Mathematicians against gene-regulatory networks

Alessia Annibale

Mathematics, King's College London

DISORDERED SYSTEMS DAYS AT KING'S COLLEGE LONDON

A workshop on disorder to celebrate Reimer Kühn

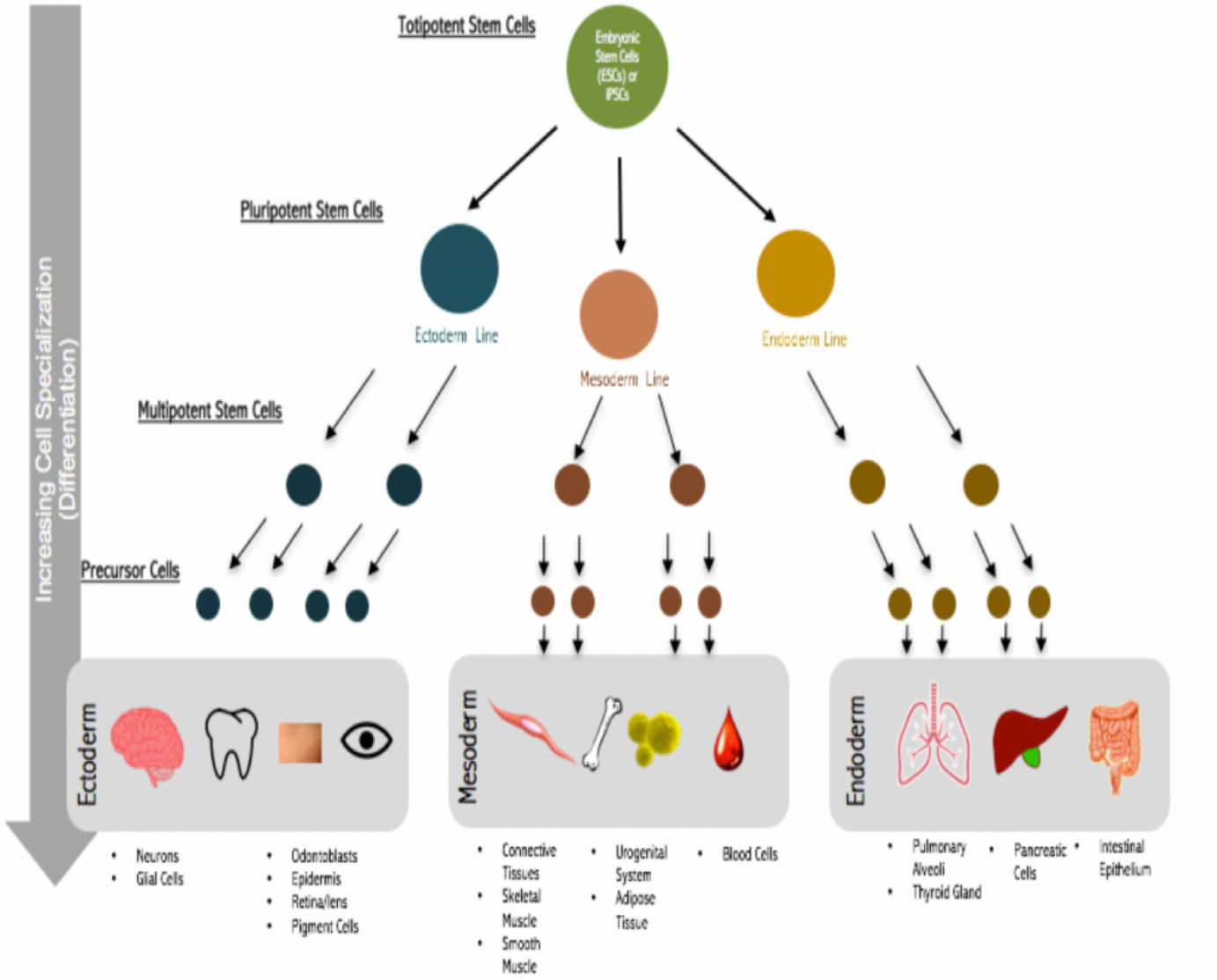
11-12 September 2023

Outline

Outline

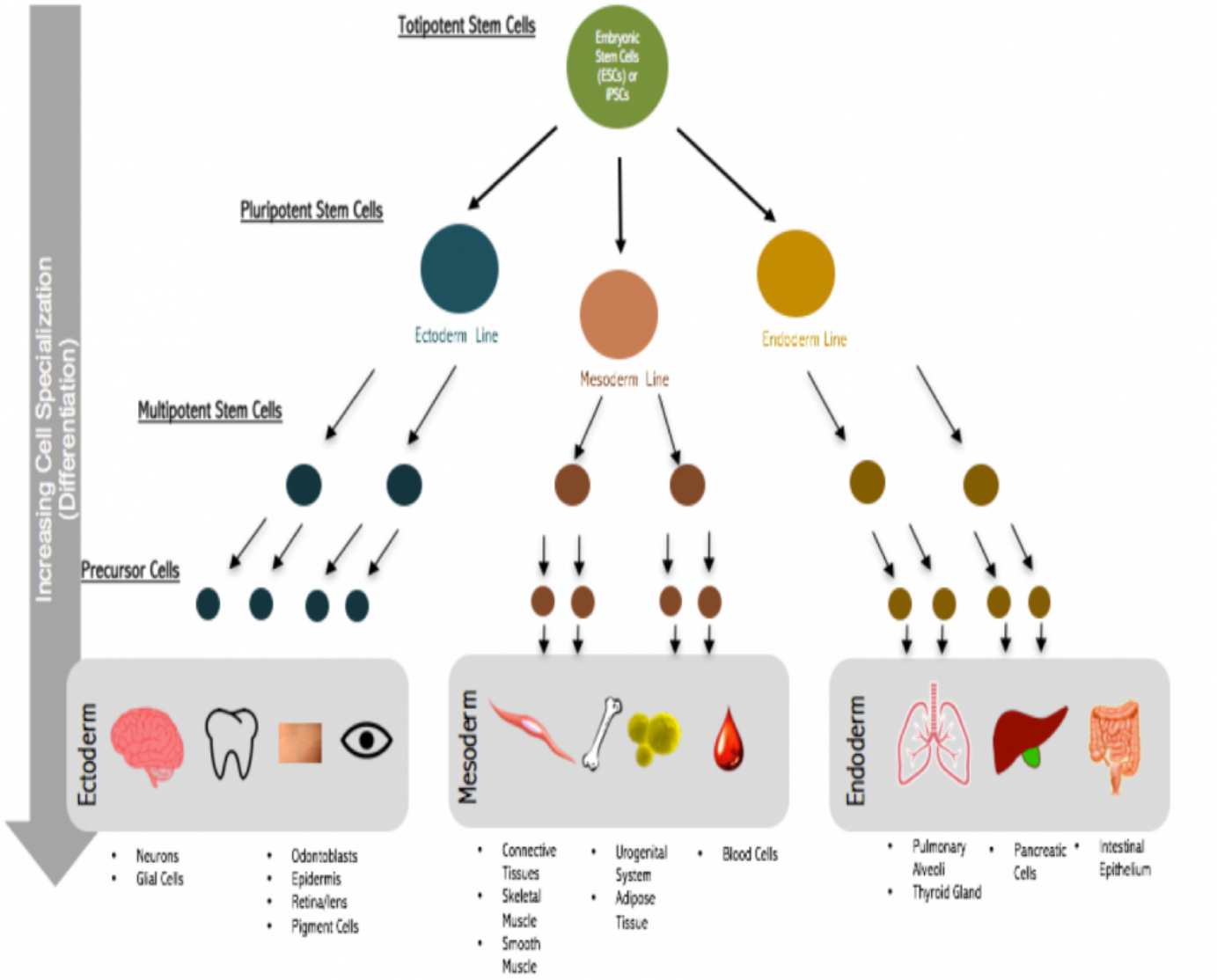
Cell differentiation

Cell differentiation



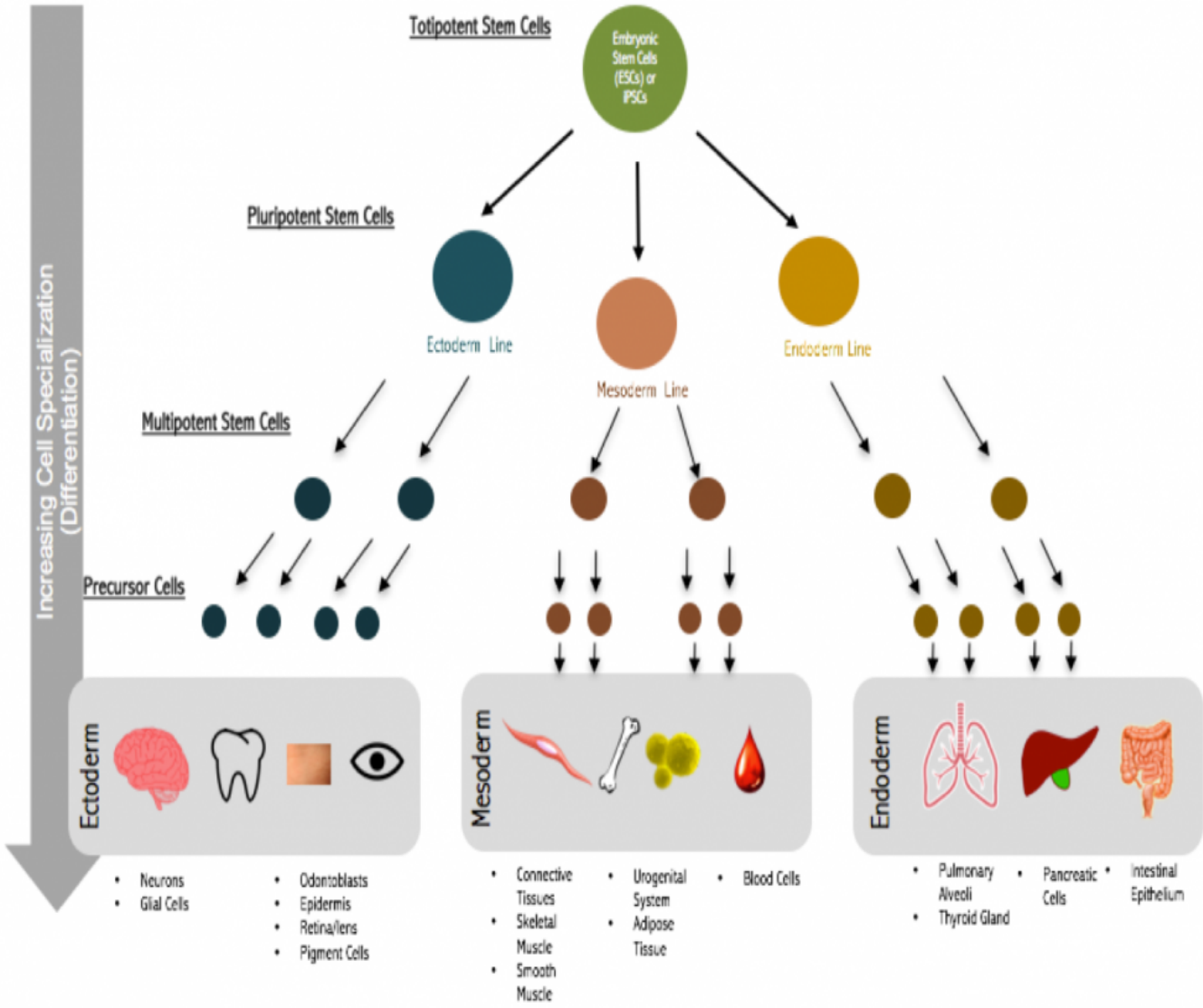
Source: ARK Investment Management LLC | ark-invest.com

Cell differentiation



Source: ARK Investment Management LLC | ark-invest.com

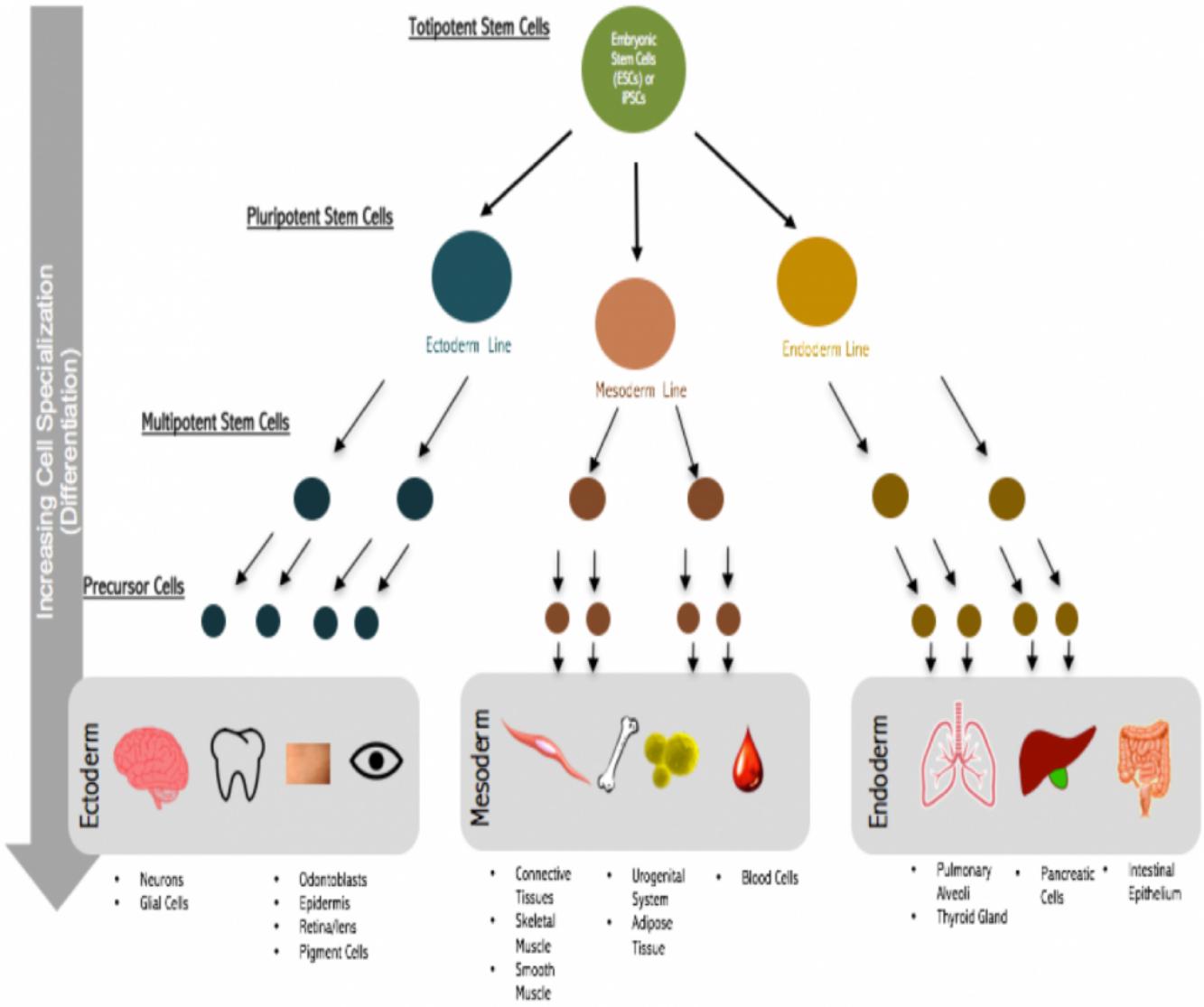
Cell differentiation



Source: ARK Investment Management LLC | ark-invest.com

- Each cell contains the same genes, $N \sim 25000$

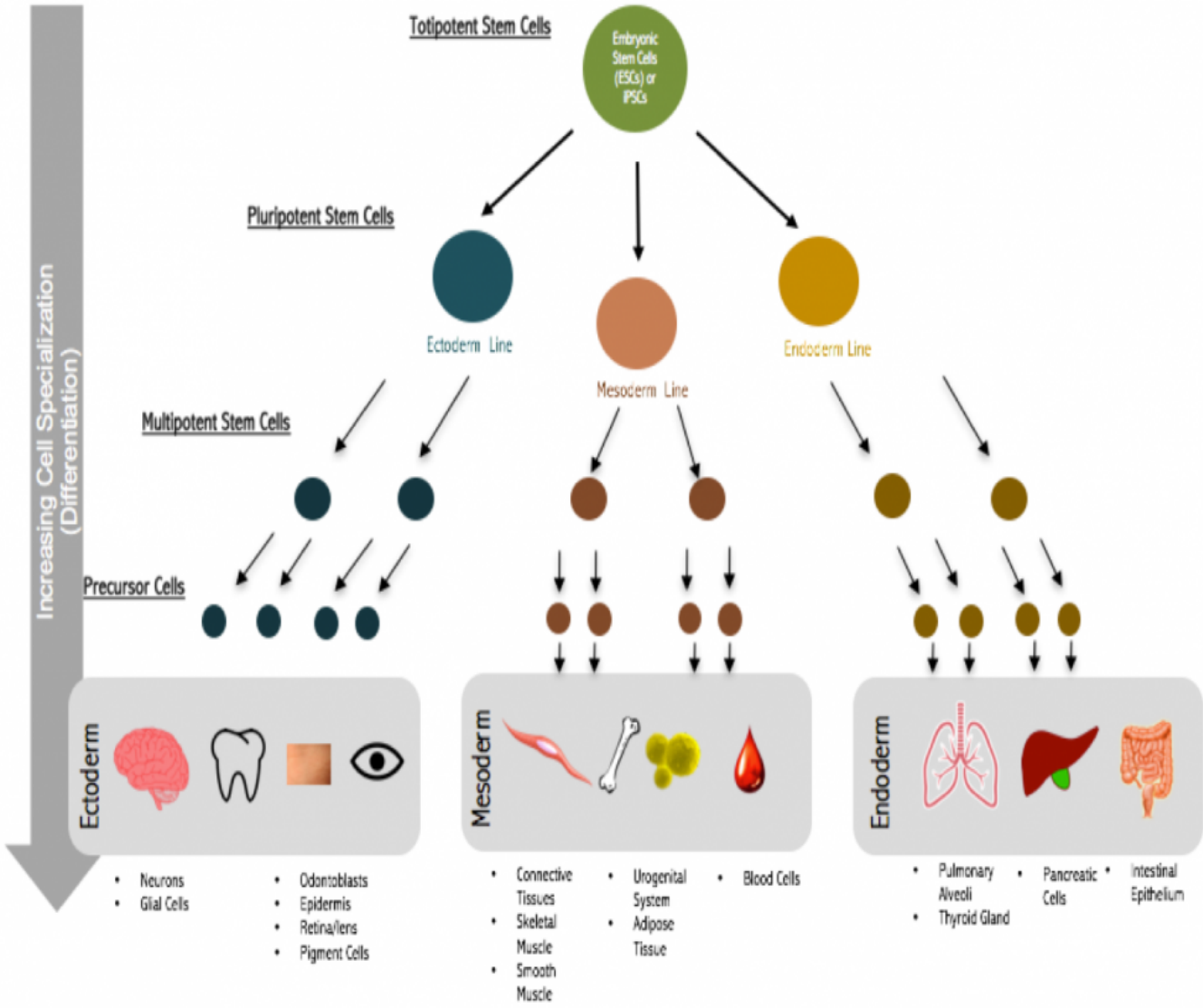
Cell differentiation



Source: ARK Investment Management LLC | ark-invest.com

- Each cell contains the same genes, $N \sim 25000$
- Different cells express different genes

Cell differentiation



Source: ARK Investment Management LLC | ark-invest.com

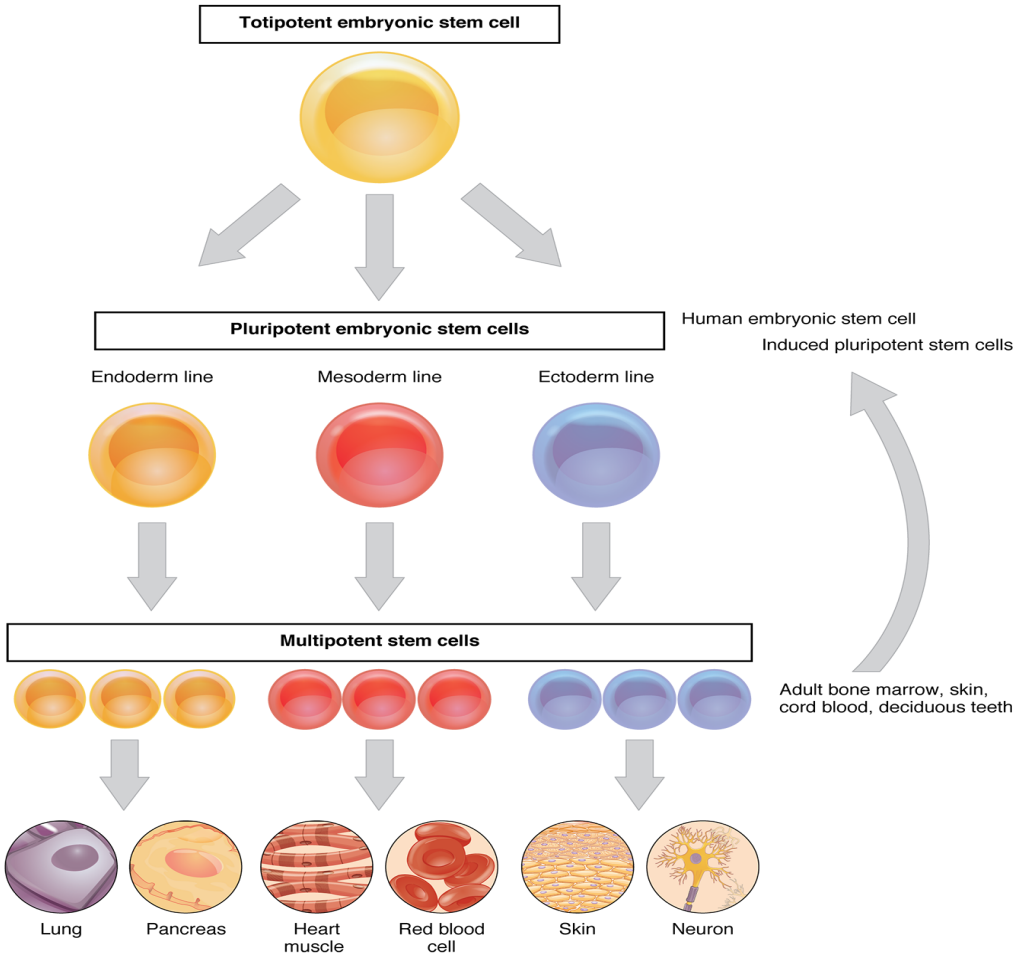
- Each cell contains the same genes, $N \sim 25000$
- Different cells express different genes
- Transcription factors (TFs) regulate expression

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

Cell reprogramming: Takahashi and Yamanaka (2006)

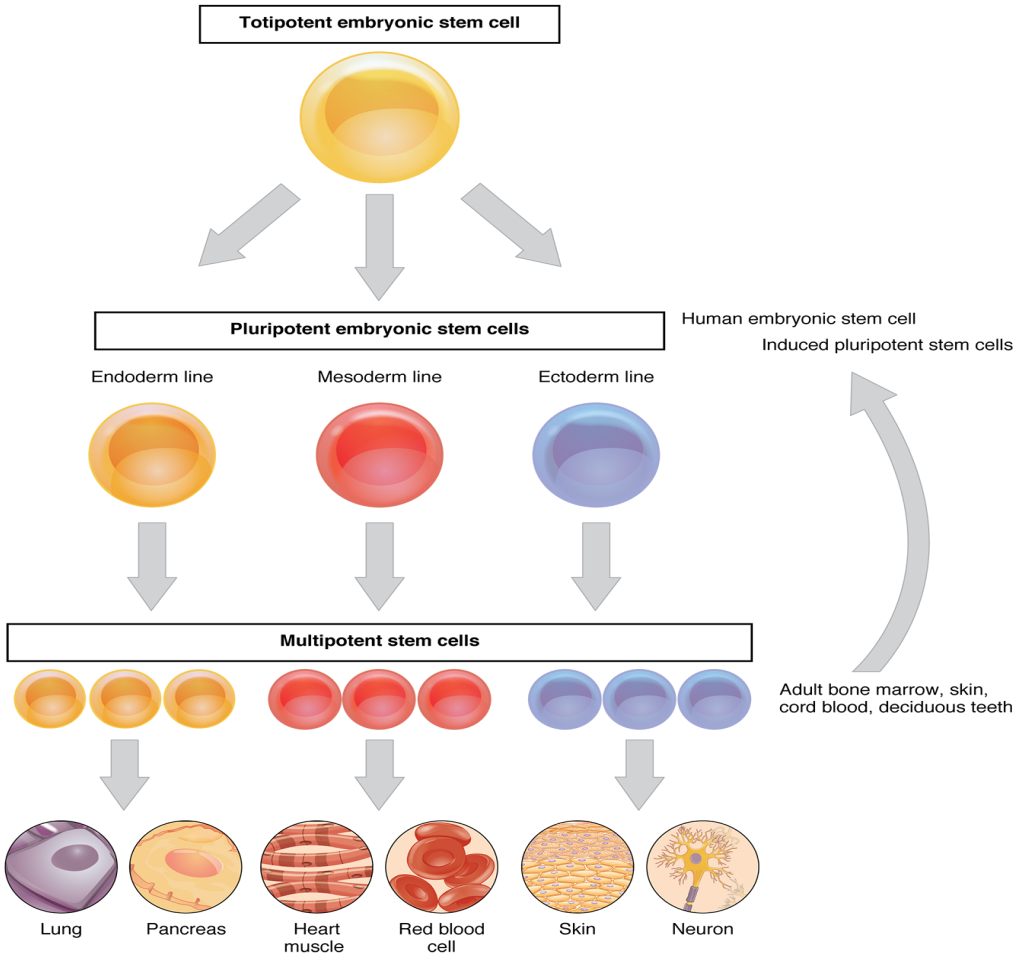
Introduce **4 TFs**, 'Yamanaka factors', into somatic cells



Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

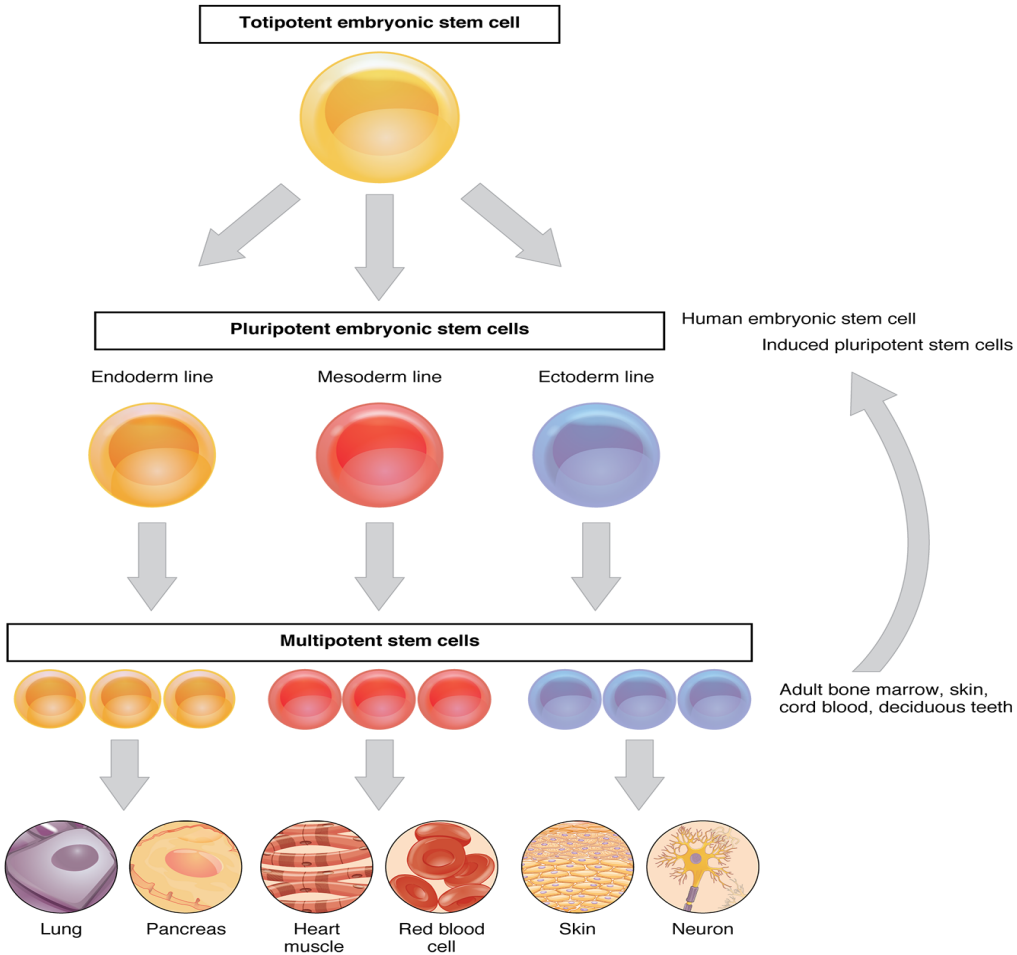
Introduce **4 TFs**, 'Yamanaka factors', into somatic cells



Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

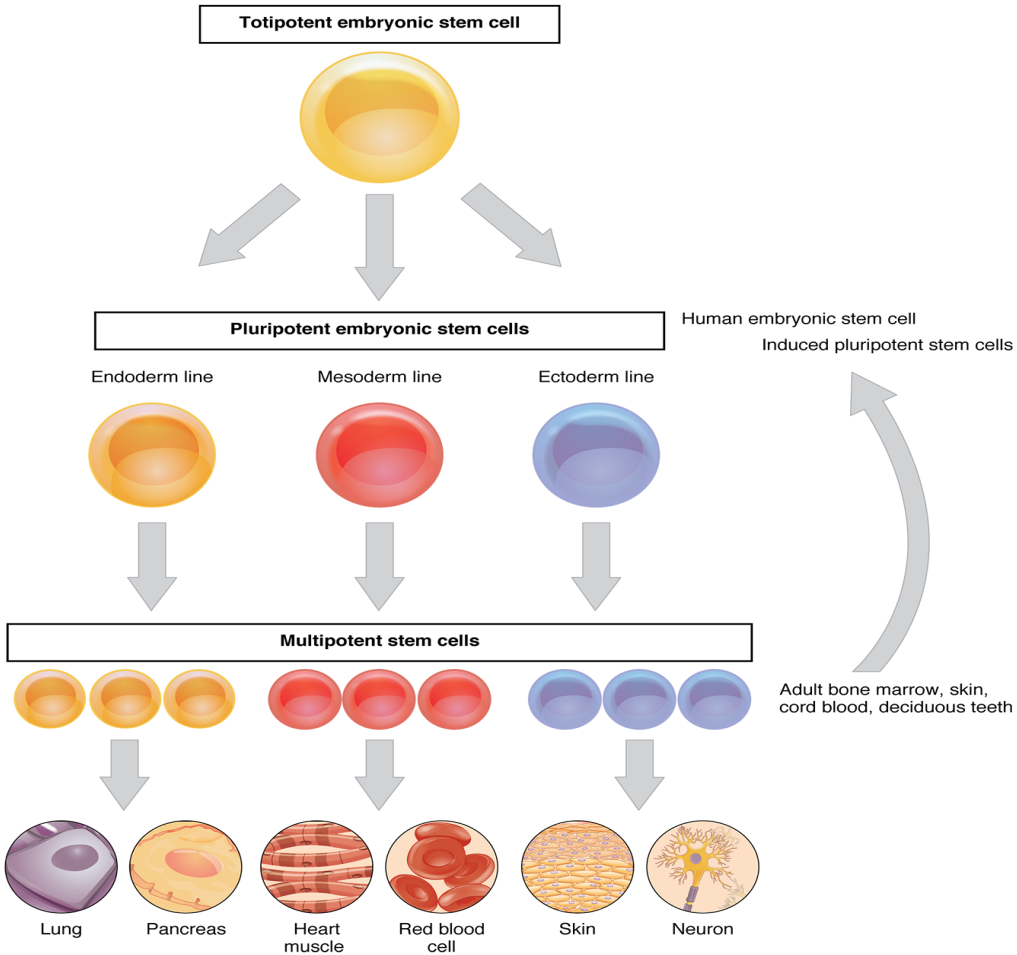
Introduce **4 TFs**, 'Yamanaka factors', into somatic cells



Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

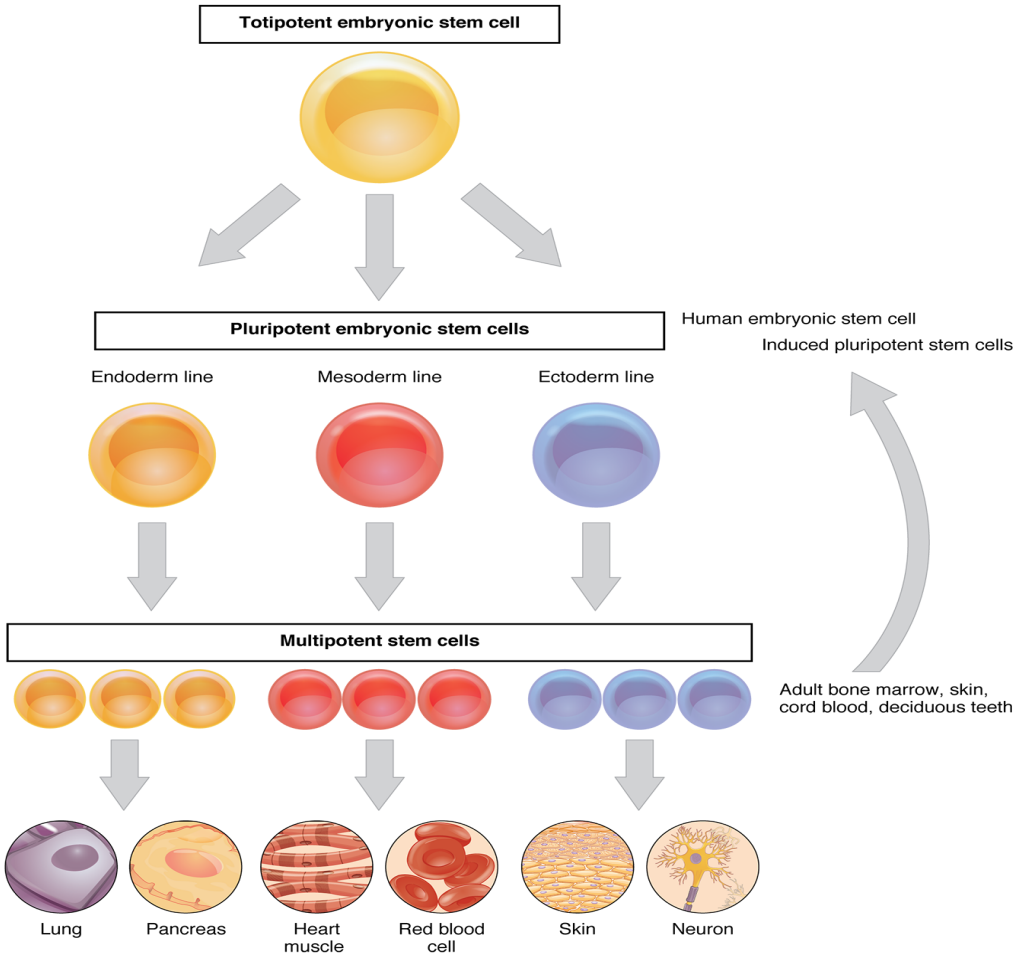


- Nobel prize 2012 for Physiology or Medicine

Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

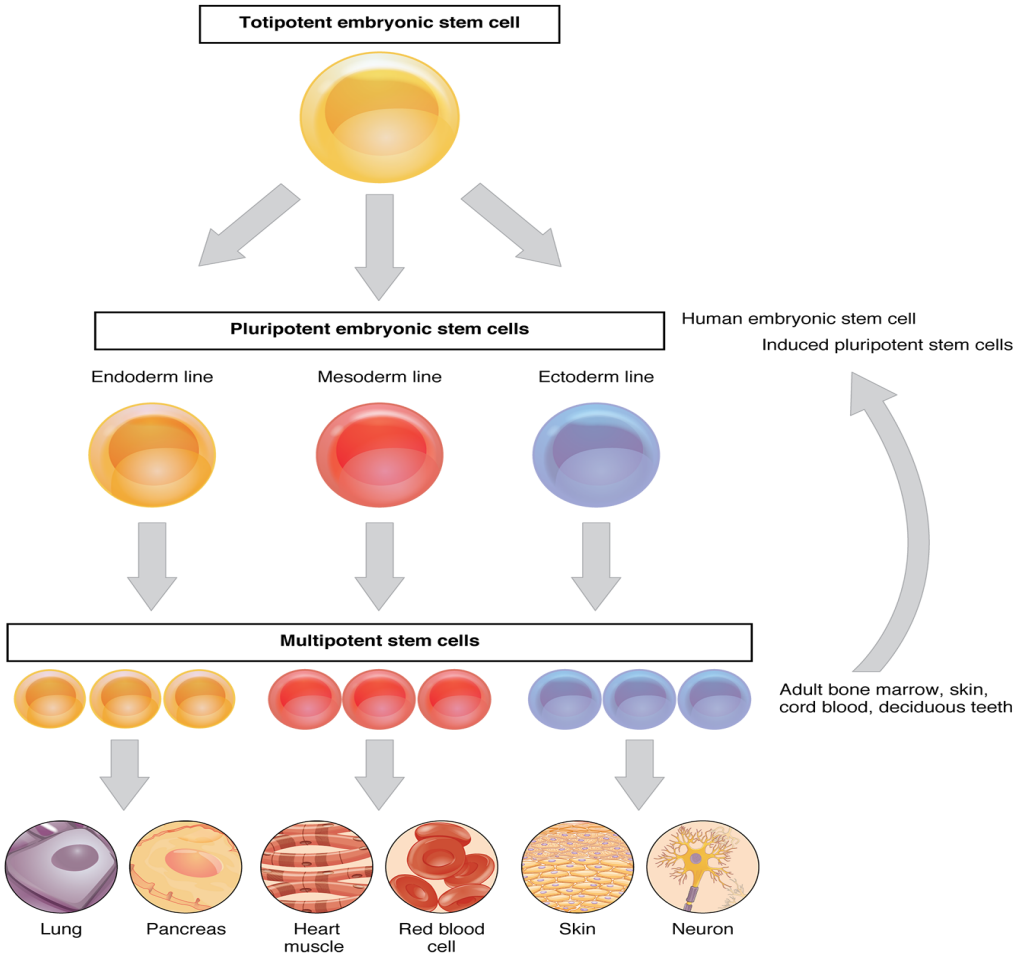


- Nobel prize 2012 for Physiology or Medicine
- $\mathcal{O}(10)$ **days** to reprogram

Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

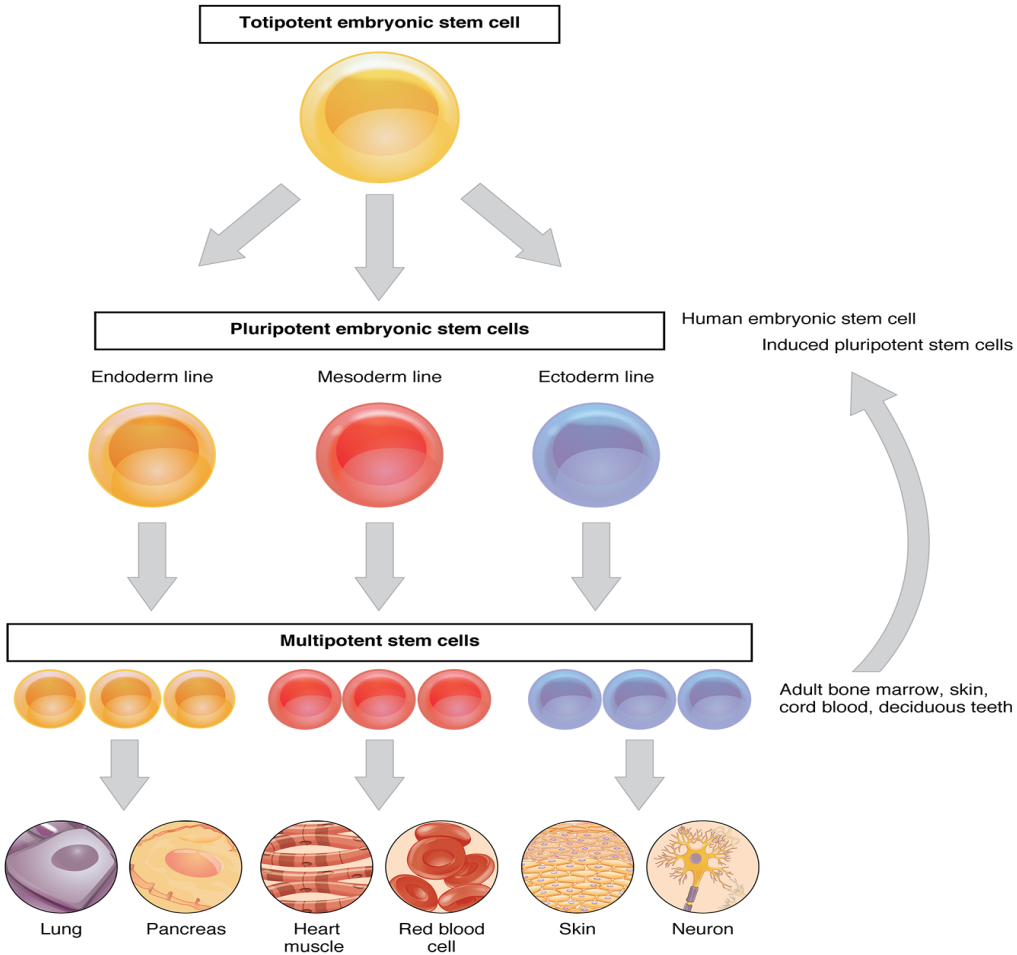


- Nobel prize 2012 for Physiology or Medicine
- $\mathcal{O}(10)$ **days** to reprogram
- many cells take '**bad trajectories**' after reprogramming.. e.g. cancer

Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

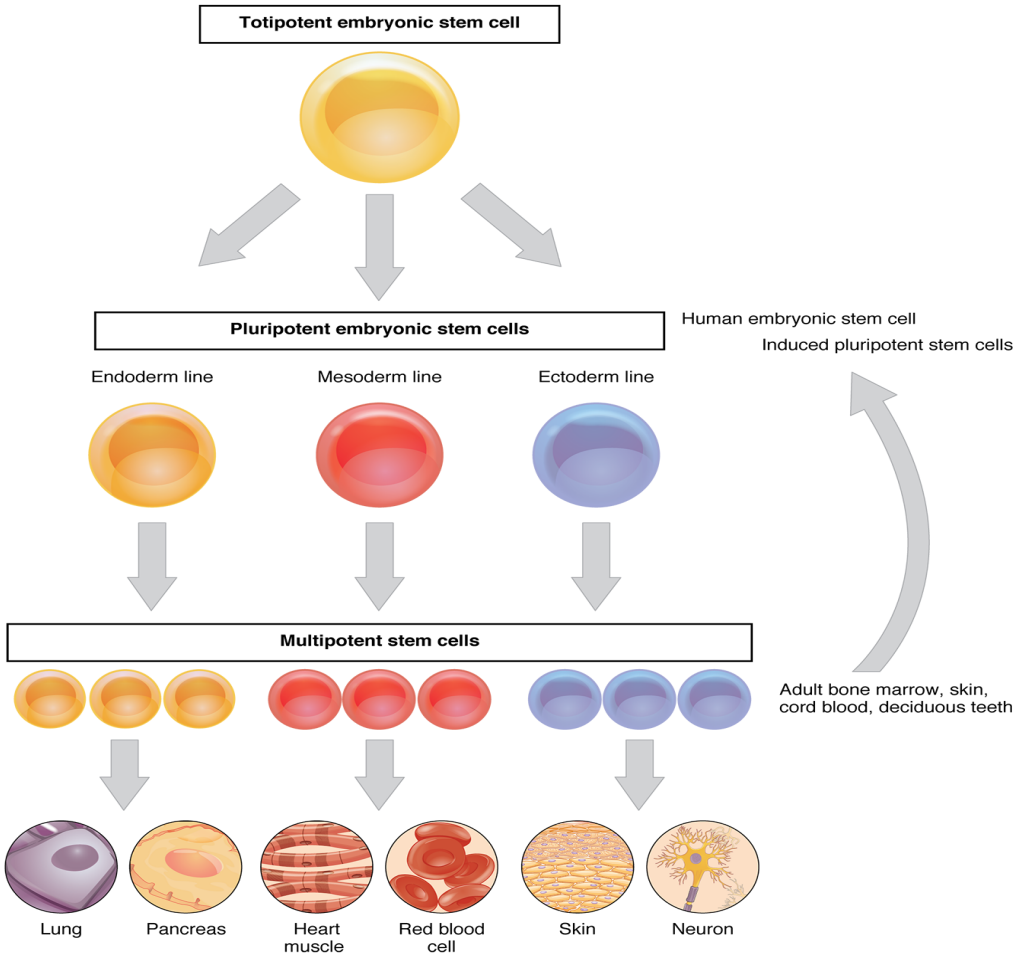


- Nobel prize 2012 for Physiology or Medicine
- $\mathcal{O}(10)$ **days** to reprogram
- many cells take '**bad trajectories**' after reprogramming.. e.g. cancer
- **Q:** How to control cell fate?

Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

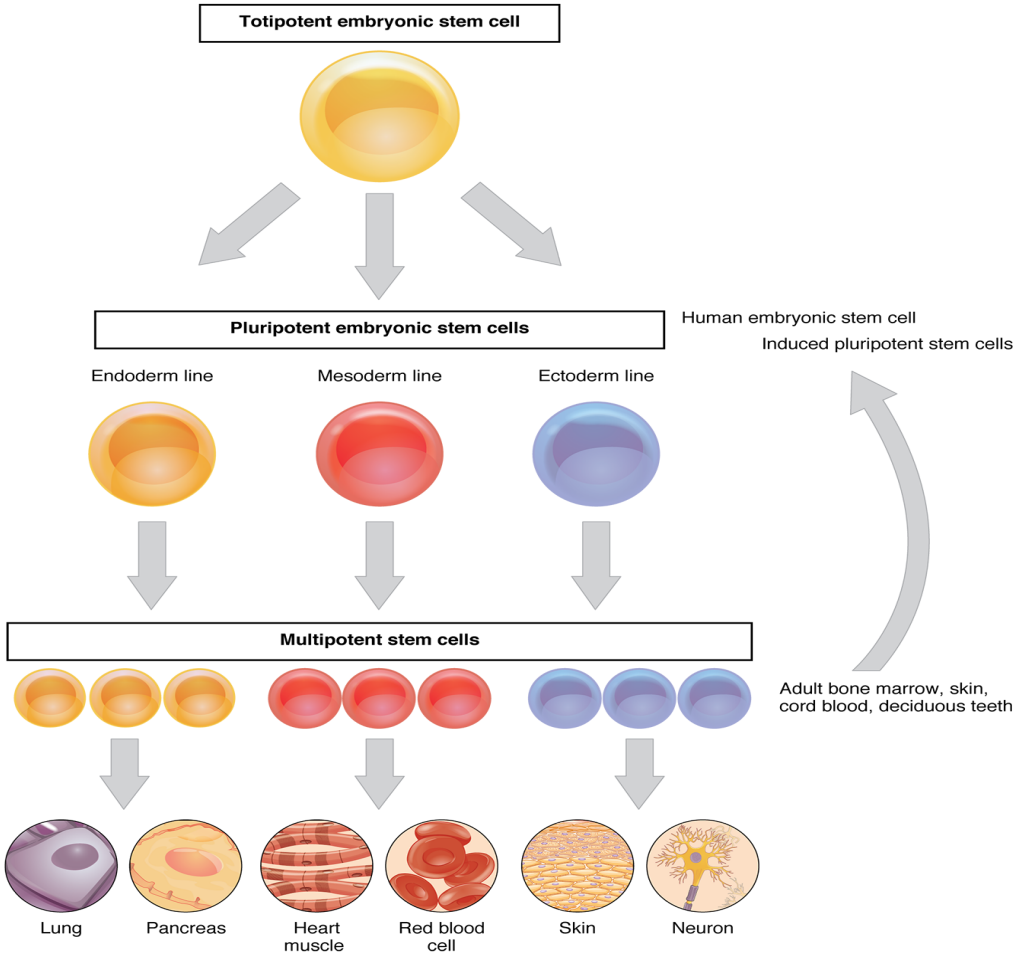


- Nobel prize 2012 for Physiology or Medicine
- $\mathcal{O}(10)$ **days** to reprogram
- many cells take '**bad trajectories**' after reprogramming.. e.g. cancer
- **Q:** How to control cell fate?
- huge dim: 'on/off' genes $\Rightarrow 2^{25,000}$ possible gene patterns

Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells

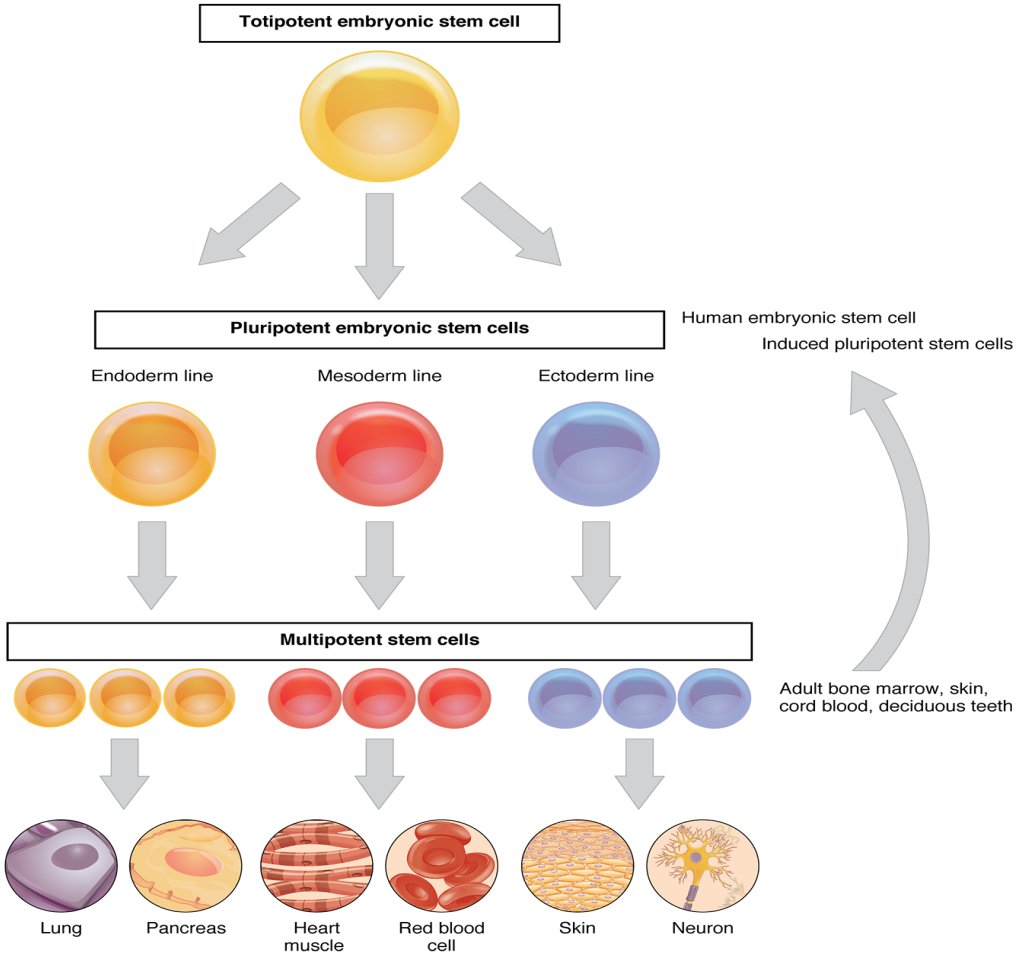


- Nobel prize 2012 for Physiology or Medicine
- $\mathcal{O}(10)$ **days** to reprogram
- many cells take '**bad trajectories**' after reprogramming.. e.g. cancer
- **Q:** How to control cell fate?
- huge dim: 'on/off' genes $\Rightarrow 2^{25,000}$ possible gene patterns
- Striking that we **only** have ~ 300 cell types..

Source: Cell Transfection — cell-transfection.com

Cell reprogramming: Takahashi and Yamanaka (2006)

Introduce **4 TFs**, 'Yamanaka factors', into somatic cells



Source: Cell Transfection — cell-transfection.com

- Nobel prize 2012 for Physiology or Medicine
- $\mathcal{O}(10)$ **days** to reprogram
- many cells take '**bad trajectories**' after reprogramming.. e.g. cancer
- **Q:** How to control cell fate?
- huge dim: 'on/off' genes $\Rightarrow 2^{25,000}$ possible gene patterns
- Striking that we **only** have ~ 300 cell types..
- Idea: cell types are attractors of **gene dynamics**, like memories for **neural dynamics**..

Outline

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases}$$

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\left[h_i(t) - z_i(t)\right]$$

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Preliminary observation

- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \leq x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]

Preliminary observation

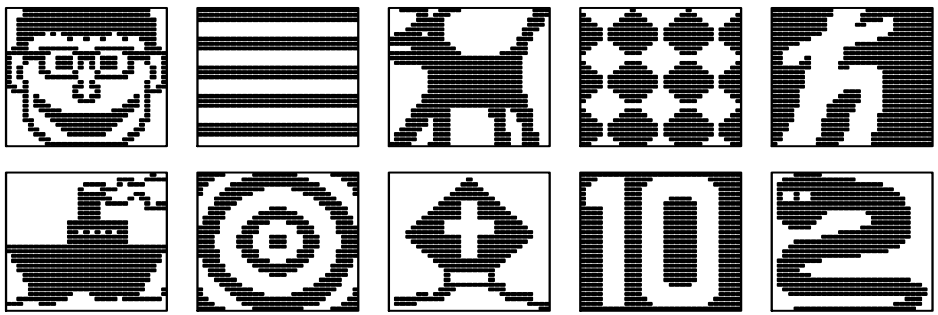
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



Theory of Neuronal Information Procesing, Coolen, Kühn, Sollich

Preliminary observation

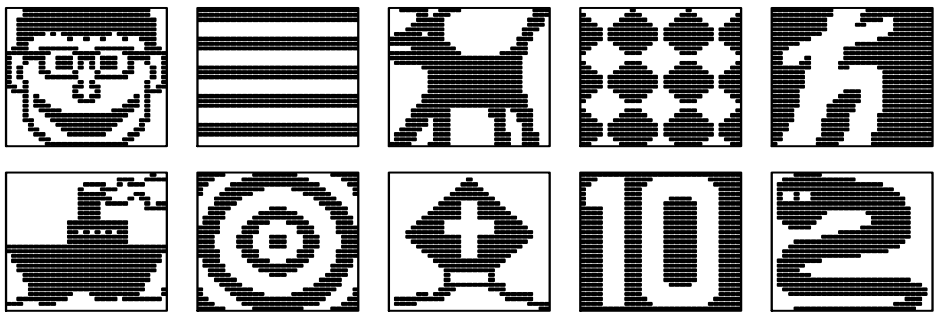
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



N pixels, $i = 1, \ldots, N$

$$\xi_i^\mu = \begin{cases} 1 & \square \\ -1 & \blacksquare \end{cases}$$

Preliminary observation

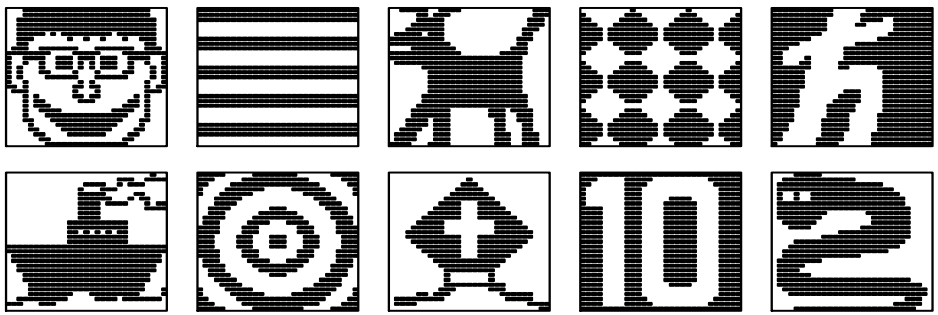
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



N pixels, $i = 1, \ldots, N$

$$\xi_i^\mu = \begin{cases} 1 & \square \\ -1 & \blacksquare \end{cases}$$

Preliminary observation

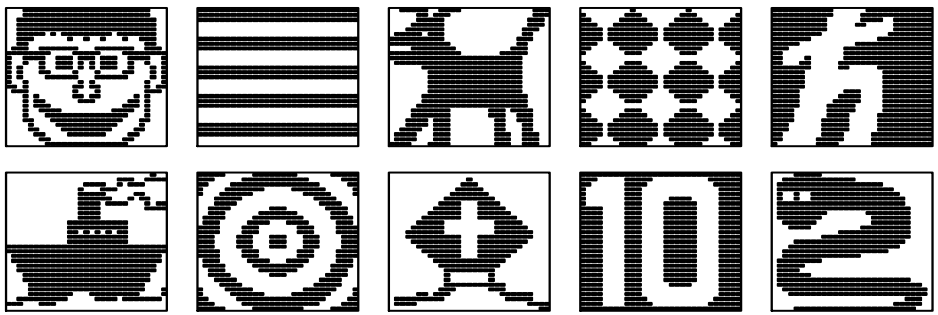
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



N pixels, $i = 1, \ldots, N$

$$\xi_i^\mu = \begin{cases} 1 & \square \\ -1 & \blacksquare \end{cases}$$

Preliminary observation

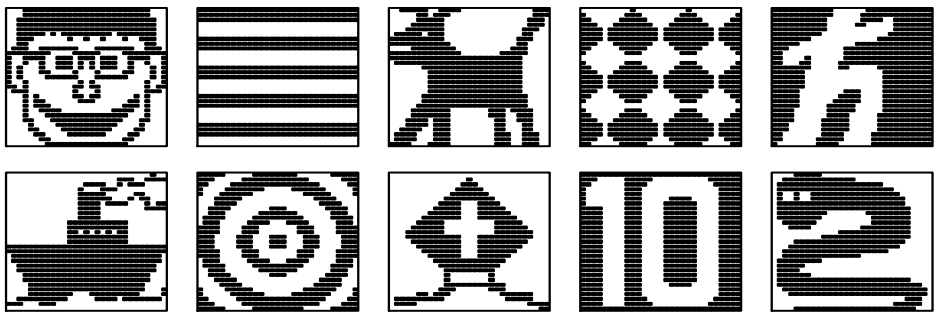
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



N pixels, $i = 1, \ldots, N$

$$\xi_i^\mu = \begin{cases} 1 & \square \\ -1 & \blacksquare \end{cases}$$

Theory of Neuronal Information Procesing, Coolen, Kühn, Sollich

$$J_{ij} = P^{-1} \sum_\mu \xi_i^\mu \xi_j^\mu$$

Preliminary observation

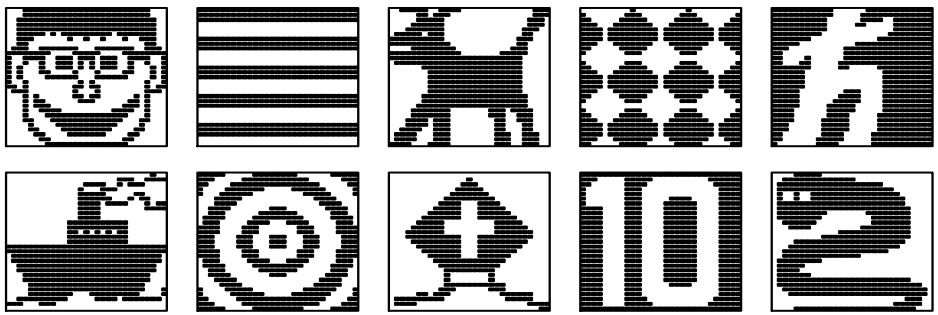
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



Theory of Neuronal Information Procesing, Coolen, Kühn, Sollich

N pixels, $i = 1, \ldots, N$

$$\xi_i^\mu = \begin{cases} 1 & \square \\ -1 & \blacksquare \end{cases}$$

$$J_{ij} = P^{-1} \sum_\mu \xi_i^\mu \xi_j^\mu$$

$$\boldsymbol{\sigma}(0) \to \ldots \to \boldsymbol{\xi}^\rho$$

Preliminary observation

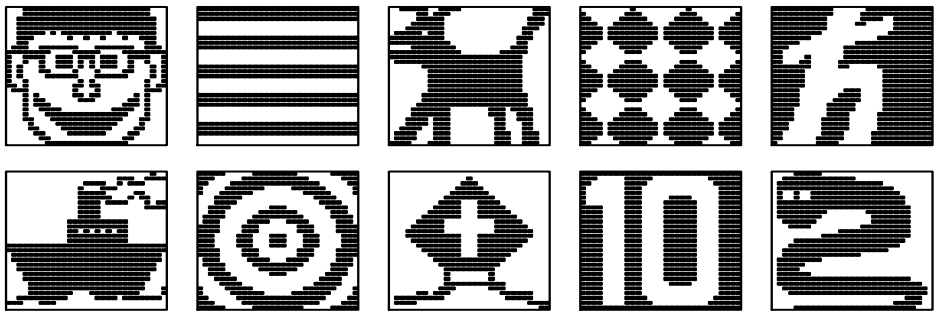
- If I knew everything about cell.. could write equations for all chemical reactions in a cell (genes, mRNA, proteins, metabolites...).
- Suppose I integrate out all degrees of freedom **except** genes $\Rightarrow$ reduced system of interacting genes $h_i(t) = \sum_j J_{ij} n_j(t)$

$$J_{ij} = \begin{cases} > 0 & \text{promote} \\ < 0 & \text{inhibit} \end{cases} \qquad n_i \in \{0,1\}: \quad n_i(t+1) = \Theta\Big[h_i(t) - z_i(t)\Big]$$

$$\text{Prob}[z \le x] = \Phi_T(x), \quad T = \text{noise level}$$

- Neural Networks $\sigma_i = \pm 1$: $\sigma_i(t+1) = \text{sgn}\Big[\sum_j J_{ij}\sigma_j(t) - z_i(t)\Big]$

Store multiple patterns $\{\boldsymbol{\xi}^1, \ldots, \boldsymbol{\xi}^P\}$ [Hopfield (1982)]



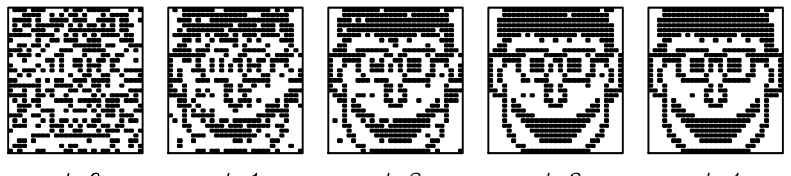
N pixels, $i = 1, \ldots, N$

$$\xi_i^\mu = \begin{cases} 1 & \square \\ -1 & \blacksquare \end{cases}$$

Theory of Neuronal Information Procesing, Coolen, Kühn, Sollich

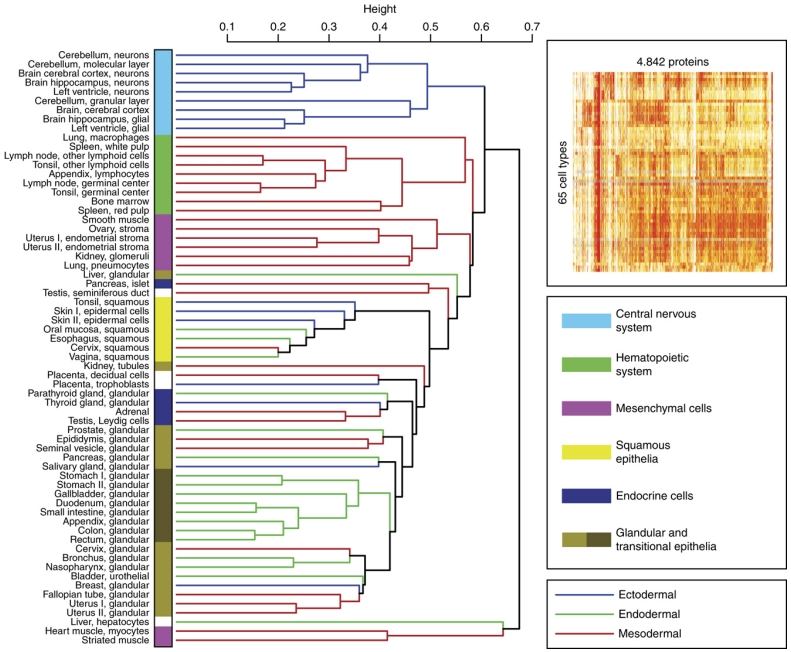
$$J_{ij} = P^{-1} \sum_\mu \xi_i^\mu \xi_j^\mu$$

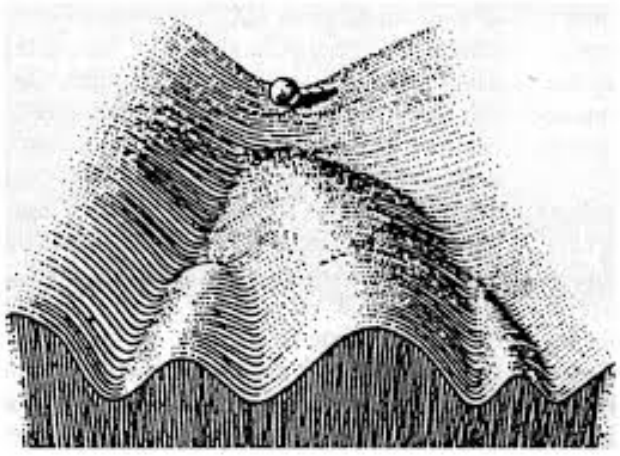
$$\boldsymbol{\sigma}(0) \to \ldots \to \boldsymbol{\xi}^\rho$$



Nature of cellular attractors

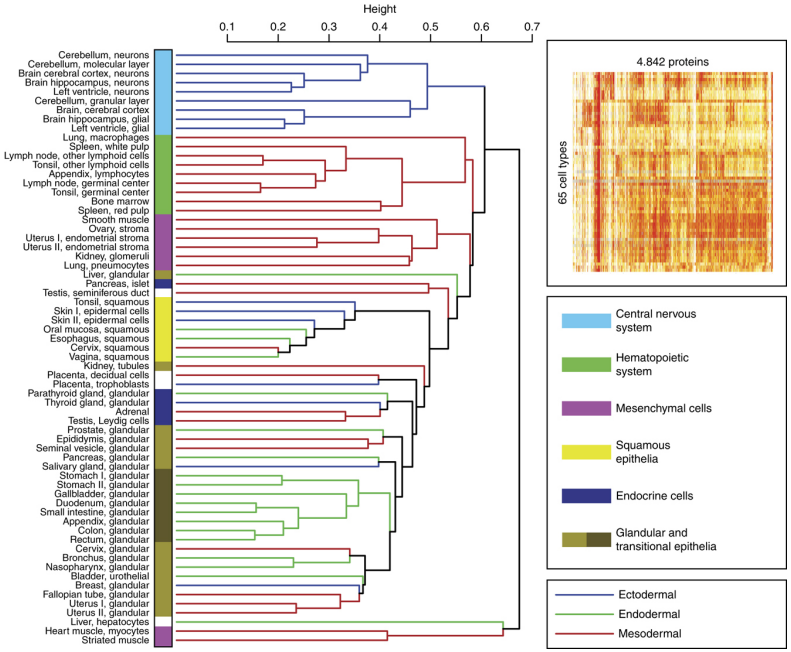
- cell types are hierarchically organized

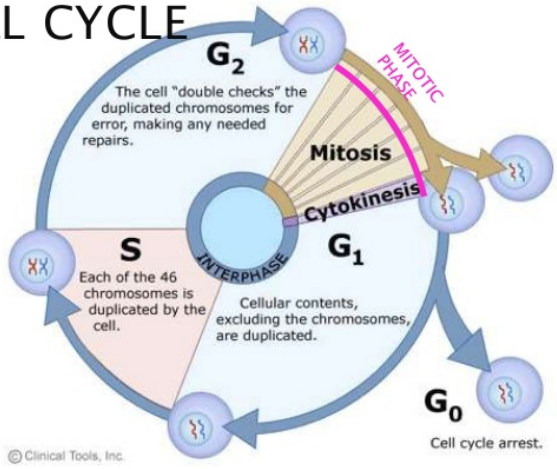




Nature of cellular attractors

- cell types are hierarchically organized

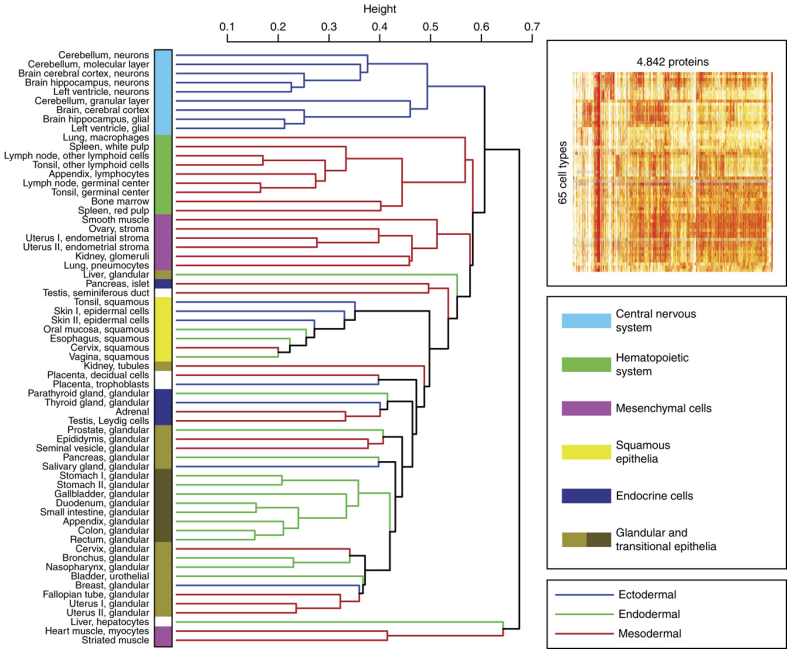


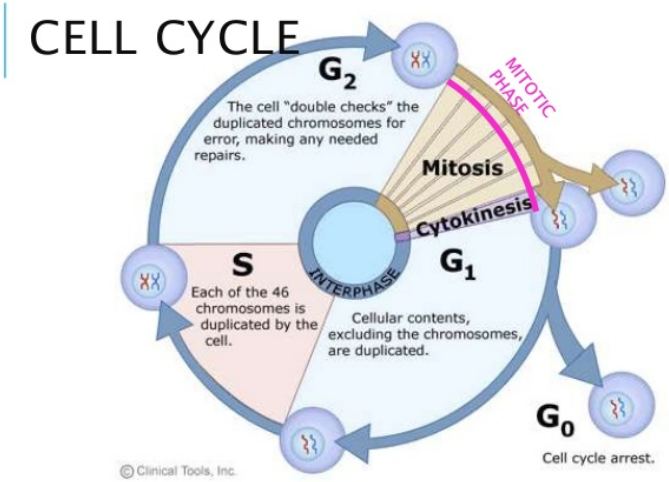


- dynamical entities i.e. they cycle ($G_1 \to S \to G_2 \to M \to G_1 \dots$)

Nature of cellular attractors

- cell types are hierarchically organized



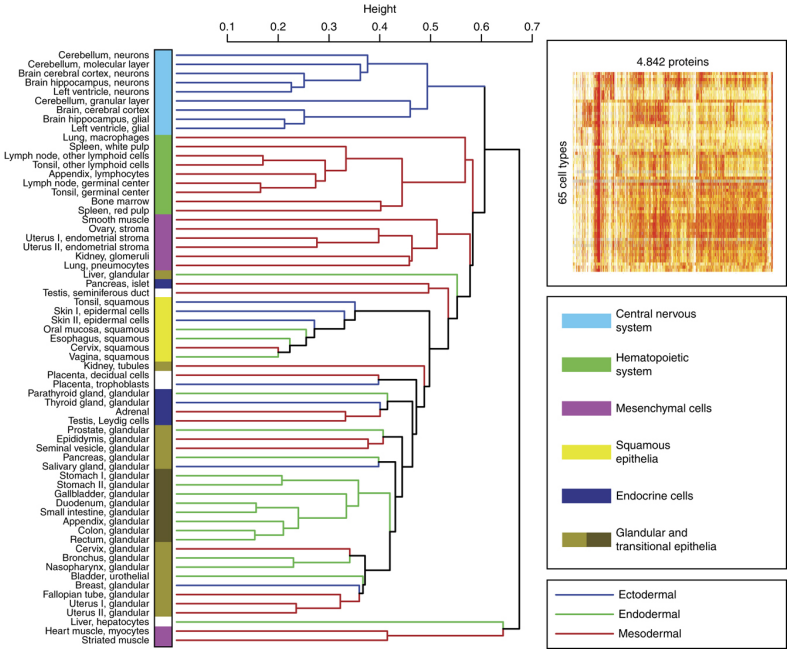


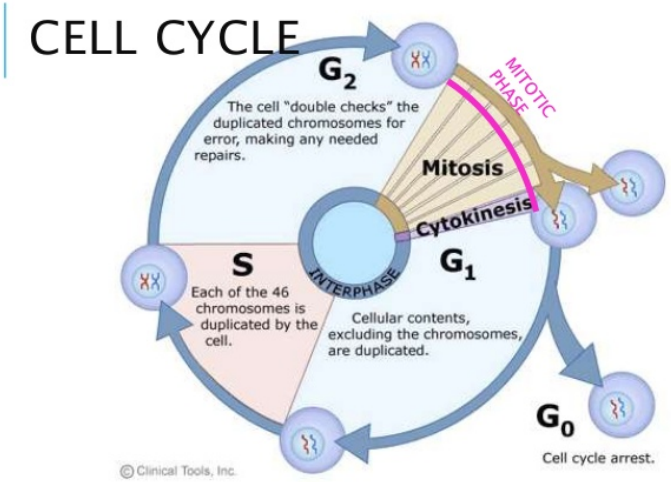
- dynamical entities i.e. they cycle ($G_1 \to S \to G_2 \to M \to G_1 \dots$)

$\Rightarrow$ Choose J_{ij} to encode **hierarchically organized cycles**

Nature of cellular attractors

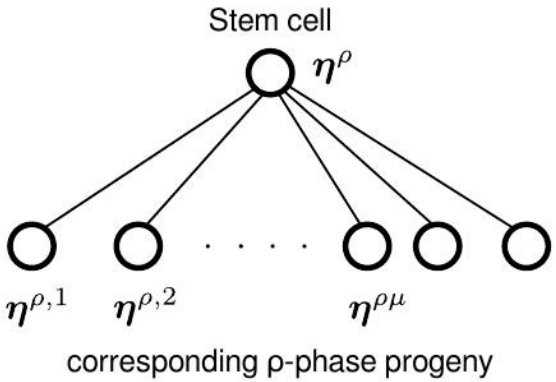
- cell types are hierarchically organized





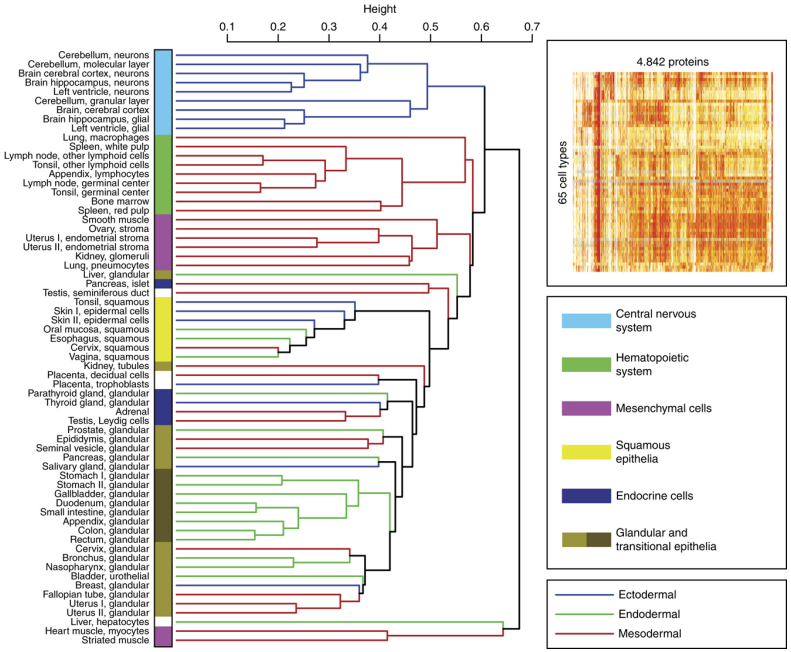
- dynamical entities i.e. they cycle ($G_1 \to S \to G_2 \to M \to G_1 \dots$)

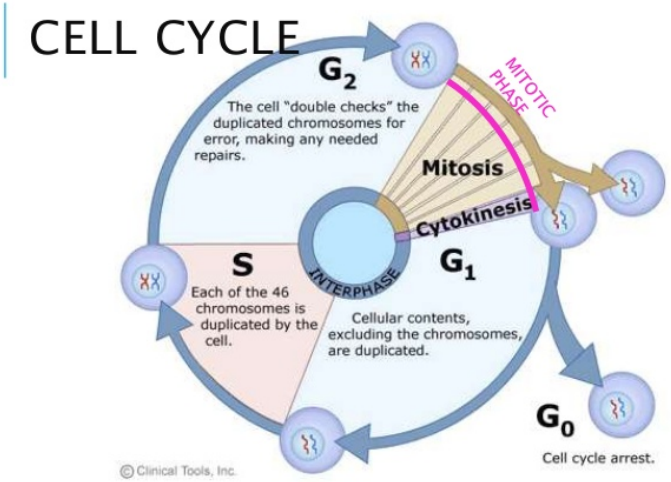
$\Rightarrow$ Choose J_{ij} to encode **hierarchically organized cycles**



Nature of cellular attractors

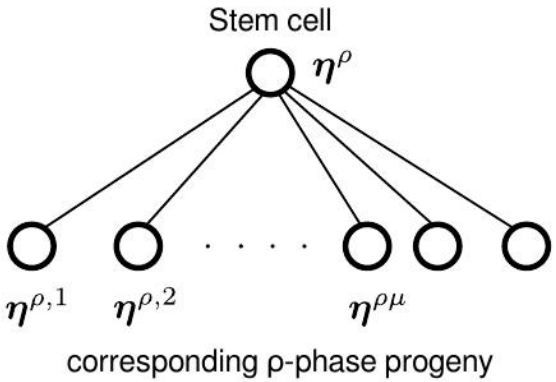
- cell types are hierarchically organized





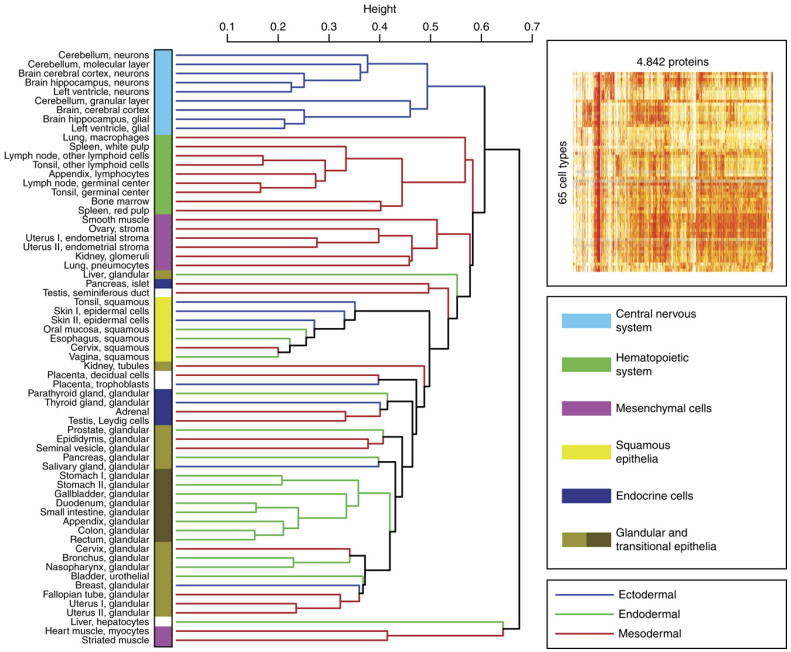
- dynamical entities i.e. they cycle ($G_1 \to S \to G_2 \to M \to G_1 \dots$)

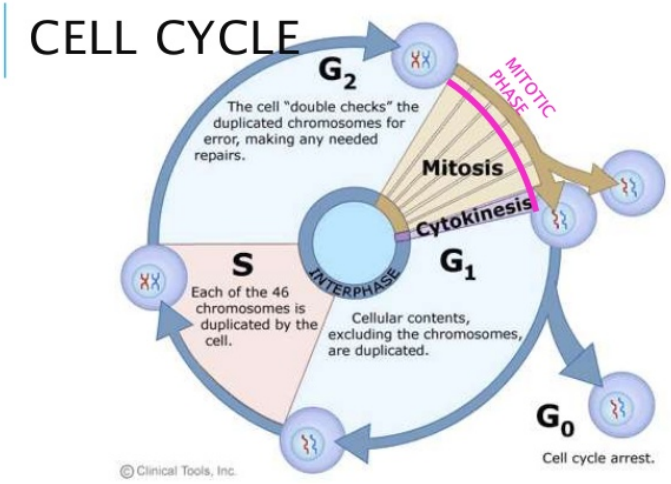
$\Rightarrow$ Choose J_{ij} to encode **hierarchically organized cycles**



Nature of cellular attractors

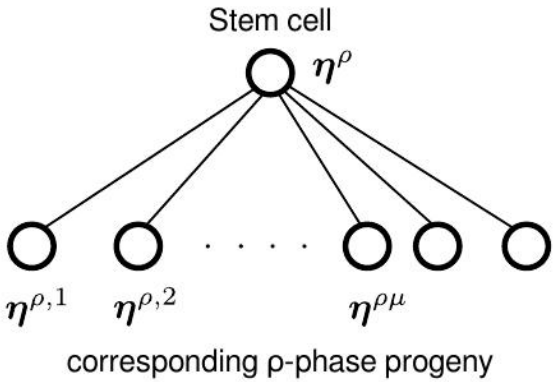
- cell types are hierarchically organized





- dynamical entities i.e. they cycle $(G_1 \to S \to G_2 \to M \to G_1 \dots)$

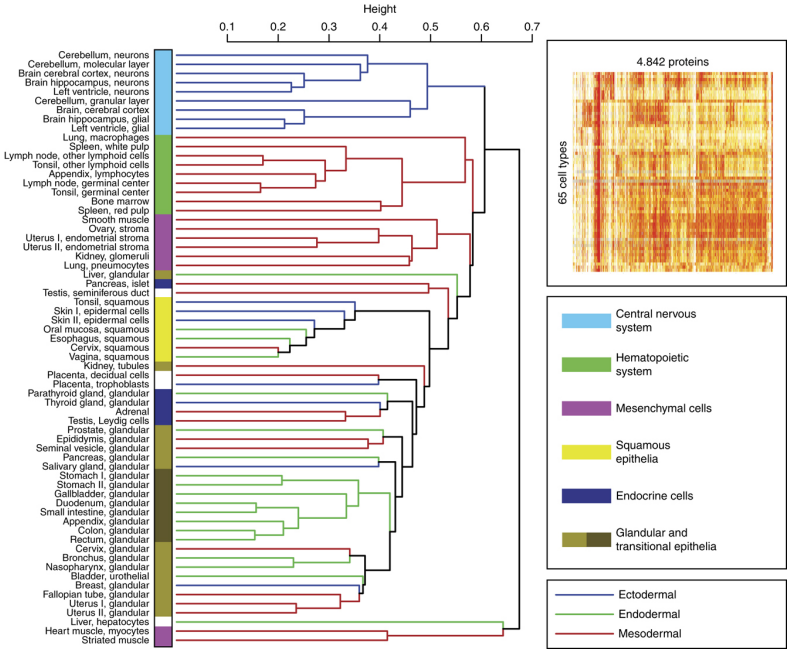
$\Rightarrow$ Choose J_{ij} to encode **hierarchically organized cycles**

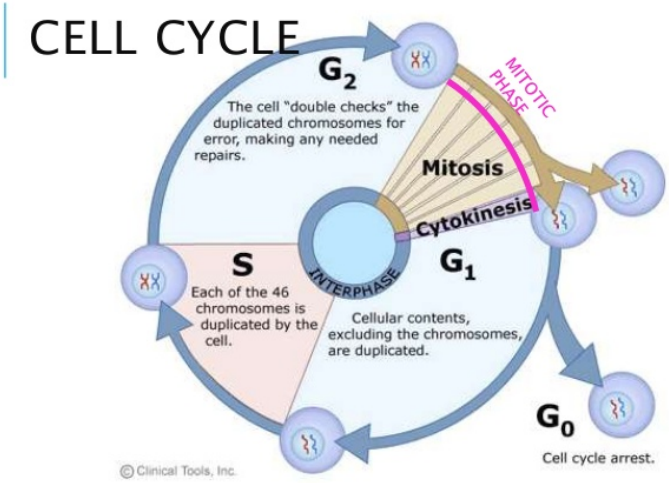


- ρ=cell-cycle stage $\in \{1 \dots C\}$

Nature of cellular attractors

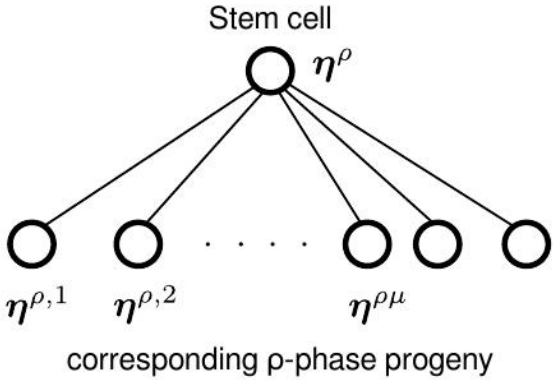
- cell types are hierarchically organized





- dynamical entities i.e. they cycle $(G_1 \to S \to G_2 \to M \to G_1 \dots)$

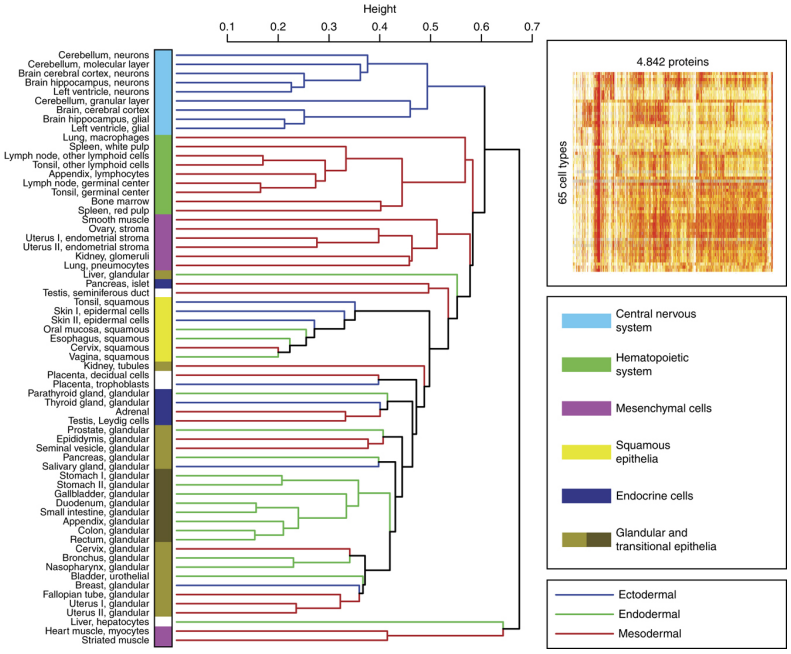
$\Rightarrow$ Choose J_{ij} to encode **hierarchically organized cycles**

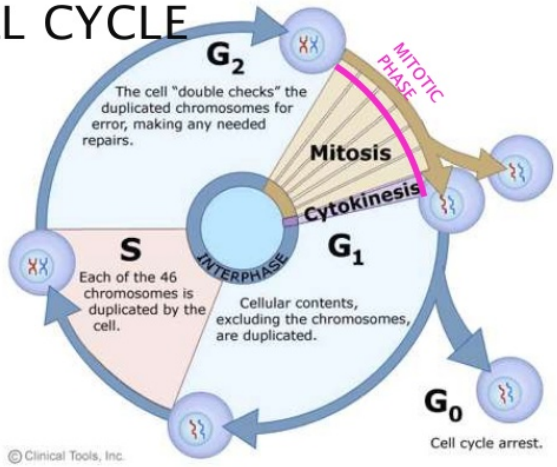


- ρ=cell-cycle stage $\in \{1 \dots C\}$
- μ = somatic cell type $\in \{1 \dots M\}$

Nature of cellular attractors

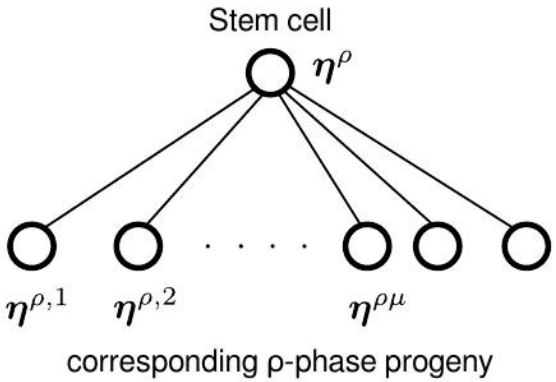
- cell types are hierarchically organized





- dynamical entities i.e. they cycle ($G_1 \to S \to G_2 \to M \to G_1 \dots$)

$\Rightarrow$ Choose J_{ij} to encode **hierarchically organized cycles**



- ρ=cell-cycle stage $\in \{1 \dots C\}$
- μ = somatic cell type $\in \{1 \dots M\}$
- $\eta_i^{\rho}, \eta_i^{\rho\mu} \in \{0, 1\}$: gene i in given cell type & phase

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

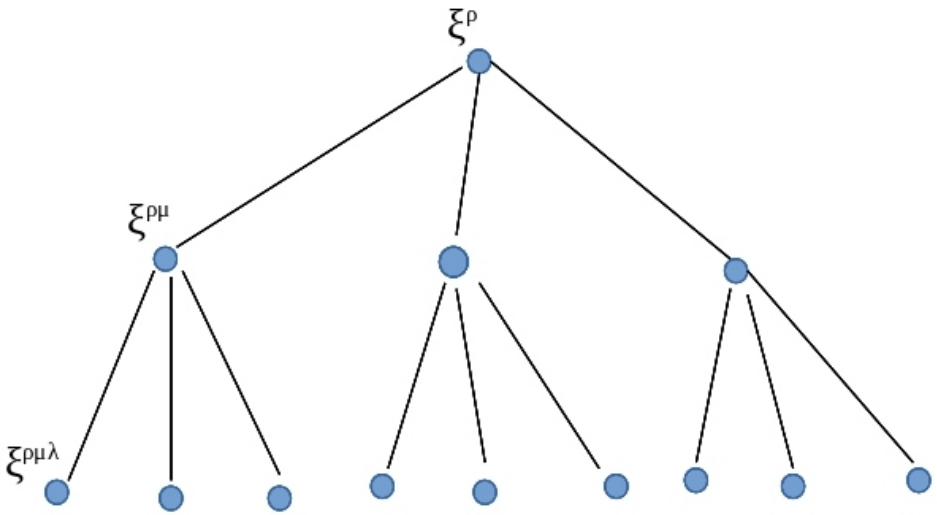
- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$

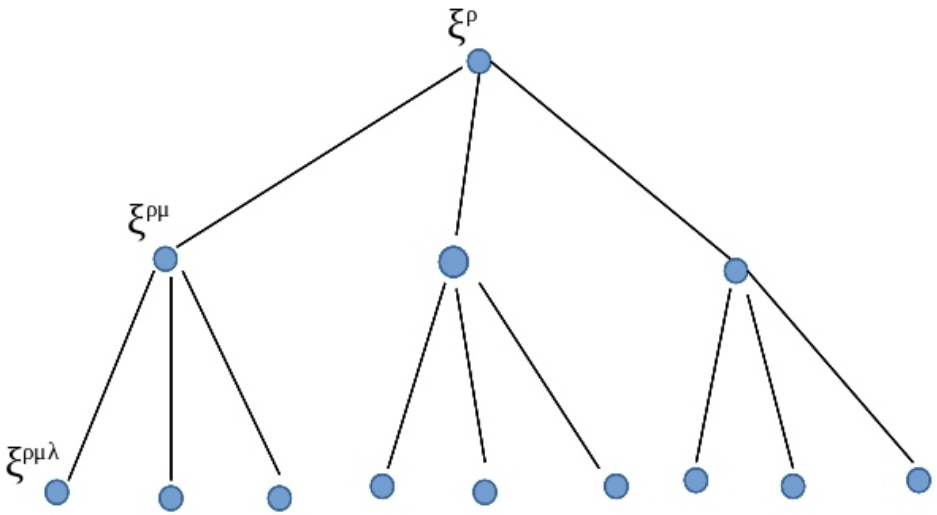


Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$



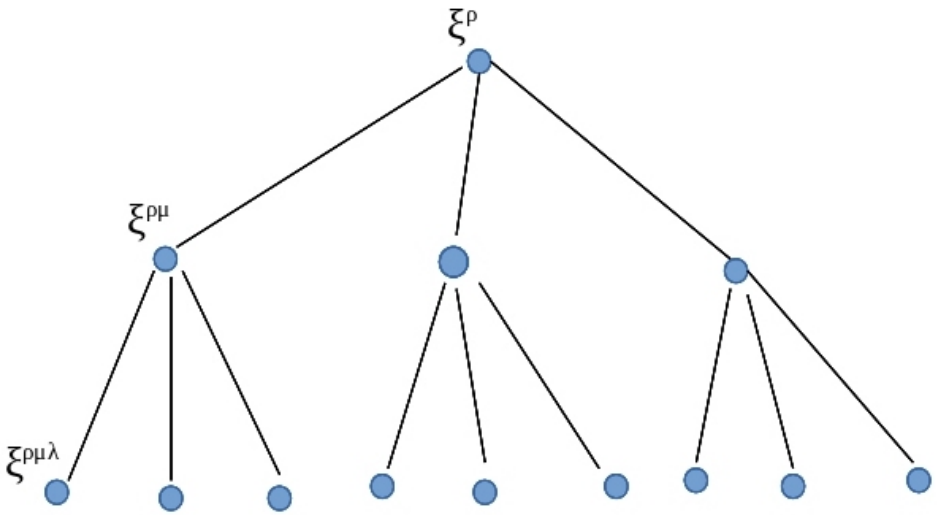
[Parga, Virasoro (1986); Krogh, Herz (1988)]

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$



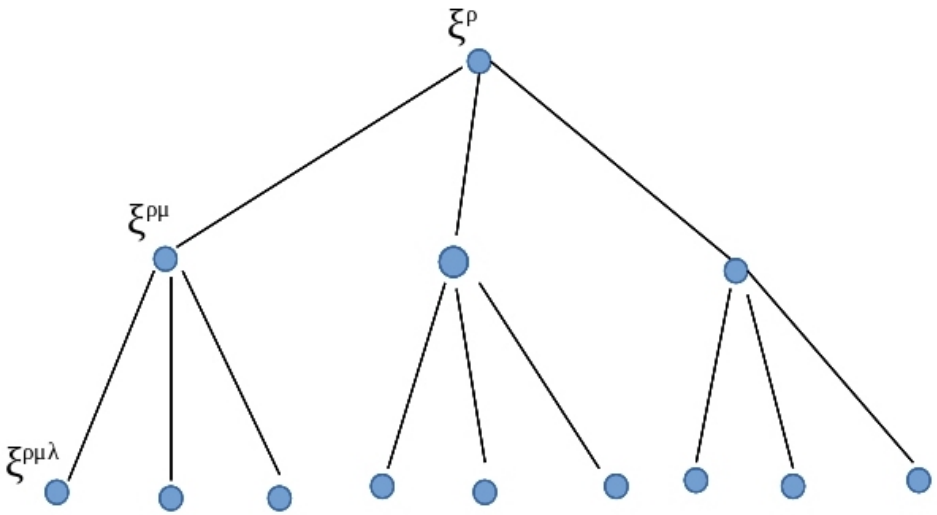
[Parga, Virasoro (1986); Krogh, Herz (1988)]

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$



[Parga, Virasoro (1986); Krogh, Herz (1988)]

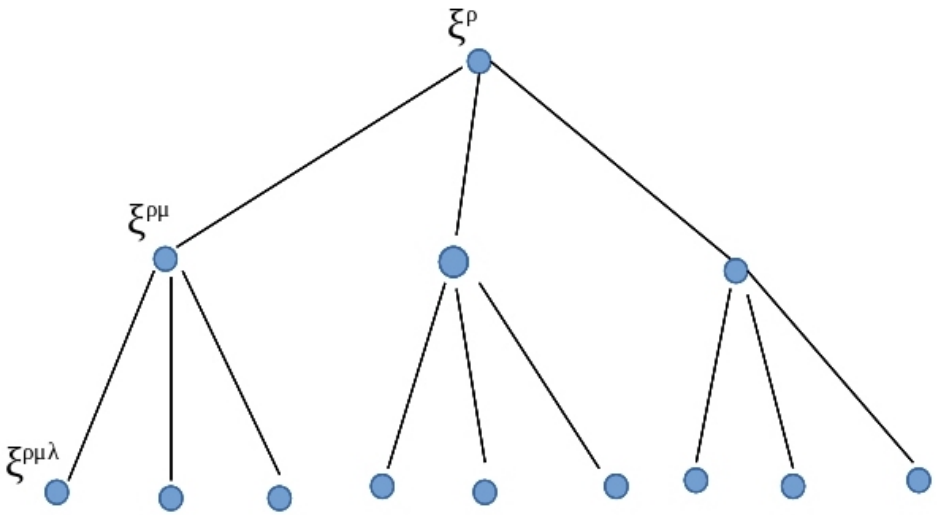
Markov process [defn as martingale]:
$W\left(\xi^{\mu_1 \ldots \mu_{k+1}} \middle| \xi^{\mu_1 \ldots \mu_k}\right)$

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$



[Parga, Virasoro (1986); Krogh, Herz (1988)]

Markov process [defn as martingale]:
$W(\xi^{\mu_1 \ldots \mu_{k+1}} | \xi^{\mu_1 \ldots \mu_k})$

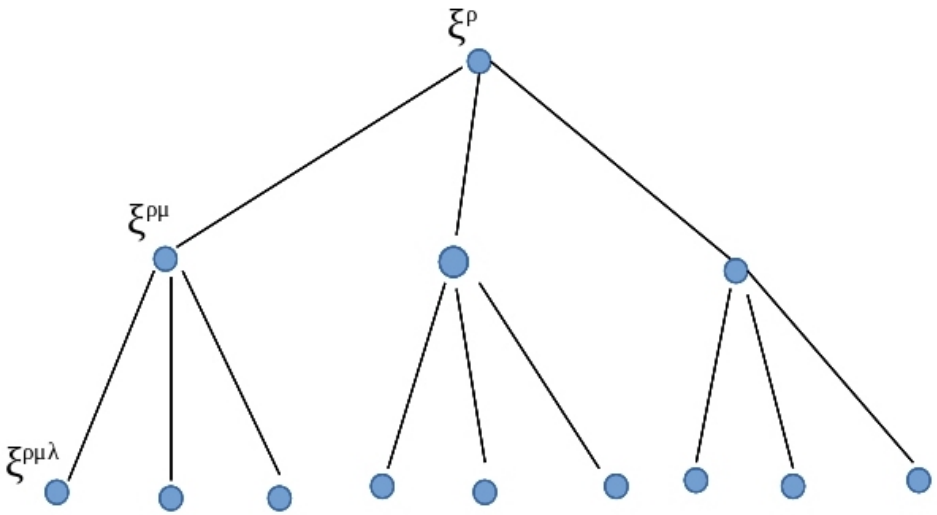
$$J_{ij} = \frac{1}{N} \left\{ \sum_{\rho=1}^{M} \frac{\xi_i^{\rho} \xi_j^{\rho}}{q_1} + \sum_{\rho\mu=1}^{M} \frac{(\xi_i^{\rho\mu} - \xi_i^{\rho})(\xi_j^{\rho\mu} - \xi_j^{\rho})}{q_2 - q_1} + \sum_{\rho\mu\lambda=1}^{M} \frac{(\xi_i^{\rho\mu\lambda} - \xi_i^{\rho\mu})(\xi_j^{\rho\mu\lambda} - \xi_j^{\rho\mu})}{1 - q_2} \right\}$$

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$



[Parga, Virasoro (1986); Krogh, Herz (1988)]

Markov process [defn as martingale]:
$W(\xi^{\mu_1 \ldots \mu_{k+1}} | \xi^{\mu_1 \ldots \mu_k})$

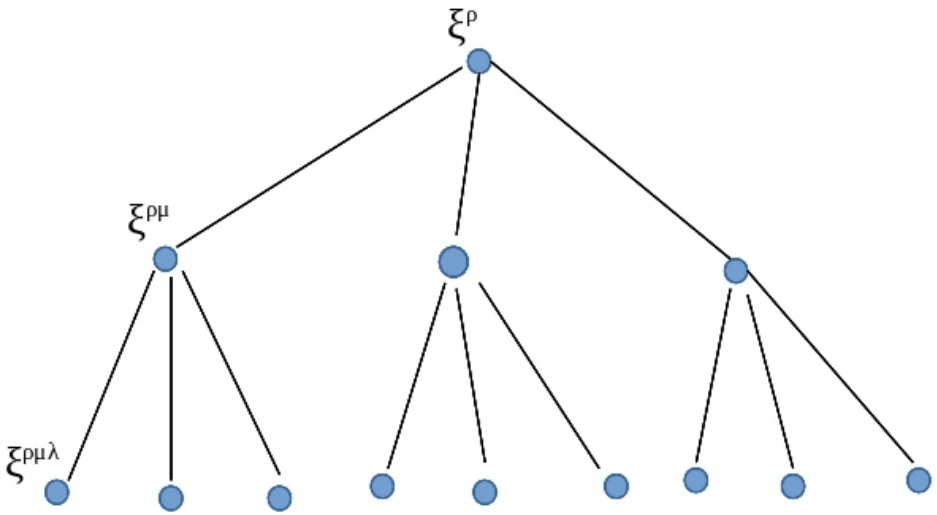
$$J_{ij} = \frac{1}{N} \left\{ \sum_{\rho=1}^{M} \frac{\xi_i^{\rho} \xi_j^{\rho}}{q_1} + \sum_{\rho\mu=1}^{M} \frac{(\xi_i^{\rho\mu} - \xi_i^{\rho})(\xi_j^{\rho\mu} - \xi_j^{\rho})}{q_2 - q_1} + \sum_{\rho\mu\lambda=1}^{M} \frac{(\xi_i^{\rho\mu\lambda} - \xi_i^{\rho\mu})(\xi_j^{\rho\mu\lambda} - \xi_j^{\rho\mu})}{1 - q_2} \right\}$$

Inspiration from neural networks

- Sequences of patterns: $\boldsymbol{\xi}^1 \to \boldsymbol{\xi}^2 \to \ldots \boldsymbol{\xi}^P$ [Sompolinsky, Kanter (1986)]

$$J_{ij} = \frac{1}{N} \sum_{\mu=1}^{P} \xi_i^{\mu+1} \xi_j^{\mu} \qquad \text{Cycles} : \boldsymbol{\xi}^{P+1} = \boldsymbol{\xi}^1$$

- Patterns hierarchically organized $\boldsymbol{\xi}^{\rho} \to \{\boldsymbol{\xi}^{\rho\mu}\} \to \{\{\boldsymbol{\xi}^{\rho\mu\lambda}\}\}$



[Parga, Virasoro (1986); Krogh, Herz (1988)]

Markov process [defn as martingale]:
$W(\xi^{\mu_1 \ldots \mu_{k+1}} | \xi^{\mu_1 \ldots \mu_k})$

$$J_{ij} = \frac{1}{N} \left\{ \sum_{\rho=1}^{M} \frac{\xi_i^{\rho} \xi_j^{\rho}}{q_1} + \sum_{\rho\mu=1}^{M} \frac{(\xi_i^{\rho\mu} - \xi_i^{\rho})(\xi_j^{\rho\mu} - \xi_j^{\rho})}{q_2 - q_1} + \sum_{\rho\mu\lambda=1}^{M} \frac{(\xi_i^{\rho\mu\lambda} - \xi_i^{\rho\mu})(\xi_j^{\rho\mu\lambda} - \xi_j^{\rho\mu})}{1 - q_2} \right\}$$

$\Rightarrow$ Combine and adapt to 0,1 variables.. (for a more general W)

Outline

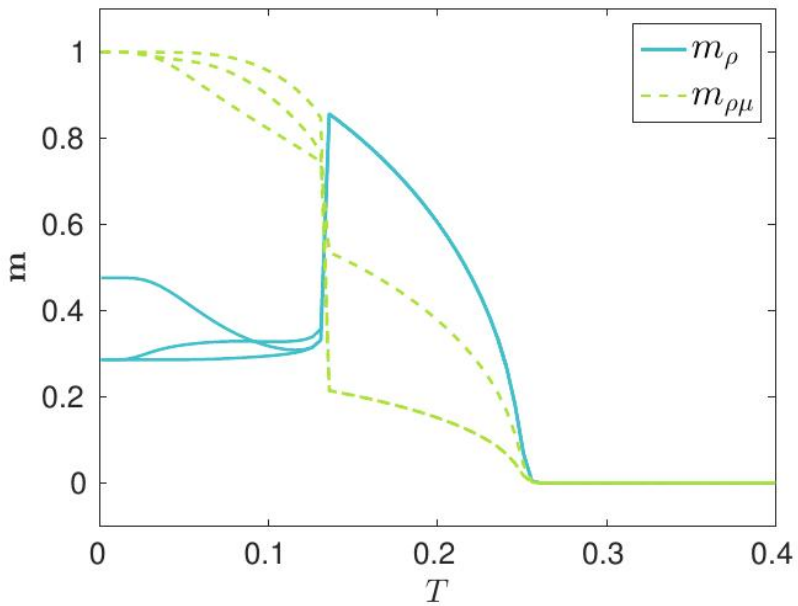
Results

get eqns. for correlations of $\mathbf{n}$ with $\boldsymbol{\eta}^{\rho}$ (m_{ρ}, full) and with $\boldsymbol{\eta}^{\rho\mu}$ ($m_{\rho\mu}$, dashed). Here $\rho = 1, 2, 3$.

Results

get eqns. for correlations of $\mathbf{n}$ with $\boldsymbol{\eta}^{\rho}$ (m_{ρ}, full) and with $\boldsymbol{\eta}^{\rho\mu}$ ($m_{\rho\mu}$, dashed). Here $\rho = 1, 2, 3$.

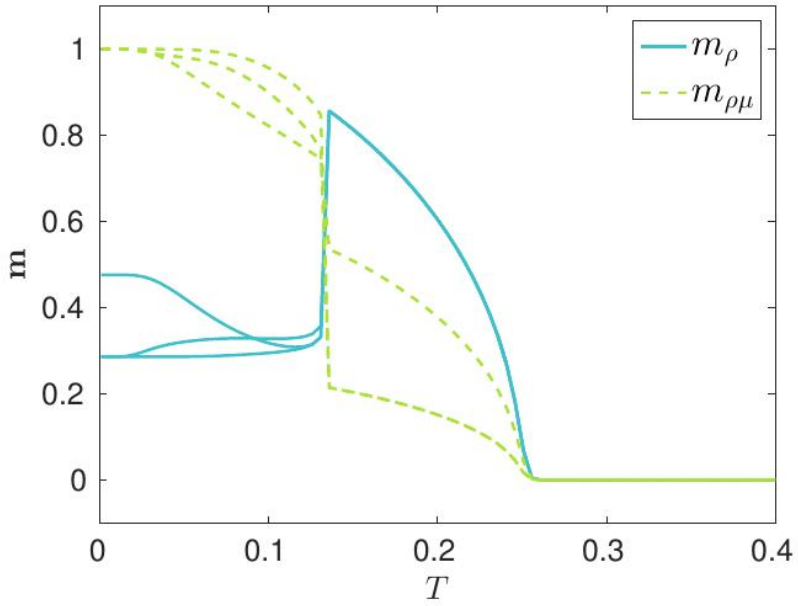
Dependence on noise level



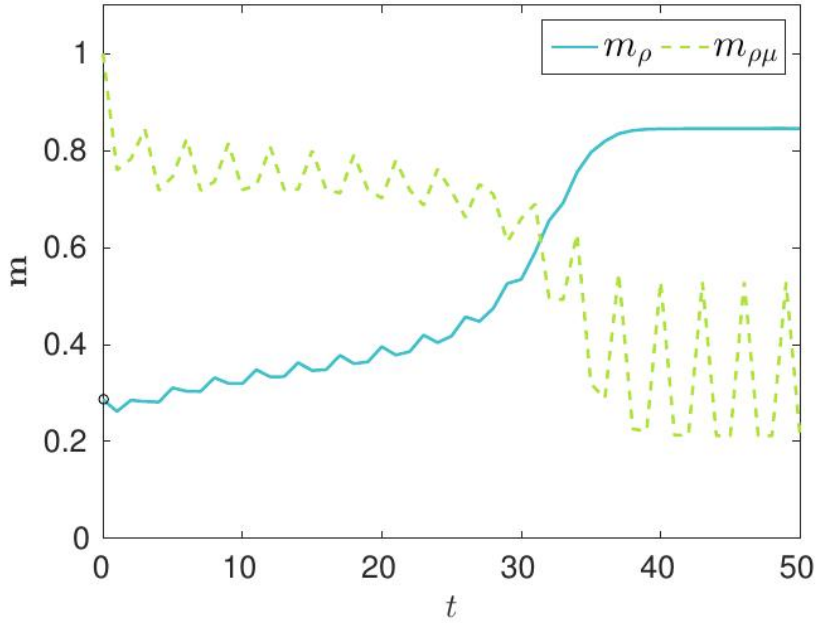
Results

get eqns. for correlations of $\mathbf{n}$ with $\boldsymbol{\eta}^{\rho}$ (m_{ρ}, full) and with $\boldsymbol{\eta}^{\rho\mu}$ ($m_{\rho\mu}$, dashed). Here $\rho = 1, 2, 3$.

Dependence on noise level



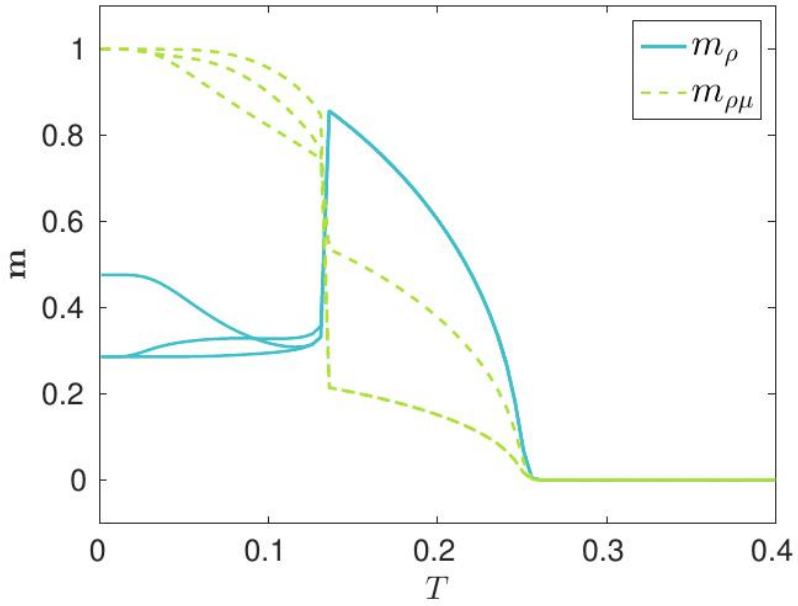
De-differentiation, T=0.14



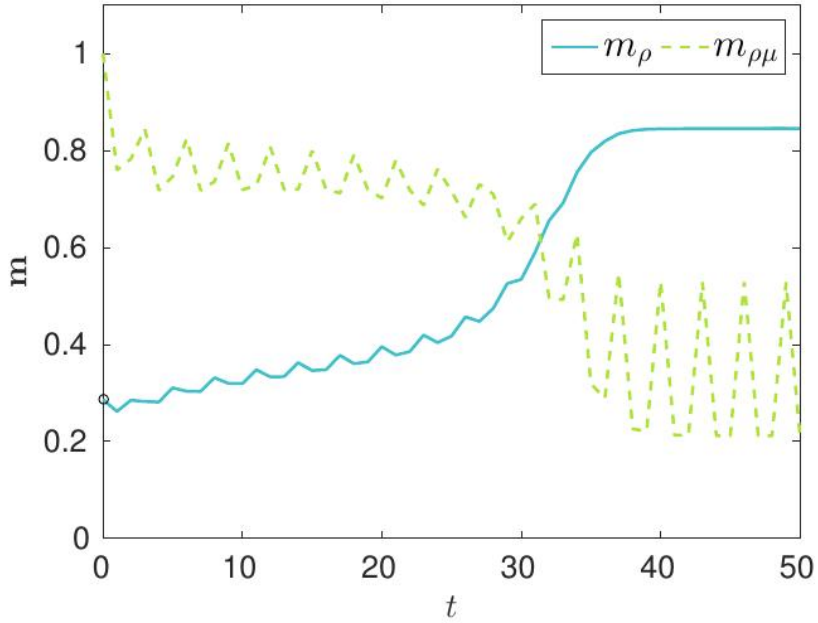
Results

get eqns. for correlations of $\mathbf{n}$ with $\boldsymbol{\eta}^{\rho}$ (m_{ρ}, full) and with $\boldsymbol{\eta}^{\rho\mu}$ ($m_{\rho\mu}$, dashed). Here $\rho = 1, 2, 3$.

Dependence on noise level



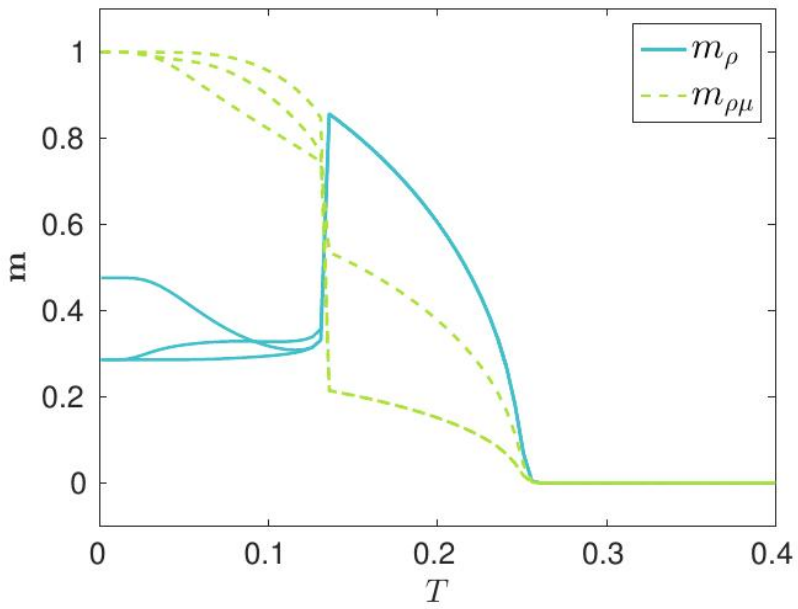
De-differentiation, T=0.14



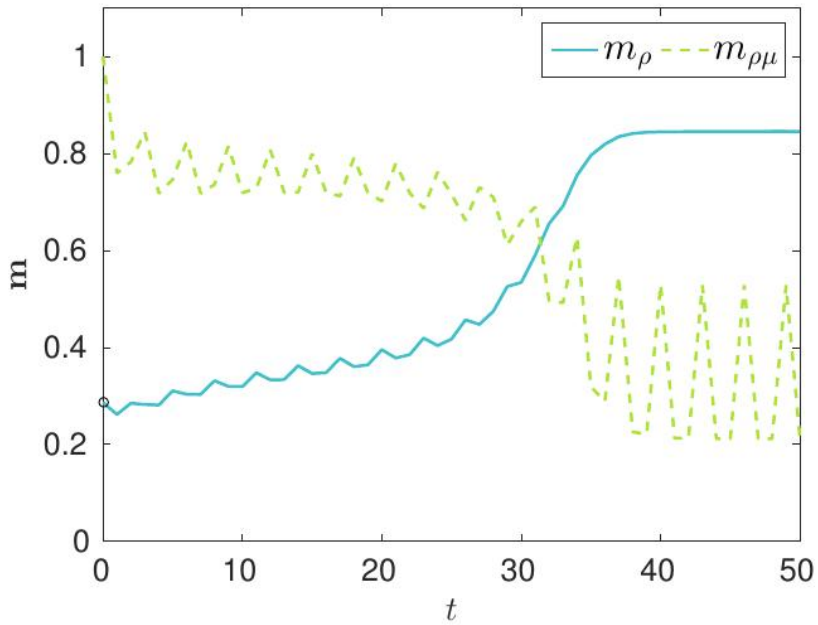
Results

get eqns. for correlations of $\mathbf{n}$ with $\boldsymbol{\eta}^{\rho}$ (m_{ρ}, full) and with $\boldsymbol{\eta}^{\rho\mu}$ ($m_{\rho\mu}$, dashed). Here $\rho = 1, 2, 3$.

Dependence on noise level



De-differentiation, T=0.14



Note: de-differentiation takes $\mathcal{O}(10)$ cycles.

[R Hannam, **AA**, R Kühn, J Phys A (2017)]

Results

- Apply direct perturbation to genes to drive transition from somatic $\rightarrow$ stem cell

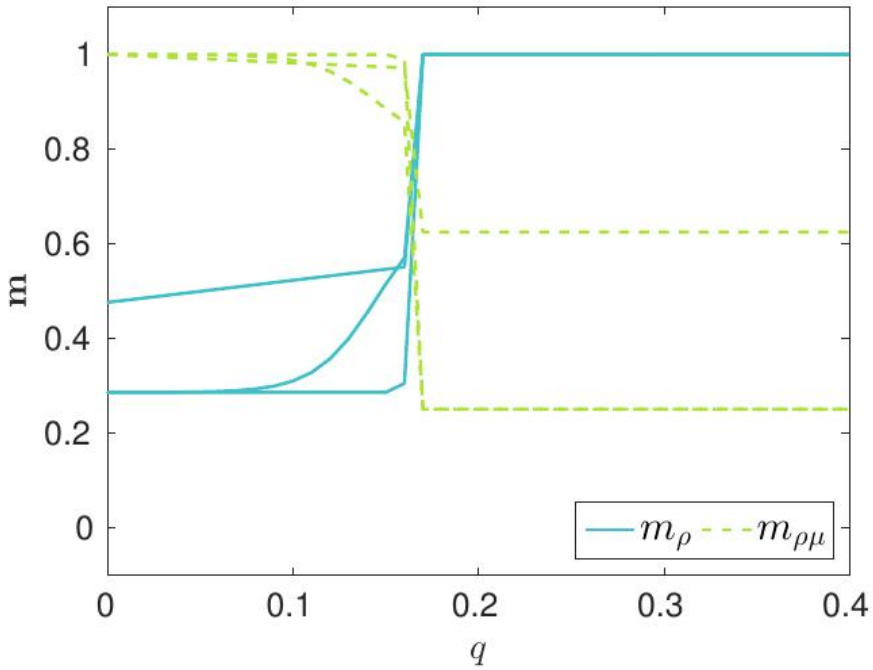
Results

- Apply direct perturbation to genes to drive transition from somatic $\rightarrow$ stem cell

Results

- Apply direct perturbation to genes to drive transition from somatic $\rightarrow$ stem cell

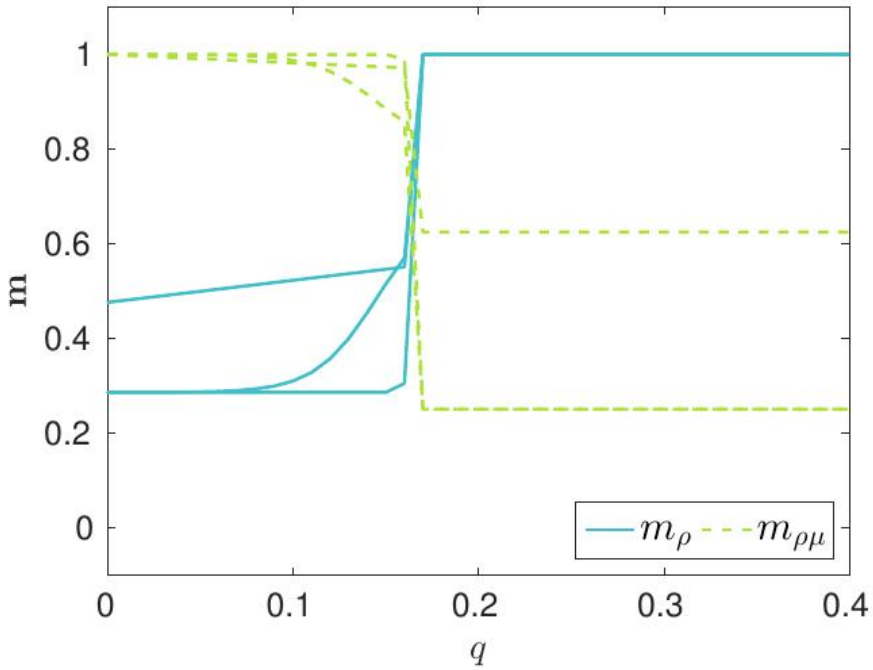
  Correlations vs fraction q of perturbed genes, $T = 0.01$.



Results

- Apply direct perturbation to genes to drive transition from somatic $\rightarrow$ stem cell

  Correlations vs fraction q of perturbed genes, $T = 0.01$.



Critical fraction of genes $q_r \in [0.1, 0.2]$ ($\searrow$ when $T \nearrow$)

[R Hannam, **AA**, R Kühn, J Phys A (2017)]

Results

- If we include in the system only regulatory genes:

  $N_r \simeq 0.1N \simeq 2,500 \qquad \Rightarrow \qquad q_r N_r \simeq$ 250 - 500 genes

Results

- If we include in the system only regulatory genes:

  $N_r \simeq 0.1N \simeq 2,500 \qquad \Rightarrow \qquad q_r N_r \simeq$ 250 - 500 genes

Results

- If we include in the system only regulatory genes:

  $N_r \simeq 0.1N \simeq 2,500 \qquad \Rightarrow \qquad q_r N_r \simeq$ 250 - 500 genes

  Also:

- each Yamanaka TF involved in regulating $\mathcal{O}(100)$ genes

Results

- If we include in the system only regulatory genes:

  $N_r \simeq 0.1N \simeq 2,500 \qquad \Rightarrow \qquad q_r N_r \simeq$ 250 - 500 genes

  Also:

- each Yamanaka TF involved in regulating $\mathcal{O}(100)$ genes

Results

- If we include in the system only regulatory genes:

  $N_r \simeq 0.1N \simeq 2,500 \qquad \Rightarrow \qquad q_r N_r \simeq$ 250 - 500 genes

  Also:

- each Yamanaka TF involved in regulating $\mathcal{O}(100)$ genes

  $\Rightarrow$ can perturb $q_r N_r$ genes with $\mathcal{O}(\mathbf{3-5})$ TFs! (**Yamanaka territory!**)

Results

- If we include in the system only regulatory genes:

  $N_r \simeq 0.1N \simeq 2,500 \qquad \Rightarrow \qquad q_r N_r \simeq$ 250 - 500 genes

  Also:

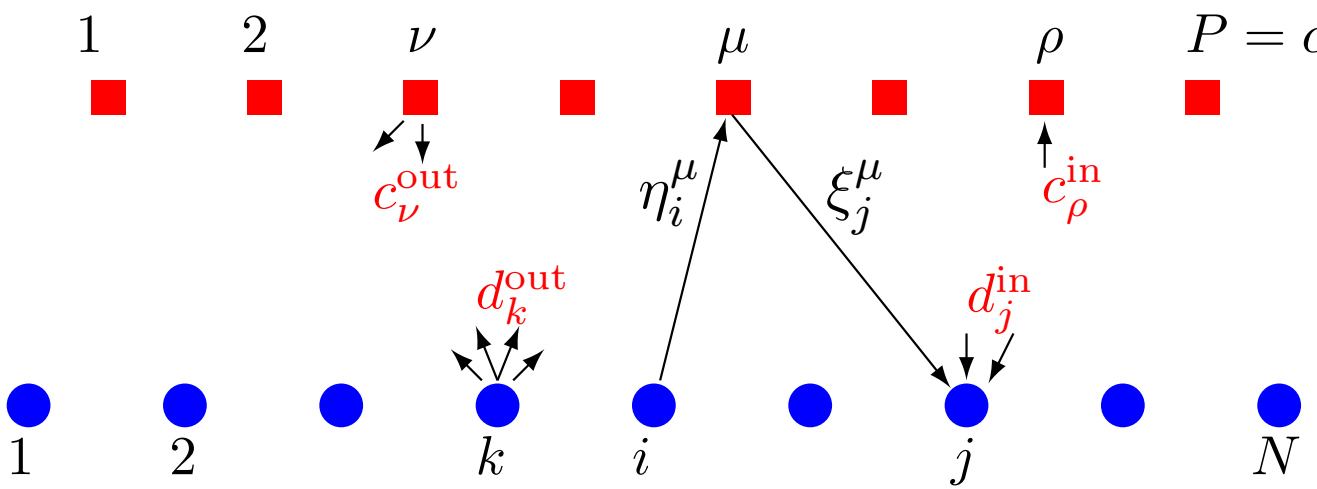
- each Yamanaka TF involved in regulating $\mathcal{O}(100)$ genes

  $\Rightarrow$ can perturb $q_r N_r$ genes with $\mathcal{O}(\mathbf{3-5})$ TFs! (**Yamanaka territory!**)

Q: Biological grounds for interactions?

Outline

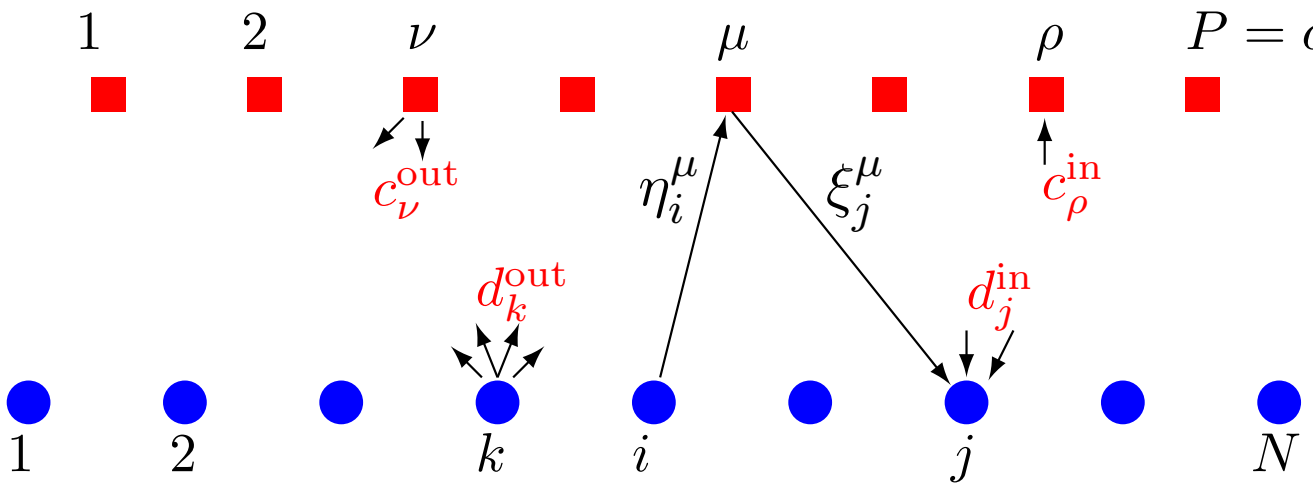
A bipartite graph approach



$$\eta_i^\mu \in \{0, 1\}$$

$$\xi_i^\mu \in \{0, \pm 1\}$$

A bipartite graph approach

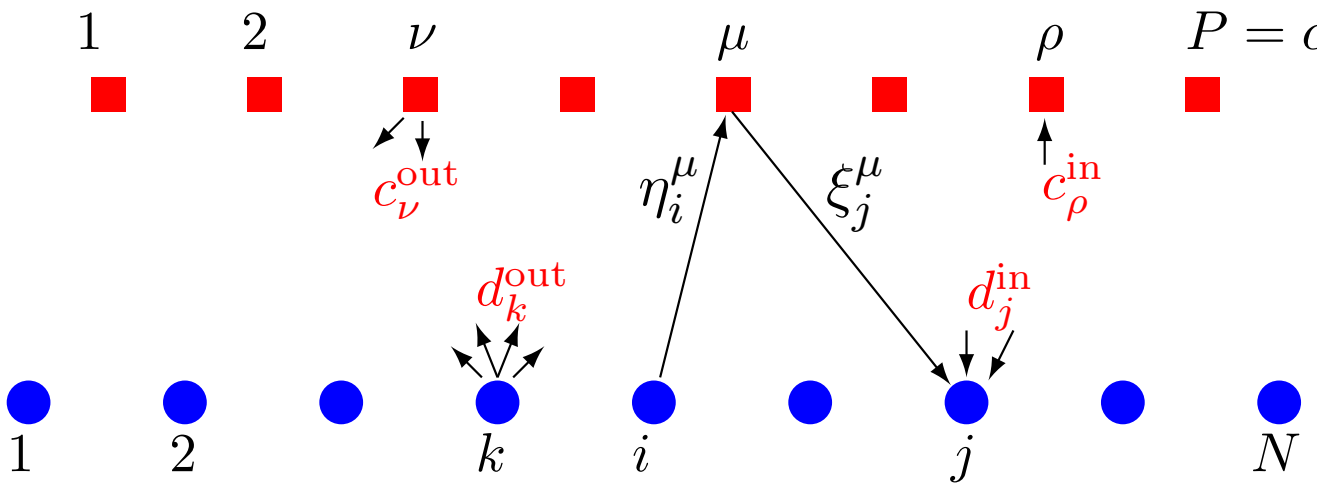


$\eta_i^\mu \in \{0, 1\}$

$\xi_i^\mu \in \{0, \pm 1\}$

$$n_i(t+1) = \Theta[\sum_\mu \xi_i^\mu \tau_\mu(t) - \vartheta_i - z_i(t)], \qquad \tau_\mu(t) = \text{concentration of TF } \mu \text{ at } t$$

A bipartite graph approach



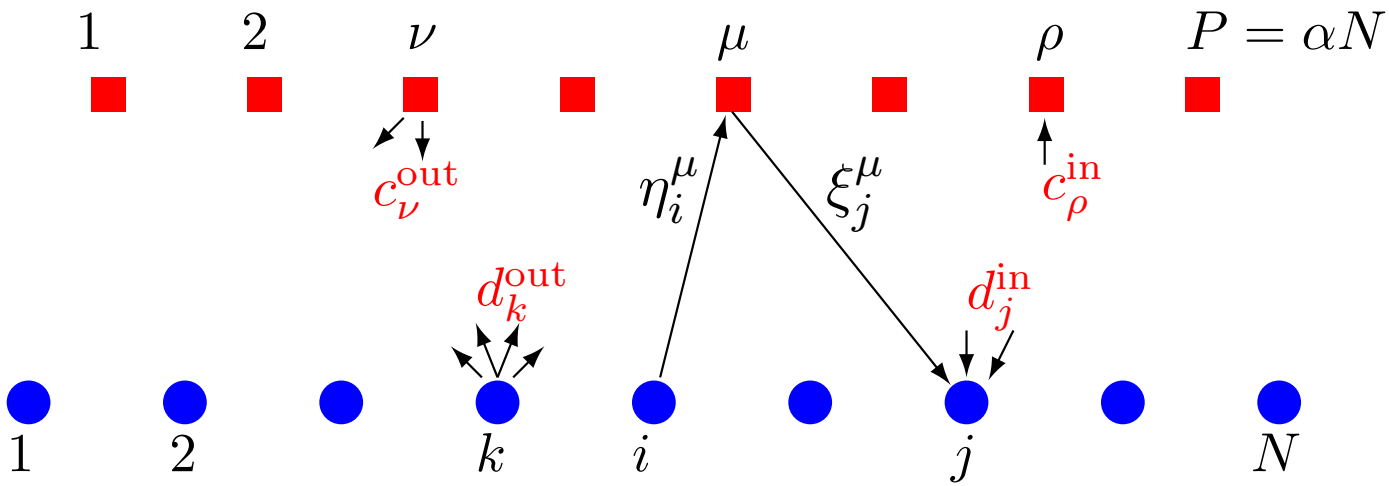
$\eta_i^\mu \in \{0, 1\}$

$\xi_i^\mu \in \{0, \pm 1\}$

$$n_i(t+1) = \Theta[\sum_\mu \xi_i^\mu \tau_\mu(t) - \vartheta_i - z_i(t)], \qquad \tau_\mu(t) = \text{concentration of TF } \mu \text{ at } t$$

Different types of logic for TFs:

A bipartite graph approach



$\eta_i^\mu \in \{0, 1\}$

$\xi_i^\mu \in \{0, \pm 1\}$

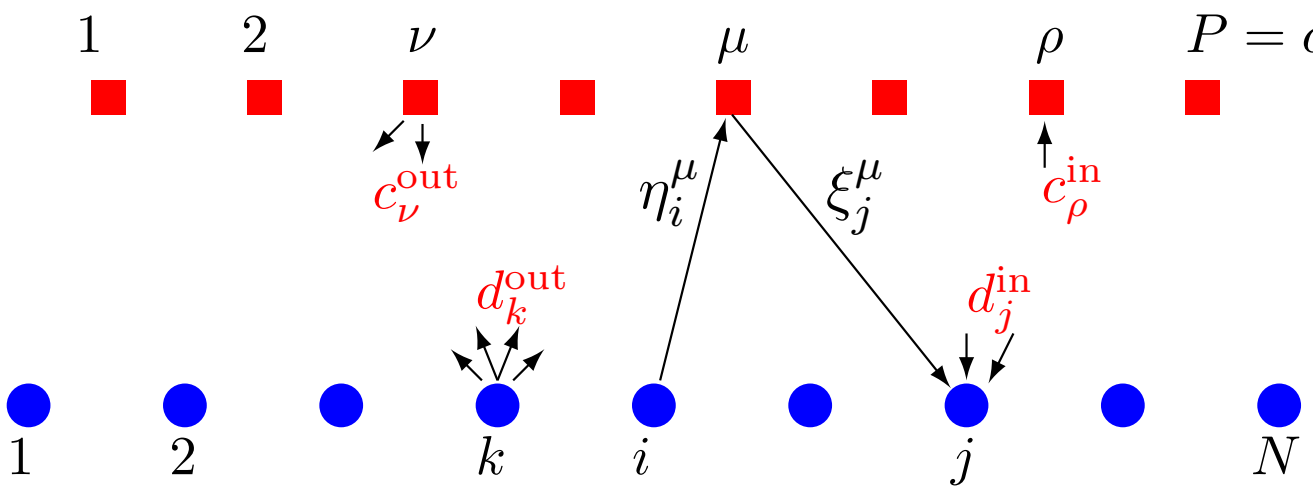
$$n_i(t+1) = \Theta[\sum_\mu \xi_i^\mu \tau_\mu(t) - \vartheta_i - z_i(t)], \qquad \tau_\mu(t) = \text{concentration of TF } \mu \text{ at } t$$

Different types of logic for TFs:

- **AND**: TF μ 'ON' if all contributing genes 'ON'

$$\tau_\mu(t) = \prod_{j:\eta_j^\mu = 1} n_j(t)$$

A bipartite graph approach



$\eta_i^\mu \in \{0, 1\}$

$\xi_i^\mu \in \{0, \pm 1\}$

$$n_i(t+1) = \Theta[\sum_\mu \xi_i^\mu \tau_\mu(t) - \vartheta_i - z_i(t)], \qquad \tau_\mu(t) = \text{concentration of TF } \mu \text{ at } t$$

Different types of logic for TFs:

- **AND**: TF μ 'ON' if all contributing genes 'ON'

$$\tau_\mu(t) = \prod_{j:\eta_j^\mu = 1} n_j(t)$$

- **OR**: TF μ 'ON' if at least one contributing gene 'ON'

$$\tau_\mu(t) = \frac{1}{c_\mu^{\text{in}}} \sum_j \eta_j^\mu n_j(t)$$

Linear vs non-linear threshold dynamics

OR: linear threshold model

Linear vs non-linear threshold dynamics

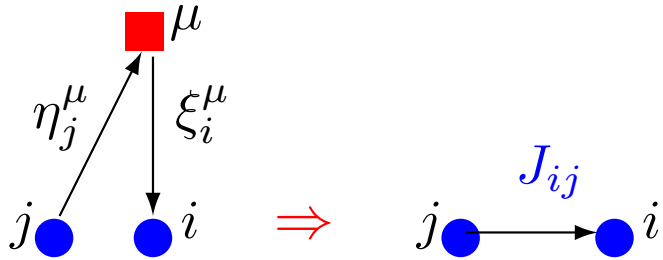
OR: linear threshold model

$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$

Linear vs non-linear threshold dynamics

OR: linear threshold model

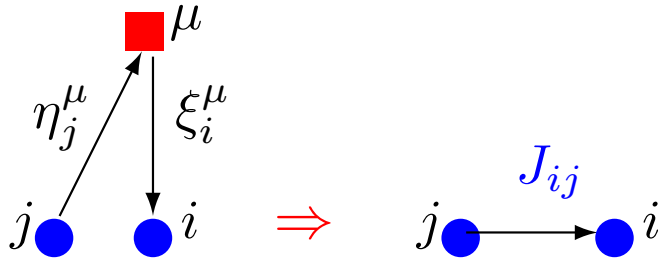
$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$



Linear vs non-linear threshold dynamics

OR: linear threshold model

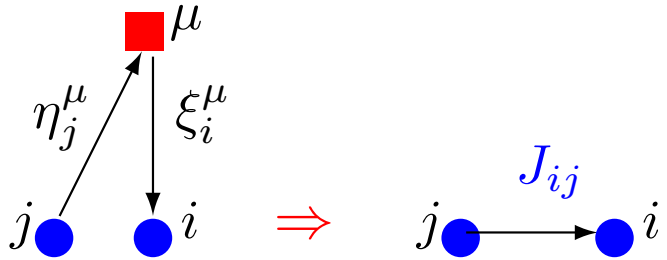
$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$



Linear vs non-linear threshold dynamics

OR: linear threshold model

$$n_i(t+1) = \Theta\Big[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)\Big]$$



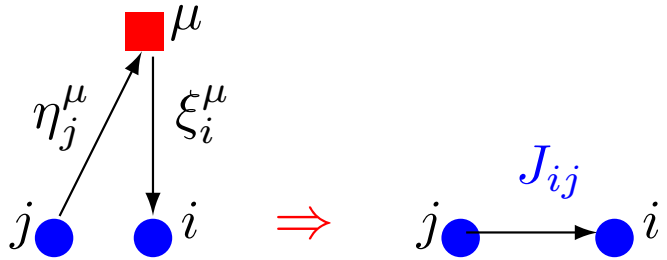
- Reminiscent of Neural networks with sparse patterns $\xi_i^\mu = 0, \pm 1$

$$J_{ij} = \sum_\mu \frac{\xi_i^\mu \xi_j^\mu}{c_\mu}$$

Linear vs non-linear threshold dynamics

OR: linear threshold model

$$n_i(t+1) = \Theta\Big[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)\Big]$$



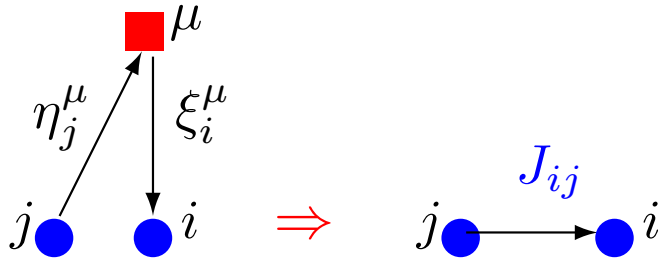
- Reminiscent of Neural networks with sparse patterns $\xi_i^\mu = 0, \pm 1$

$$J_{ij} = \sum_\mu \frac{\xi_i^\mu \xi_j^\mu}{c_\mu}$$

Linear vs non-linear threshold dynamics

OR: linear threshold model

$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$



- Reminiscent of Neural networks with sparse patterns $\xi_i^\mu = 0, \pm 1$

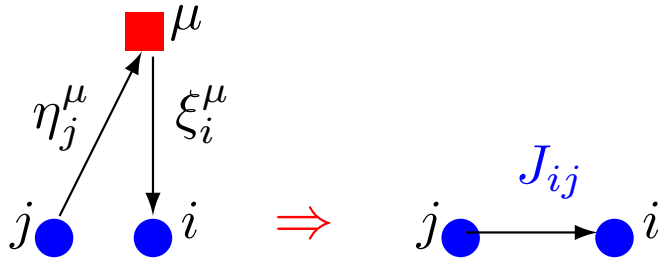
$$J_{ij} = \sum_\mu \frac{\xi_i^\mu \xi_j^\mu}{c_\mu} \quad \Rightarrow \quad \textbf{parallel retrieval of patterns}$$

[Agliari, **AA**, Barra, Coolen, Tantari, JPA (2013)]
[Sollich, Tantari, **AA**, Barra, PRL (2014)]

Linear vs non-linear threshold dynamics

OR: linear threshold model

$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$



- Reminiscent of Neural networks with sparse patterns $\xi_i^\mu = 0, \pm 1$

$$J_{ij} = \sum_\mu \frac{\xi_i^\mu \xi_j^\mu}{c_\mu} \quad \Rightarrow \quad \textbf{parallel retrieval of patterns}$$

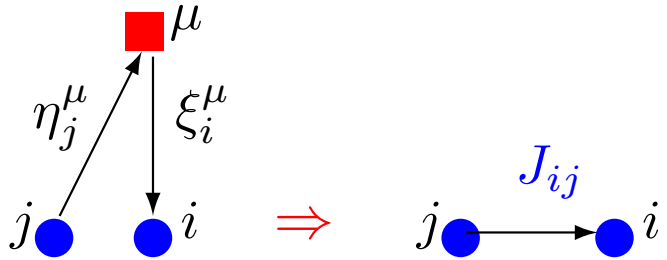
[Agliari, **AA**, Barra, Coolen, Tantari, JPA (2013)]
[Sollich, Tantari, **AA**, Barra, PRL (2014)]

BUT: here $\xi_i^\mu \in \{0, \pm 1\}$ and $\eta_i^\mu \in \{0, 1\} \Rightarrow J_{ij} \neq J_{ji}$

Linear vs non-linear threshold dynamics

OR: linear threshold model

$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$



- Reminiscent of Neural networks with sparse patterns $\xi_i^\mu = 0, \pm 1$

$$J_{ij} = \sum_\mu \frac{\xi_i^\mu \xi_j^\mu}{c_\mu} \quad \Rightarrow \quad \textbf{parallel retrieval of patterns}$$

[Agliari, **AA**, Barra, Coolen, Tantari, JPA (2013)]
[Sollich, Tantari, **AA**, Barra, PRL (2014)]

BUT: here $\xi_i^\mu \in \{0, \pm 1\}$ and $\eta_i^\mu \in \{0, 1\} \Rightarrow J_{ij} \neq J_{ji}$

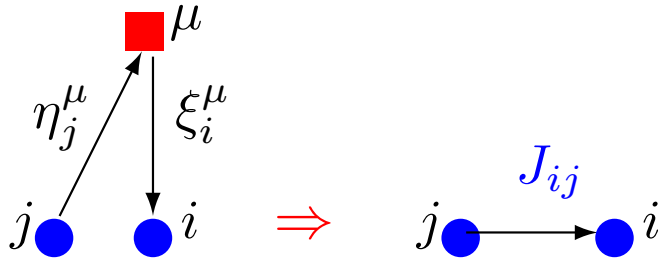
AND: non-linear threshold model

$$n_i(t+1) = \Theta[\sum_\mu \xi_i^\mu \prod_{j:\eta_i^\mu = 1} n_j(t) - \vartheta_i - z_i(t)]$$

Linear vs non-linear threshold dynamics

OR: linear threshold model

$$n_i(t+1) = \Theta[\sum_j \underbrace{\sum_\mu \frac{\xi_i^\mu \eta_j^\mu}{c_\mu^{\text{in}}}}_{J_{ij}} n_j(t) - \vartheta_i - z_i(t)]$$



- Reminiscent of Neural networks with sparse patterns $\xi_i^\mu = 0, \pm 1$

$$J_{ij} = \sum_\mu \frac{\xi_i^\mu \xi_j^\mu}{c_\mu} \quad \Rightarrow \quad \textbf{parallel retrieval} \text{ of patterns}$$

[Agliari, **AA**, Barra, Coolen, Tantari, JPA (2013)]
[Sollich, Tantari, **AA**, Barra, PRL (2014)]

BUT: here $\xi_i^\mu \in \{0, \pm 1\}$ and $\eta_i^\mu \in \{0, 1\} \Rightarrow J_{ij} \neq J_{ji}$

AND: non-linear threshold model

$$n_i(t+1) = \Theta[\sum_\mu \xi_i^\mu \prod_{j:\eta_i^\mu = 1} n_j(t) - \vartheta_i - z_i(t)]$$

- asymmetric multi-node interactions (as opposed to pairwise)

Outline

Percolation

Q: For which model's parameters can have non-trivial attractors?

Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes

Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes

Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes

- GC *only* stable solution if $\boxed{\alpha \langle c^{\text{out}} \rangle_{\text{TF}} \, P_{\text{TF}}(c^{\text{in}} = 1) > 1}$

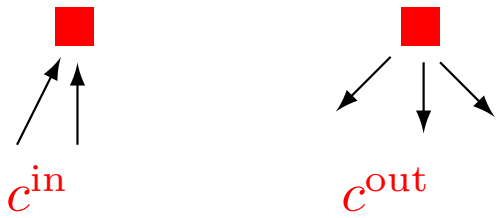
Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes



- GC *only* stable solution if $\boxed{\alpha \langle c^{\text{out}} \rangle_{\text{TF}} \, P_{\text{TF}}(c^{\text{in}} = 1) > 1}$

  $\Rightarrow$ TFs should regulate sufficiently many genes

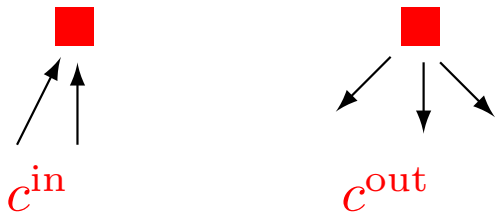
Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes



- GC *only* stable solution if $\boxed{\alpha \langle c^{\text{out}} \rangle_{\text{TF}} \, P_{\text{TF}}(c^{\text{in}} = 1) > 1}$

  $\Rightarrow$ TFs should regulate sufficiently many genes

OR:

- GC only stable option for $\boxed{\alpha \langle c^{\text{in}} \rangle_{\text{TF}} \langle c^{\text{out}} \rangle_{\text{TF}} > 1}$

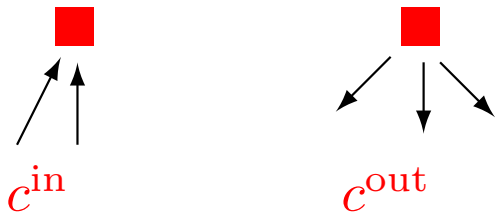
Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes



- GC *only* stable solution if $\boxed{\alpha \langle c^{\text{out}} \rangle_{\text{TF}} \, P_{\text{TF}}(c^{\text{in}} = 1) > 1}$

  $\Rightarrow$ TFs should regulate sufficiently many genes

OR:

- GC only stable option for $\boxed{\alpha \langle c^{\text{in}} \rangle_{\text{TF}} \langle c^{\text{out}} \rangle_{\text{TF}} > 1}$

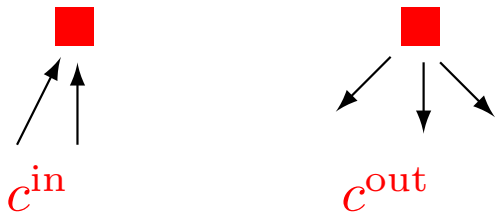
Percolation

Q: For which model's parameters can have non-trivial attractors?
A: Noisy finite systems are *ergodic* $\Rightarrow$ **Giant Component (GC)** for multiplicity of attractors $\Rightarrow$ Percolation theory

AND:

- GC stable if $\boxed{\langle c^{\text{in}} \rangle_{\text{TF}} \, P_G(d^{\text{in}} = 1) < 1}$

  $\Rightarrow$ TFs should be small complexes



- GC *only* stable solution if $\boxed{\alpha \langle c^{\text{out}} \rangle_{\text{TF}} \, P_{\text{TF}}(c^{\text{in}} = 1) > 1}$

  $\Rightarrow$ TFs should regulate sufficiently many genes

OR:

- GC only stable option for $\boxed{\alpha \langle c^{\text{in}} \rangle_{\text{TF}} \langle c^{\text{out}} \rangle_{\text{TF}} > 1}$

TFs **indeed** small complexes which regulate many genes!

[Hannam, Kühn, **AA**, JPA (2019); Torrisi, Kühn, **AA**, JSTAT (2020)]

Outline

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

Dynamics

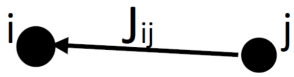
- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

Dynamics

- Linear threshold model

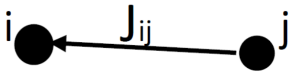
$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$



Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

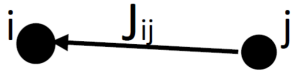


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$



$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

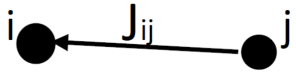
- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$



$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

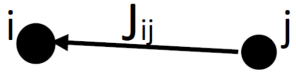
$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$



$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

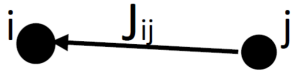
$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

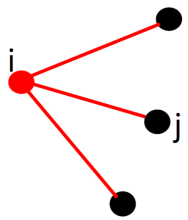


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

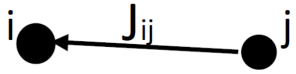
- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

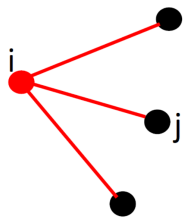


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i},\, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i},\, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]

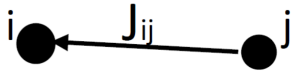


$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

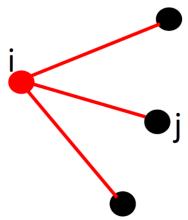


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i},\, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i},\, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]

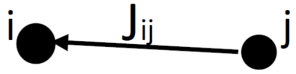


$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

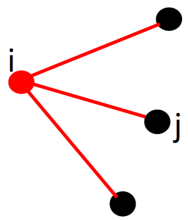


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



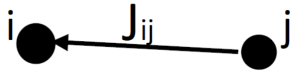
$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

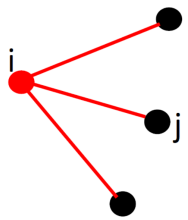


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



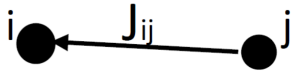
$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]

Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

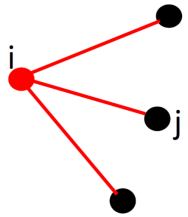


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

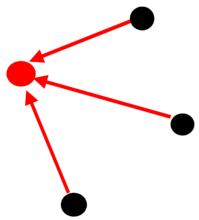
$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

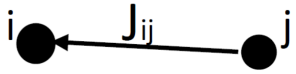
- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]



Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

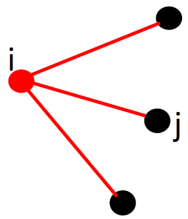


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

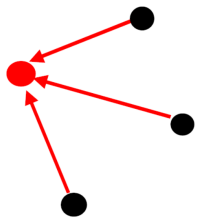
- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]

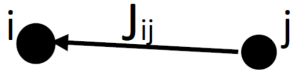
$$P(\boldsymbol{n}_{\partial_i}, t) = P^{(i)}(\boldsymbol{n}_{\partial_i}, t)$$



Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

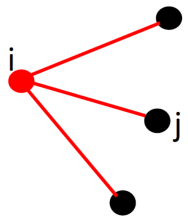


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

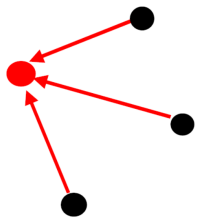
- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]

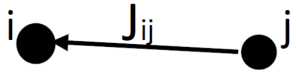
$$P(\boldsymbol{n}_{\partial_i}, t) = P^{(i)}(\boldsymbol{n}_{\partial_i}, t)$$



Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

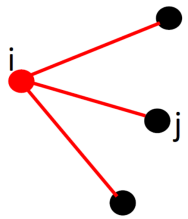


$z_i(t)$ random with Prob$[z \leq x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i}, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

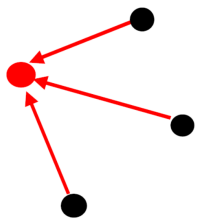
- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]

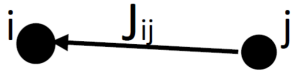
$$P(\boldsymbol{n}_{\partial_i}, t) = P^{(i)}(\boldsymbol{n}_{\partial_i}, t) = \prod_{j \in \partial_i} P_j(n_j, t)$$



Dynamics

- Linear threshold model

$$n_i(t+1) = \Theta\left[h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) - \vartheta_i - z_i(t)\right] \qquad h_i\left(\boldsymbol{n}_{\partial_i}(t)\right) = \sum_j J_{ij}\, n_j(t)$$

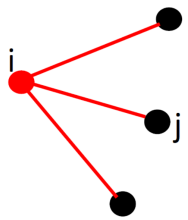


$z_i(t)$ random with Prob$[z \le x] = \Phi_T(x)$

- Interested in activation probability $P_i(t) = \text{Prob}(n_i(t) = 1)$

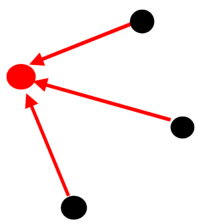
$$P_i(t+1) = \left\langle \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right)\right\rangle_{\boldsymbol{n}_{\partial_i},\, t} \qquad \left\langle \ldots \right\rangle_{\boldsymbol{n}_{\partial_i},\, t} = \sum_{\boldsymbol{n}_{\partial_i}} \ldots P(\boldsymbol{n}_{\partial_i}, t)$$

- Bethe lattice: Cavity method [Mezard & Parisi, (2001)]



$$P^{(i)}(\boldsymbol{n}_{\partial_i}) = \prod_{j \in \partial_i} P_j^{(i)}(n_j)$$

- Fully-asymmetric $J_{ij} J_{ji} = 0$ [Neri & Bollé, JSTAT (2009)]



$$P(\boldsymbol{n}_{\partial_i}, t) = P^{(i)}(\boldsymbol{n}_{\partial_i}, t) = \prod_{j \in \partial_i} P_j(n_j, t)$$

$$P_i(t+1) = \sum_{\boldsymbol{n}_{\partial_i}} \Phi_T\left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i\right) \prod_{j \in \partial_i} P_j(t)^{n_j} \left[1 - P_j(t)\right]^{1-n_j}$$

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

  via **Dynamic programming**

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

  via **Dynamic programming**

  [Torrisi, **AA**, Kühn, PRE (2021)] https://github.com/g-torr/

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

  via **Dynamic programming**

  [Torrisi, **AA**, Kühn, PRE (2021)] https://github.com/g-torr/

- Power-law: $p(k) \sim \gamma k^{-\gamma-1}$, $\gamma = 2.81$; $N = 2 \cdot 10^5$, $k_{\max} = 800$

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

  via **Dynamic programming**

  [Torrisi, **AA**, Kühn, PRE (2021)] https://github.com/g-torr/

- Power-law: $p(k) \sim \gamma k^{-\gamma-1}$, $\gamma = 2.81$; $N = 2 \cdot 10^5$, $k_{\max} = 800$

Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

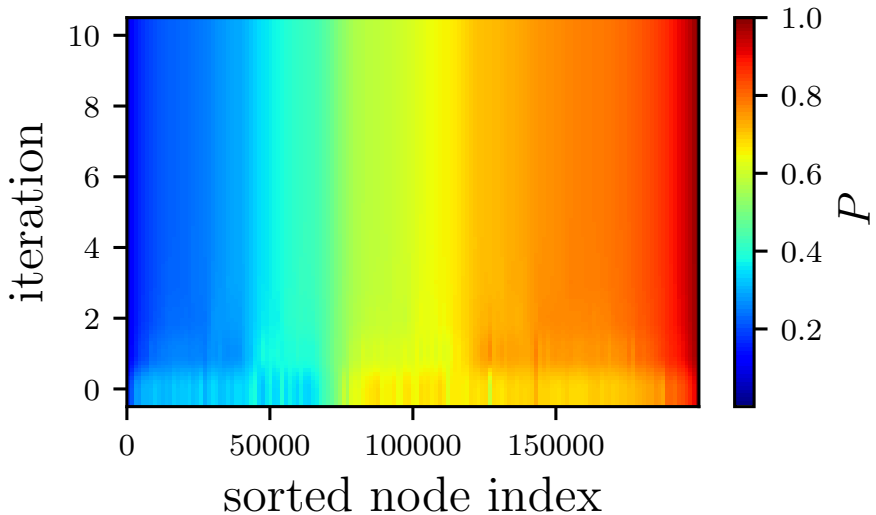
- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \quad \Rightarrow \quad \mathcal{O}(\sum_i |\partial_i|^2)$

  via **Dynamic programming**

  [Torrisi, **AA**, Kühn, PRE (2021)] https://github.com/g-torr/

- Power-law: $p(k) \sim \gamma k^{-\gamma-1}$, $\gamma = 2.81$; $N = 2 \cdot 10^5$, $k_{\max} = 800$

  Dynamics: $P_i(t)$ vs i, t



Dynamic programming

Solve iteratively: $t = 0, 1, \ldots$ BUT exponential complexity: $2^{|\partial_i|}$

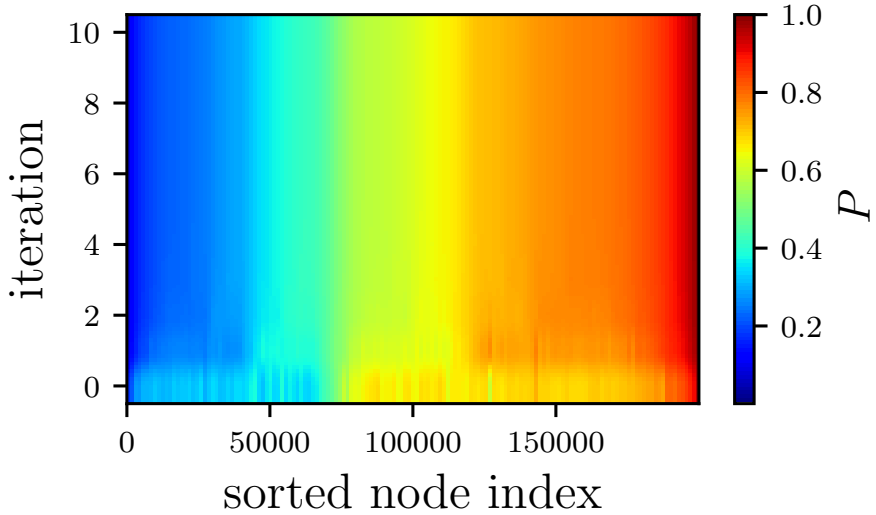
- For **binary** $J_{ij} = \pm J$: $\mathcal{O}(\sum_i 2^{|\partial_i|}) \Rightarrow \mathcal{O}(\sum_i |\partial_i|^2)$

  via **Dynamic programming**

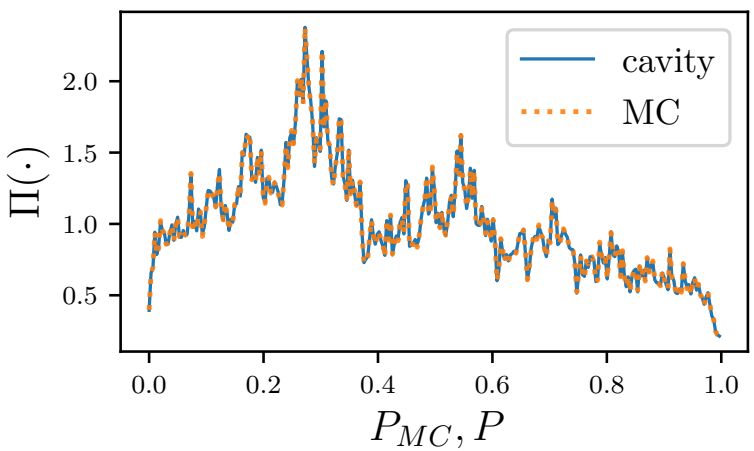
  [Torrisi, **AA**, Kühn, PRE (2021)] https://github.com/g-torr/

- Power-law: $p(k) \sim \gamma k^{-\gamma-1}$, $\gamma = 2.81$; $N = 2 \cdot 10^5$, $k_{\max} = 800$

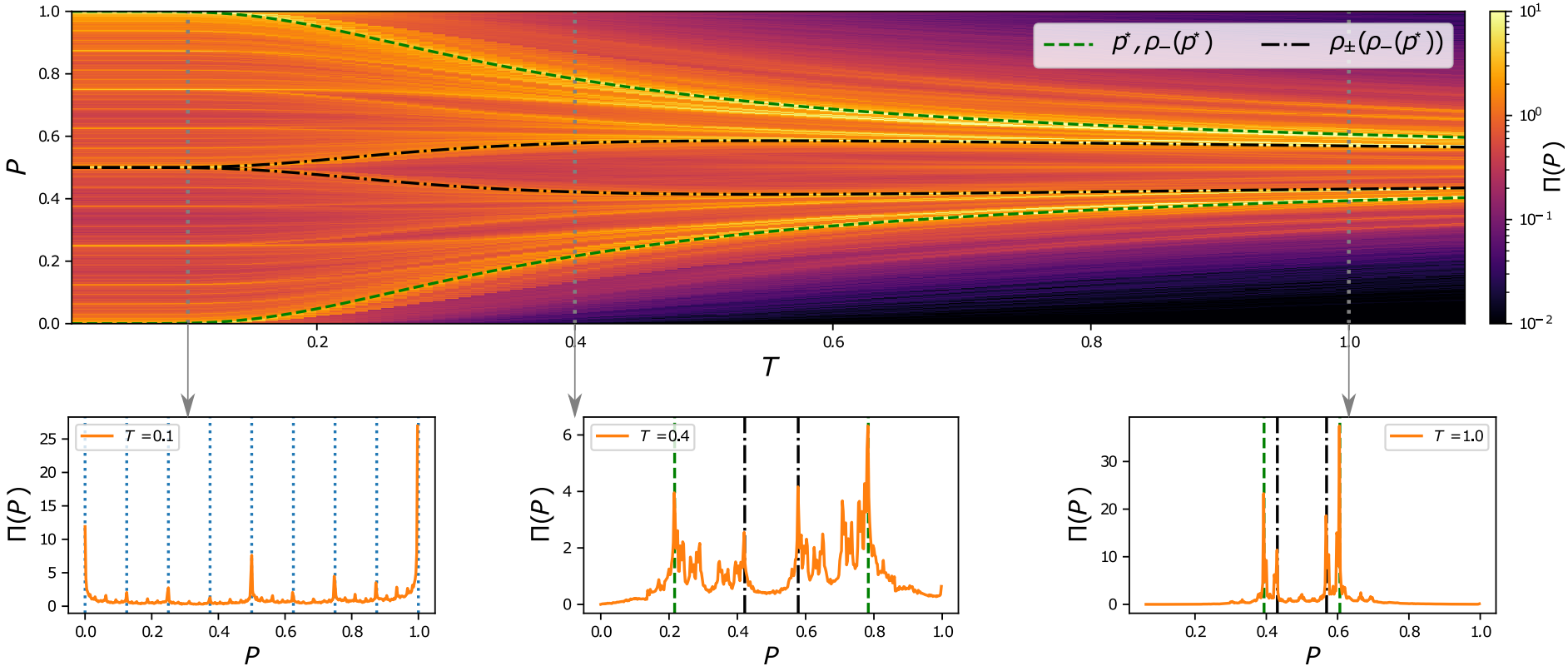
  Dynamics: $P_i(t)$ vs i, t



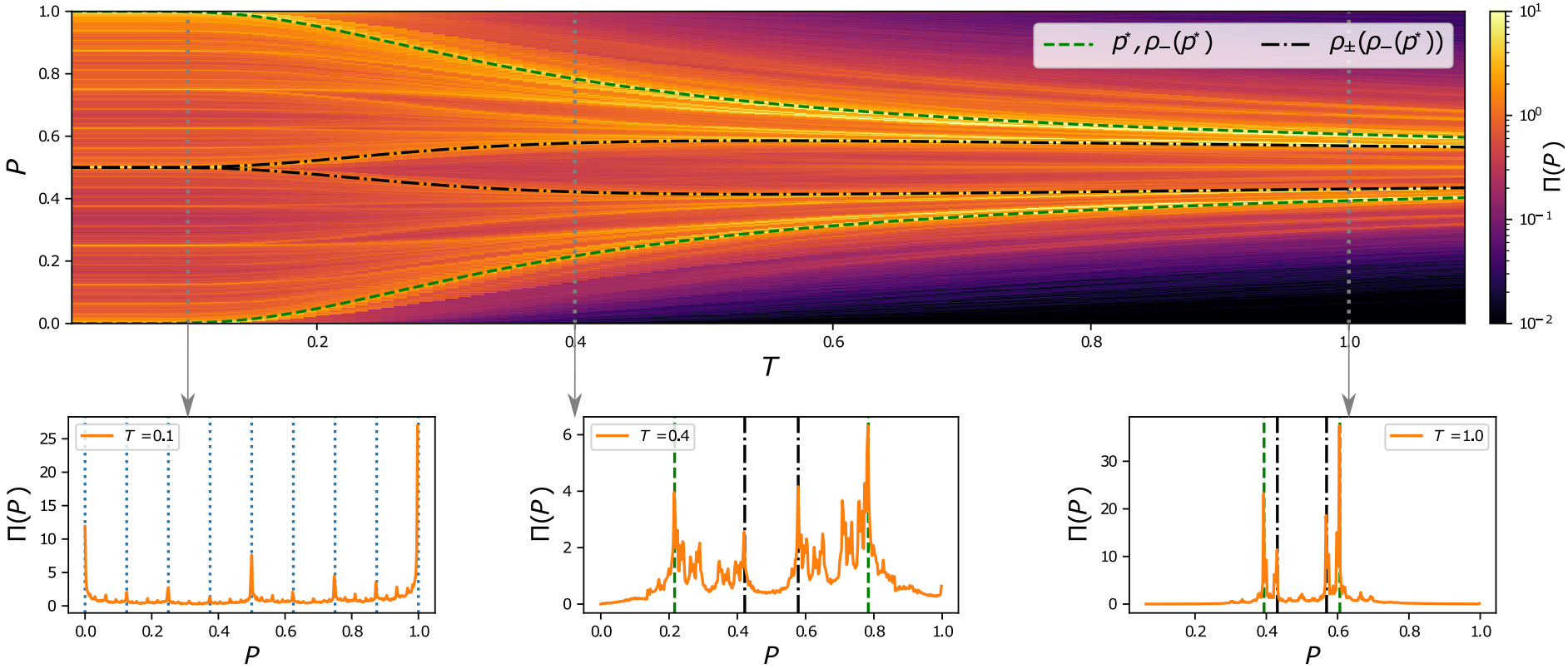
Stationarity: $\Pi(P) = N^{-1} \sum_i \delta(P - P_i)$



Results $P(J_{ij} \neq 0) = \eta\delta(J_{ij} - J) + (1-\eta)\delta(J_{ij} + J), \quad \eta = 0.62$

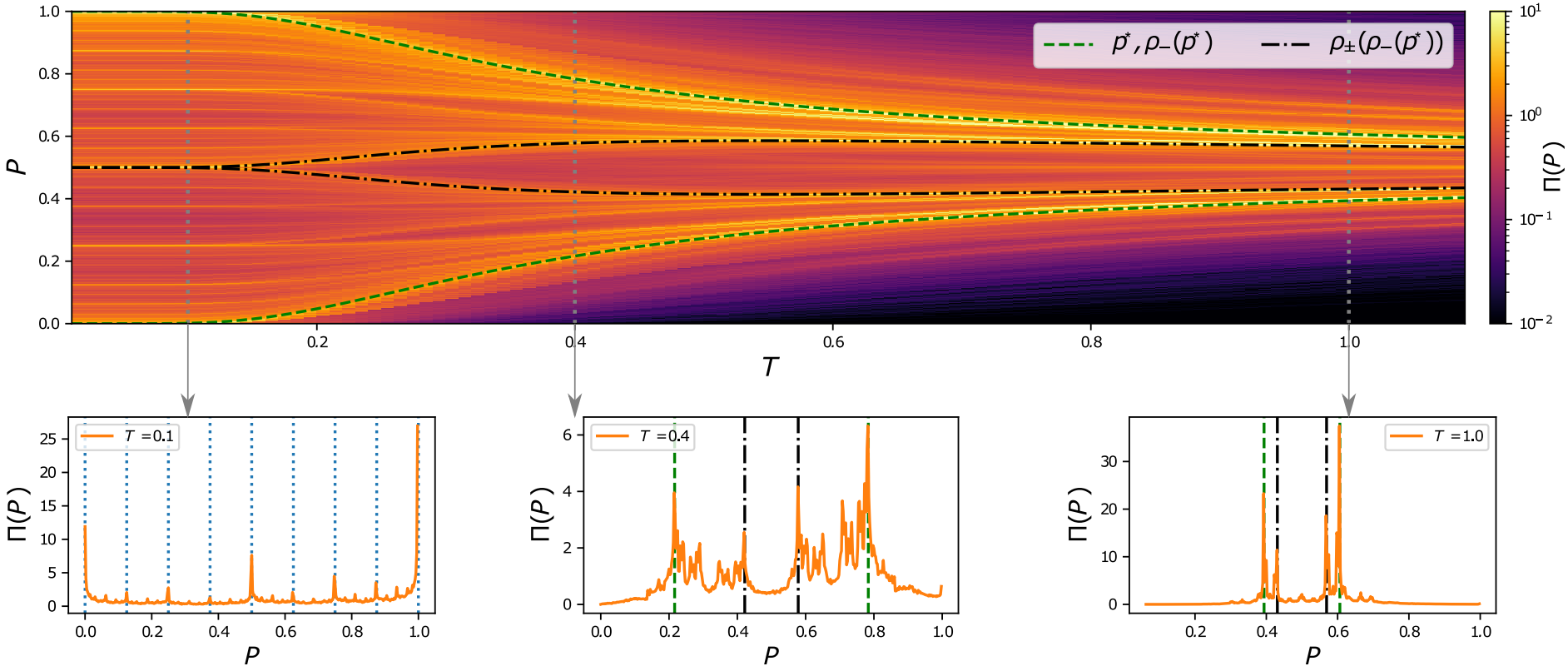


Results $P(J_{ij} \neq 0) = \eta\delta(J_{ij} - J) + (1-\eta)\delta(J_{ij} + J), \quad \eta = 0.62$



- Multimodal distribution $\Rightarrow$ node **heterogeneities**

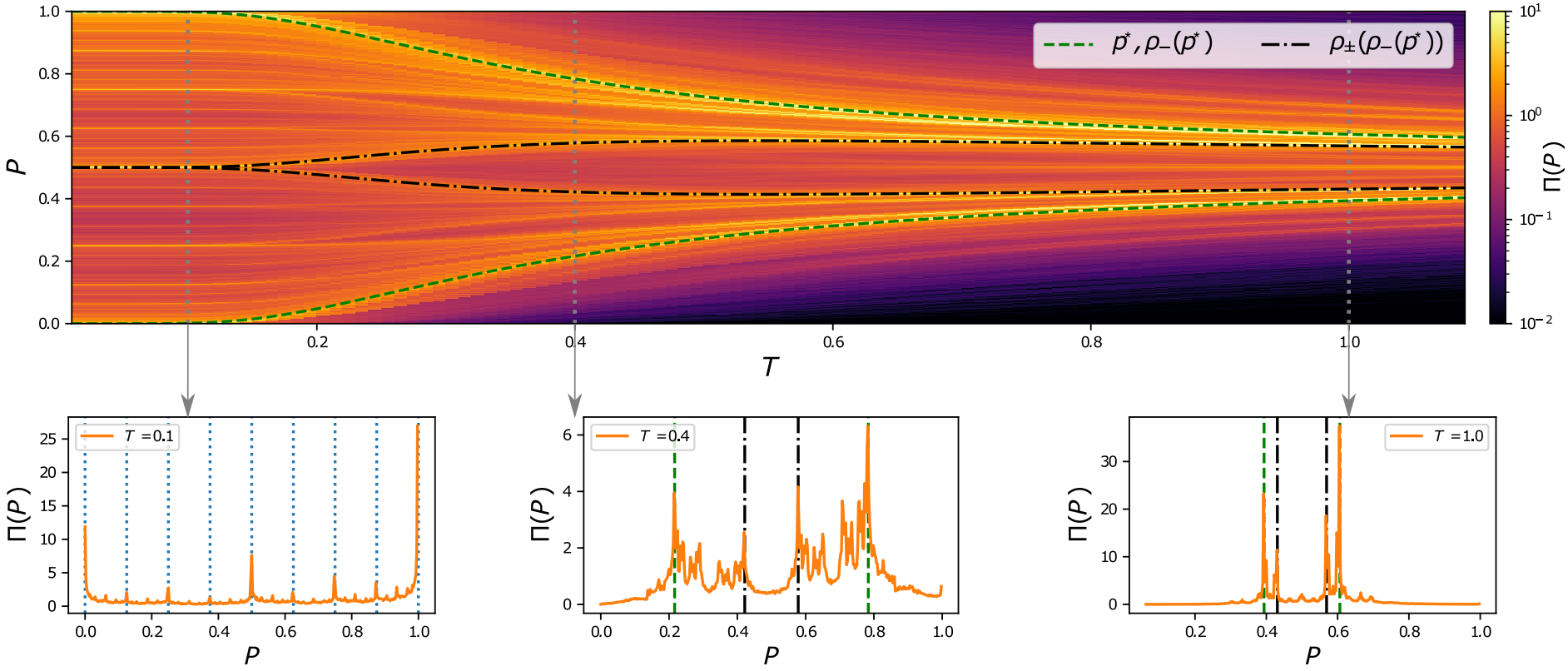
Results $P(J_{ij} \neq 0) = \eta\delta(J_{ij} - J) + (1-\eta)\delta(J_{ij} + J), \quad \eta = 0.62$



- Multimodal distribution $\Rightarrow$ node **heterogeneities**
- Let $k_i = 1$, $\partial_i = \{j\}$, $P_j = P$:

$$P_i = P\Phi_T(\pm J - \vartheta) + (1 - P)\Phi_T(-\vartheta) \equiv \rho_\pm(P)$$

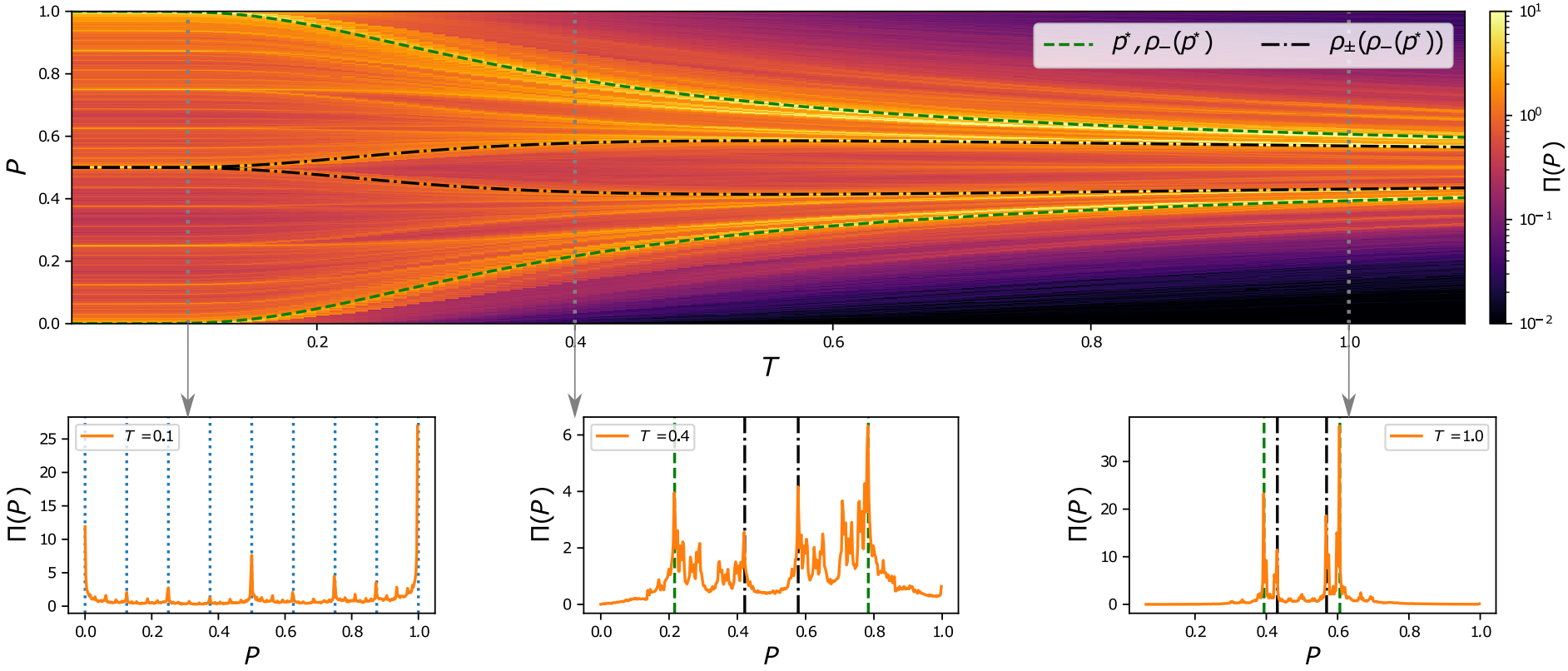
Results $P(J_{ij} \neq 0) = \eta\delta(J_{ij} - J) + (1-\eta)\delta(J_{ij} + J), \quad \eta = 0.62$



- Multimodal distribution $\Rightarrow$ node **heterogeneities**
- Let $k_i = 1$, $\partial_i = \{j\}$, $P_j = P$:

$$P_i = P\Phi_T(\pm J - \vartheta) + (1 - P)\Phi_T(-\vartheta) \equiv \rho_\pm(P)$$

Results $P(J_{ij}\neq 0) = \eta\delta(J_{ij}-J) + (1-\eta)\delta(J_{ij}+J), \quad \eta = 0.62$

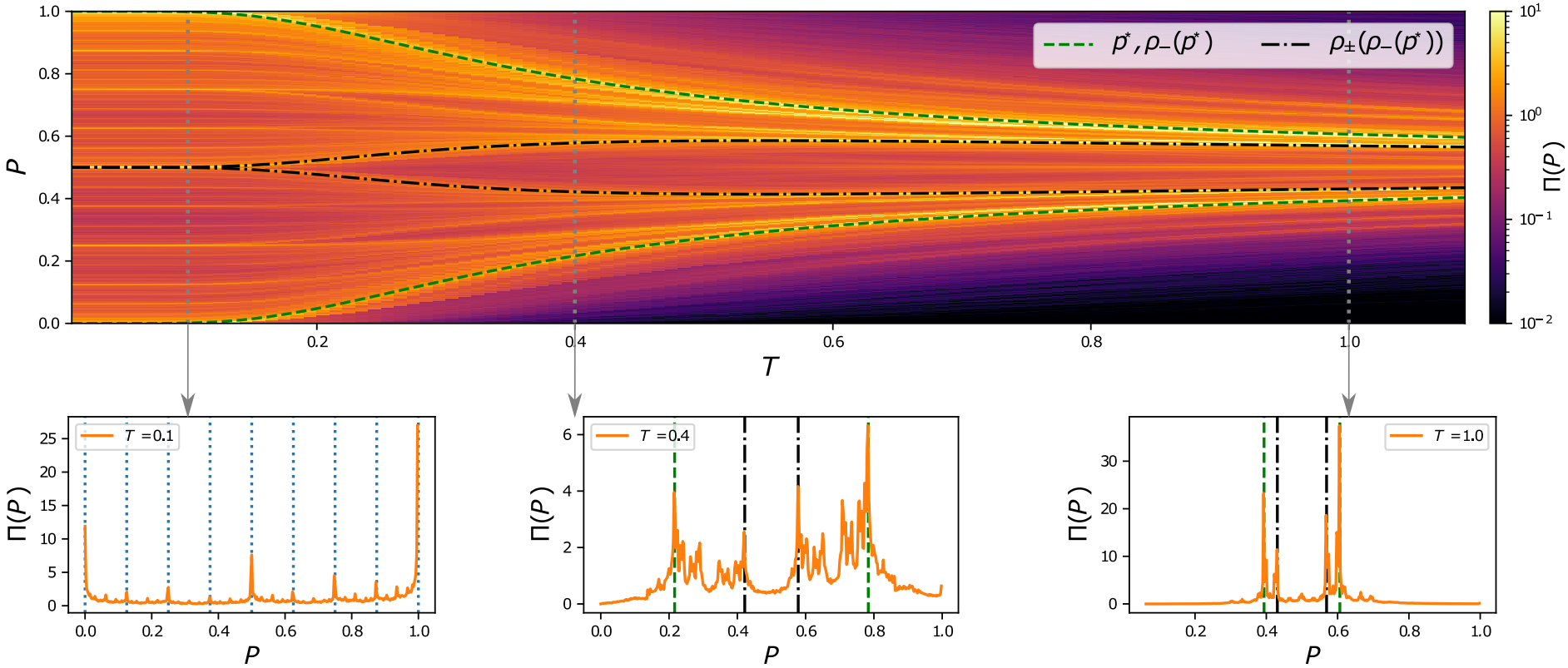


- Multimodal distribution $\Rightarrow$ node **heterogeneities**
- Let $k_i = 1$, $\partial_i = \{j\}$, $P_j = P$:

$$P_i = P\Phi_T(\pm J - \vartheta) + (1-P)\Phi_T(-\vartheta) \equiv \rho_\pm(P)$$

e.g. in FM chains $P_{i+1} = \rho_+(P_i) \quad \Rightarrow p^* = \rho_+(p^*)$

Results $P(J_{ij} \neq 0) = \eta\delta(J_{ij} - J) + (1-\eta)\delta(J_{ij} + J), \quad \eta = 0.62$



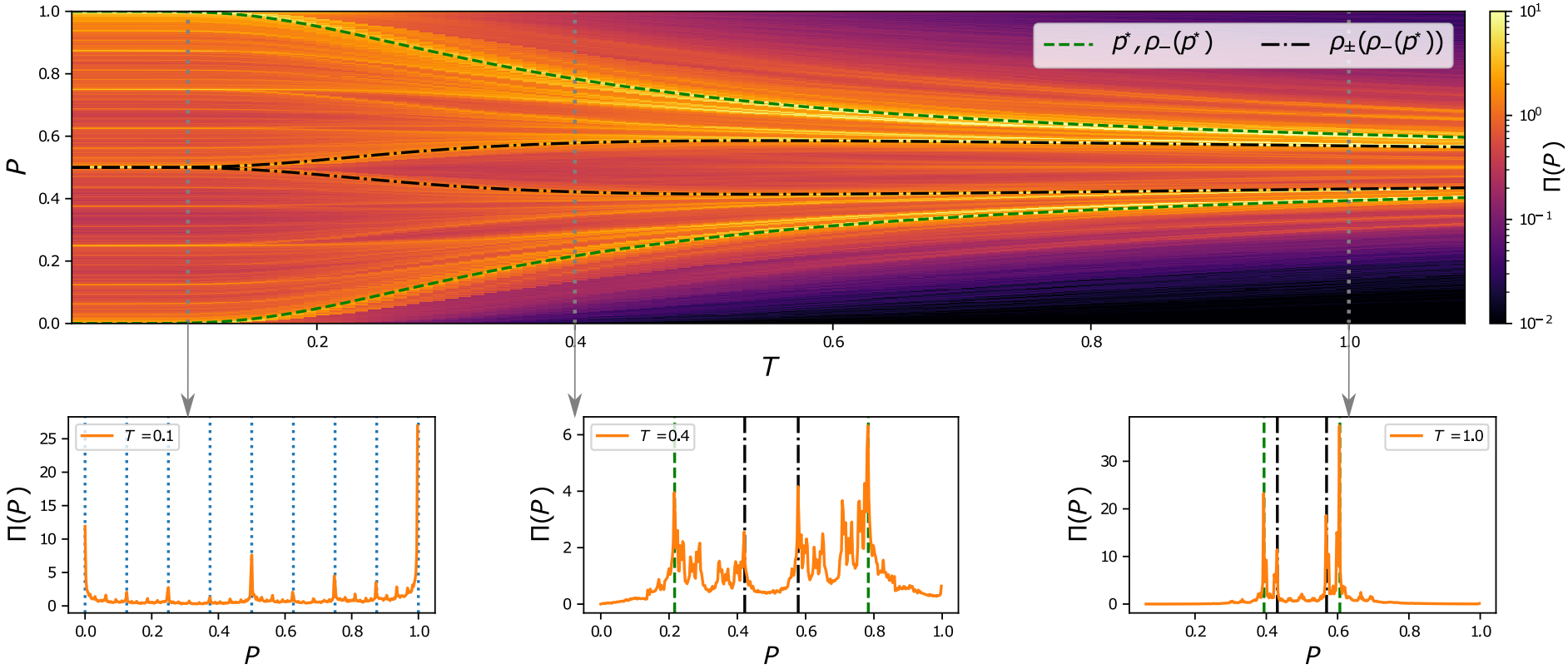
- Multimodal distribution $\Rightarrow$ node **heterogeneities**
- Let $k_i = 1$, $\partial_i = \{j\}$, $P_j = P$:

$$P_i = P\Phi_T(\pm J - \vartheta) + (1 - P)\Phi_T(-\vartheta) \equiv \rho_\pm(P)$$

e.g. in FM chains $P_{i+1} = \rho_+(P_i) \quad \Rightarrow p^* = \rho_+(p^*)$

Note: in fully asymmetric nets only **one** attractor

Results $P(J_{ij} \neq 0) = \eta\delta(J_{ij} - J) + (1-\eta)\delta(J_{ij} + J), \quad \eta = 0.62$



- Multimodal distribution $\Rightarrow$ node **heterogeneities**
- Let $k_i = 1$, $\partial_i = \{j\}$, $P_j = P$:

$$P_i = P\Phi_T(\pm J - \vartheta) + (1 - P)\Phi_T(-\vartheta) \equiv \rho_\pm(P)$$

e.g. in FM chains $P_{i+1} = \rho_+(P_i) \quad \Rightarrow p^* = \rho_+(p^*)$

Note: in fully asymmetric nets only **one** attractor [Torrisi, **AA**, Kühn, PRE (2021)]

Outline

One-time approximation

With **bi-directional** links $P^{(i)}(\boldsymbol{n}_{\partial_i}, t) \neq \prod_{j \in \partial_i} P^{(i)}(n_j, t)$ BUT

One-time approximation

With **bi-directional** links $P^{(i)}(\boldsymbol{n}_{\partial_i}, t) \neq \prod_{j\in\partial_i} P^{(i)}(n_j, t)$ BUT

$$P^{(i)}(\mathbf{n}_{\partial i}^{0\ldots T}|n_i^{0\ldots T}) = \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\ldots T}|n_i^{0\ldots T})$$

One-time approximation

With **bi-directional** links $P^{(i)}(\boldsymbol{n}_{\partial_i}, t) \neq \prod_{j\in\partial_i} P^{(i)}(n_j, t)$ BUT

$$P^{(i)}(\mathbf{n}_{\partial i}^{0\ldots T}|n_i^{0\ldots T}) = \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\ldots T}|n_i^{0\ldots T})$$

Have to work with *trajectories*

$$P_i(n_i^{0,\ldots,t}) = \sum_{\boldsymbol{n}_{\partial_i}^{0\ldots T}} (\ldots) \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\ldots T}|n_i^{0\ldots T})$$

One-time approximation

With **bi-directional** links $P^{(i)}(\boldsymbol{n}_{\partial_i}, t) \neq \prod_{j\in\partial_i} P^{(i)}(n_j, t)$ BUT

$$P^{(i)}(\mathbf{n}_{\partial i}^{0\ldots T}|n_i^{0\ldots T}) = \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\ldots T}|n_i^{0\ldots T})$$

Have to work with *trajectories*

$$P_i(n_i^{0,\ldots,t}) = \sum_{\boldsymbol{n}_{\partial_i}^{0\ldots T}} (\ldots) \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\ldots T}|n_i^{0\ldots T})$$

Additional complexity in time.. One-time step approximation [Neri Bollé 2009]

$$P^{(i)}(n_j^{0\ldots T}|n_i^{0\ldots T}) \simeq P_j^{(i)}(n_j^0) \prod_{s=1}^{t} P_j^{(i)}(n_j^s|n_i^{s-1})$$

One-time approximation

With **bi-directional** links $P^{(i)}(\boldsymbol{n}_{\partial_i}, t) \neq \prod_{j\in\partial_i} P^{(i)}(n_j, t)$ BUT

$$P^{(i)}(\mathbf{n}_{\partial i}^{0\dots T}|n_i^{0\dots T}) = \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\dots T}|n_i^{0\dots T})$$

Have to work with *trajectories*

$$P_i(n_i^{0,\dots,t}) = \sum_{\boldsymbol{n}_{\partial_i}^{0\dots T}} (\dots) \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\dots T}|n_i^{0\dots T})$$

Additional complexity in time.. One-time step approximation [Neri Bollé 2009]

$$P^{(i)}(n_j^{0\dots T}|n_i^{0\dots T}) \simeq P_j^{(i)}(n_j^0) \prod_{s=1}^{t} P_j^{(i)}(n_j^s|n_i^{s-1})$$

$\Rightarrow$ recursion for cavity marginals $P_i^{(\ell)}(n_i^t|n_\ell^{t-1})$ in terms of $P_j^{(i)}(n_j^{t-1}|n_i^{t-2})$

One-time approximation

With **bi-directional** links $P^{(i)}(\boldsymbol{n}_{\partial_i}, t) \neq \prod_{j\in\partial_i} P^{(i)}(n_j, t)$ BUT

$$P^{(i)}(\mathbf{n}_{\partial i}^{0\dots T}|n_i^{0\dots T}) = \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\dots T}|n_i^{0\dots T})$$

Have to work with *trajectories*

$$P_i(n_i^{0,\dots,t}) = \sum_{\boldsymbol{n}_{\partial_i}^{0\dots T}} (\dots) \prod_{j\in\partial_i} P_j^{(i)}(n_j^{0\dots T}|n_i^{0\dots T})$$

Additional complexity in time.. One-time step approximation [Neri Bollé 2009]

$$P^{(i)}(n_j^{0\dots T}|n_i^{0\dots T}) \simeq P_j^{(i)}(n_j^0) \prod_{s=1}^{t} P_j^{(i)}(n_j^s|n_i^{s-1})$$

$\Rightarrow$ recursion for cavity marginals $P_i^{(\ell)}(n_i^t|n_\ell^{t-1})$ in terms of $P_j^{(i)}(n_j^{t-1}|n_i^{t-2})$

Similar equation for $P_i(n_i^t)$ .. **Both** benefit from dynamic programming!

Symmetry breaking

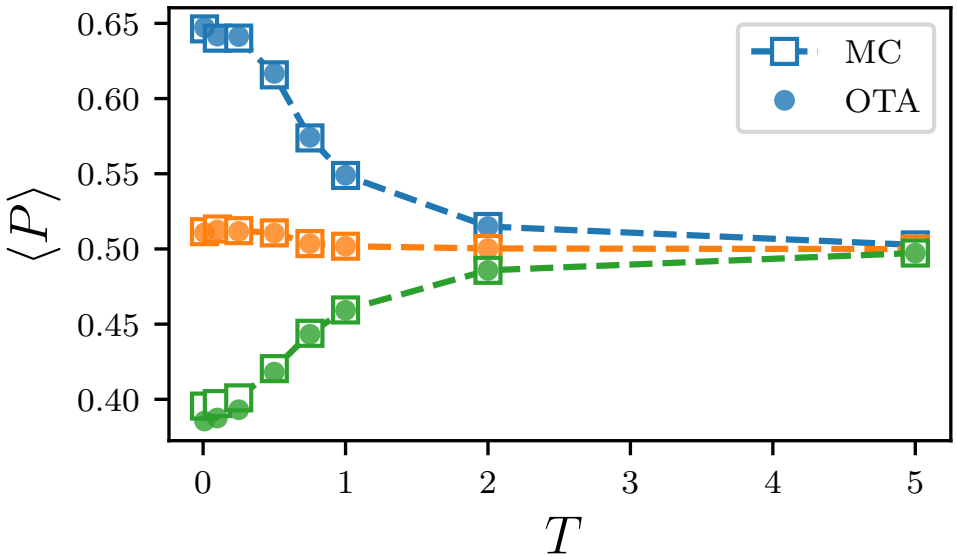
Consider (i) $J_{ij} = J_{ji}$; (ii) $J_{ij} = -J_{ji}$; (iii) $P(J_{ij}, J_{ji}) = P(J_{ij})P(J_{ji})$

Symmetry breaking

Consider (i) $J_{ij} = J_{ji}$; (ii) $J_{ij} = -J_{ji}$; (iii) $P(J_{ij}, J_{ji}) = P(J_{ij})P(J_{ji})$

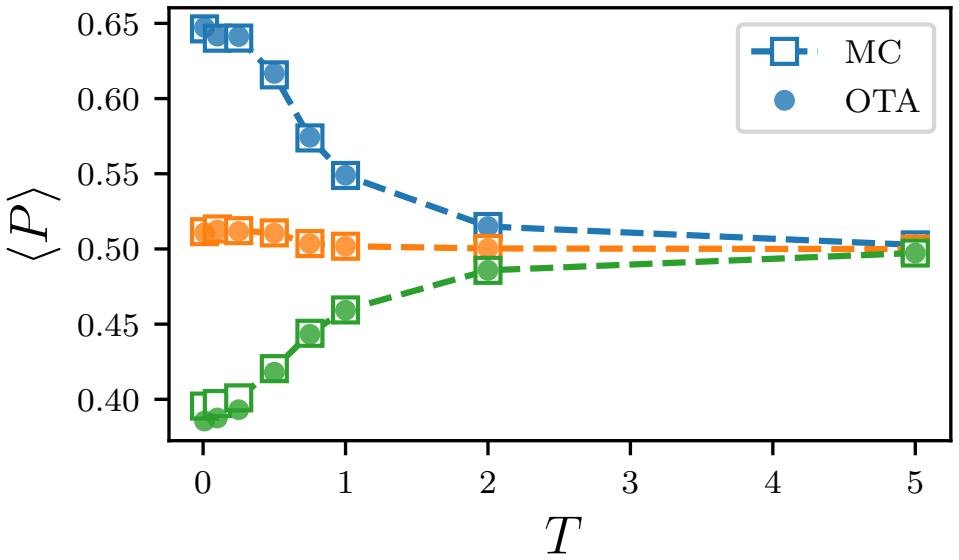
Symmetry breaking

Consider (i) $J_{ij} = J_{ji}$; (ii) $J_{ij} = -J_{ji}$; (iii) $P(J_{ij}, J_{ji}) = P(J_{ij})P(J_{ji})$



Symmetry breaking

Consider (i) $J_{ij} = J_{ji}$; (ii) $J_{ij} = -J_{ji}$; (iii) $P(J_{ij}, J_{ji}) = P(J_{ij})P(J_{ji})$



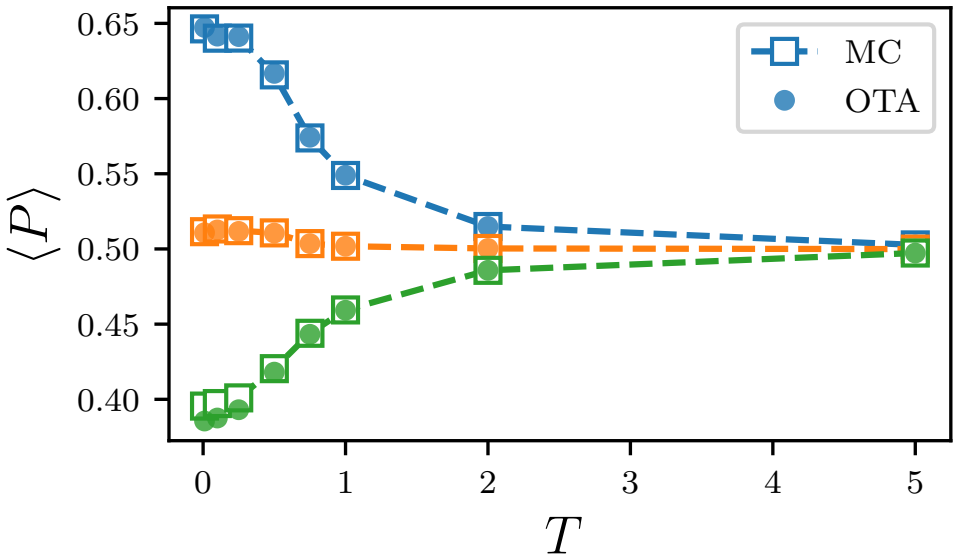
$$\langle P \rangle = N^{-1} \sum_i P_i$$

$$\theta = 0$$

$$\langle J_{ij} \rangle = 0$$

Symmetry breaking

Consider (i) $J_{ij} = J_{ji}$; (ii) $J_{ij} = -J_{ji}$; (iii) $P(J_{ij}, J_{ji}) = P(J_{ij})P(J_{ji})$



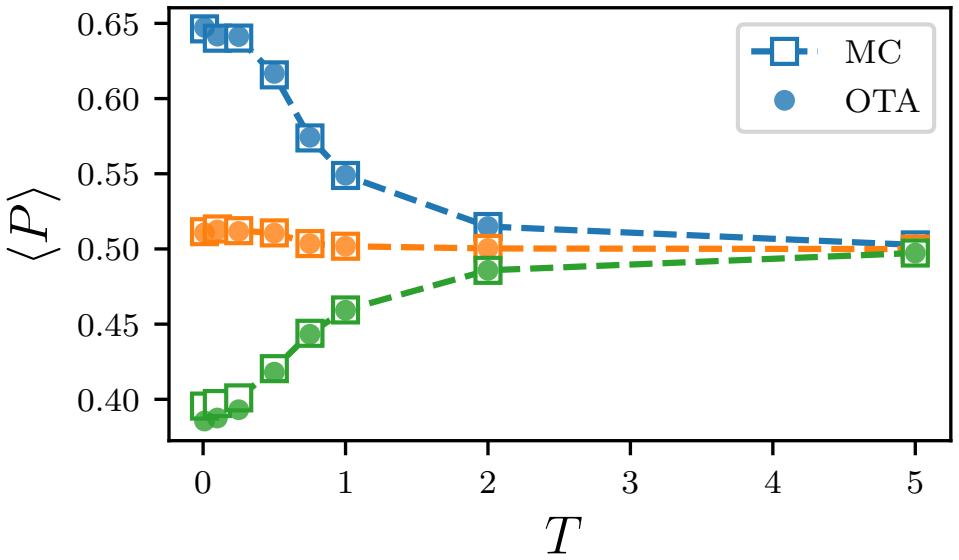
$$\langle P \rangle = N^{-1} \sum_i P_i$$

$$\theta = 0$$

$$\langle J_{ij} \rangle = 0$$

Symmetry breaking

Consider (i) $J_{ij} = J_{ji}$; (ii) $J_{ij} = -J_{ji}$; (iii) $P(J_{ij}, J_{ji}) = P(J_{ij})P(J_{ji})$



$$\langle P \rangle = N^{-1} \sum_i P_i$$

$$\theta = 0$$

$$\langle J_{ij} \rangle = 0$$

$\Rightarrow$ bias towards activation or quiescence $\Rightarrow$ **Symmetry breaking**

[G Torrisi, R Kühn, **AA**, JSTAT (2022)]

Outline

Extensions: Multi-node and self-interactions

Extensions: Multi-node and self-interactions

Extensions: Multi-node and self-interactions

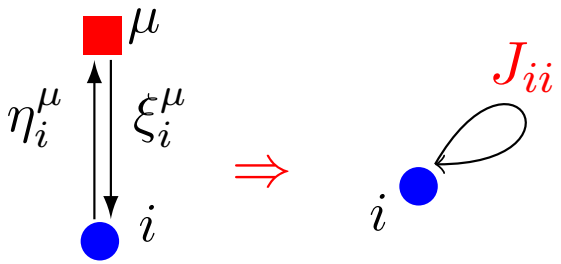
- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

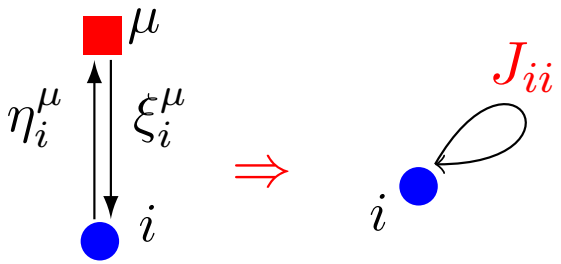
  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

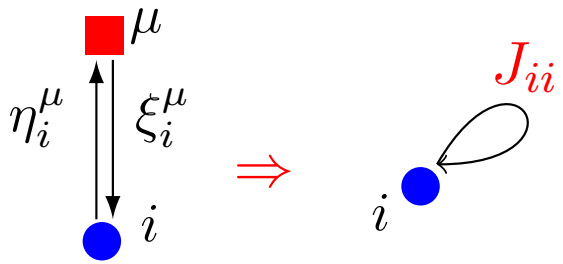
  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$

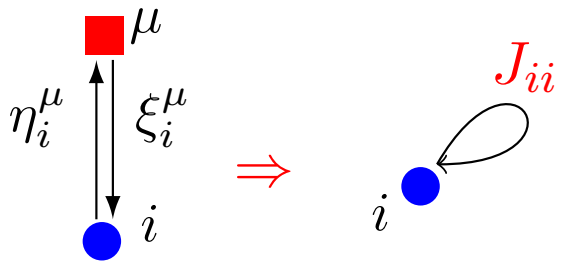


- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$

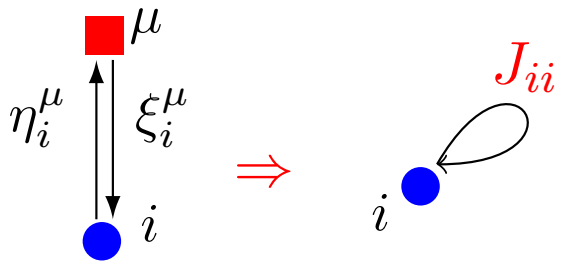


- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$

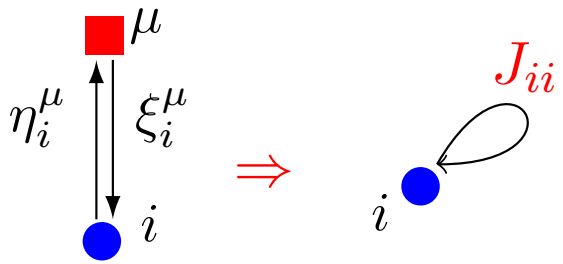


- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$ **X**

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$

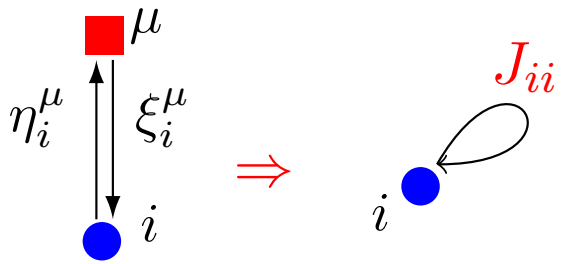


- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$ **X**

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



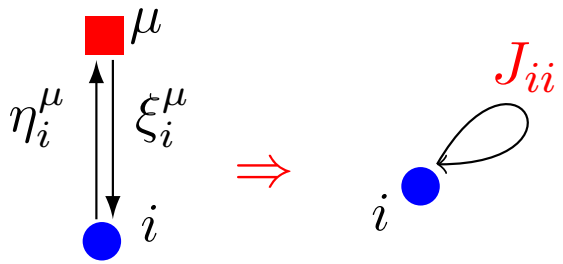
- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)]$, $\quad J_{ii} \neq 0$ **X**

  Map: N **nodes** and **self-interactions** $\Rightarrow 2N$ and **bi-directional** links

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)]$, $\quad J_{ii} \neq 0$ **X**

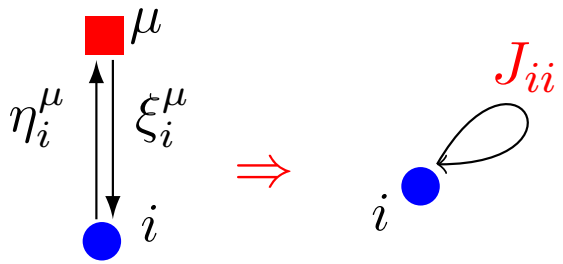
  Map: N **nodes** and **self-interactions** $\Rightarrow 2N$ and **bi-directional** links

  ... evolving according to linear threshold model **V**

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$ **X**

  Map: N **nodes** and **self-interactions** $\Rightarrow 2N$ and **bi-directional** links

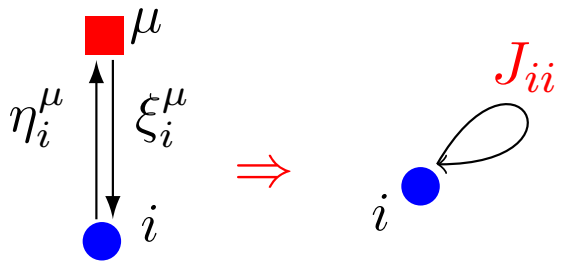
  ... evolving according to linear threshold model **V**

- AND: $\tau_\mu(t) = \prod_{j:\eta_j^\mu = 1} n_j(t) \;\Rightarrow\; \tau_\mu(t) = \Theta[\sum_j \eta_j^\mu n_j(t) - c_\mu + \epsilon]$

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$ **X**

  Map: N **nodes** and **self-interactions** $\Rightarrow 2N$ and **bi-directional** links

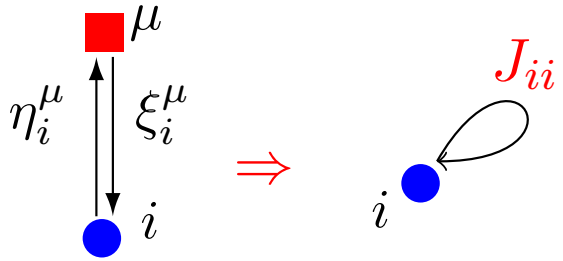
  ... evolving according to linear threshold model **V**

- AND: $\tau_\mu(t) = \prod_{j:\eta_j^\mu = 1} n_j(t) \;\Rightarrow\; \tau_\mu(t) = \Theta[\sum_j \eta_j^\mu n_j(t) - c_\mu + \epsilon]$

Extensions: Multi-node and self-interactions

- Bi-directional links in *bipartite* graphs:

  $P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) = p$



- OR: $n_i(t+1) = \Theta[\sum_j J_{ij} n_j(t) - \vartheta_i - z_i(t)], \quad J_{ii} \neq 0$ **X**

  Map: N **nodes** and **self-interactions** $\Rightarrow 2N$ and **bi-directional** links

  ... evolving according to linear threshold model **V**

- AND: $\tau_\mu(t) = \prod_{j:\eta_j^\mu = 1} n_j(t) \;\Rightarrow\; \tau_\mu(t) = \Theta[\sum_j \eta_j^\mu n_j(t) - c_\mu + \epsilon]$

  $N + P$ nodes, bi-directional links, linear threshold model **V**

Extensions: Multi-node and self-interactions

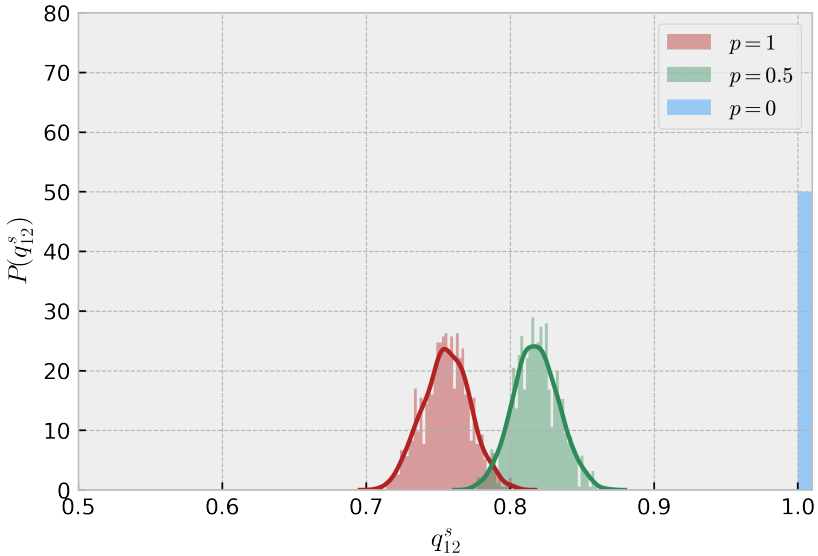
- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors

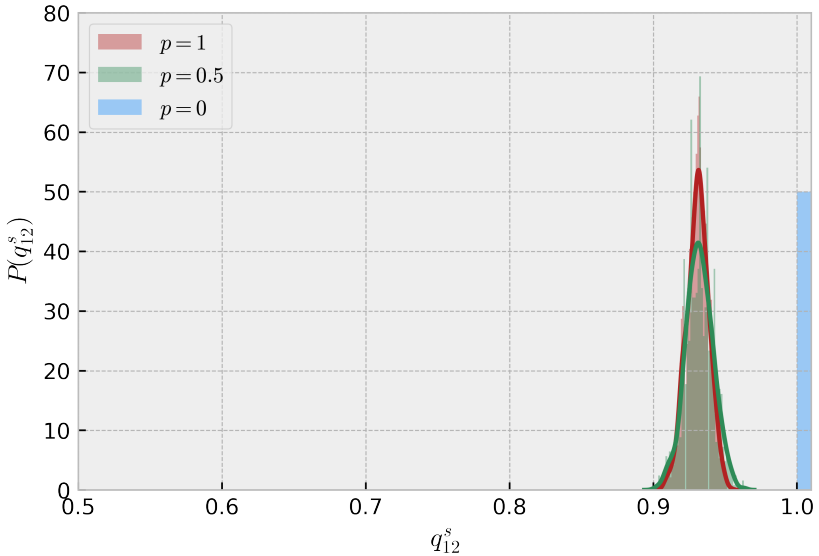
Extensions: Multi-node and self-interactions

- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors
- Similarity of attractors: AND (left), OR (right)

Extensions: Multi-node and self-interactions

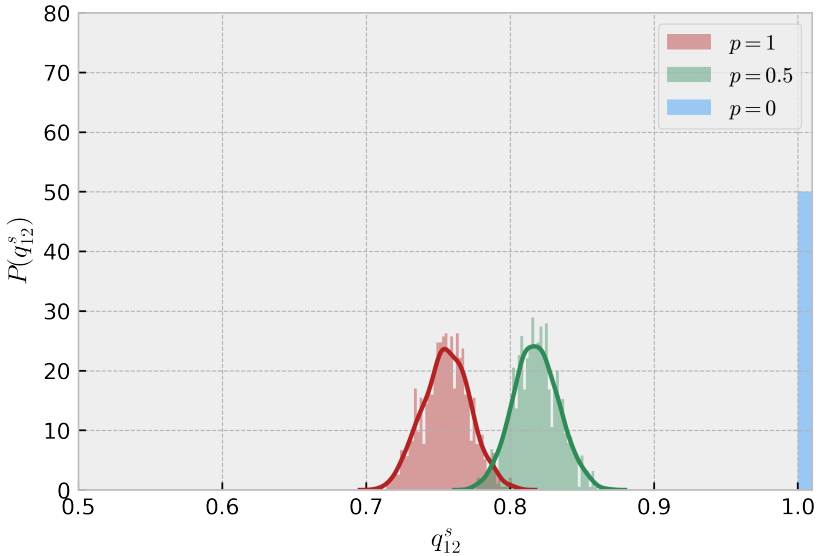
- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors
- Similarity of attractors: AND (left), OR (right)

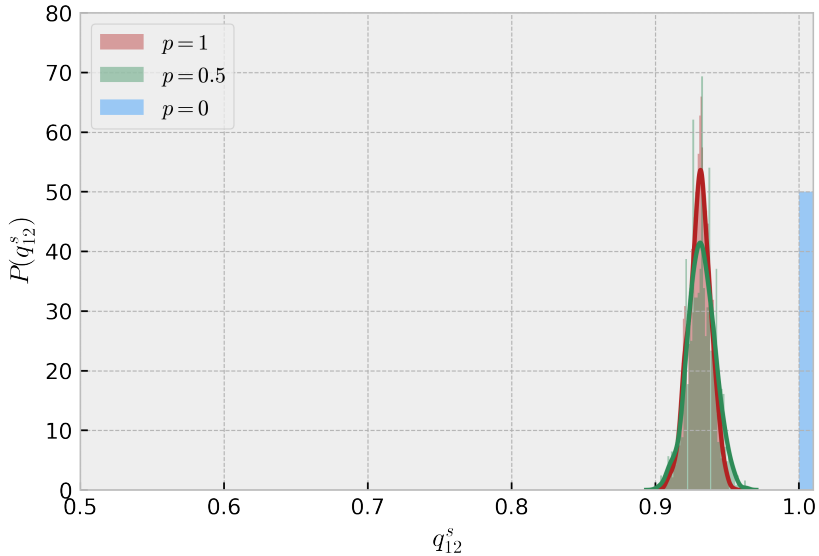




Extensions: Multi-node and self-interactions

- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors
- Similarity of attractors: AND (left), OR (right)

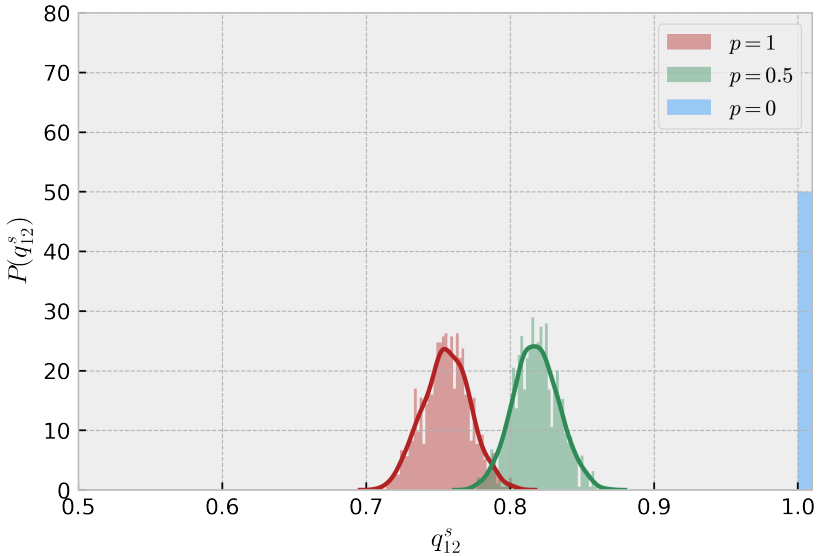


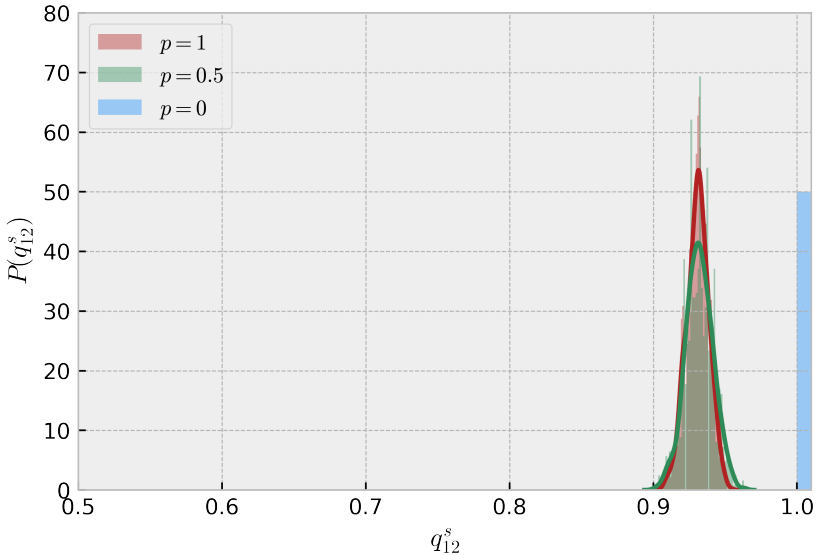


Self-regulation crucial for **multiplicity** of attractors

Extensions: Multi-node and self-interactions

- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors
- Similarity of attractors: AND (left), OR (right)



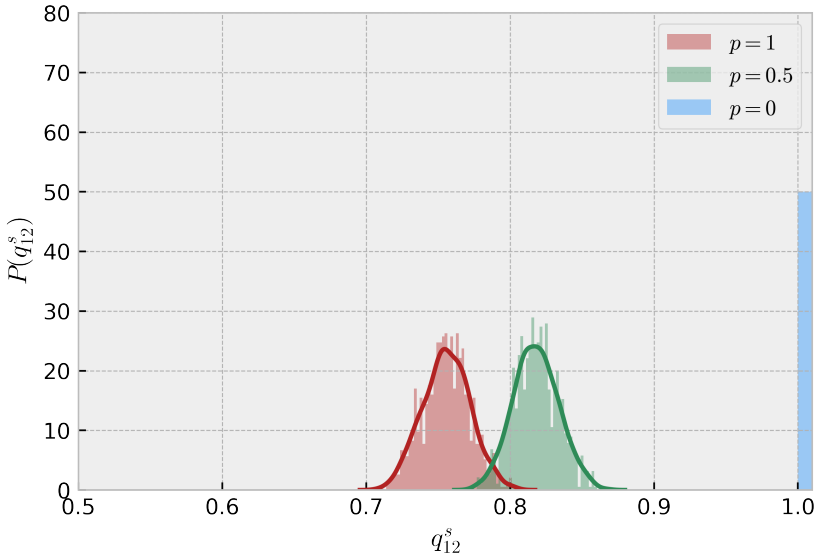


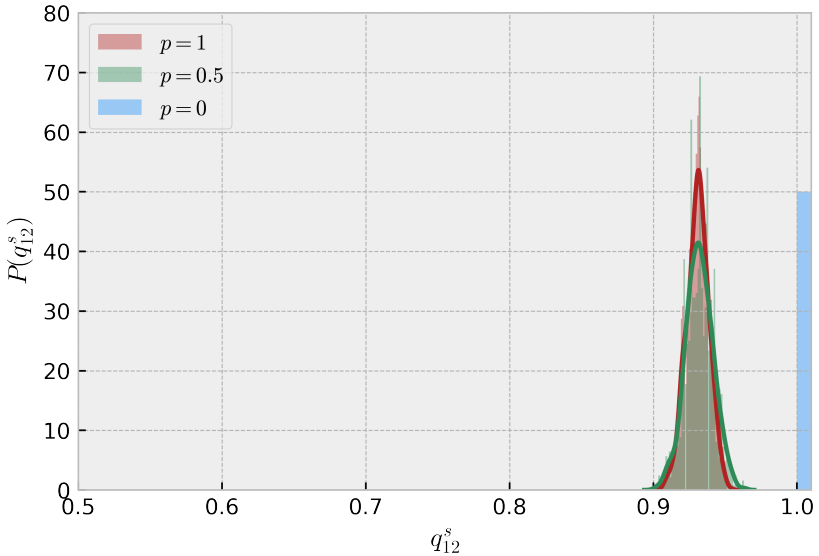
Self-regulation crucial for **multiplicity** of attractors

Cooperativity promotes **diversity**

Extensions: Multi-node and self-interactions

- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors
- Similarity of attractors: AND (left), OR (right)





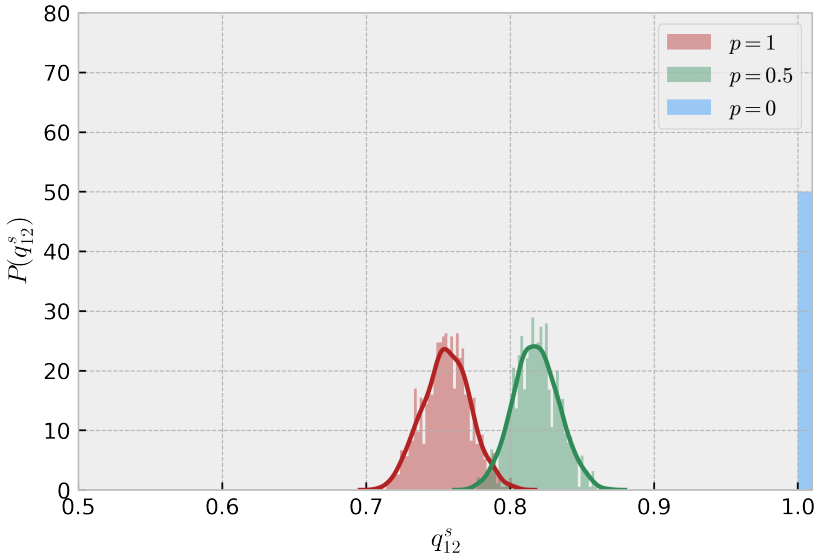
Self-regulation crucial for **multiplicity** of attractors

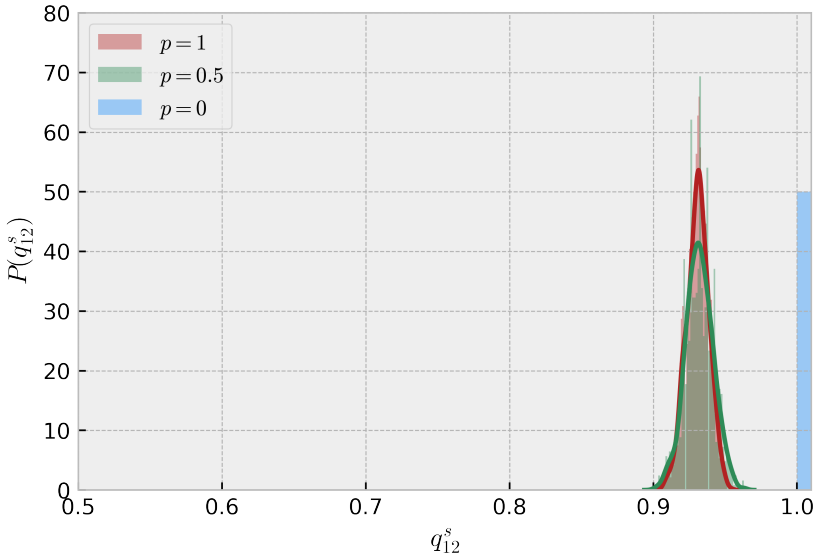
Cooperativity promotes **diversity**

Both common features of GRNs $\Rightarrow$ Sustain **multi-cellular life**?

Extensions: Multi-node and self-interactions

- $p = P(\xi_i^\mu \neq 0 | \eta_i^\mu = 1) \Rightarrow$ at low T **multiplicity** of attractors
- Similarity of attractors: AND (left), OR (right)





Self-regulation crucial for **multiplicity** of attractors

Cooperativity promotes **diversity**

Both common features of GRNs $\Rightarrow$ Sustain **multi-cellular life**?

[Hurry, Mozeika, AA, JPA (2022)]

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps
  - extensions to bi-directional links via OTA

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps
  - extensions to bi-directional links via OTA
  - Unbiased interactions can sustain activation or quiescence

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps
  - extensions to bi-directional links via OTA
  - Unbiased interactions can sustain activation or quiescence
  - Extensions to multi-node & self-interactions

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps
  - extensions to bi-directional links via OTA
  - Unbiased interactions can sustain activation or quiescence
  - Extensions to multi-node & self-interactions
  - Multiplicity of attractors at low T

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps
  - extensions to bi-directional links via OTA
  - Unbiased interactions can sustain activation or quiescence
  - Extensions to multi-node & self-interactions
  - Multiplicity of attractors at low T
  - Multi-node interactions favour diversity of attractors

Discussion

- Constructed a minimal model with hierarchically organized cell cycles inspired by **Neural Networks**
- Investigated reprogramming:
  - Takes several cycles
  - Can be achieved with realistic numbers of TFs
- **Bipartite models**
  - Percolation: for a cell to exist, TFs typically small protein complexes that regulate many genes
  - Dynamics: efficient algorithm based on dynamic programming to calculate gene expression profile
  - for fully asymmetric networks, distribution of node activation has rich structure; Salient features rationalised in terms of discrete stochastic maps
  - extensions to bi-directional links via OTA
  - Unbiased interactions can sustain activation or quiescence
  - Extensions to multi-node & self-interactions
  - Multiplicity of attractors at low T
  - Multi-node interactions favour diversity of attractors
  - Still many unanswered questions... the fight Maths vs GRNs continues!

Acknowledgement

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Many thanks Reimer!

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Many thanks Reimer!

Many thanks for listening!

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Many thanks Reimer!

Many thanks for listening!

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Many thanks Reimer!

Many thanks for listening!

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Many thanks Reimer!

Many thanks for listening!

Acknowledgement

PhD Students
Ryan Hannam
Giuseppe Torrisi
Christian Hurry

Collaborators
Elena Agliari (Sapienza, Rome)
Adriano Barra (Unisalento, Lecce)
Anthony Coolen (Radboud, Nijmegen)
Reimer Kühn (KCL, London)
Alexander Mozeika (KCL, London)
Peter Sollich (ITP, Göttingen)
Daniele Tantari (Università di Bologna)

Many thanks Reimer!

Many thanks for listening!

Dynamic programming

$$P_i(t+1) = \langle \Phi_T \left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i \right) \rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad h_i(\boldsymbol{n}_{\partial_i}) = \sum_j J_{ij} n_j$$

- Let $\partial_i = \{1, \ldots, k_i\}$ and def. average over **subset** of nodes

$$f_i(\ell, \tilde{h}) = \left\langle \Phi_T \left(\tilde{h} + \sum_{j=\ell}^{k_i} J_{ij} n_j - \vartheta_i \right) \right\rangle_{n_\ell, \ldots, k_i, t} \qquad \Rightarrow \qquad P_i(t+1) = f_i(1, 0)$$

$$\tilde{h} = \text{auxiliary field}$$

Dynamic programming

$$P_i(t+1) = \langle \Phi_T \left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i \right) \rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad h_i(\boldsymbol{n}_{\partial_i}) = \sum_j J_{ij} n_j$$

- Let $\partial_i = \{1, \ldots, k_i\}$ and def. average over **subset** of nodes

$$f_i(\ell, \tilde{h}) = \left\langle \Phi_T \left(\tilde{h} + \sum_{j=\ell}^{k_i} J_{ij} n_j - \vartheta_i \right) \right\rangle_{n_\ell, \ldots, k_i, t} \qquad \Rightarrow \qquad P_i(t+1) = f_i(1, 0)$$

$$\tilde{h} = \text{auxiliary field}$$

Dynamic programming

$$P_i(t+1) = \langle \Phi_T \left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i \right) \rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad h_i(\boldsymbol{n}_{\partial_i}) = \sum_j J_{ij} n_j$$

- Let $\partial_i = \{1, \ldots, k_i\}$ and def. average over **subset** of nodes

$$f_i(\ell, \tilde{h}) = \left\langle \Phi_T \left(\tilde{h} + \sum_{j=\ell}^{k_i} J_{ij} n_j - \vartheta_i \right) \right\rangle_{n_\ell, \ldots, k_i, t} \quad \Rightarrow \quad P_i(t+1) = f_i(1, 0)$$

$$\tilde{h} = \text{auxiliary field}$$

$f_i(\ell, \tilde{h})$ obtained from **backward recursion**

$$f_i(\ell, \tilde{h}) = P_\ell(t)\, f_i(\ell+1, \tilde{h} + J_{i\ell}) + \left(1 - P_\ell(t)\right) f_i(\ell+1, \tilde{h})$$

Dynamic programming

$$P_i(t+1) = \langle \Phi_T \left(h_i(\boldsymbol{n}_{\partial_i}) - \vartheta_i \right) \rangle_{\boldsymbol{n}_{\partial_i}, t} \qquad h_i(\boldsymbol{n}_{\partial_i}) = \sum_j J_{ij} n_j$$

- Let $\partial_i = \{1, \ldots, k_i\}$ and def. average over **subset** of nodes

$$f_i(\ell, \tilde{h}) = \left\langle \Phi_T \left(\tilde{h} + \sum_{j=\ell}^{k_i} J_{ij} n_j - \vartheta_i \right) \right\rangle_{n_\ell, \ldots, k_i, t} \quad \Rightarrow \quad P_i(t+1) = f_i(1, 0)$$

$$\tilde{h} = \text{auxiliary field}$$

$f_i(\ell, \tilde{h})$ obtained from **backward recursion**

$$f_i(\ell, \tilde{h}) = P_\ell(t)\, f_i(\ell+1, \tilde{h} + J_{i\ell}) + \big(1 - P_\ell(t)\big)\, f_i(\ell+1, \tilde{h})$$

with **terminal boundary condition** $f_i(k_i + 1, \tilde{h}) = \Phi_T \left(\tilde{h} - \vartheta_i \right)$

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell + 1, \tilde{h})$ and $f_i(\ell + 1, \tilde{h} \pm J)$

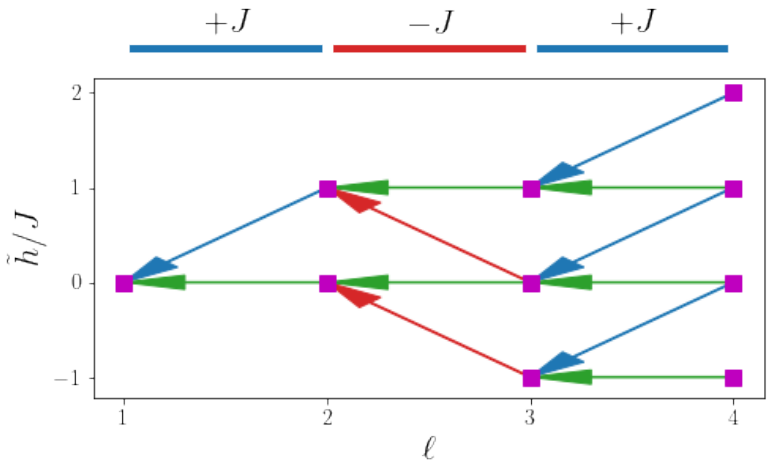
$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid

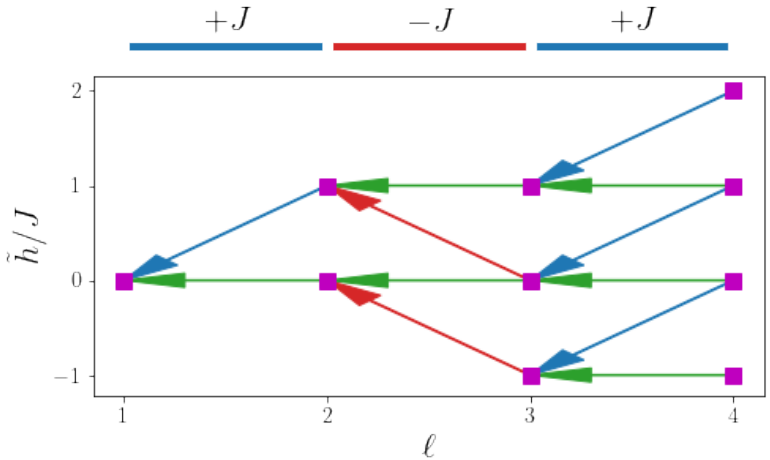


Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid



$$k_i = 3, \quad \partial_i = \{1, 2, 3\}$$

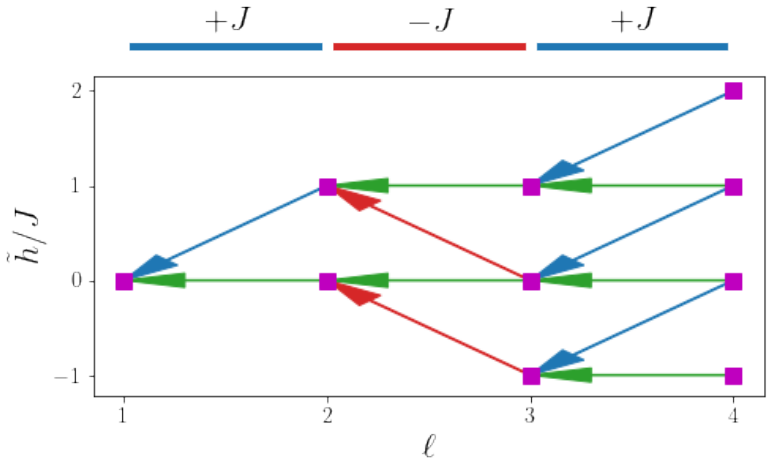
$$J_{i1} = +J; \; J_{i2} = -J; \; J_{i3} = +J$$

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid



$$k_i = 3, \quad \partial_i = \{1, 2, 3\}$$

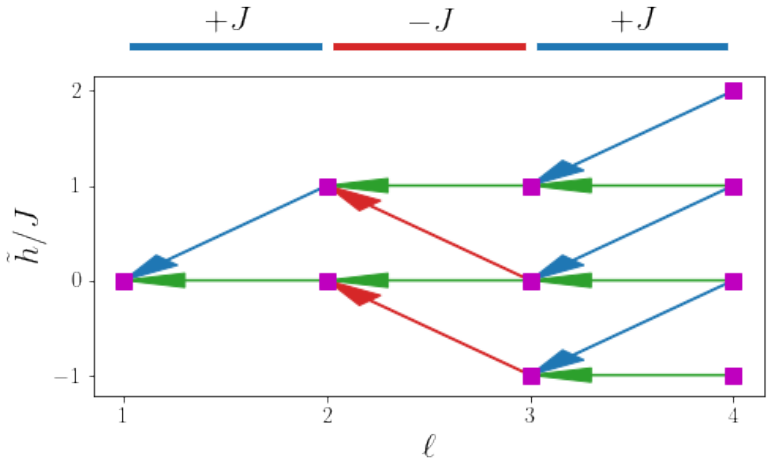
$$J_{i1} = +J; \; J_{i2} = -J; \; J_{i3} = +J$$

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid



$$k_i = 3, \quad \partial_i = \{1, 2, 3\}$$

$$J_{i1} = +J; \; J_{i2} = -J; \; J_{i3} = +J$$

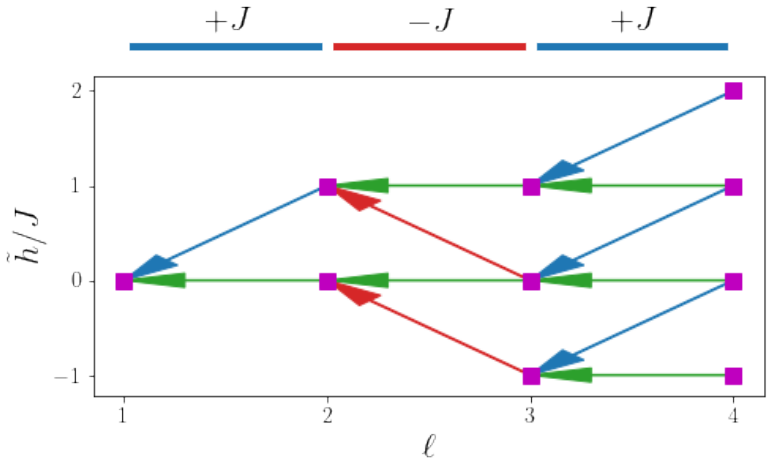
$$\text{Nr eval.} = \sum_{\ell=1}^{k_i+1} \ell = (k_i+1)(k_i+2)/2 \qquad \forall \; i$$

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid



$$k_i = 3, \quad \partial_i = \{1, 2, 3\}$$

$$J_{i1} = +J; \; J_{i2} = -J; \; J_{i3} = +J$$

$$\text{Nr eval.} = \sum_{\ell=1}^{k_i+1} \ell = (k_i+1)(k_i+2)/2 \qquad \forall\, i$$

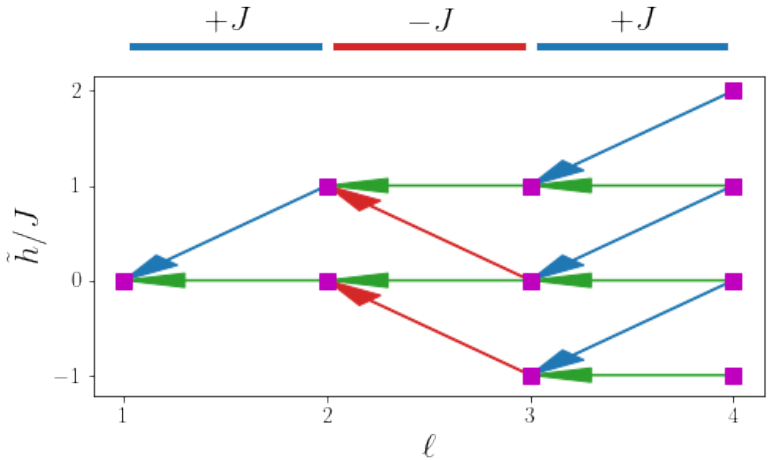
Complexity $\mathcal{O}(\sum_i 2^{k_i}) \quad \Rightarrow \quad \mathcal{O}(\sum_i k_i^2)$

Dynamic programming [Torrisi, **AA**, Kühn, PRE (2021)]

For $J_{ij} \in \{0, \pm J\}$:

at each ℓ, $f_i(\ell, \tilde{h})$ requires $f_i(\ell+1, \tilde{h})$ and $f_i(\ell+1, \tilde{h} \pm J)$

$\Rightarrow$ evaluation of $f_i(\ell, \tilde{h})$ only required on discrete grid



$$k_i = 3, \quad \partial_i = \{1, 2, 3\}$$

$$J_{i1} = +J; \; J_{i2} = -J; \; J_{i3} = +J$$

$$\text{Nr eval.} = \sum_{\ell=1}^{k_i+1} \ell = (k_i+1)(k_i+2)/2 \qquad \forall\ i$$

Complexity $\mathcal{O}(\sum_i 2^{k_i}) \quad \Rightarrow \quad \mathcal{O}(\sum_i k_i^2)$

Similar reduction for $J_{ij} \in \{-r_i J_i, \ldots, -J_i, 0, J_i, \ldots, s_i J_i\}$