Literature review:

Bayesian modelling for the study of regulatory interactions

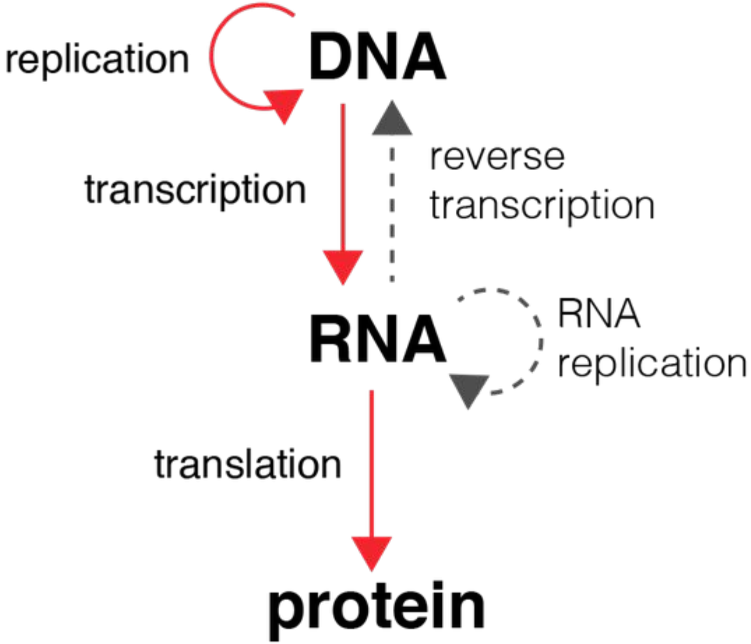
Transcriptional regulation logic
Graphical models
Bayesian networks
Parameter inference

University of Massachusetts Boston - September 28th 2018 - Argenis Arriojas

Content

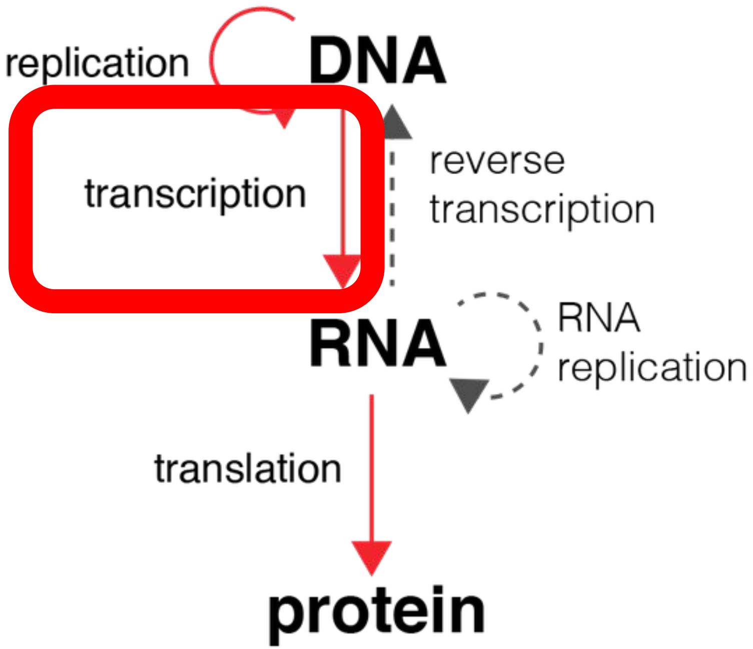
- Introduction
- Inferring regulatory logic https://doi.org/10.1093/bioinformatics/bti388
- Molecular causes of transcriptional response https://doi.org/10.1093/bioinformatics/btt557
- Model-based gene set analysis https://doi.org/10.1093/nar/gkq045
- Our challenges
- Markov Chain Monte Carlo
  - PyMC3/PyMC4 - a library for MCMC modeling and parameter inference
    - Limitations to implement our model
  - My attempts to implement a sampler
  - Object Oriented Approach

Introduction



- Proteins are complex molecules produced from DNA
- These have many different and very specific functions within an organism
  - Antibody
  - Enzyme
  - Messenger
  - Structural component
  - Transport/storage
- Transcription factors (TFs) regulate when and when not to produce certain proteins
- This regulation depends on many factors and is harmoniously orchestrated to achieve cellular objectives

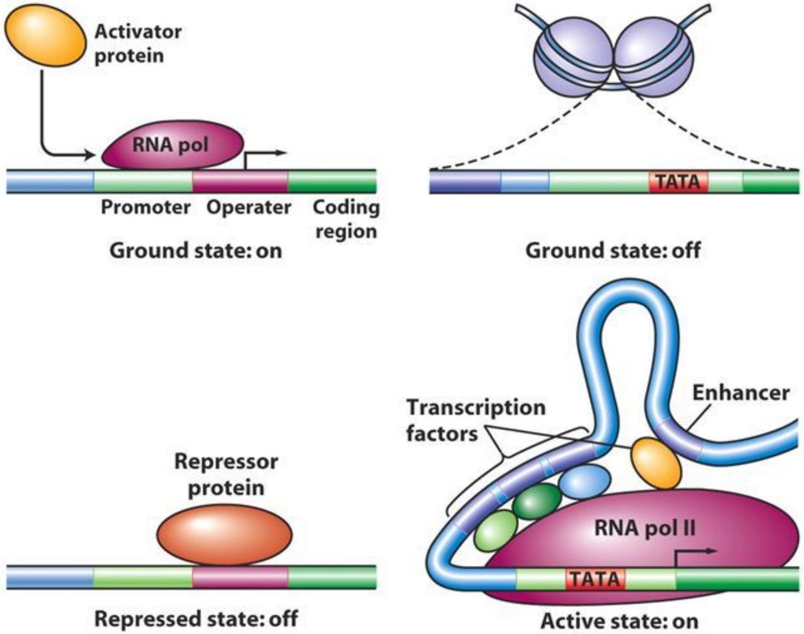
Introduction



- Proteins are complex molecules produced from DNA
- These have many different and very specific functions within an organism
  - Antibody
  - Enzyme
  - Messenger
  - Structural component
  - Transport/storage
- Transcription factors (TFs) regulate when and when not to produce certain proteins
- This regulation depends on many factors and is harmoniously orchestrated to achieve cellular objectives

Transcriptional regulation

Transcription factors bind to promoter region of genes and may either activate or repress expression of a gene



Transcriptional regulation

Transcription factors bind to promoter region of genes and may either activate or repress expression of a gene

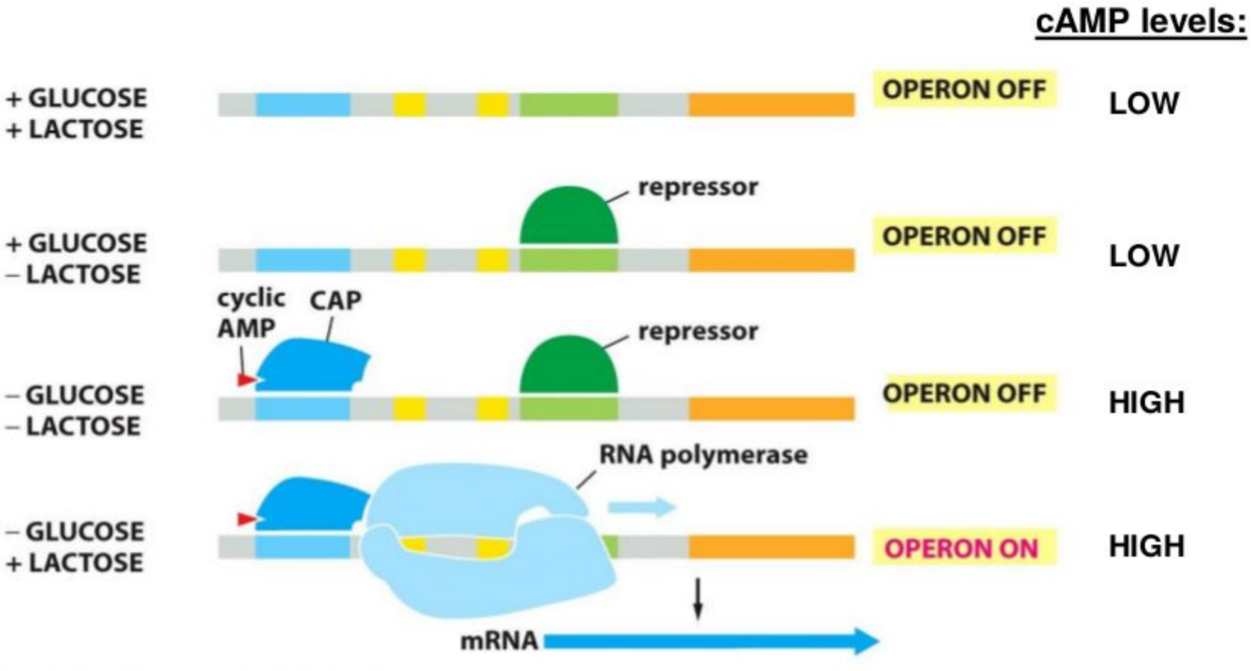
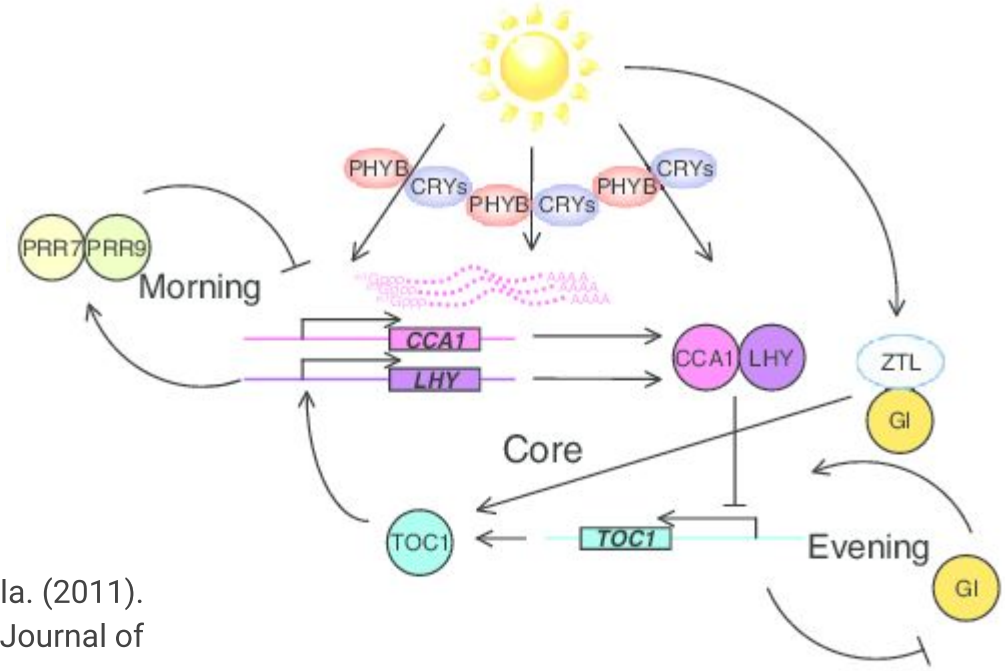


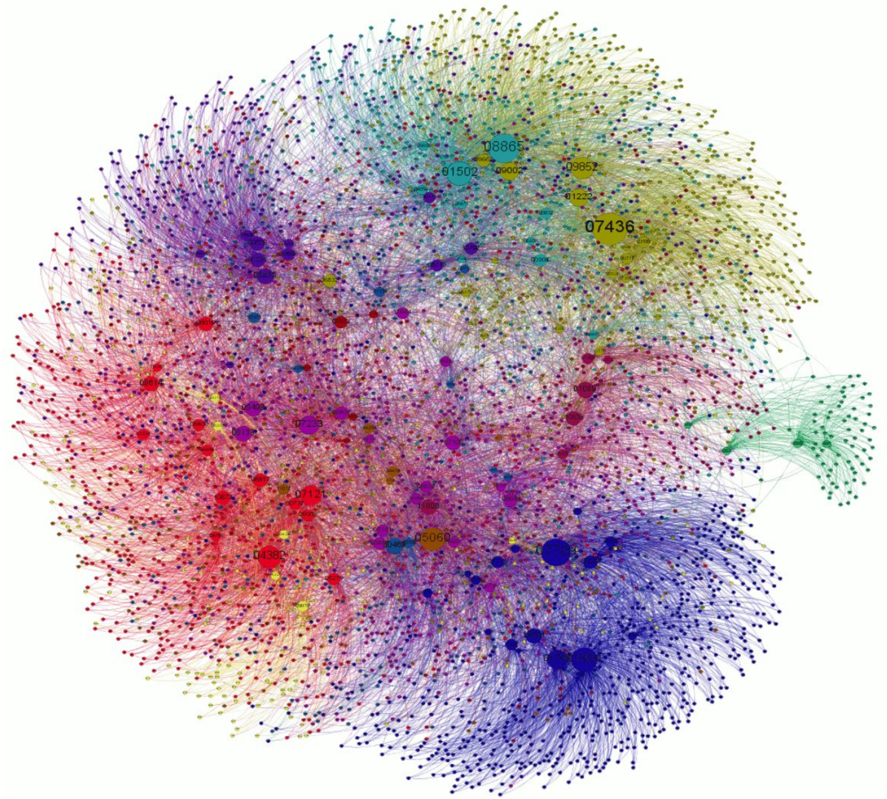
Figure 8-9 Essential Cell Biology, 4th ed. (© Garland Science 2014)

Transcriptional and post-transcriptional regulation

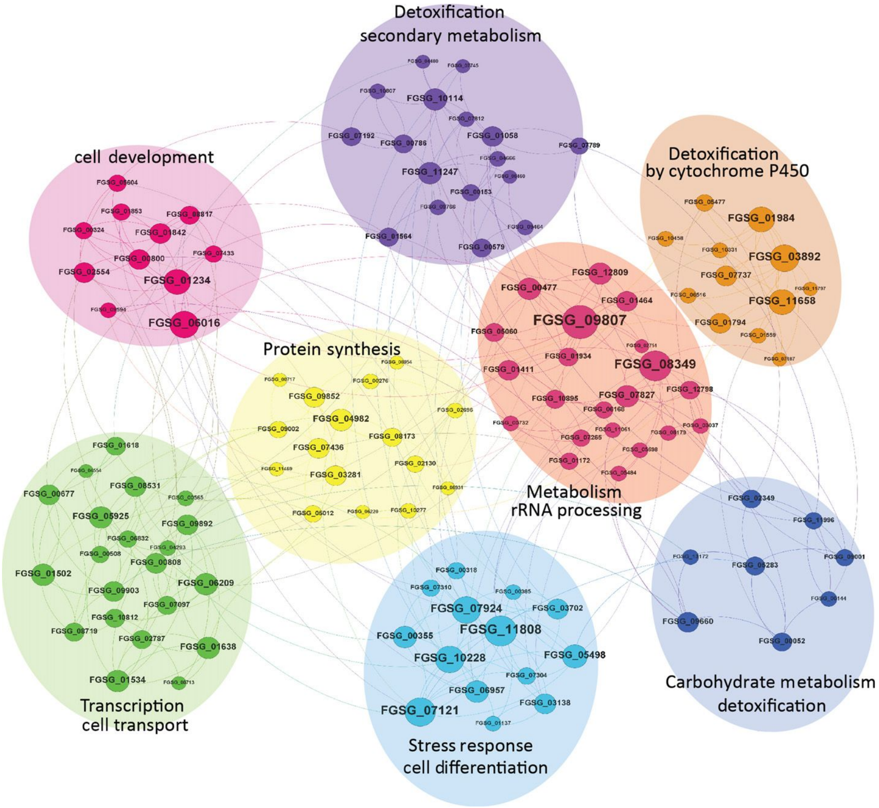


Kojima, Shihoko & L Shingle, Danielle & Green, Carla. (2011). Post-transcriptional control of circadian rhythms. Journal of cell science. 124. 311-20. 10.1242/jcs.065771.

Regulatory networks are complex



Regulatory networks are complex

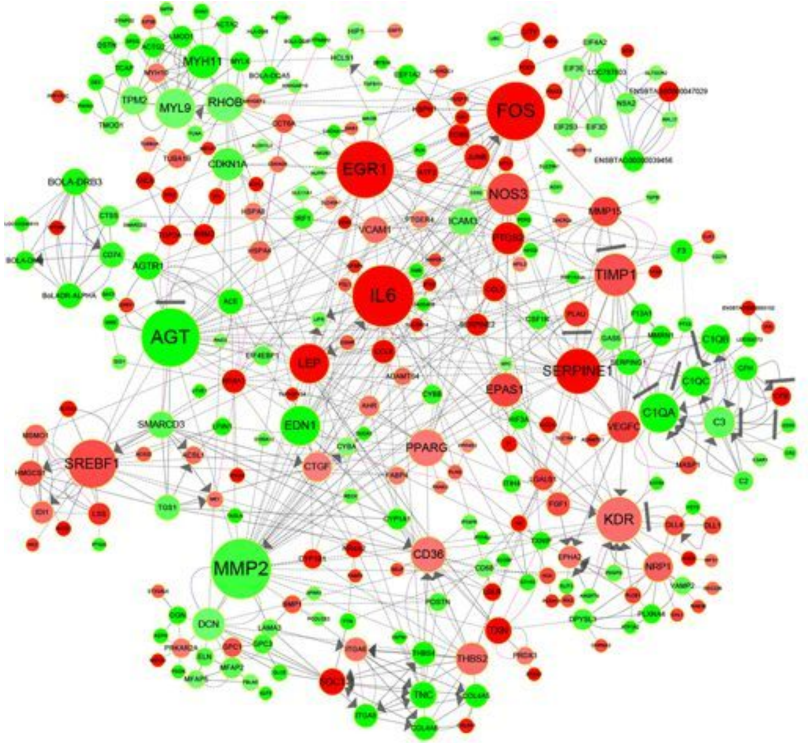


Guo, L. , Zhao, G. , Xu, J. , Kistler, H. C., Gao, L. and Ma, L. (2016), Compartmentalized gene regulatory network of the pathogenic fungus Fusarium graminearum. New Phytol, 211: 527-541. doi:10.1111/nph.13912

Problem:

We want to identify TFs and determine their role in cellular behaviour

Using data to discover TFs



What data?

Differentially expressed genes

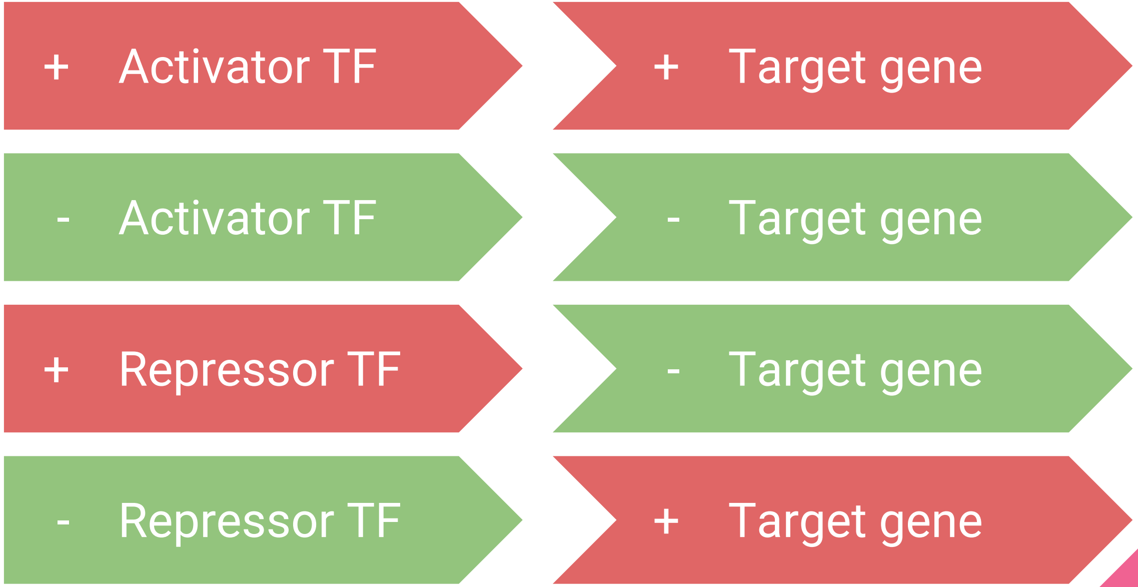
Differentially expressed genes

- Micro-arrays, RNA-seq
    - Measures gene expression levels at RNA level
    - This is a good measure of the activity of a gene
    - Contrast between two conditions
        - Wild-type
        - Special condition

If there is a change in expression level of a gene, there may be some TF responsible for it

Differentially expressed genes

If there is a change in expression level of a gene, there may be some TF responsible for it



+ Activator TF	+ Target gene

- Activator TF	- Target gene

+ Repressor TF	- Target gene

- Repressor TF	+ Target gene

Inferring genetic regulatory logic from expression data

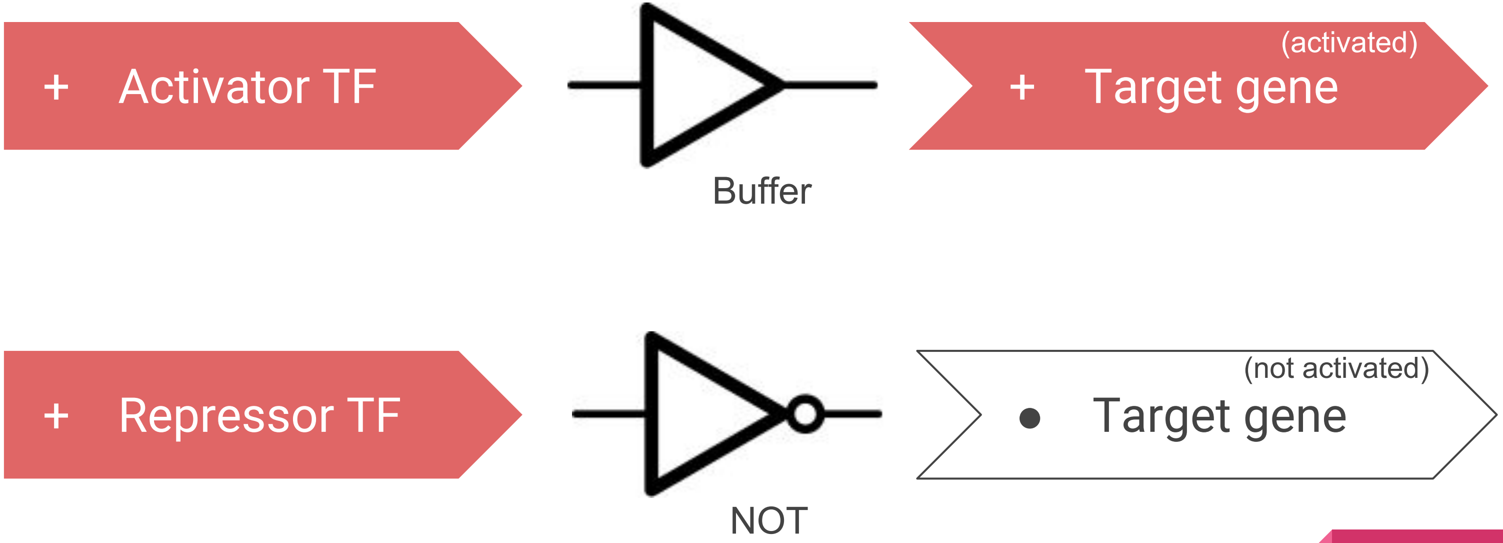
Inferring genetic regulatory logic from expression data

Objective: Identify active TFs

The idea: Model regulatory interactions like digital circuits

Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic



+ Activator TF

Buffer

(activated)

+ Target gene

+ Repressor TF

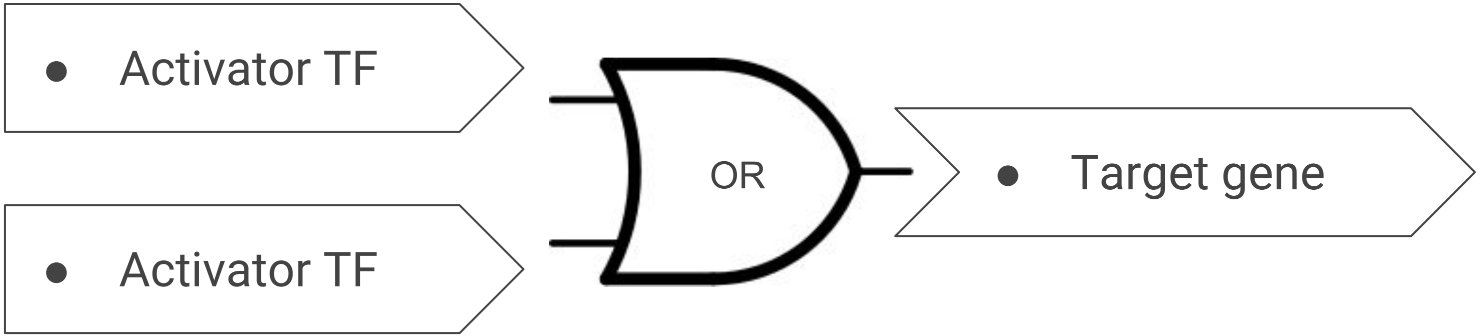
NOT

(not activated)

• Target gene

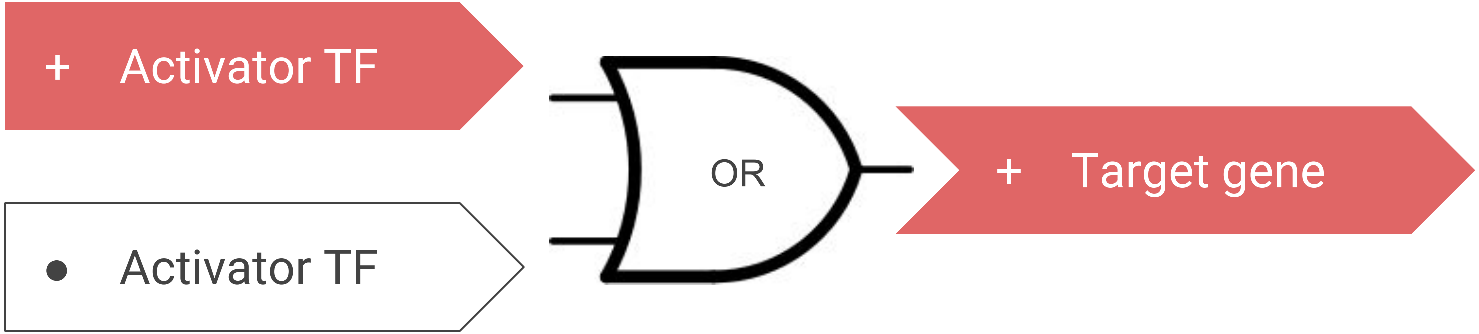
Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic



Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic



+ Activator TF

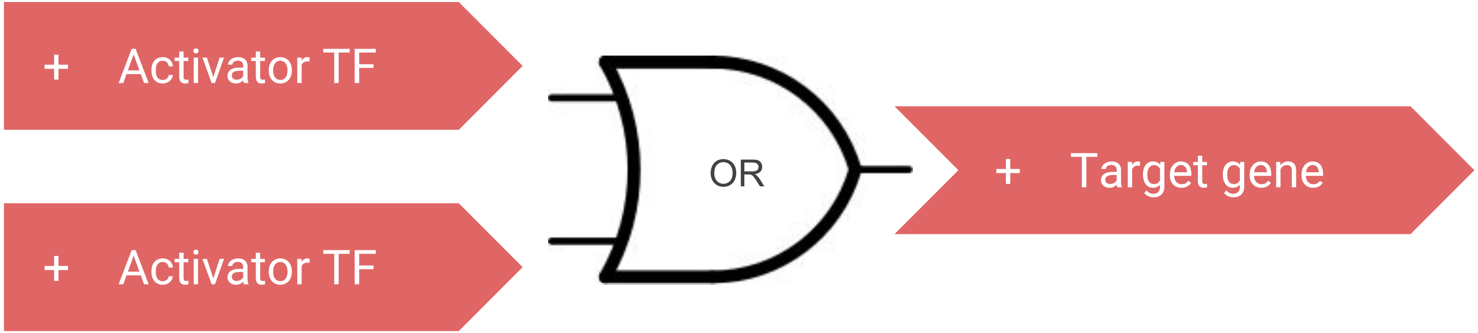
● Activator TF

OR

+ Target gene

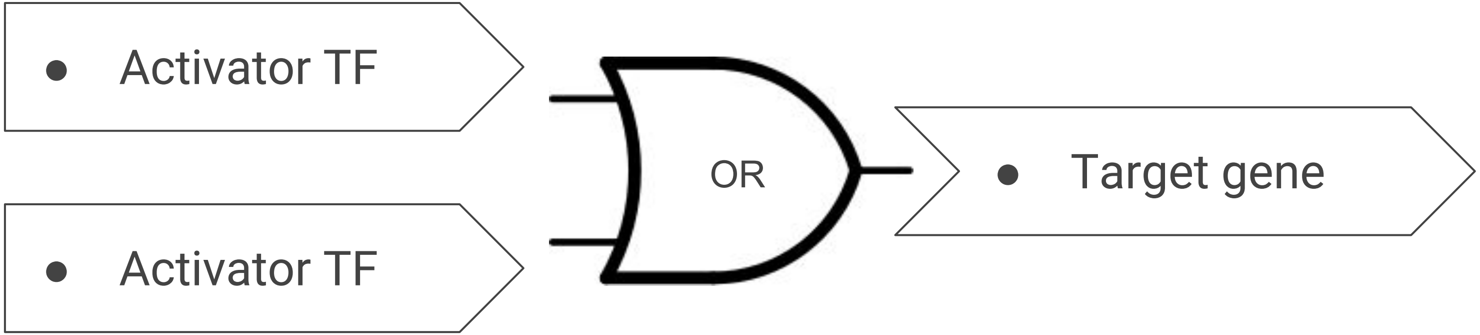
Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic



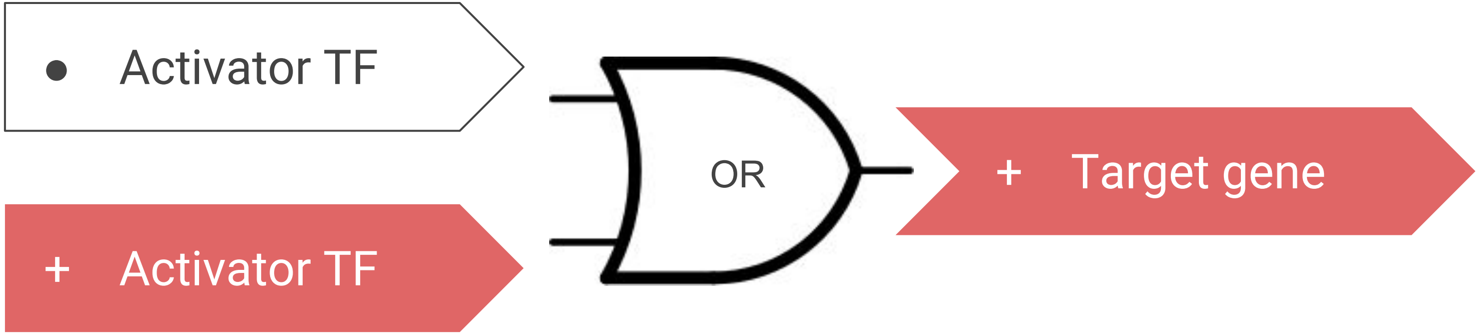
Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic



Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic



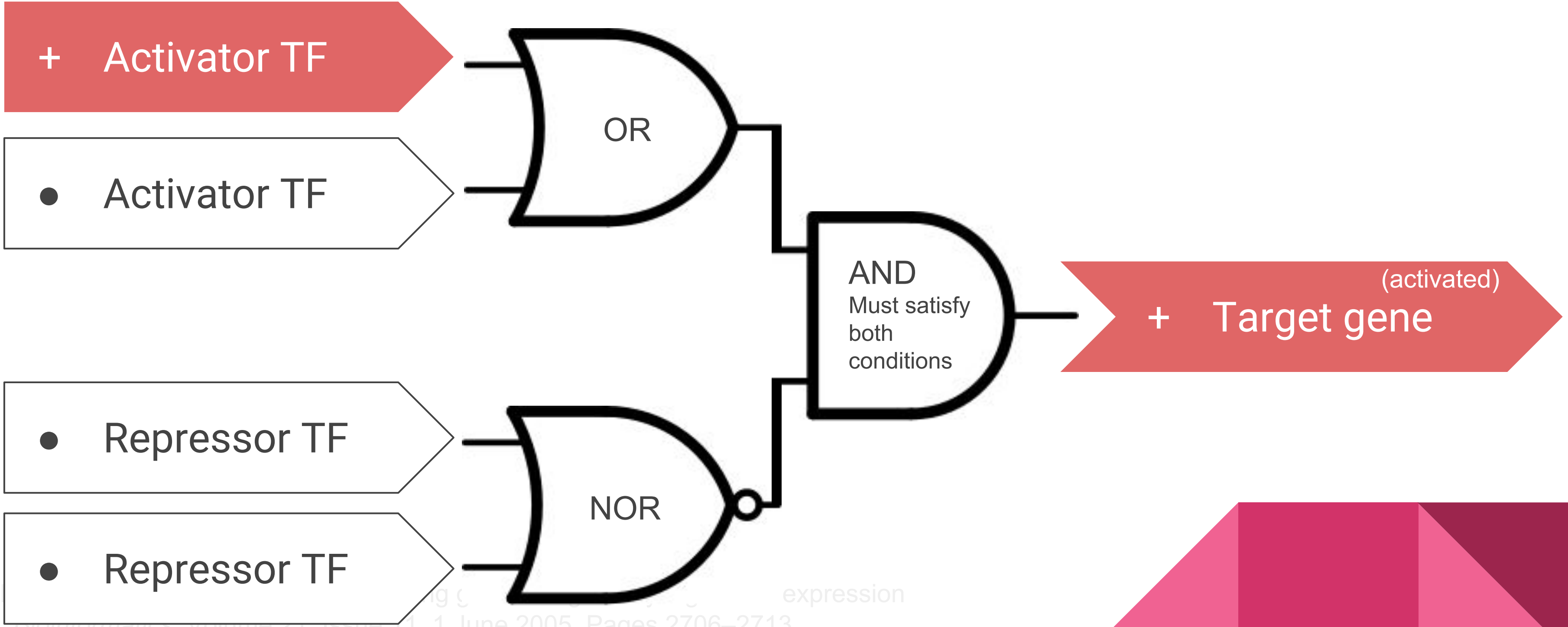
Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Cases of transcriptional regulation

- Single activator (Buffer)
- Single repressor (NOT)
- Multiple possible activators (OR)
- Multiple necessary activators (AND)
- Multiple possible repressors (NOR)
- Multiple necessary repressors (NAND)
- Multiple possible activators, multiple possible repressors (OR-NOR)
- Other combinations ...

Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic - the OR-NOR model



+ Activator TF

● Activator TF

● Repressor TF

● Repressor TF

OR

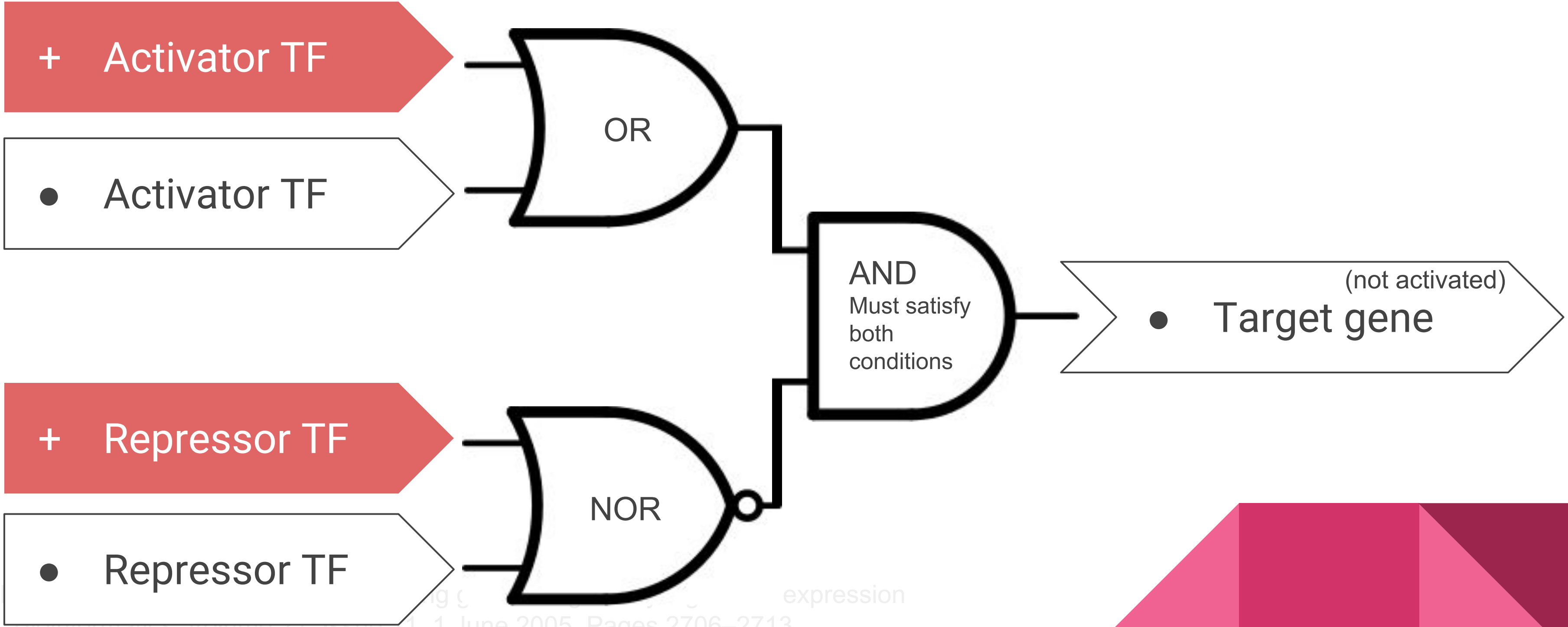
NOR

AND
Must satisfy both conditions

(activated)
+ Target gene

Svetl [illegible] expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Regulatory logic - the OR-NOR model



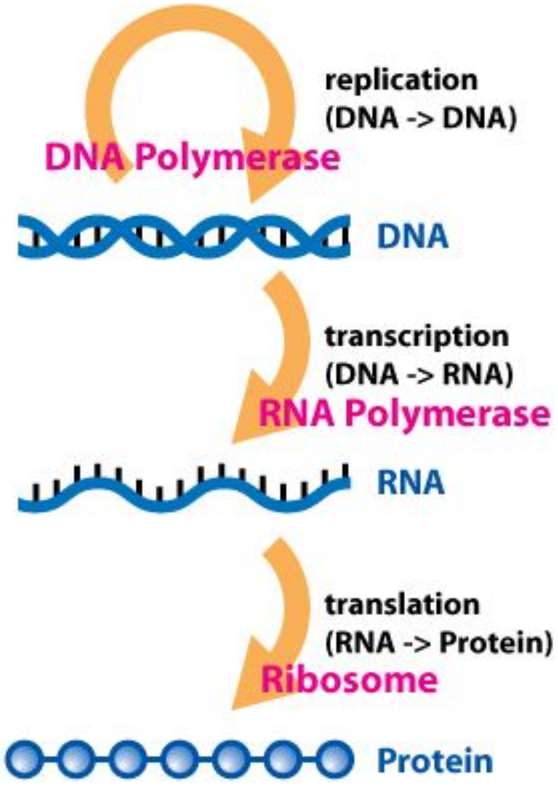
Svetl … ng g … expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Approach of the authors

- Use RNA levels as buffer for TFs levels
- For a target gene, look for TFs activation within the same data, considering all other genes as candidate regulators
- Use several experiments for parameter inference
- Yeast expression data
  - Simultaneous
  - Time-delay
- Bayesian model
- Markov Chain Monte Carlo for sampling posterior distribution

Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

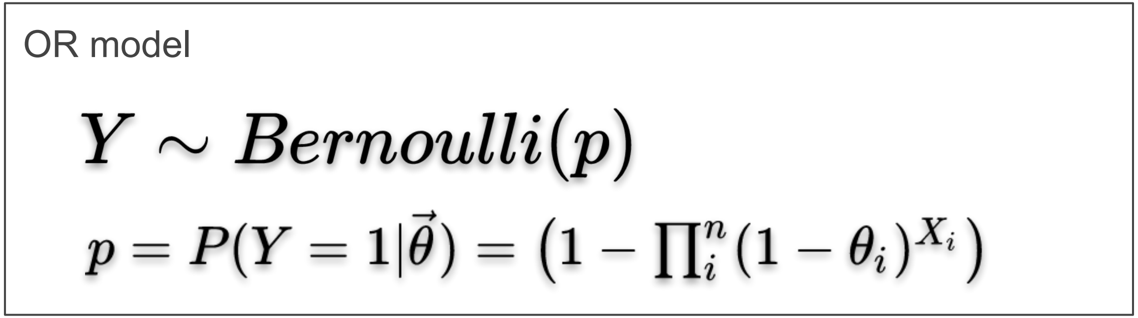
mRNA as buffer for proteins?



This means that by measuring levels of RNA, we indirectly measure levels of proteins for the same gene

This way, TF levels would be estimated through its corresponding RNA levels

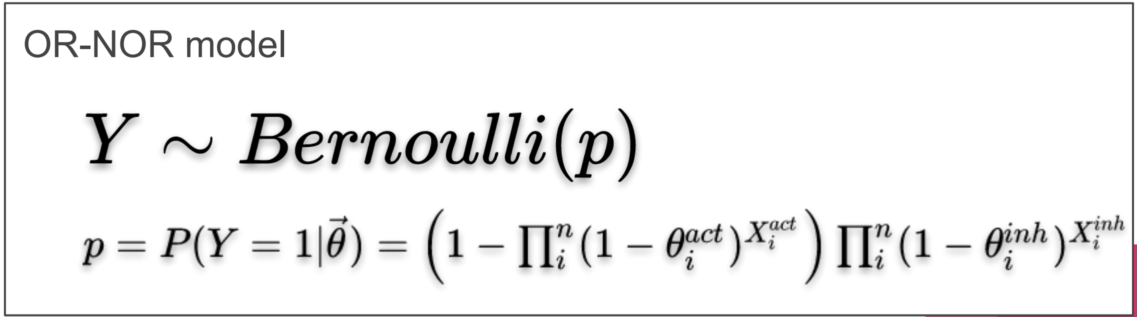
Implementation of models



OR model

$$Y \sim Bernoulli(p)$$

$$p = P(Y = 1|\vec{\theta}) = \left(1 - \prod_i^n (1 - \theta_i)^{X_i}\right)$$



OR-NOR model

$$Y \sim Bernoulli(p)$$

$$p = P(Y = 1|\vec{\theta}) = \left(1 - \prod_i^n (1 - \theta_i^{act})^{X_i^{act}}\right) \prod_i^n (1 - \theta_i^{inh})^{X_i^{inh}}$$

Svetlana Bulashevska, Roland Eils; Inferring genetic regulatory logic from expression data, *Bioinformatics*, Volume 21, Issue 11, 1 June 2005, Pages 2706–2713, https://doi.org/10.1093/bioinformatics/bti388

Shortcomings

- Focus on one target gene at a time
- Doesn’t use prior biological knowledge to build up gene network
- Needs several experiments to perform inference

Molecular causes of transcriptional response: a Bayesian prior knowledge approach

Molecular causes of transcriptional response: a Bayesian prior knowledge approach

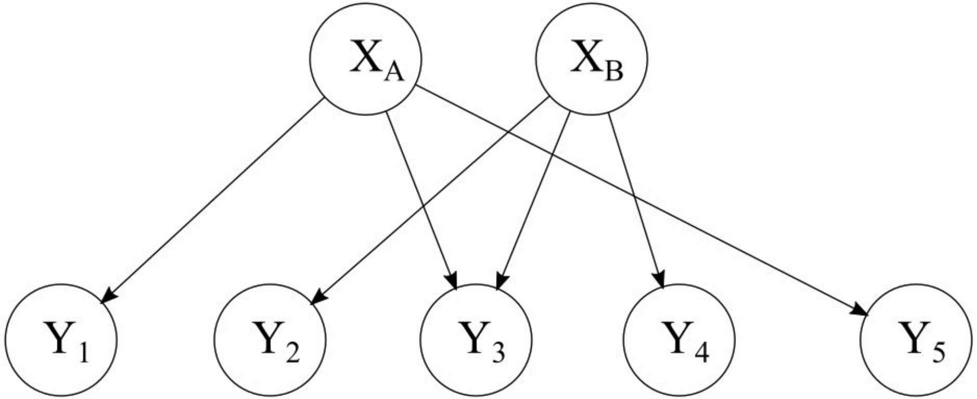
Objective: Identify active TFs

The idea: Use prior biological knowledge to build Bayesian network of regulatory interactions

Zarringhalam, K., Enayetallah, A., Gutteridge, A., Sidders, B., & Ziemek, D. (2013). Molecular causes of transcriptional response: a Bayesian prior knowledge approach. *Bioinformatics*, *29*(24), 3167–3173. http://doi.org/10.1093/bioinformatics/btt557

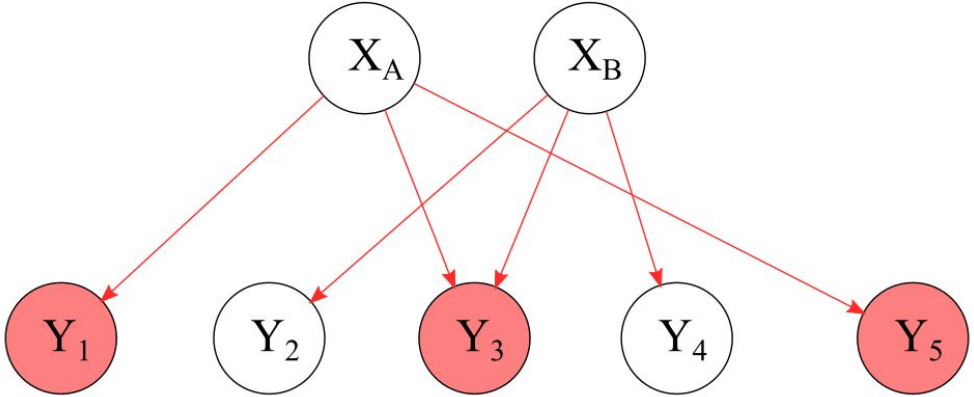
Use of causal relations

Given a network of causal relations and a DEG pattern, it is possible to estimate the most likely active TF



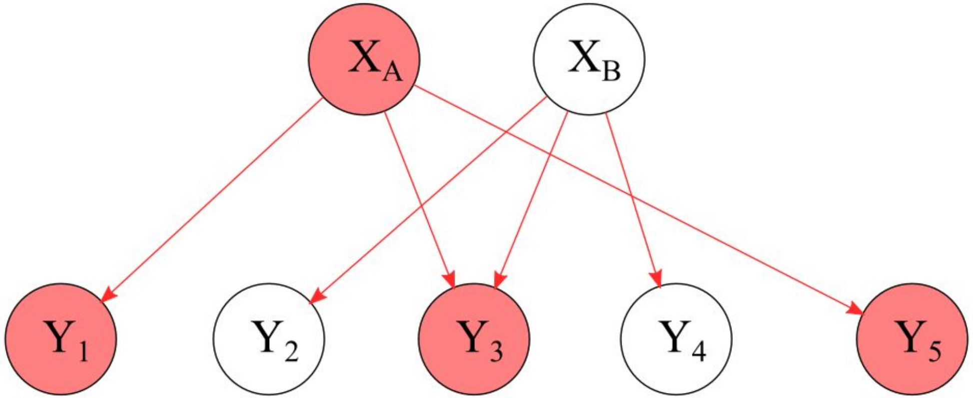
Use of causal relations

Given a network of causal relations and a DEG pattern, it is possible to estimate the most likely active TF



Use of causal relations

Given a network of causal relations and a DEG pattern, it is possible to estimate the most likely active TF



Bayesian network of molecular interactions

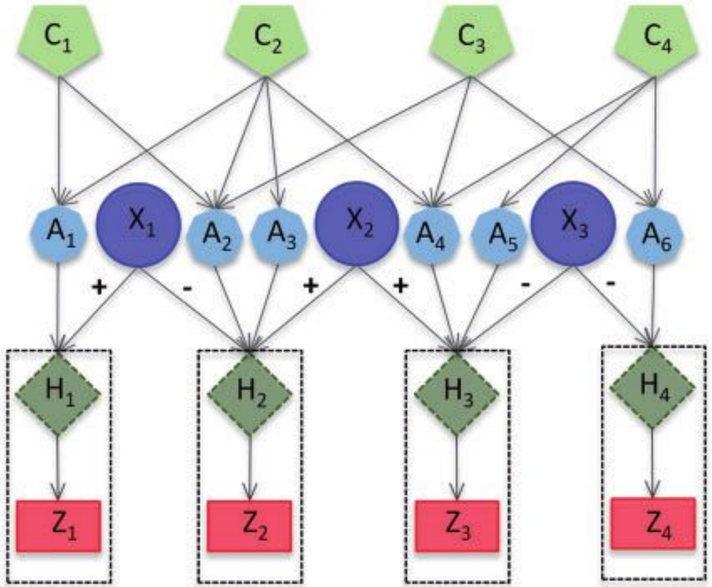
- Use of known causal relationships between molecules
- ~ 450.000 causal relations from literature

- Takes into account expression data for all genes in active contexts
- Contexts are defined through enrichment analysis of non-zero network
- MeSH terms of known causal relations are used for enrichment analysis

Zarringhalam, K., Enayetallah, A., Gutteridge, A., Sidders, B., & Ziemek, D. (2013). Molecular causes of transcriptional response: a Bayesian prior knowledge approach. *Bioinformatics*, *29*(24), 3167–3173. http://doi.org/10.1093/bioinformatics/btt557

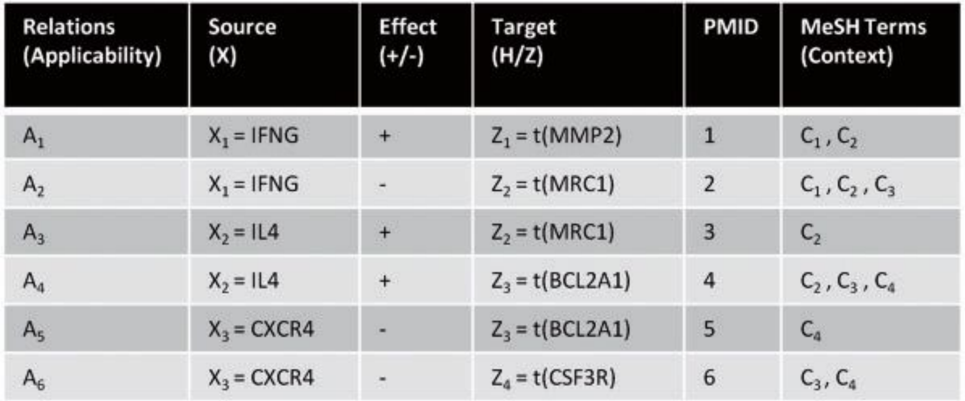
Bayesian network of molecular interactions

- Z: Observed transcript levels for a gene (Gene expression data)
- H: **True** state for corresponding transcript. Not directly seen
- X: Regulator (TF) **True** state. This is what is to be inferred
- A: Applicability of interaction. Given by context
- C: Context (MeSH terms)

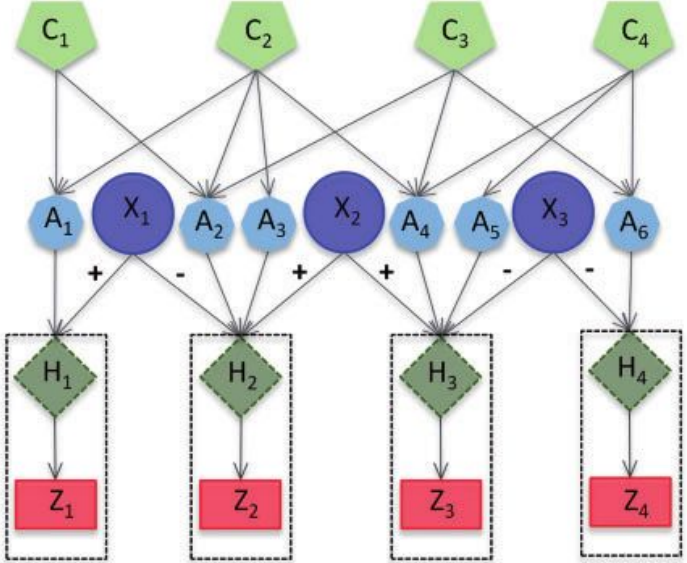


Zarringhalam, K., Enayetallah, A., Gutteridge, A., Sidders, B., & Ziemek, D. (2013). Molecular causes of transcriptional response: a Bayesian prior knowledge approach. *Bioinformatics*, *29*(24), 3167–3173. http://doi.org/10.1093/bioinformatics/btt557

Bayesian network of molecular interactions



Relations (Applicability)	Source (X)	Effect (+/-)	Target (H/Z)	PMID	MeSH Terms (Context)
A_1	X_1 = IFNG	+	Z_1 = t(MMP2)	1	C_1 , C_2
A_2	X_1 = IFNG	-	Z_2 = t(MRC1)	2	C_1 , C_2 , C_3
A_3	X_2 = IL4	+	Z_2 = t(MRC1)	3	C_2
A_4	X_2 = IL4	+	Z_3 = t(BCL2A1)	4	C_2 , C_3 , C_4
A_5	X_3 = CXCR4	-	Z_3 = t(BCL2A1)	5	C_4
A_6	X_3 = CXCR4	-	Z_4 = t(CSF3R)	6	C_3, C_4



Zarringhalam, K., Enayetallah, A., Gutteridge, A., Sidders, B., & Ziemek, D. (2013). Molecular causes of transcriptional response: a Bayesian prior knowledge approach. *Bioinformatics*, *29*(24), 3167–3173. http://doi.org/10.1093/bioinformatics/btt557

Building noise in the Bayesian network

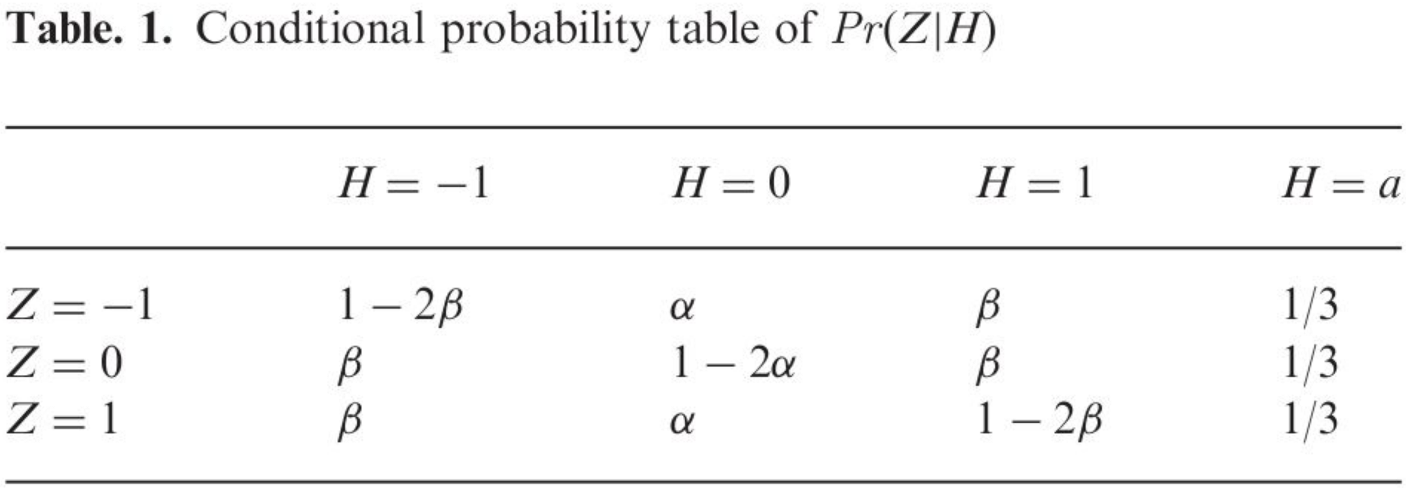


Table. 1. Conditional probability table of $Pr(Z|H)$

	$H = -1$	$H = 0$	$H = 1$	$H = a$
$Z = -1$	$1 - 2\beta$	α	β	$1/3$
$Z = 0$	β	$1 - 2\alpha$	β	$1/3$
$Z = 1$	β	α	$1 - 2\beta$	$1/3$

Zarringhalam, K., Enayetallah, A., Gutteridge, A., Sidders, B., & Ziemek, D. (2013). Molecular causes of transcriptional response: a Bayesian prior knowledge approach. *Bioinformatics*, *29*(24), 3167–3173. http://doi.org/10.1093/bioinformatics/btt557

Shortcoming

- Relies on knowledge of causal relations
    - Big curated network of ~450.000 statements is licensed
- Doesn’t provide a way to identify or test unknown causal relations

GOing Bayesian: model-based gene set analysis of genome-scale data

GOing Bayesian: model-based gene set analysis of genome-scale data

Objective: Identify relevant active categories/terms

The idea: Use Bayesian network of terms associated with DEGs

Bauer, S., Gagneur, J., & Robinson, P. N. (2010). GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Research*, *38*(11), 3523–3532. http://doi.org/10.1093/nar/gkq045

Bayesian network vs enrichment analysis

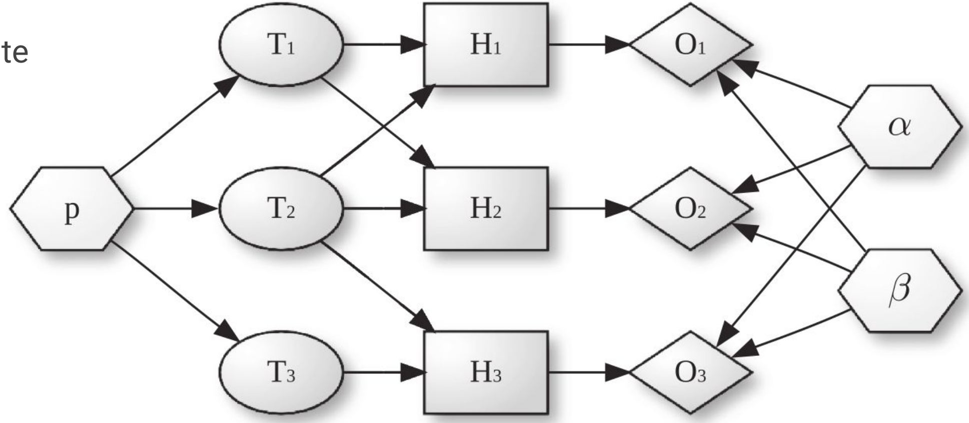
- Enrichment analysis is performed on one category at a time
- Knowledge bases such as Gene Ontology contain hundreds of thousands of categories with very high overlap between categories
- Enrichment analysis often returns large numbers of correlated categories

- Model-based gene set analysis (MGSA) considers all the categories at once
- Bayesian modelling naturally takes category overlap into account

Bauer, S., Gagneur, J., & Robinson, P. N. (2010). GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Research*, *38*(11), 3523–3532. http://doi.org/10.1093/nar/gkq045

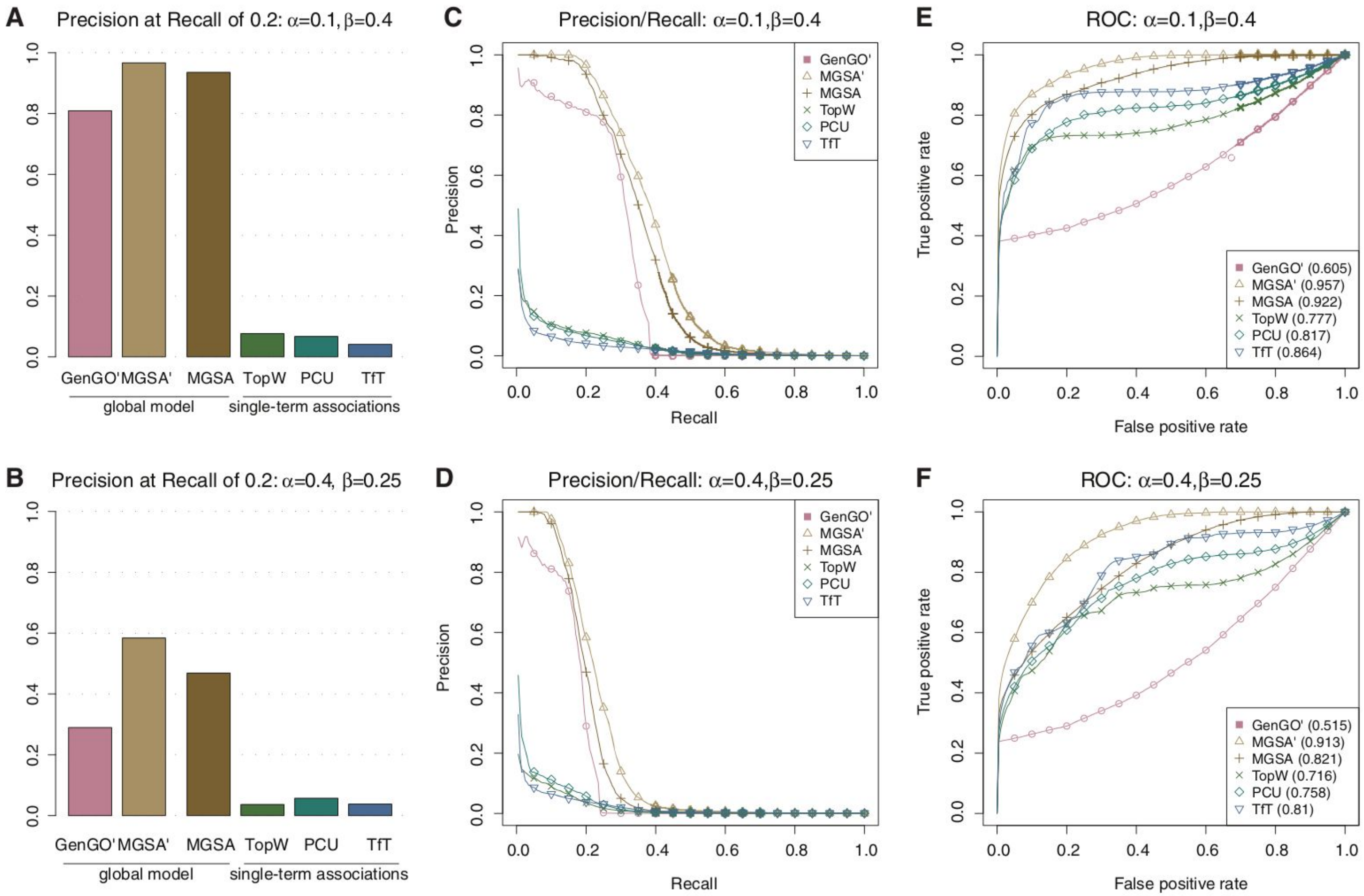
MGSA: Bayesian network of category associations

- T: Terms layer
    - Boolean nodes
    - Represents categories/terms activation state
    - This is what is being inferred
- H: Hidden layer
    - Boolean nodes
    - True activation state of genes
    - Measured indirectly through O
- O: Observed layer
    - Boolean nodes
    - Measures activation state of genes
- The parameter set
    - Continuous in [0, 1]
    - Parametrize the distributions of O and T



Bauer, S., Gagneur, J., & Robinson, P. N. (2010). GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Research*, *38*(11), 3523–3532. http://doi.org/10.1093/nar/gkq045

Performance comparison (simulated data)



Possible shortcomings

- Computation time?
  - No mention on this
  - Would like to compare to other methods

Building on top

Combining ideas

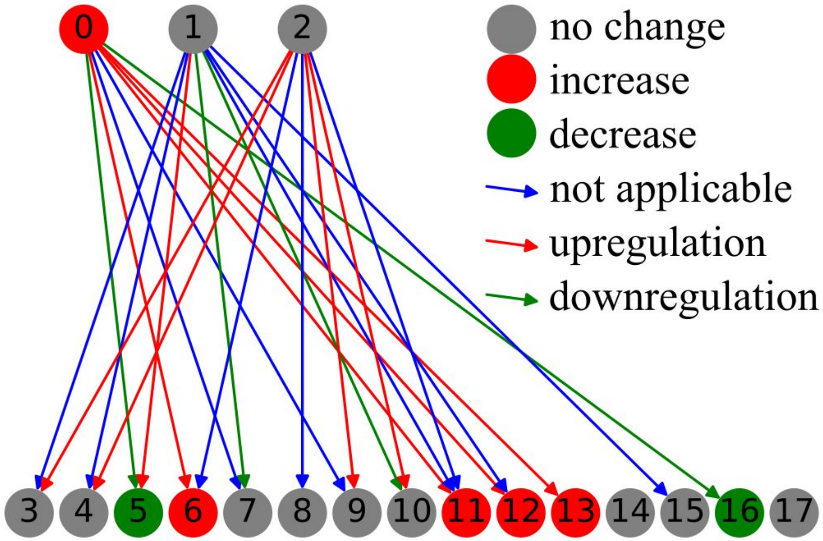
Objectives:

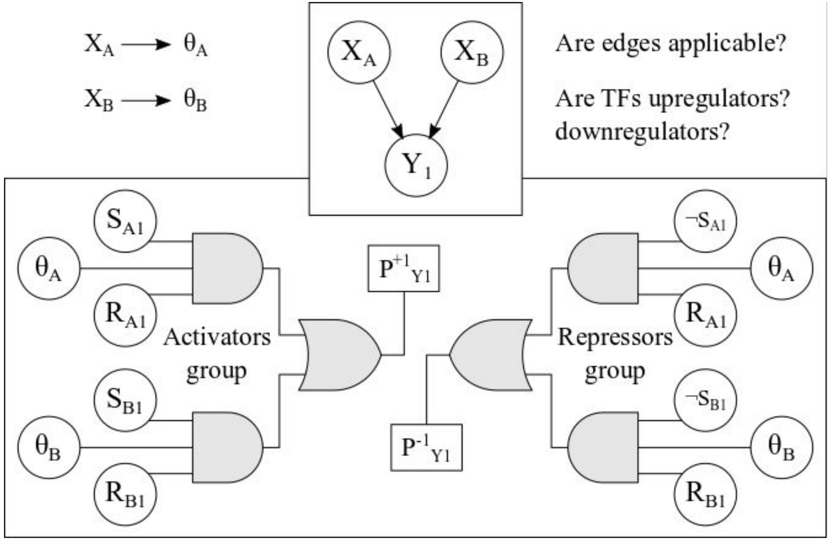
- Identify active TFs
- Identify regulatory interactions between molecules

Ideas to combine:

- Regulatory logic
- Use Bayesian network
- Use MGSA instead of enrichment analysis

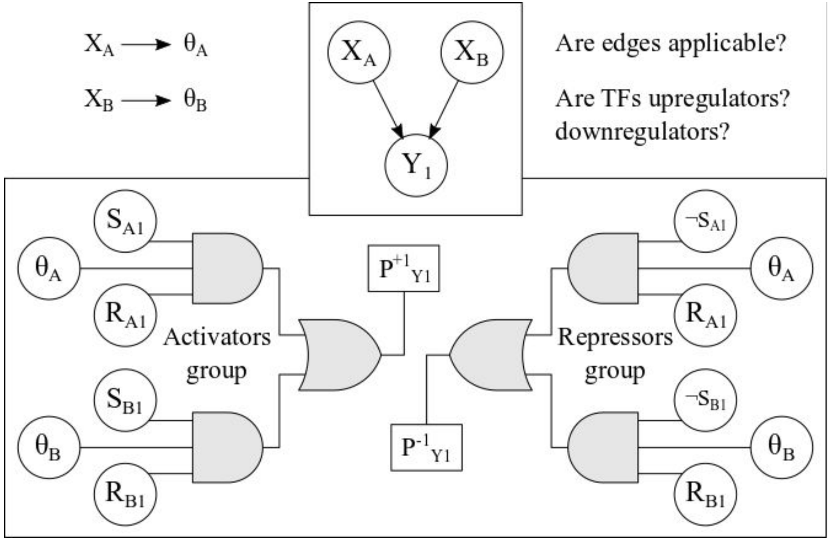
Embedding regulatory logic in a Bayesian network





Embedding regulatory logic in a Bayesian network

OR-NOR model



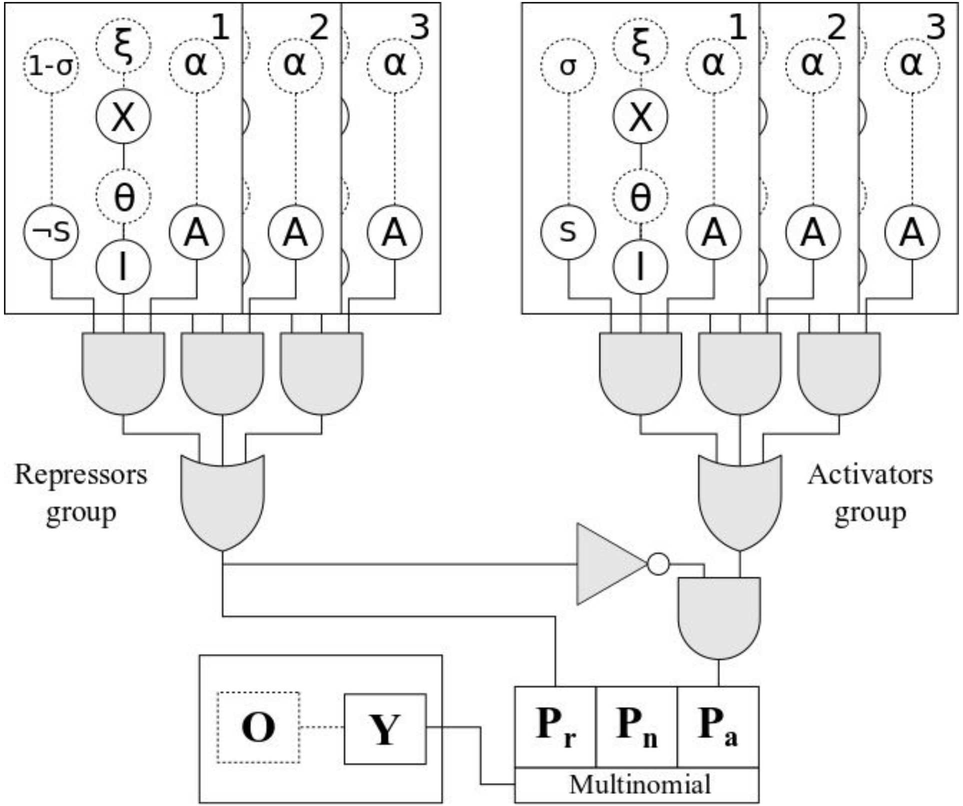
$$Y_j \sim Multinomial(P_{Y_j}^{-1}, P_{Y_j}^{0}, P_{Y_j}^{+1})$$

$$P_{Y_j}^{-1} = 1 - \prod_i (1 - \theta_i (1 - S_{ij}) R_{ij})$$

$$P_{Y_j}^{+1} = [1 - \prod_i (1 - \theta_i S_{ij} R_{ij})] \prod_i (1 - \theta_i (1 - S_{ij}) R_{ij})$$

$$P_{Y_j}^{0} = 1 - P_{Y_j}^{+1} - P_{Y_j}^{-1}$$

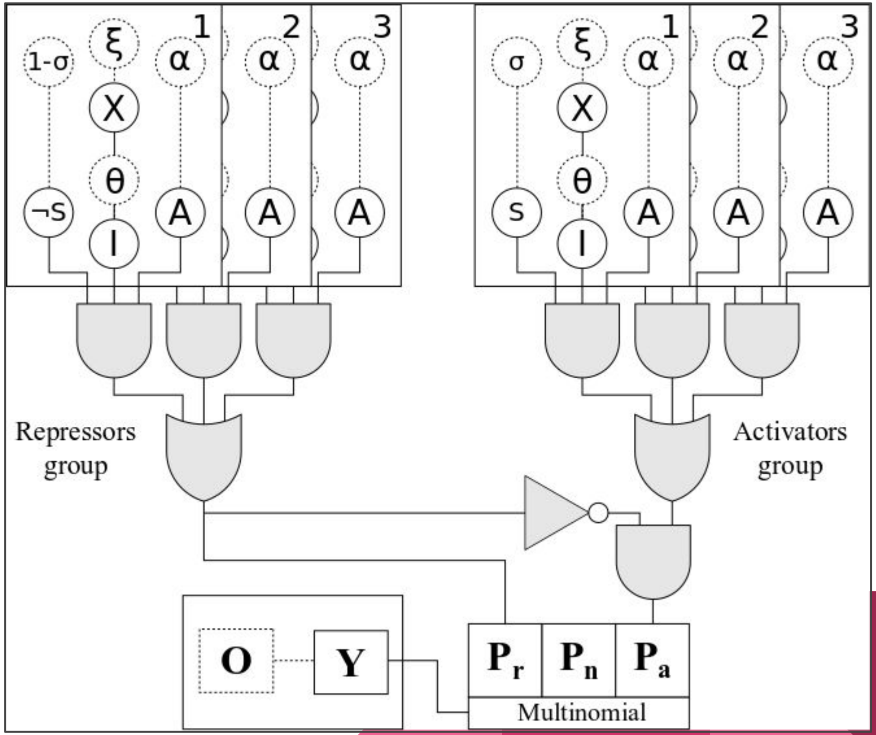
Enhancing our model



- Incorporate noise at the DEG level
- Refine roles of parameters for X estimation

Nodes in the new proposal

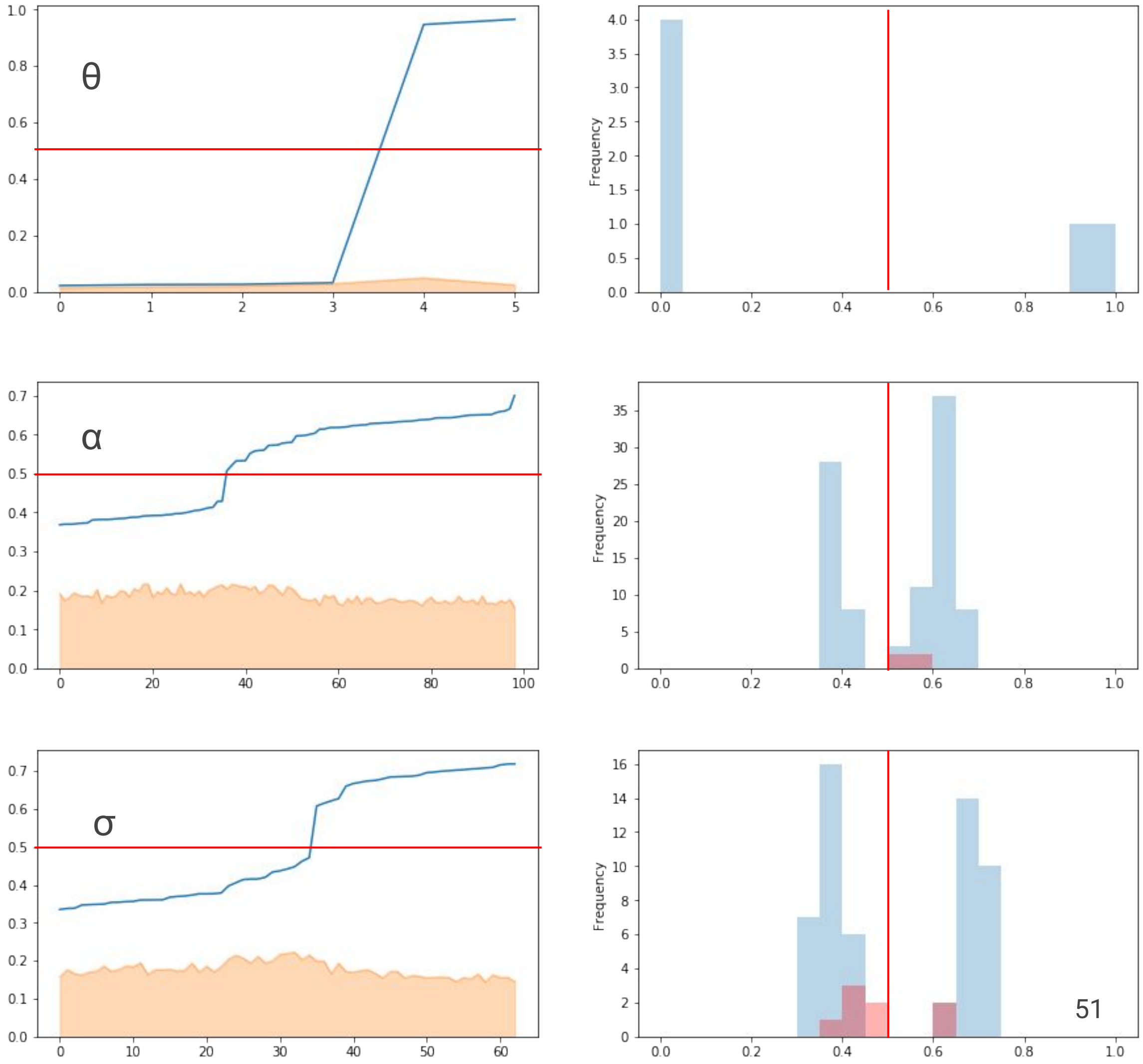
- $X_i \in \{0, 1\}$: True state of activation of i-th TF. This variable doesn’t participate directly in our model
- $\theta_i \in [0, 1]$: Probability that Xi regulates target gene
- $A_i \in \{0, 1\}$: True applicability of a regulation interaction between i-th TF and gene Y
- $\alpha_i \in [0, 1]$: Noisy representation of A i
- $S_i \in \{0, 1\}$: Sign of regulation for an applicable regulation interaction. Here, Si = 0 and Si = 1 indicate downregulation and upregulation interactions respectively
- $\sigma_i \in [0, 1]$: Noisy representation of Si
- $O \in \{-1, 0, 1\}$: The observed differential expression for gene Y
- $Y \in \{-1, 0, 1\}$: The true differential expression for gene Y



Model implementations

- Initially used PyMC3
  - Theano based python library
  - Works nicely for small networks
  - Limitations
    - Large compilation time for larger networks
    - Lack of sparse tensors. Limits the way in which models can be specified
- Possibly in the future: PyMC4
  - Being implemented on TensorFlow
- Currently working on Python based sampler
  - Allows for flexibility in the model specification and sampling strategies
  - Using Object Oriented programming
  - Using Dynamic programming to improve efficiency
  - Once it’s tested, can be ported to Cython or C/C++ for increasing efficiency

An example of results for small network

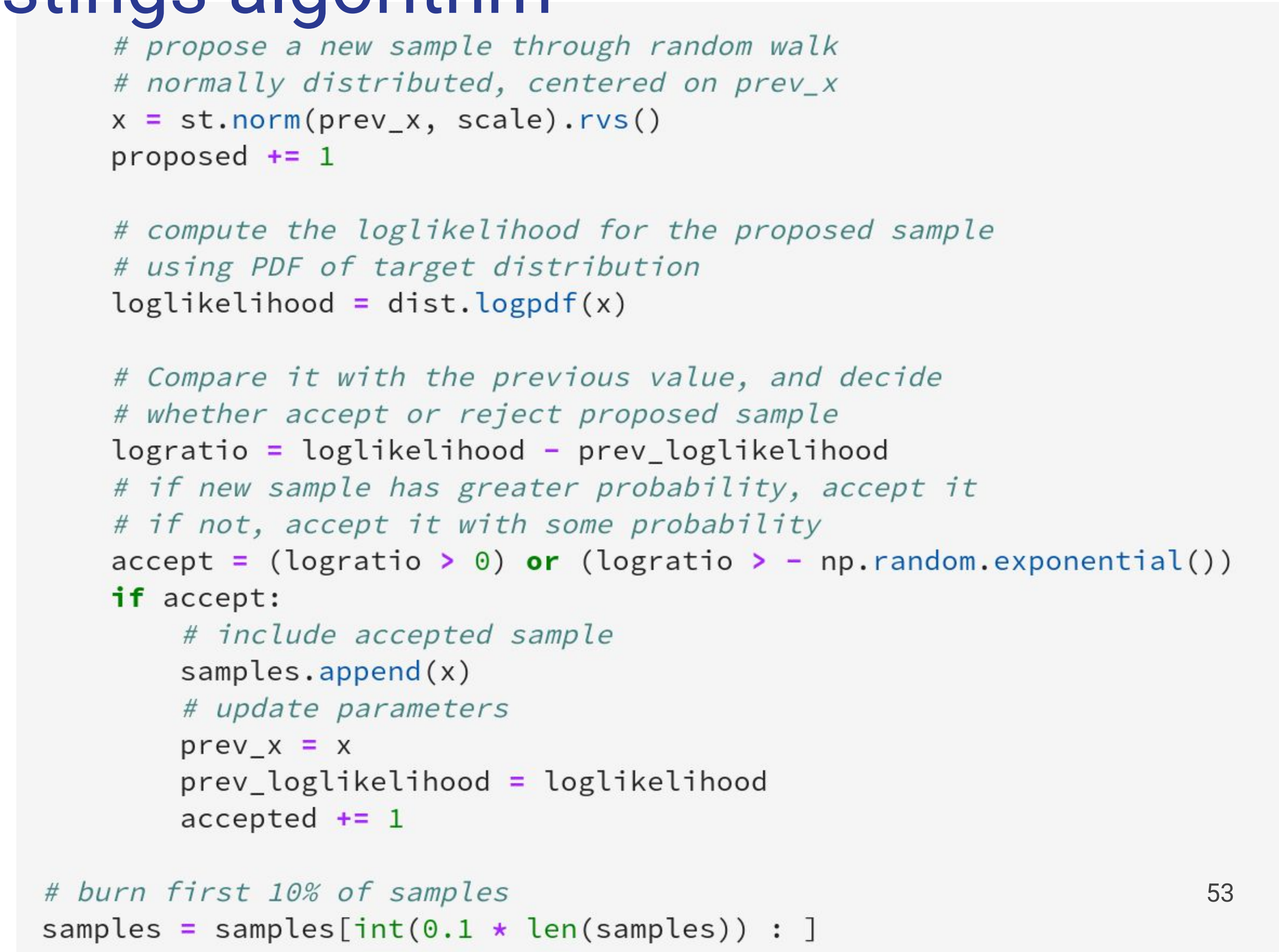


An example of Metropolis-Hastings

Sampling from arbitrary distributions

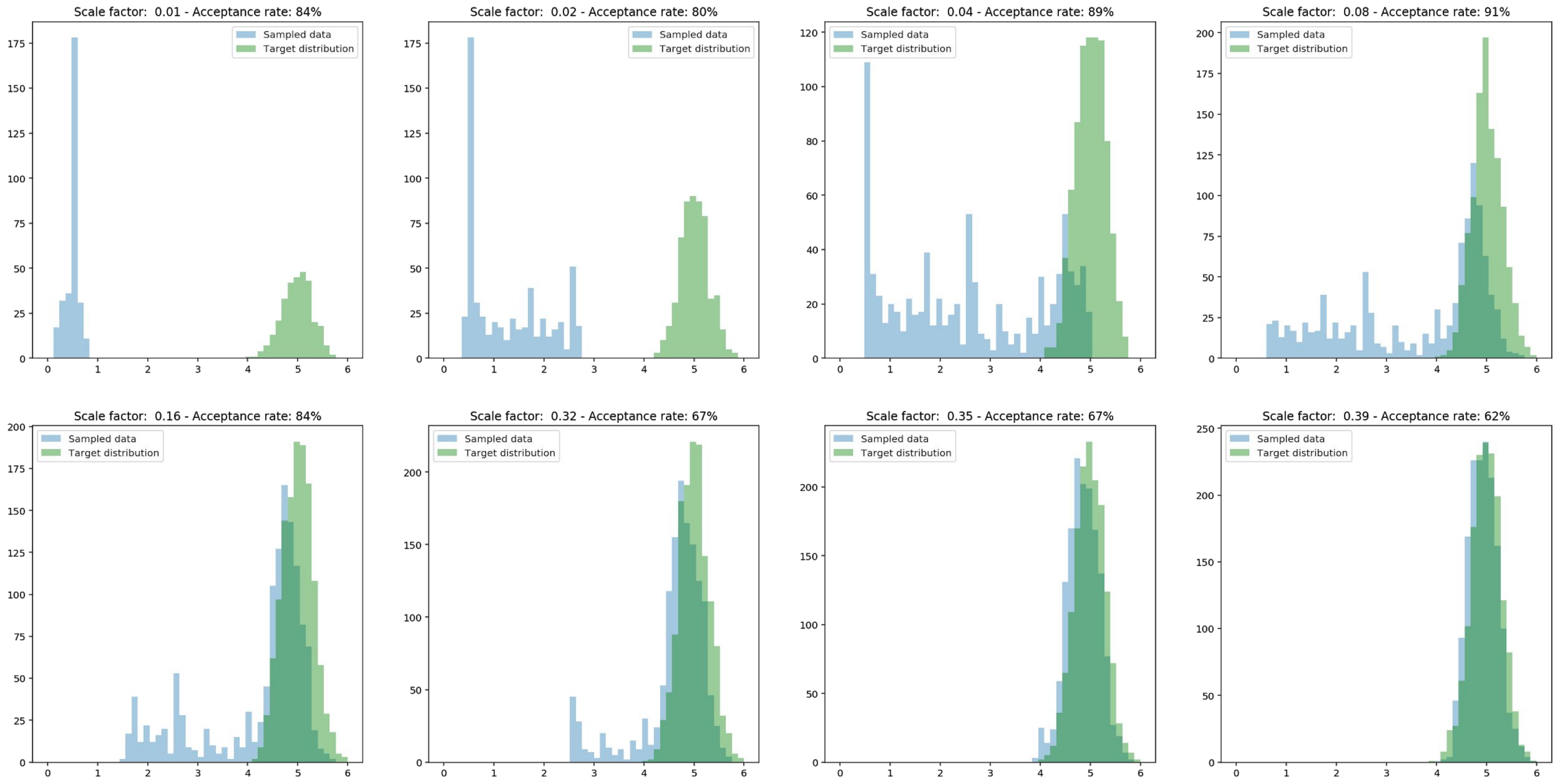
(You'll still need to compute likelihoods)

Metropolis-Hastings algorithm

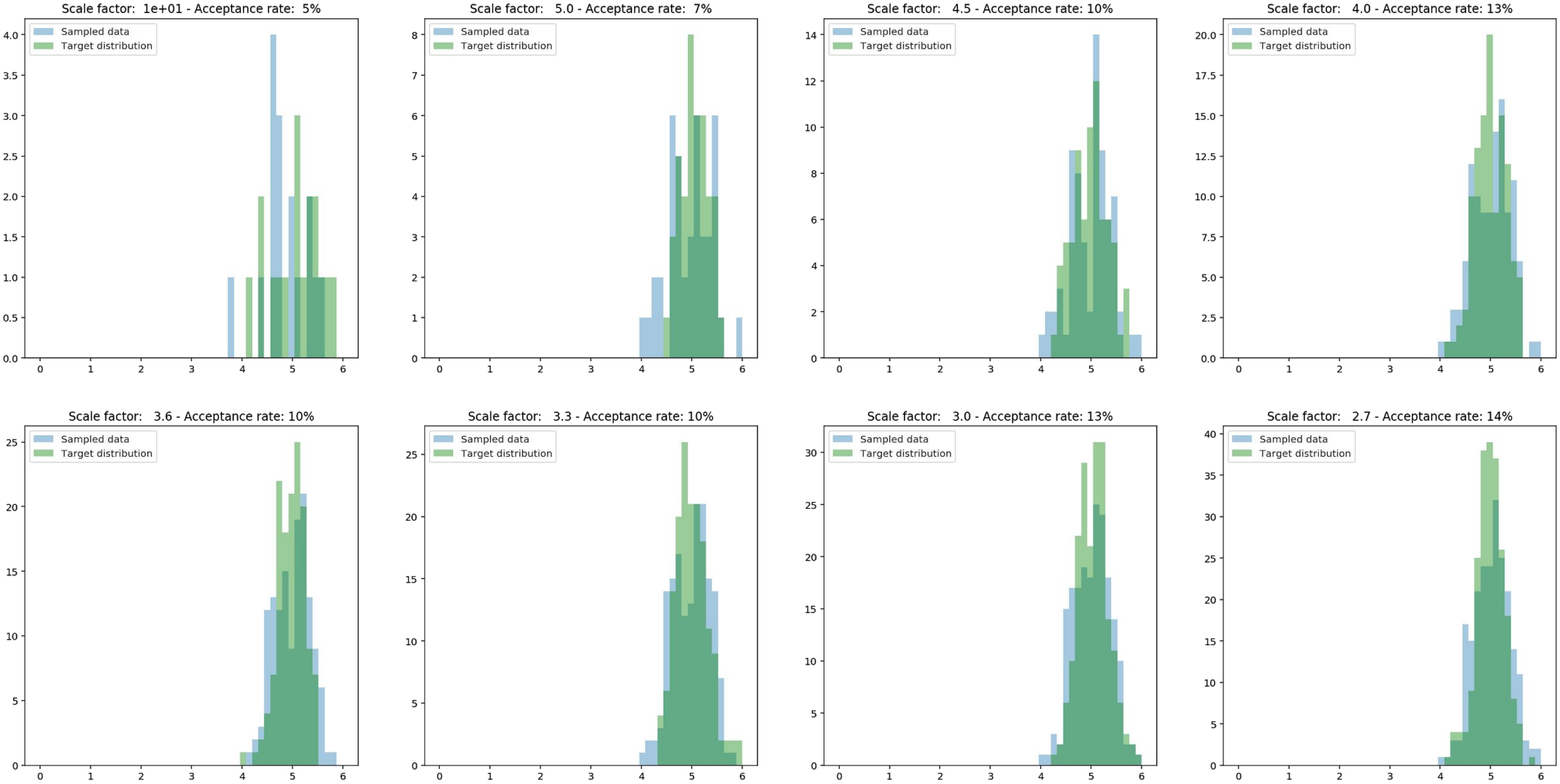


```
    # propose a new sample through random walk
    # normally distributed, centered on prev_x
    x = st.norm(prev_x, scale).rvs()
    proposed += 1

    # compute the loglikelihood for the proposed sample
    # using PDF of target distribution
    loglikelihood = dist.logpdf(x)

    # Compare it with the previous value, and decide
    # whether accept or reject proposed sample
    logratio = loglikelihood - prev_loglikelihood
    # if new sample has greater probability, accept it
    # if not, accept it with some probability
    accept = (logratio > 0) or (logratio > - np.random.exponential())
    if accept:
        # include accepted sample
        samples.append(x)
        # update parameters
        prev_x = x
        prev_loglikelihood = loglikelihood
        accepted += 1

# burn first 10% of samples
samples = samples[int(0.1 * len(samples)) : ]
```

Use of logarithms is often needed because computed probability densities may be very small for numeric precision



An example of MCMC with Metropolis-Hastings sampling



An example of MCMC with Metropolis-Hastings sampling

Q&A

Thank you