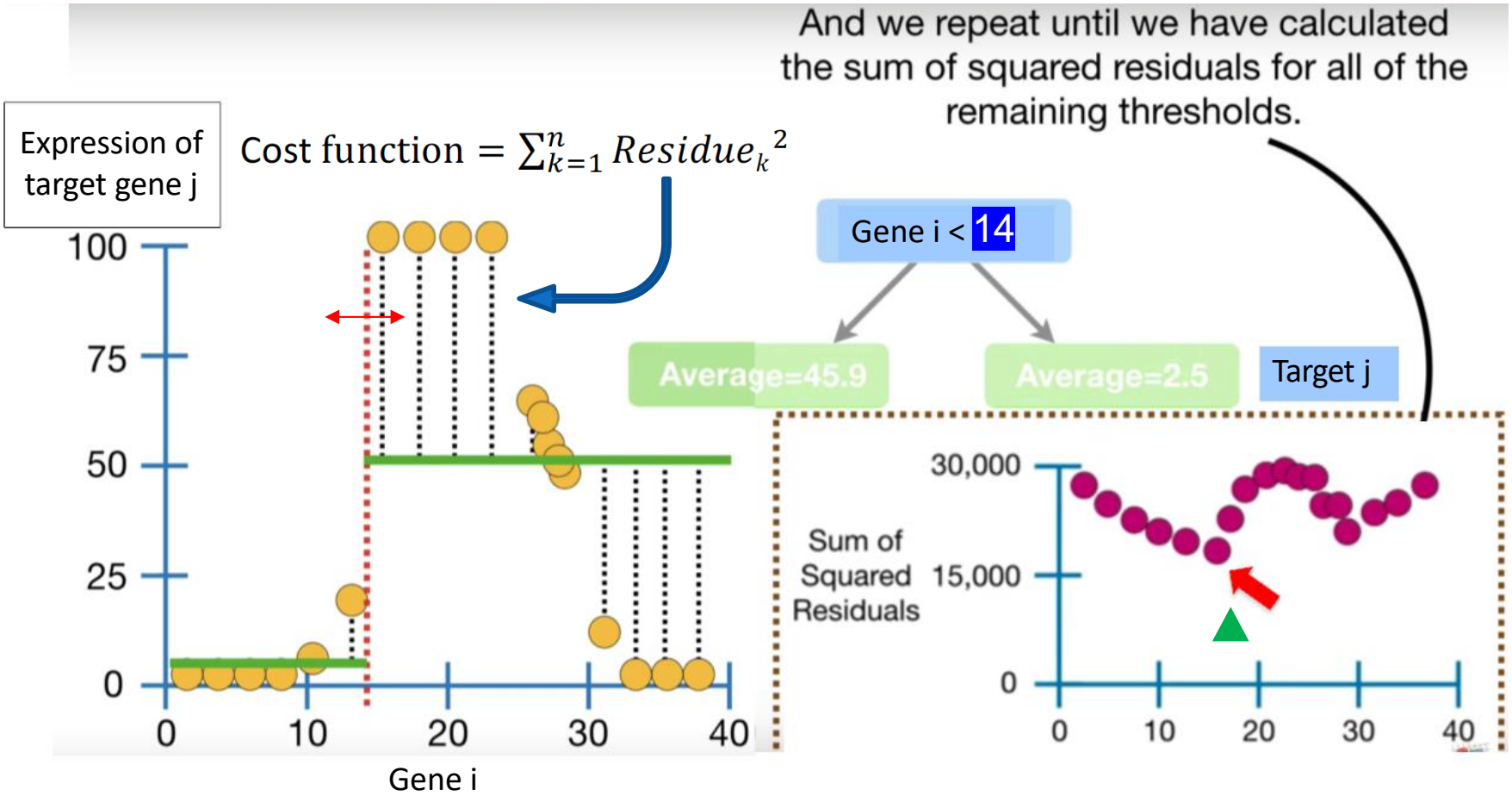


How about non-linear situations…

How a regression tree works

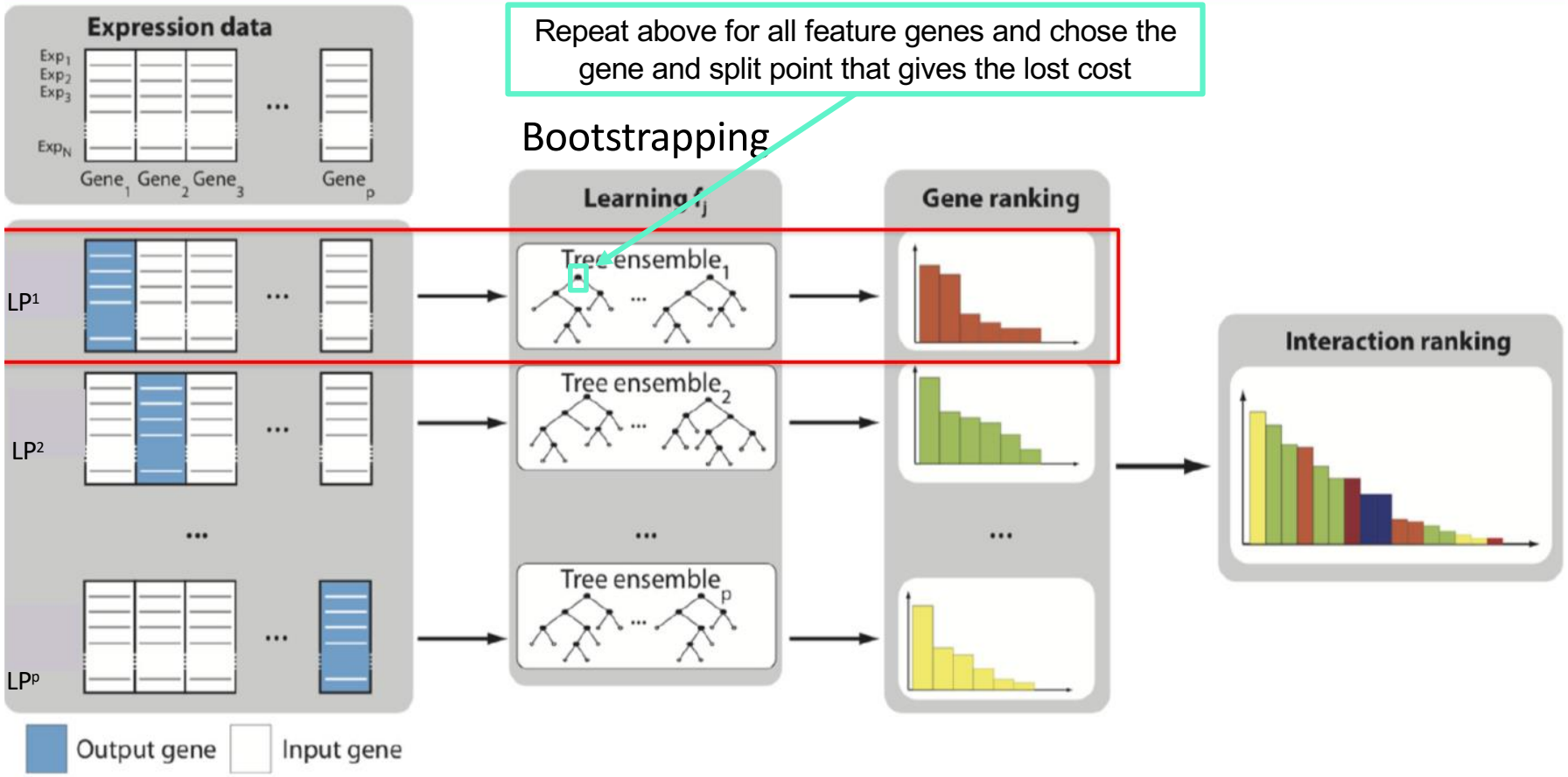


How a regression tree works



NIAID

Bootstrapping / ensemble of trees



Build a tree from multiple variables

Grow a regression tree

1. For a given target gene j
2. Iterate through each of the feature gene. Chose the one that gives the lowest cost.
3. For each of the branch, do the same as above to determine next branch points.

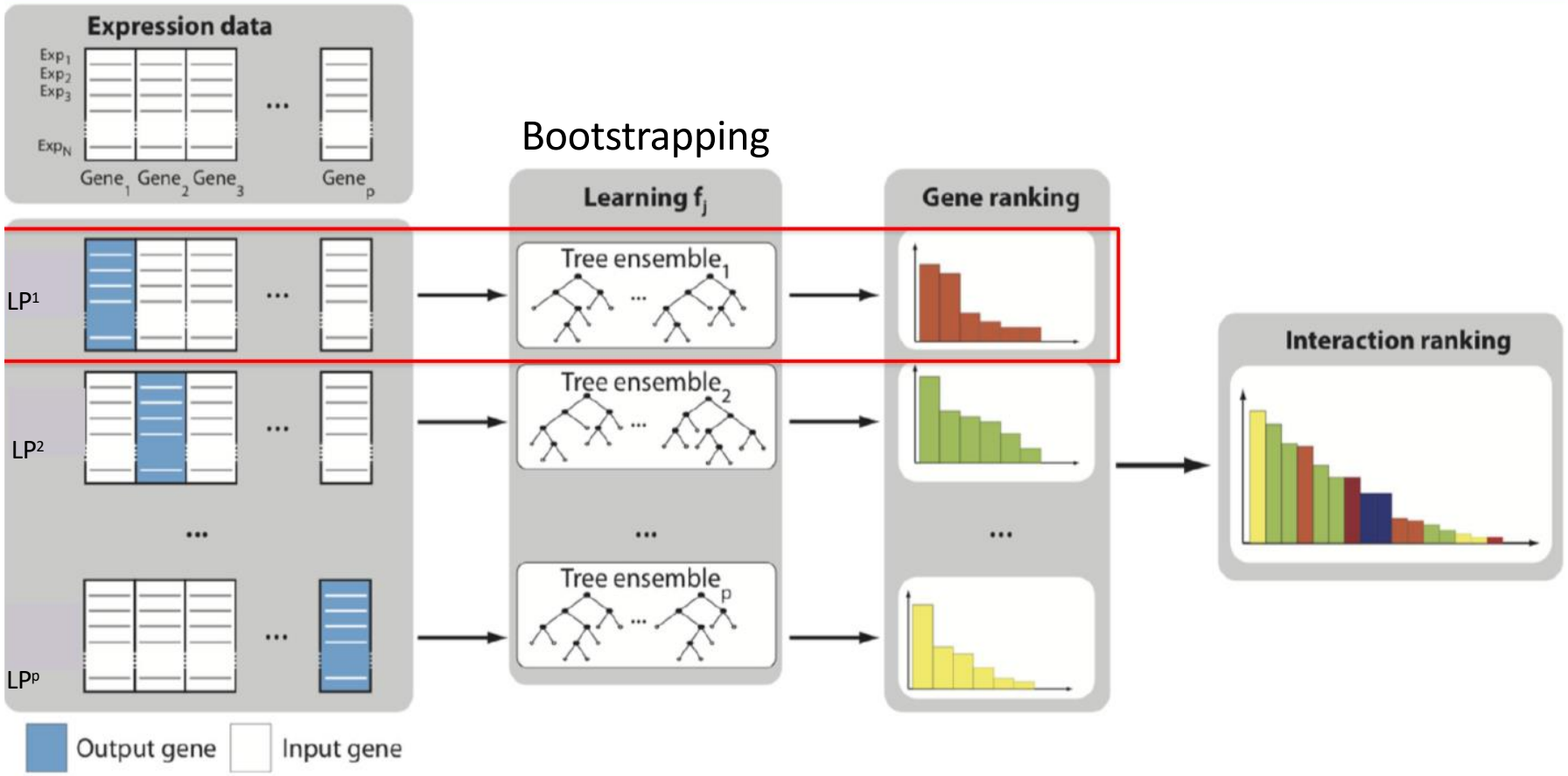
Determining feature importance

1. In a regression tree, **sum up the reduction of variances due to splitting** by each specific gene in all branches. That is the score for the feature importance.
2. For all trees, average feature importance for each feature. Rank the feature genes by scales.

Combine feature importance for all the target genes as a rank of all interactions.

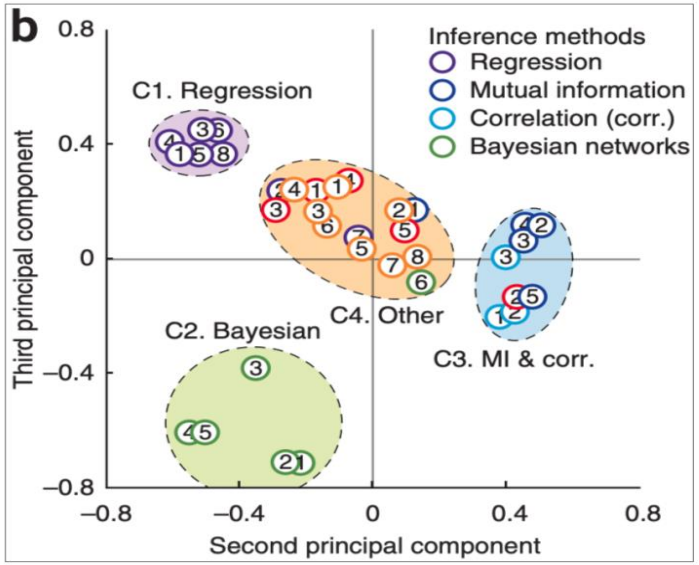
1. For each of the target gene, repeat through all the steps above, until all the feature importance for all target genes are determined.
2. Combine all interactions and rank them by the scores.

Bootstrapping / ensemble of trees



Discussion on Genie3

- Efficiently handle large dataset.
  - Parallel processing
  - Implemented in R and Python
- Has been joined with cis-Target to identify TF-target regulation in single-cell data.



Method 4, iRegulon – prediction of upstream transcriptional factors based on promoter sequences

OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

iRegulon: From a Gene List to a Gene Regulatory Network Using Large Motif and Track Collections

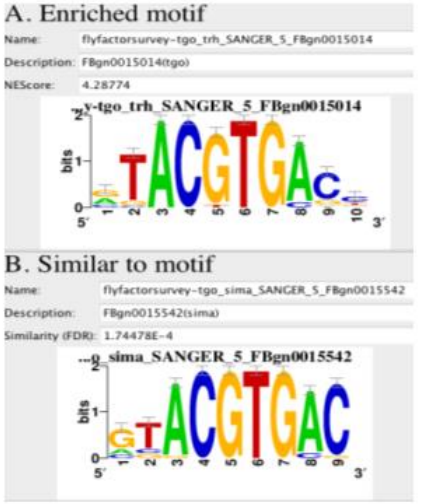
Rekin's Janky[1☯], Annelien Verfaillie[1☯], Hana Imrichová[1], Bram Van de Sande[1], Laura Standaert[2,3], Valerie Christiaens[1], Gert Hulselmans[1], Koen Herten[1], Marina Naval Sanchez[1], Delphine Potier[1], Dmitry Svetlichnyy[1], Zeynep Kalender Atak[1], Mark Fiers[3], Jean-Christophe Marine[2,3], Stein Aerts[1]*

1 Laboratory of Computational Biology, KU Leuven Center for Human Genetics, Leuven, Belgium, **2** Laboratory for Molecular Cancer Biology, KU Leuven Center for Human Genetics, Leuven, Belgium, **3** VIB Center for the Biology of Disease, Laboratory for Molecular Cancer Biology, Leuven, Belgium

Abstract

Identifying master regulators of biological processes and mapping their downstream gene networks are key challenges in systems biology. We developed a computational method, called iRegulon, to reverse-engineer the transcriptional regulatory network underlying a co-expressed gene set using *cis*-regulatory sequence analysis. iRegulon implements a genome-wide ranking-and-recovery approach to detect enriched transcription factor motifs and their optimal sets of direct targets. We increase the accuracy of network inference by using very large motif collections of up to ten thousand position weight matrices collected from various species, and linking these to candidate human TFs via a *motif2TF* procedure. We validate iRegulon on gene sets derived from ENCODE ChIP-seq data with increasing levels of noise, and we compare iRegulon with existing motif discovery methods. Next, we use iRegulon on more challenging types of gene lists, including microRNA target sets, protein-protein interaction networks, and genetic perturbation data. In particular, we over-activate p53 in breast cancer cells, followed by RNA-seq and ChIP-seq, and could identify an extensive up-regulated network controlled directly by p53. Similarly we map a repressive network with no indication of direct p53 regulation but rather an indirect effect via E2F and NFY. Finally, we generalize our computational framework to include regulatory tracks such as ChIP-seq data and show how motif and track discovery can be combined to map functional regulatory interactions among co-expressed genes. iRegulon is available as a Cytoscape plugin from http://iregulon.aertslab.org.

iRegulon/cis-Target

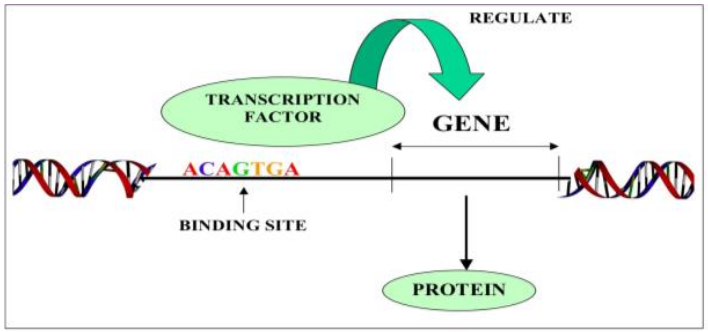


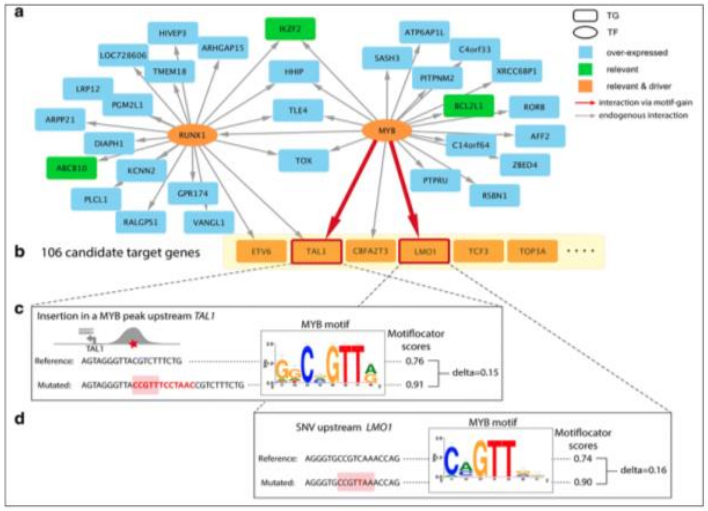
Using database to predict upstream transcriptional regulators

NIAID

Introduction to iRegulon

- Transcriptional activity correlates with TF binding on promoter of target genes.
- One transcriptional factor activates a group of genes to achieve a biological function.
- TF expression increases it activity.
- Transcriptional factor has conserved binding sequence/motif.
- Many of the binding sequences have been validated.





Source of the TFBS/Motif database

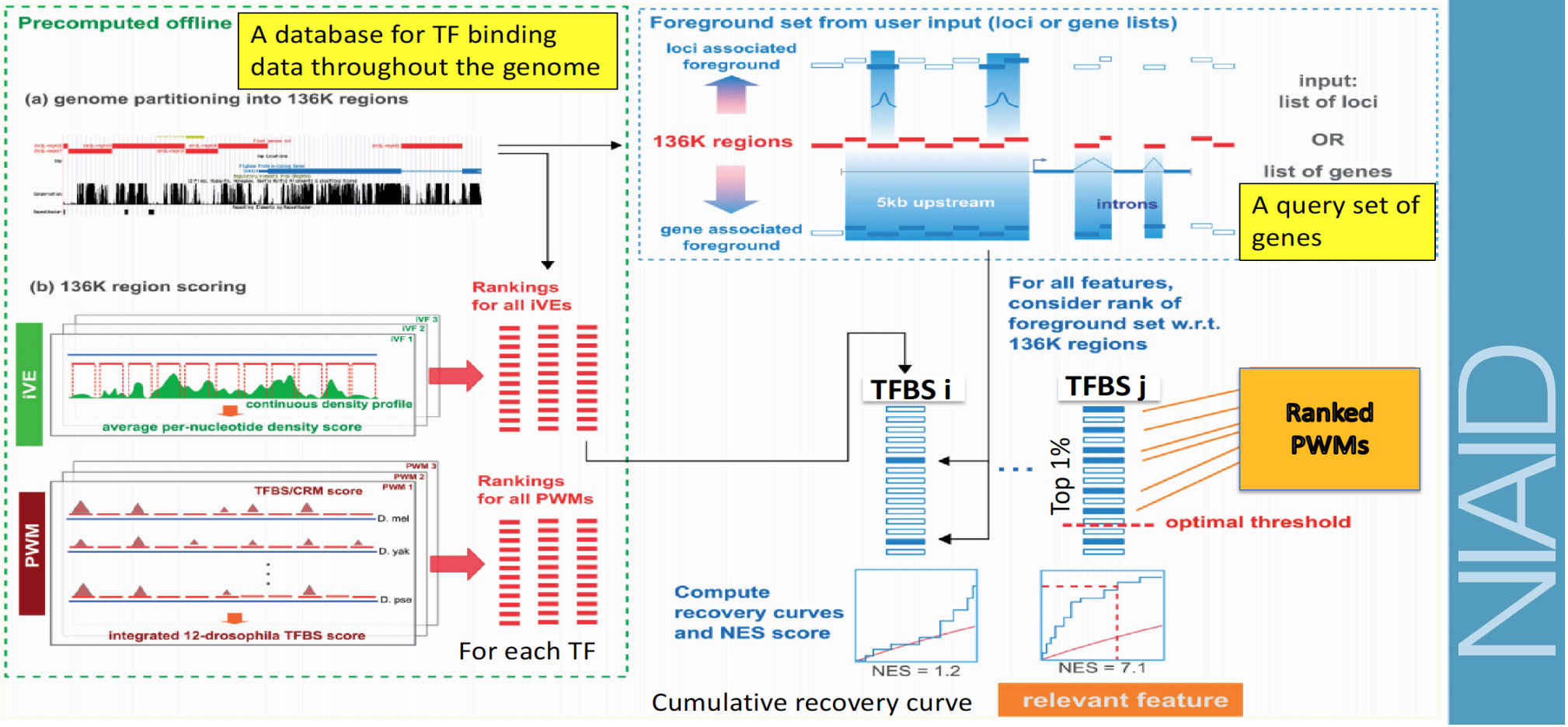
Table 1. Description of the motif and track collections used.

Source	Organism(s)	Type of motif	# motifs "6K"	# motifs "10K"	# tracks "1K ChIP"
Elemento [73]	Drosophila	Predicted (conserved)[a]	371	371	-
FlyFactorSurvey [75]	Drosophila	B-1H, others (e.g., FlyReg)	614	652	-
hPDI [77]	Human	Experimental	437	437	-
Jaspar [21]	Multiple species	Curated	1315	1315	-
SelexConsensus [76]	Drosophila	Curated (FlyReg)	38	38	-
Stark [74]	Drosophila	Predicted (conserved)[a]	228	228	-
Tiffin [76]	Drosophila	Predicted (gene sets)[a]	120	120	-
TRANSFAC PUBLIC [5]	Multiple species	Curated, ChIP-chip	398	398	-
TRANSFAC PRO [5]	Multiple species	Curated, ChIP-chip	1153	1850	-
YetFasco [78]	Yeast	Uniprobe, Curated, ChIP-chip	1709	1709	-
ENCODE [79]	Human	Predicted (from DHS)[a]	-	683	-
Factorbook [46]	Human	ENCODE ChIP-Seq motifs	-	79	-
Taipale [132]	Human, Mouse	HT-Selex	-	820	-
iDMMPMM [133]	Human	footprints, Selex, b1h, peaks	-	39	-
SwissRegulon [134]	Human	Curated	-	190	-
Wolfe [135]	Drosophila	ZFP motifs	-	36	-
HOMER [116]	Multiple species	ChIP-Seq Motifs, others (e.g. ENCODE)	-	1865	-
Dimers [136]	Human	Predicted dimers	-	603	-
ENCODE ChIP-Seq [23]	Human	-	-	-	999
Taipale ChIP-Seq [24]	Human	-	-	-	117
p53 and control ChIP-Seq (this study)	Human	-	-	-	2
Total			**6383**	11611 (**9713** nr)	**1118**

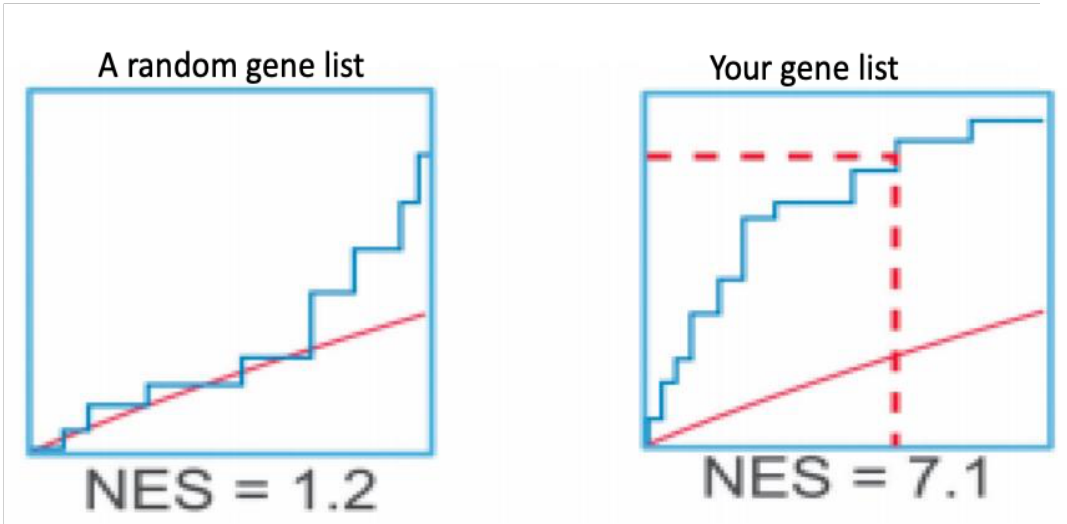
[a]Orphan motifs (unknown TFs).
nr = non-redundant.
doi:10.1371/journal.pcbi.1003731.t001

NIAID

How does cisTarget Work?

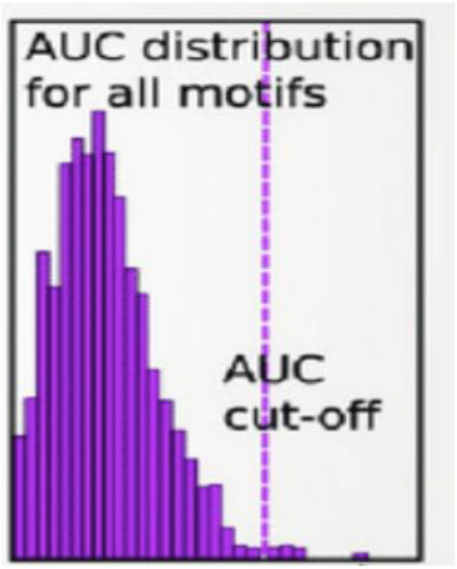


Normalized Enrichment Scores



For all TFBS/CRMs, Area Under the Curve (AUC) is calculated.
For top 1% of the rank. NES were calculated.

$$NES = \frac{AUC - AUCmean}{AUCstd}$$



Ranking of likelihood of all PWMs in all 137k loci, as a target for a TFBS.
(The ranking is based on a combination of motif clustering and comparative genomics.)

NIAID

Demo on iRegulon

- How to import the gene list to Cytoscape
- Set parameters for iRegulon to run
- Interpret the output

git clone https://github.com/niaid/BESTversion/Gene_Regulatory_Networks

Pro and cons

- Pros
  - Easy to generate hypothesis
  - Based on many coordinated genes in your data
  - Allow generation of a rich hypothesis
    - Binding sequence ---mutagenesis to test the TF binding.
    - Mechanism-- PCR to test the regulation of gene expression by the TF
    - Well informed on the biology as TF function is relatively well annotated
- Cons
  - The binding is mainly predicted by sequence enrichment, not guaranteed
  - Binding is dependent not only on sequence but also on DNA/histone modifications.
  - multiple TFs can bind to the same conserved sequence.

Integrate GENIE3 co-expression module detection with cis-Target detection of TF regulation

SCENIC: single-cell regulatory network inference and clustering

Sara Aibar[1,2], Carmen Bravo González-Blas[1,2], Thomas Moerman[3,4], Vân Anh Huynh-Thu[5], Hana Imrichova[1,2], Gert Hulselmans[1,2], Florian Rambow[6,7], Jean-Christophe Marine[6,7], Pierre Geurts[5], Jan Aerts[3,4], Joost van den Oord[8], Zeynep Kalender Atak[1,2], Jasper Wouters[1,2,8] & Stein Aerts[1,2]

We present SCENIC, a computational method for simultaneous gene regulatory network reconstruction and cell-state identification from single-cell RNA-seq data (http://scenic.aertslab.org). On a compendium of single-cell data from tumors and brain, we demonstrate that *cis*-regulatory analysis can be exploited to guide the identification of transcription factors and cell states. SCENIC provides critical biological insights into the mechanisms driving cellular heterogeneity.

The transcriptional state of a cell emerges from an underlying gene regulatory network (GRN) in which a limited number of transcription factors (TFs) and cofactors regulate each other and their downstream target genes. Recent advances in single-cell transcriptome profiling have provided exciting opportunities for high-resolution identification of transcriptional states and of transitions between states—for example, during differentiation[1,2]. Statistical techniques and bioinformatics methods that are optimized for single-cell RNA-seq have led to new biological insights[3], but it is still unclear whether specific and robust GRNs underlying stable cell states can be determined. This may indeed be challenging given that at the single-cell level, gene expression may be partially disconnected from the dynamics of TF inputs on account of stochastic variation of gene expression from tran-

and thus optimize the discovery and characterization of cell states. To this end, we developed single-cell regulatory network inference and clustering (SCENIC) to map GRNs and then identify stable cell states by evaluating the activity of the GRNs in each cell. The SCENIC workflow consists of three steps (**Fig. 1a**, **Supplementary Fig. 1** and see Online Methods). In the first step, sets of genes that are coexpressed with TFs are identified using GENIE3 (ref. 8) (**Supplementary Fig. 1a**). Since the GENIE3 modules are only based on coexpression, they may include many false positives and indirect targets. To identify putative direct-binding targets, each coexpression module is subjected to *cis*-regulatory motif analysis using RcisTarget (**Supplementary Fig. 1b** and see Online Methods). Only modules with significant motif enrichment of the correct upstream regulator are retained, and they are pruned to remove indirect targets lacking motif support. We refer to these processed modules as regulons.

As part of SCENIC, we developed the AUCell algorithm to score the activity of each regulon in each cell (**Supplementary Figs. 1c** and **2**, and see Online Methods). For a given regulon, comparing AUCell scores across cells makes it possible to identify which cells have significantly higher subnetwork activity. The resulting binary activity matrix has 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 (**Supplementary Fig. 3**).

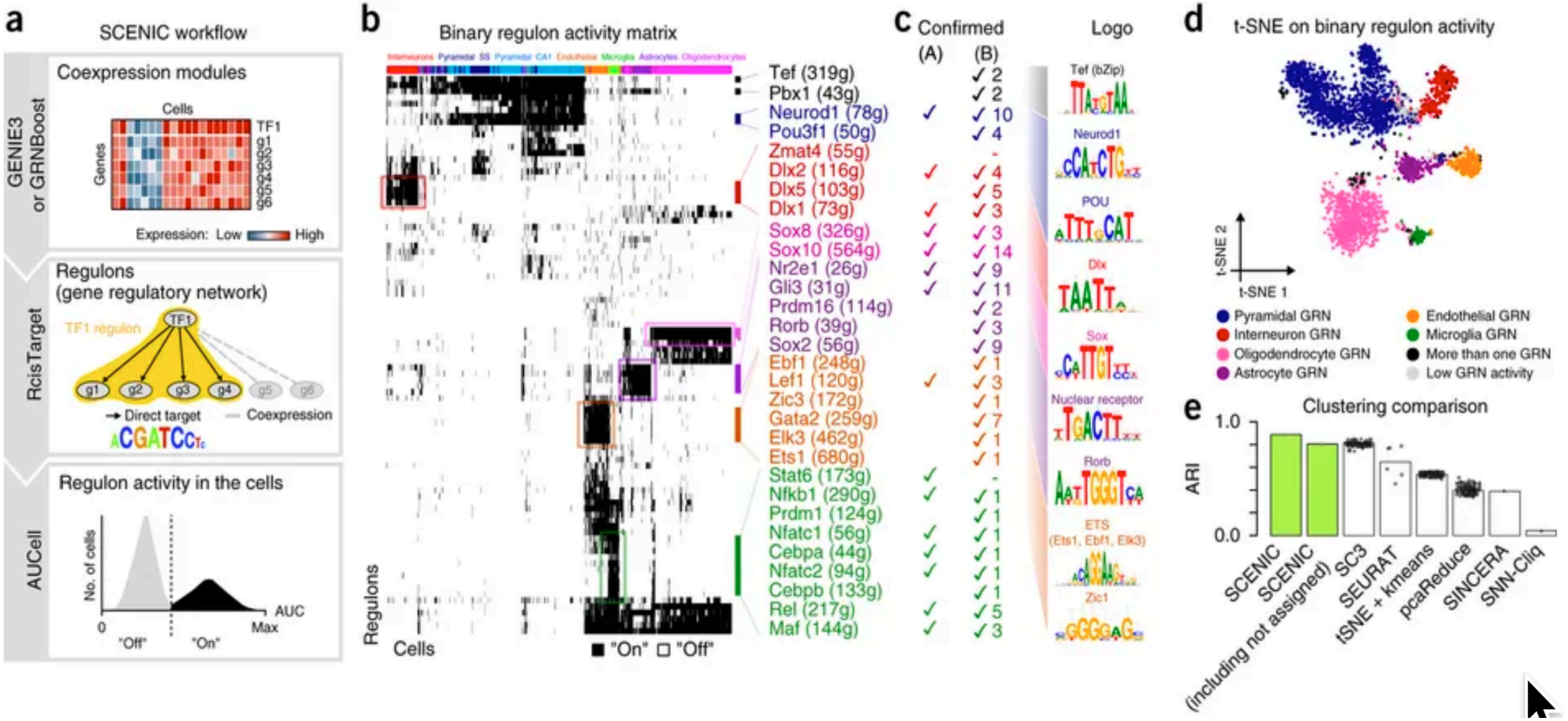
To evaluate the performance of SCENIC, we applied it to an scRNA-seq data set with well-known cell types from the adult mouse brain[9] (**Fig. 1b–e**). This analysis provided 151 regulons—out of 1,046 initial coexpression modules—with significantly enriched motifs for the corresponding TFs (7% of the initial TFs). Scoring regulon activity for each cell revealed the expected cell types (**Fig. 1d,e**) alongside a list of potential master regulators for each cell type (e.g., the microglia network in **Supplementary Fig. 4**). Clustering by cell type (overall sensitivity of 0.88, specificity of 0.99, and adjusted Rand index (ARI) > 0.80) is more accurate than many dedicated single-cell clustering methods[10].

To assess the robustness of SCENIC, we reanalyzed the mouse brain data: the full data set; samples of 100 randomly

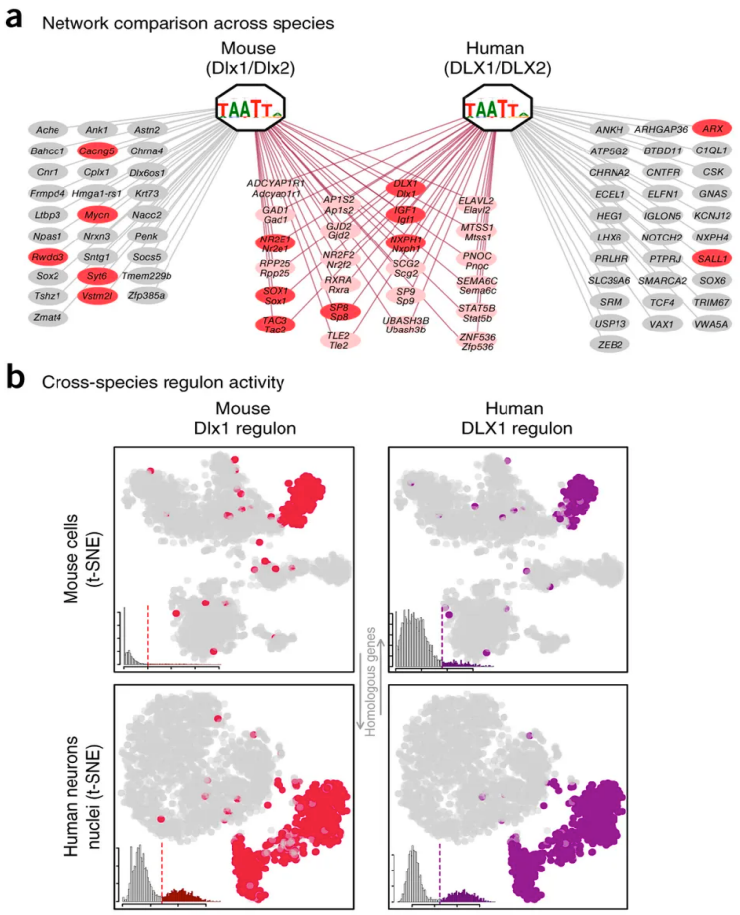
NIH National Institute of Allergy and Infectious Diseases

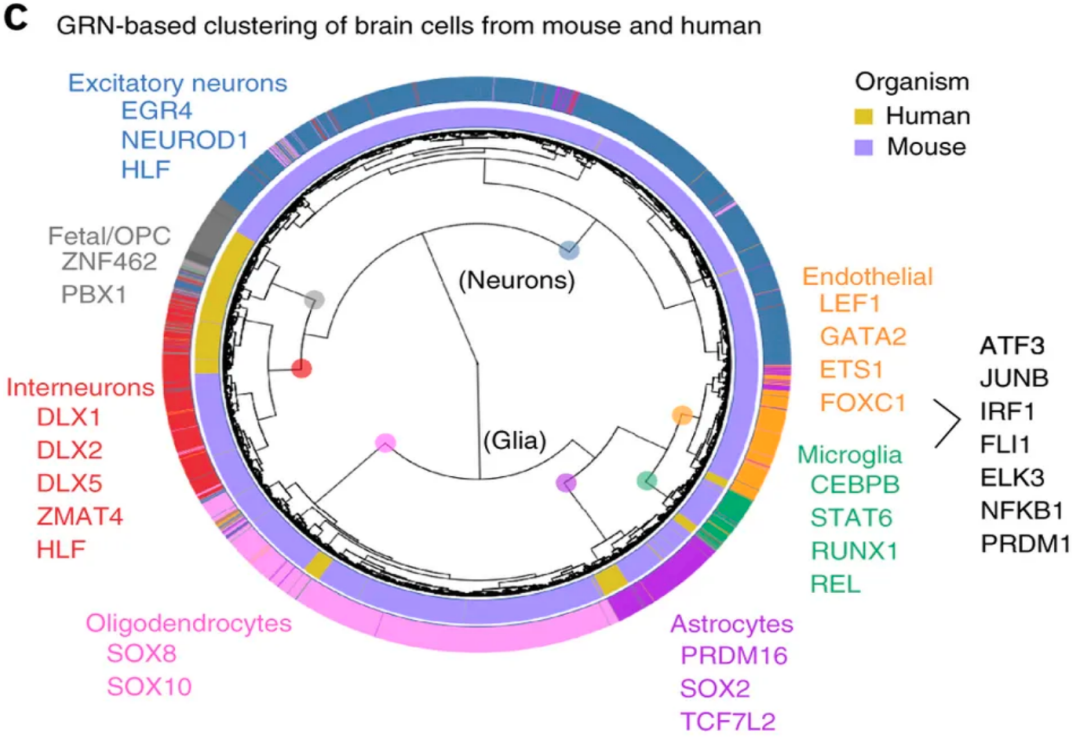
NIAID

SCENIC workflow and results



What do you achieve?





NIAID

Running SCENIC

- Run the installation may spent 2 hrs.
- Convert files to loom files.
- Download the cis-target file may take a long time. And how to set up these files before hand?
  - 1.1 G for mm9 mouse. Take 30 minutes to download.
- A dry run before your own data.

git clone https://github.com/niaid/Gene_Regulatory_Networks

A side note: Object Oriented Programing in R

- R is a procedure oriented programing language.
  - A <- 4
  - B <- A+5
  - General/defined functions: B<- function(A)
- Many of the genomic R packages /Bioconductor packages uses OOP concepts in R.
  - S3, S4, R5, R6 objects in R are object oriented, to avoid to have too many environment variables.
  - Each has slots identified by ”@” operator, and each slot can have “$” fields.
  - Stored values are extracted by custom defined methods
  - Values can be mutated by applied methods and return back to its slot
  - If A is an OOP object, Function (A) will change some defined variable in certain slot in A.

Object-oriented programming (OOP)– internal structure of an S4 object: scenicOptions

@inputDatabaseInfo

getDatabaseInfo(scenicOptions, "org")

```
$org
[1] "mgi"
$datasetTitle
[1] ""
$cellInfo
[1] "int/cellInfo.Rds"
$colVars
[1] "int/colVars.Rds" NA
$int_01
[1] "int/cellColorNgenes.Rds" NA
```

OOP objects in R

- Commonly used in genomic analysis
- Classified to S3,S4, R5, R6 objects
- Customized names for class
- Structure of S4 objects
  - @ slots
  - $ separate fields within a slot
  - Manually defined Methods that operation on the object

@settings

getSettings(scenicOptions, "nCores")
getDatabases(scenicOptions)
getDbAnnotation(scenicOptions)

```
$dbs
10kb "mm9-tss-centered-10kb-7species.mc9nr.feather"
$dbDir
[1] "./cisTarget_databases"
$db_mcVersion
[1] "v9"
$verbose
[1] TRUE
$nCores
[1] 10
$seed
[1] 123
$devType
[1] "pdf"
$modules
$modules$weightThreshold
[1] 0.001
$regulons
list()
$aucell
$aucell$smallestPopPercent
[1] 0.25
$defaultTsne
$defaultTsne$dims
[1] 50
$defaultTsne$perpl
[1] 50
$defaultTsne$aucType
[1] "AUC"
$tSNE_filePrefix
[1] "int/tSNE"
```

@filenames

getIntName(scenicOptions, "genesKept")
loadInt(scenicOptions, "genie3ll", ifNotExits="null")

$int

	fileName
genesKept	"int/1.1_genesKept.Rds"
corrMat	"int/1.2_corrMat.Rds"
genie3wm	"int/1.3_GENIE3_weightMatrix.Rds"
genie3ll	"int/1.4_GENIE3_linkList.Rds"
genie3weighPlot	"int/1.5_weightPlot"
tfModules_asDF	"int/1.6_tfModules_asDF.Rds"
tfModules_forEnrichment	"int/2.1_tfModules_forMotifEnrichmet.Rds"
motifs_AUC	"int/2.2_motifs_AUC.Rds"
motifEnrichment_full	"int/2.3_motifEnrichment.Rds"
motifEnrichment_selfMotifs_wGenes	"int/2.4_motifEnrichment_selfMotifs_wGenes.Rds"
regulonTargetsInfo	"int/2.5_regulonTargetsInfo.Rds"
regulons	"int/2.6_regulons_asGeneSet.Rds"
regulons_incidMat	"int/2.6_regulons_asIncidMat.Rds"
aucell_regulons	"int/3.1_regulons_forAUCell.Rds"
aucell_genesStatsPlot	"int/3.2_aucellGenesStats"
aucell_rankings	"int/3.3_aucellRankings.Rds"
aucell_regulonAUC	"int/3.4_regulonAUC.Rds"
aucell_thresholds	"int/3.5_AUCellThresholds.Rds"
aucell_thresholdsTxt	"int/3.5_AUCellThresholds_Info.tsv"
aucell_binary_full	"int/4.1_binaryRegulonActivity.Rds"
aucell_binary_nonDupl	"int/4.2_binaryRegulonActivity_nonDupl.Rds"
aucell_regulonSelection	"int/4.3_regulonSelections.Rds"
aucell_binaryRegulonOrder	"int/4.4_binaryRegulonOrder.Rds"

getOutName(scenicOptions, "s2_motifEnrichment")

$output

	fileName
s2_motifEnrichment	"output/Step2_MotifEnrichment.tsv"
s2_motifEnrichmentHtml	"output/Step2_MotifEnrichment_preview.html"
s2_regulonTargetsInfo	"output/Step2_regulonTargetsInfo.tsv"
s3_AUCheatmap	"output/Step3_RegulonActivity_heatmap"
s3_AUCtSNE_colAct	"output/Step3_RegulonActivity_tSNE_colByActivity"
s4_boxplotBinaryActivity	"output/Step4_BoxplotActiveCellsRegulon"
s4_binaryActivityHeatmap	"output/Step4_BinaryRegulonActivity_Heatmap_"
s4_binarytSNE_colAct	"output/Step4_BinaryRegulonActivity_tSNE_colByActivity"
s4_binarytSNE_colProps	"output/Step4_BinaryRegulonActivity_tSNE_colByCellProps"
loomFile	"output/scenic.loom"

NIAID

Summary of what SCENIC is doing

- 1. set the environment and bring up the packages
- 2. load data from a .loom file, and filter it.
  - Get cell Annotation (why?)
- 3. set the manager for the parameters, and reference database to be used in the run.
- 4. run spearman correlation to find out the positive vs negative correlation. which Genie3 won't find.
- 5. run GENIE3 and convert the output to coexpression modules runSCENIC_1_coexNetwork2modules()
- 6. RcisTarget (prune co-expression modules using TF-motif enrichment analysis)

New papers about GRN

nature methods

ANALYSIS

https://doi.org/10.1038/s41592-019-0690-6

Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data Beeline, Boolean

Aditya Pratapa[1], Amogh P. Jalihal[2], Jeffrey N. Law[2], Aditya Bharadwaj[1] and T. M. Murali[1]*

We present a systematic evaluation of state-of-the-art algorithms for inferring gene regulatory networks from single-cell transcriptional data. As the ground truth for assessing accuracy, we use synthetic networks with predictable trajectories, literature-curated Boolean models and diverse transcriptional regulatory networks. We develop a strategy to simulate single-cell transcriptional data from synthetic and Boolean networks that avoids pitfalls of previously used methods. Furthermore, we collect networks from multiple experimental single-cell RNA-seq datasets. We develop an evaluation framework called BEELINE. We find that the area under the precision-recall curve and early precision of the algorithms are moderate. The methods are better in recovering interactions in synthetic networks than Boolean models. The algorithms with the best early precision values for Boolean models also perform well on experimental datasets. Techniques that do not require pseudotime-ordered cells are generally more accurate. Based on these results, we present recommendations to end users. BEELINE will aid the development of gene regulatory network inference algorithms.

Single-cell RNA-sequencing technology has made it possible to trace cellular lineages during differentiation and to identify new cell types[1,2]. A central question that arises now is whether we can discover the gene regulatory networks (GRNs) that control cellular differentiation and drive transitions from one cell type to another. In such a GRN, each edge connects a transcription factor (TF) to a gene it regulates. Ideally, the edge is directed from the TF to the target gene, represents direct rather than indirect regulation and corresponds to activation or inhibition.

Single-cell expression data are especially promising for computing GRNs because, unlike bulk transcriptomic data, they do not obscure biological signals by averaging over all the cells in a sample. However, these data have features that pose significant difficulties; for example, substantial cellular heterogeneity[3], cell-to-cell variation in sequencing depth, the high sparsity caused by dropouts[4] and cell-cycle-related effects[5]. Despite these challenges, over a dozen meth-

Results

Overview of algorithms. We surveyed the literature and bioRxiv preprints for papers that either published a new GRN inference algorithm or used an existing approach. We ignored methods that did not assign weights or ranks to the interactions, required additional datasets or supervision, or sought to discover cell-type-specific networks. We selected 12 algorithms using these criteria (Methods).

We used BEELINE to evaluate these approaches on over 400 simulated datasets (across six synthetic networks and four curated Boolean models) and five experimental human or mouse single-cell RNA-seq datasets. Since eight algorithms require pseudotime-ordered cells, we used datasets (both simulated and real) that focus on cell differentiation and development, processes in which there is a meaningful temporal progression of cell states. We did not study GRNs relevant to other biological processes; for example, changes in disease states or differences among cell types.

Cell Systems

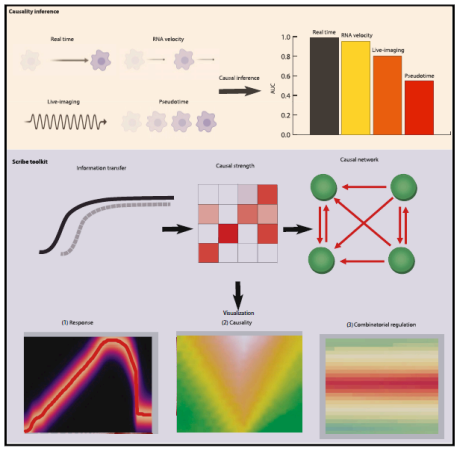
Velocity → trajectory

employs *Restricted Directed Information*

Report

Inferring Causal Gene Regulatory Networks from Coupled Single-Cell Expression Dynamics Using Scribe

Graphical Abstract



Highlights

- Scribe detects causal regulatory networks between genes in diverse single-cell datasets
- Scribe uses restricted directed information to identify regulators and their targets
- Inferring causal regulatory networks requires temporal coupling between measurements
- RNA velocity outperforms pseudotime, but neither perform as well as true time-series data

Authors

Xiaojie Qiu, Arman Rahimzamani, Li Wang, ..., Lauren Saunders, Cole Trapnell, Sreeram Kannan

Correspondence

coletrap@uw.edu (C.T.), ksreeram@uw.edu (S.K.)

In Brief

Qiu et al. present Scribe (https://github.com/aristoteleo/Scribe-py), a toolkit for detecting and visualizing causal regulatory networks between genes in diverse single-cell datasets. They use Scribe to understand how causal network reconstruction depends on temporal coupling between measurements. They show that while pseudotime-ordered single-cell data fail to capture much of the information present in true temporal couplings, RNA velocity measurements restore much of this information.

NIAID

Summary

Software	ARACNE	NetworkInference/ PIDC	bnlearn	GENIE3	iRegulon	SCENIC
semantics	Mutual information	Partial Information Decomposition	Bayes theory	Random Forest, Regression tree	Promoter and TF binding sequence, database	Combination of regression and promoter sequence
years published	2006	2017	2009	2010	2014	2017
No. of cited	2179	82	894	658	337	265
FullName/ explanation	Algorithm for the Reconstruction of Accurate Cellular Networks	Using proportional unique contribution (PUC) to a target gene	Bayes net structure and parameter learning, causality	GEne Network Inference with Ensemble of trees	reverse-engineer the transcriptional regulatory network with regulatory sequence analysis	single-cell regulatory network inference and clustering
Implementation	GUI (geWorkbench)	Julia	R	R	GUI (Cytoscape)	R, Python
type of experiment	Microarray, bulk RNA-seq	Single cell data	General	single cell data	a list of gene names	single cell data
input format	csv	csv	csv	csv	a list	csv/loom file
output	network file	network file	directed network file	network file	network file/binding sequences	network file/heatmap

NIAID

Thank you!

- Questions
  - GitHub: https://github.com/niaid/Gene_Regulatory_Networks
  - My email: zhuy16@nih.gov
  - BCBB email: bioinformatics@niaid.nih.gov
  - general tips on learning bioinformatics: https://github.com/zhuy16/learning_notes

Further feedback or questions? Enter them here in the Googledoc:

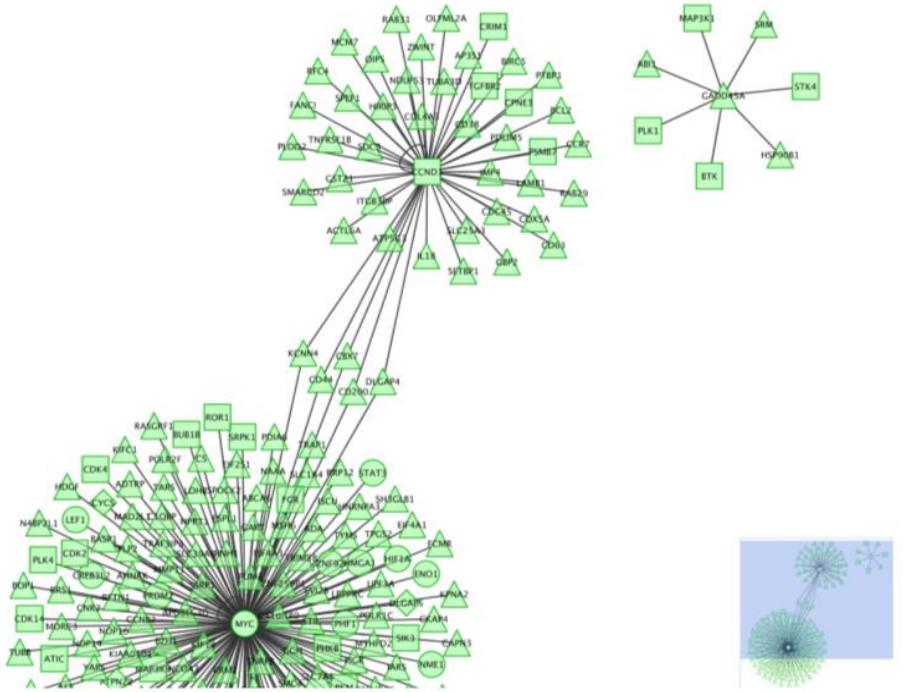
https://tinyurl.com/GRN-best

NIH National Institute of Allergy and Infectious Diseases

NIAID

Tutorial on ARACNE

- Using a microarray data
  - The interface of geWorkbench
  - setting parameters
  - Run ARACNE
  - Export from geWorkbench
  - Import to Cytoscape
  - Visualization



Key references

- Intuition understanding about entropy,
  - https://www.youtube.com/watch?v=2s3aJfRr9gE
- Calculation of mutual information using a concrete data sample.
  - https://www.youtube.com/watch?v=3iplfAfGzzI
- CS262a: Learning and reasoning with Bayesian Networks Adnan Darwiche
  - 11a. Learning Parameters: Complete Data (Chapter 17)
    - https://www.youtube.com/watch?v=gRVg0lZgLUg
  - 11b. Learning Parameters: Incomplete Data (Chapter 17)
    - https://www.youtube.com/watch?v=NDoHheP2ww4
  - 12a. Learning Network Structure I (Chapter 17)
    - https://www.youtube.com/watch?v=RV2lInyq_bI
  - 12b. Learning Network Structure II (Chapter 17)
    - https://www.youtube.com/watch?v=o-urnZRNnbY
- Bnlearn https://www.youtube.com/watch?v=4JkddqxGrO0

Other implementations of ARACNE

Bioinformatics. 2016 Jul 15;32(14):2233-5. doi: 10.1093/bioinformatics/btw216. Epub 2016 Apr 23.

ARACNe-AP: gene network reverse engineering through adaptive partitioning inference of mutual information.

Lachmann A[1], Giorgi FM[1], Lopez G[1], Califano A[1].

Author information

1 Department of Systems Biology, Columbia University, New York, NY, USA.

Abstract

The accurate reconstruction of gene regulatory networks from large scale molecular profile datasets represents one of the grand challenges of Systems Biology. The Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNe) represents one of the most effective tools to accomplish this goal. However, the initial Fixed Bandwidth (FB) implementation is both inefficient and unable to deal with sample sets providing largely uneven coverage of the probability density space. Here, we present a completely new implementation of the algorithm, based on an Adaptive Partitioning strategy (AP) for estimating the Mutual Information. The new AP implementation (ARACNe-AP) achieves a dramatic improvement in computational performance (200× on average) over the previous methodology, while preserving the Mutual Information estimator and the Network inference accuracy of the original algorithm. Given that the previous version of ARACNe is extremely demanding, the new version of the algorithm will allow even researchers with modest computational resources to build complex regulatory networks from hundreds of gene expression profiles.

AVAILABILITY AND IMPLEMENTATION: A JAVA cross-platform command line ex
detailed usage guide are freely available on Sourceforge (http://sourceforge.net/

CONTACT: califano@c2b2.columbia.edu

SUPPLEMENTARY INFORMATION: Supplementary data are available at Bioinfo

Bioinformatics. 2019 Jun 1;35(12):2165-2166. doi: 10.1093/bioinformatics/bty907.

SJARACNe: a scalable software tool for gene network reverse engineering from big data.

Khatamian A[1], Paull EO[2], Califano A[2], Yu J[1].

Author information

1 Department of Computational Biology, St. Jude Children's Research Hospital, Memphis, TN, USA.

2 Department of Systems Biology, Columbia University, New York, NY, USA.

Abstract

SUMMARY: Over the last two decades, we have observed an exponential increase in the number of generated array or sequencing-based transcriptomic profiles. Reverse engineering of biological networks from high-throughput gene expression profiles has been one of the grand challenges in systems biology. The Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNe) represents one of the most effective and widely-used tools to address this challenge. However, existing ARACNe implementations do not efficiently process big input data with thousands of samples. Here we present an improved implementation of the algorithm, SJARACNe, to solve this big data problem, based on sophisticated software engineering. The new scalable SJARACNe package achieves a dramatic improvement in computational performance in both time and memory usage and implements new features while preserving the network inference accuracy of the original algorithm. Given that large-sampled transcriptomic data is increasingly available and ARACNe is extremely demanding for network reconstruction, the scalable SJARACNe will allow even researchers with modest computational resources to efficiently construct complex regulatory and signaling networks from thousands of gene expression profiles.

AVAILABILITY AND IMPLEMENTATION: SJARACNe is implemented in C++ (computational core) and Python (pipelining scripting wrapper, ≥3.6.1). It is freely available at https://github.com/jyyulab/SJARACNe.

https://github.com/califano-lab/ARACNe-AP

NIAID

Build a regression tree from multiple variables

Grow a regression tree

1. For a given target gene j
2. For each of the feature gene i, determine the cost values of each possible split. Chose the lowest cost value as a potential split point.
3. ***Iterate through each of the rest features, get the lowest cost value.***
4. ***Choose the gene that provide the lowest cost to make a first split.***
5. For **each of the branch**, do the same 2à4 and iterate, until all the branch reaches the minimal number of samples.

Determining feature importance

1. In a regression tree, **sum up the reduction of variances due to splitting** by each specific gene in all the branches. That is the score for the feature importance of that gene.
2. Average feature importance for each feature gene. Rank the feature genes by the scaled importance.

Combine feature importance for all the target genes as a rank of all interactions.

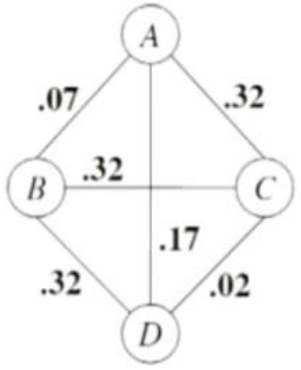
1. For each of the target gene, repeat through all the steps above, until all the feature importance for all target genes are determined.
2. Combine all interactions and rank them by the scores.

Other resources

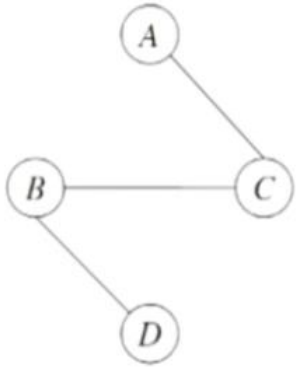
- GraphPlot, http://juliagraphs.github.io/GraphPlot.jl/
- Gamma distribution, https://towardsdatascience.com/gamma-distribution-intuition-derivation-and-examples-55f407423840
- R packages
  - `install.packages("igraph")`
  - `install.packages("bnlearn")`
  - `BiocManager::install("Rgraphviz")`
- iGraph for network analysis https://www.bioss.ac.uk/people/helen/igraphIntro.html

Learning the skeleton of a Baysian network

- Constructing a tree structure (each node can only have one parent node.
- Step 1, learning the Maximum Spanning Tree.
- Add direction through starting from a randomly node.
- The resulting directed graph will be used as seed for other algorithms.
- Optimization of a graph, a commonly used approach constrained on a penalty on the complexity (||G||) of the graph.



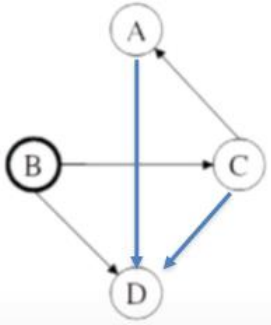
(a) mutual information graph

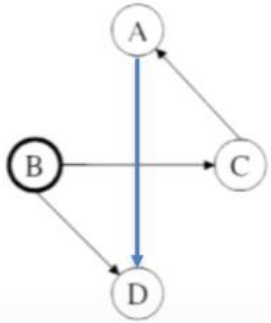


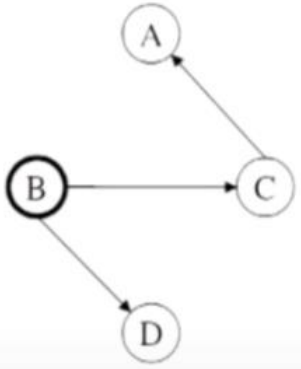
(b) maximum spanning tree

$$\text{Score}(G|\mathcal{D}) \overset{def}{=} \text{LL}(G|\mathcal{D}) - \psi(N) \cdot ||G||$$

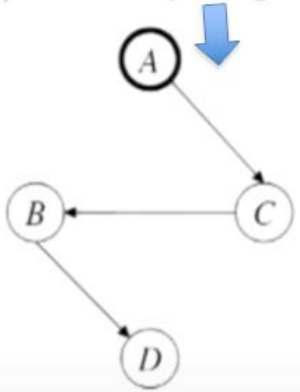
BIC score = Bayesian Information Criterion







(c) maximum likelihood tree



(d) maximum likelihood tree

Objective function for optimizing structures

The problem — Step 1 — Step 2

$$\underbrace{\mathrm{P}(\mathcal{M} \mid \mathcal{D}) = \mathrm{P}(\mathcal{G}, \Theta \mid \mathcal{D})}_{\text{learning}} = \underbrace{\mathrm{P}(\mathcal{G} \mid \mathcal{D})}_{\text{structure learning}} \cdot \underbrace{\mathrm{P}(\Theta \mid \mathcal{G}, \mathcal{D})}_{\text{parameter learning}}$$

Probability of a model (M) given the data set (D).

$$\mathrm{MI}_{\mathfrak{D}}(X, U) \stackrel{def}{=} \sum_{x,u} \mathrm{Pr}_{\mathfrak{D}}(x, u) \log \frac{\mathrm{Pr}_{\mathfrak{D}}(x, u)}{\mathrm{Pr}_{\mathfrak{D}}(x)\mathrm{Pr}_{\mathfrak{D}}(u)}$$

$$\mathrm{Score}(G|\mathfrak{D}) \stackrel{def}{=} \mathrm{LL}(G|\mathfrak{D}) - \psi(N) \cdot ||G||$$

Degree of a network
Sum of the number of possible states each node can possibly take

$$||G|| \stackrel{def}{=} \sum_{i=1}^{n} ||X_i \mathbf{U}_i||$$

$$||X_i \mathbf{U}_i|| \stackrel{def}{=} (X_i{}^{\#} - 1)\mathbf{U}_i{}^{\#}$$

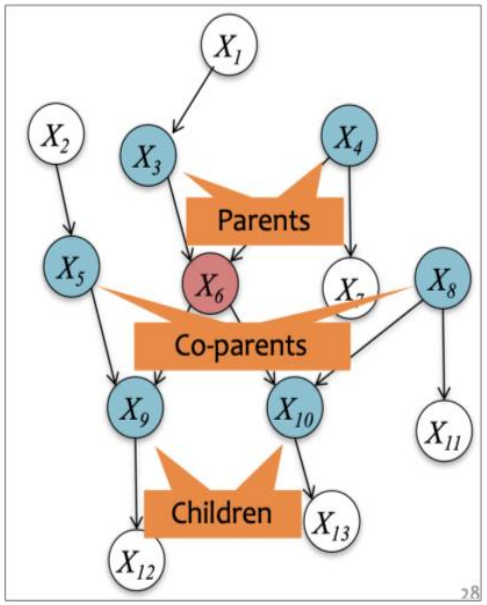
N: the number of nodes in the network

NIH National Institute of Allergy and Infectious Diseases

NIAID

Computation: how to break up a huge network to small local problems

- Optimization of the local variables
  - For computation purpose, you need to break a huge network into smaller pieces.
  - for a node in a graphical model, Markov blanket contains all the variables needed to do approximation, that shield the node from the rest of the network.



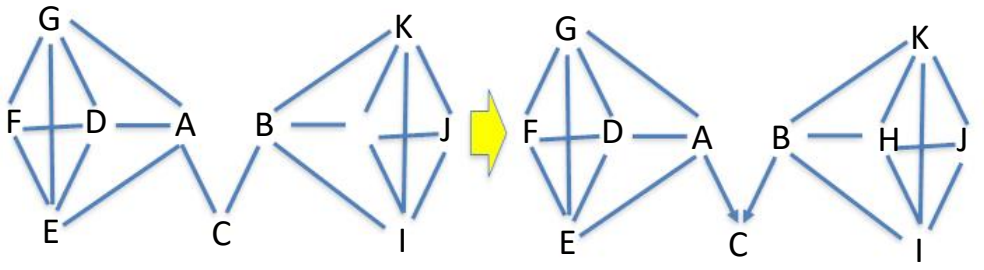
Ref: Learning Bayesian network structure using Markov blanket decomposition, Bui 2012

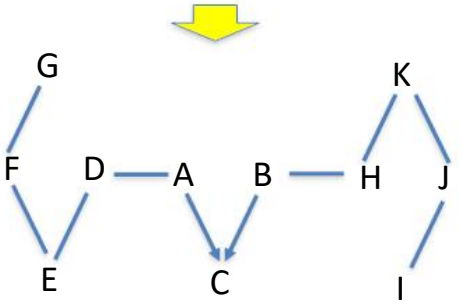
Assign directions: Induction of Causality algorithm

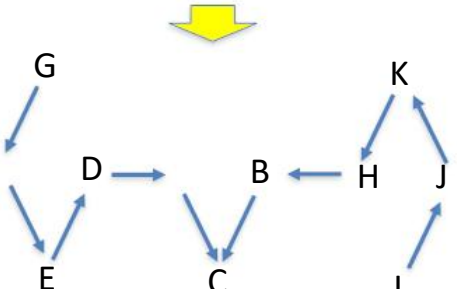
The Induction of Causality Algorithm:
If,

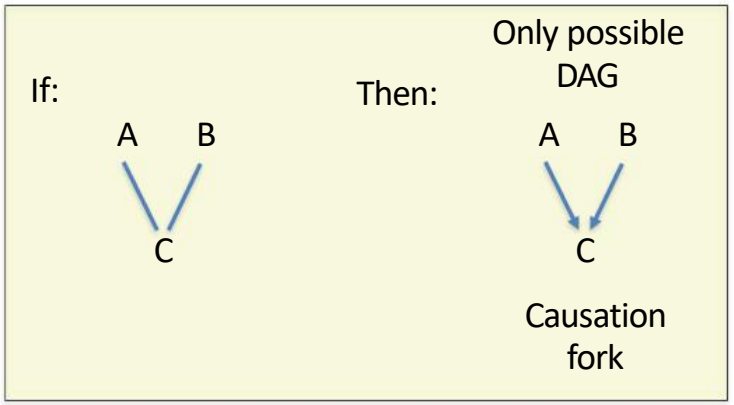
1. A, B are not connected through any other connection.
2. A and B have a common connection to C

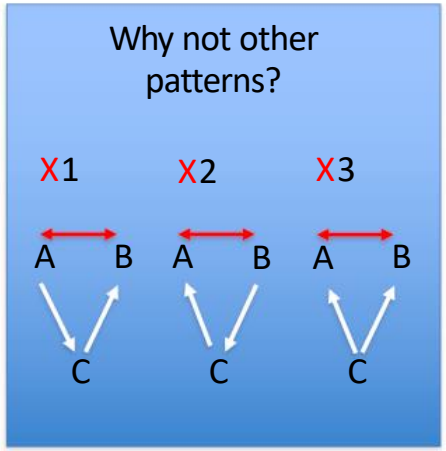
Then:
arrows point from A and B to C











Infer the parameters in Bayesian network

Expectation Maximization

- Assign a set of random parameters θ^0.
- Based on evidence samples, calculate the distribution of each sample.
- Go through all the samples.
- Construct the averaged distribution.
- Based on the averaged distribution, recalculate the set of parameters θ^1.
- Repeat the previous step until θ^{K+1} and θ^K converges.

Using log-likelihood as a objective function to maximize

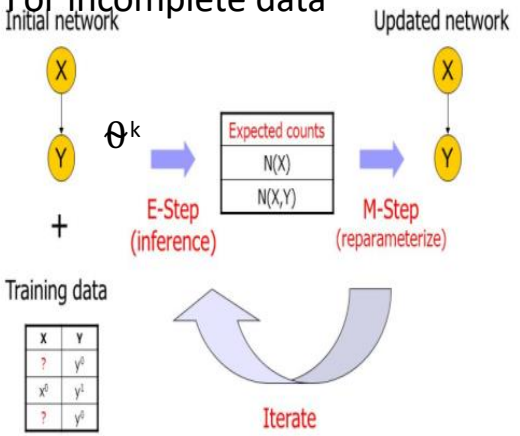
- Construct Log Likelihood formula.
- Use classical methods such as gradient decent to optimize the parameter with a assigned learning rate, to iterate and reach a local maximum.

For complete data

$$\theta^\star = \arg\max_\theta \mathrm{L}(\theta|\mathcal{D})$$

$$\text{iff } \theta^\star_{x|\mathbf{u}} = \Pr_{\mathcal{D}}(x|\mathbf{u})$$

For incomplete data



National Institute of Allergy and Infectious Diseases

NIAID