NETADIS

Statistical Physics Approaches to Networks Across Disciplines

Learning and inference in equilibrium and non-equilibrium Ising models

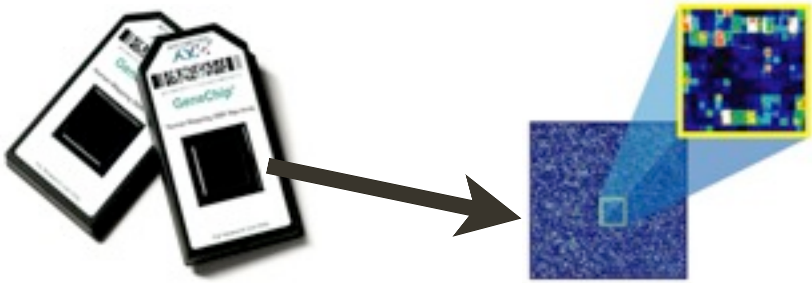
Yasser Roudi

Kavli Institute, Trondheim, Norway
NORDITA, Stockholm, Sweden

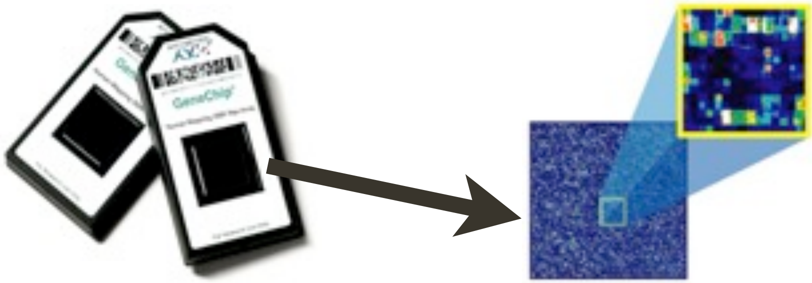
The Kavli Institute for Systems Neuroscience
Centre for the biology of memory

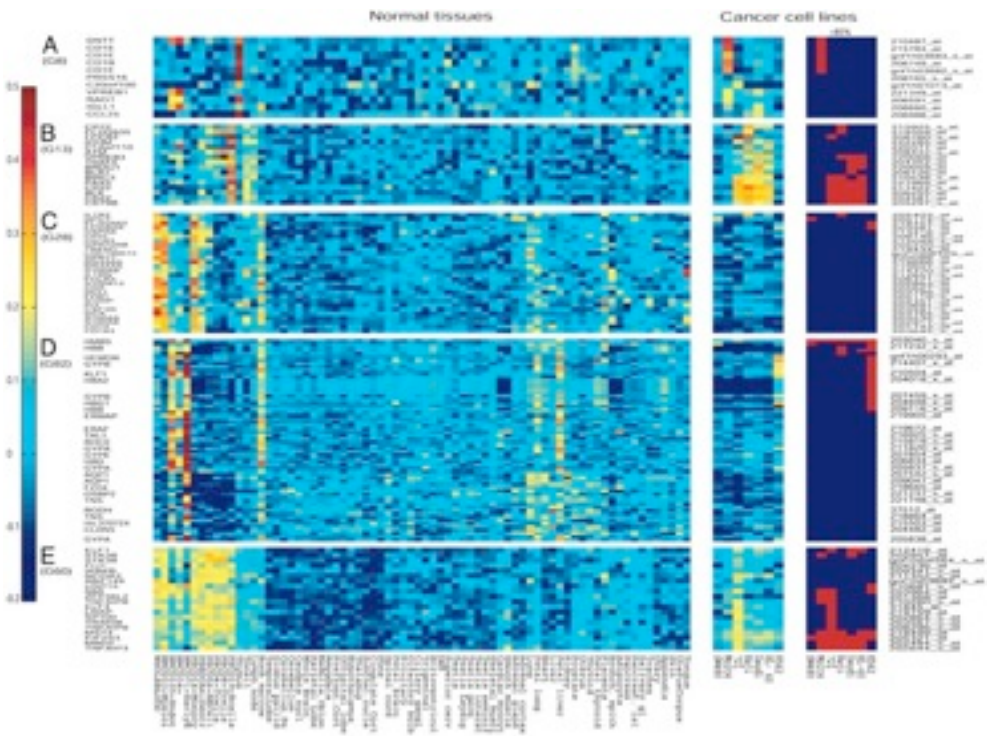
NORDITA
Nordic Institute for Theoretical Physics

high-throughput data in biology

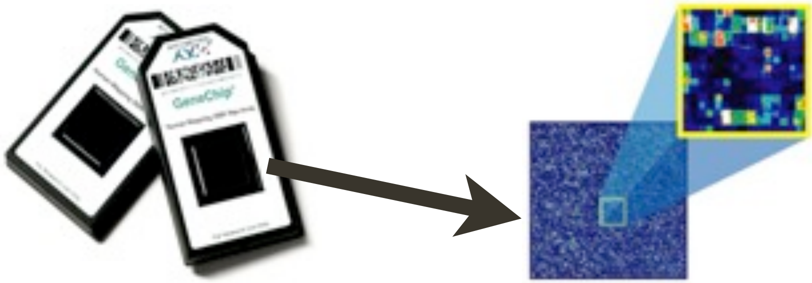


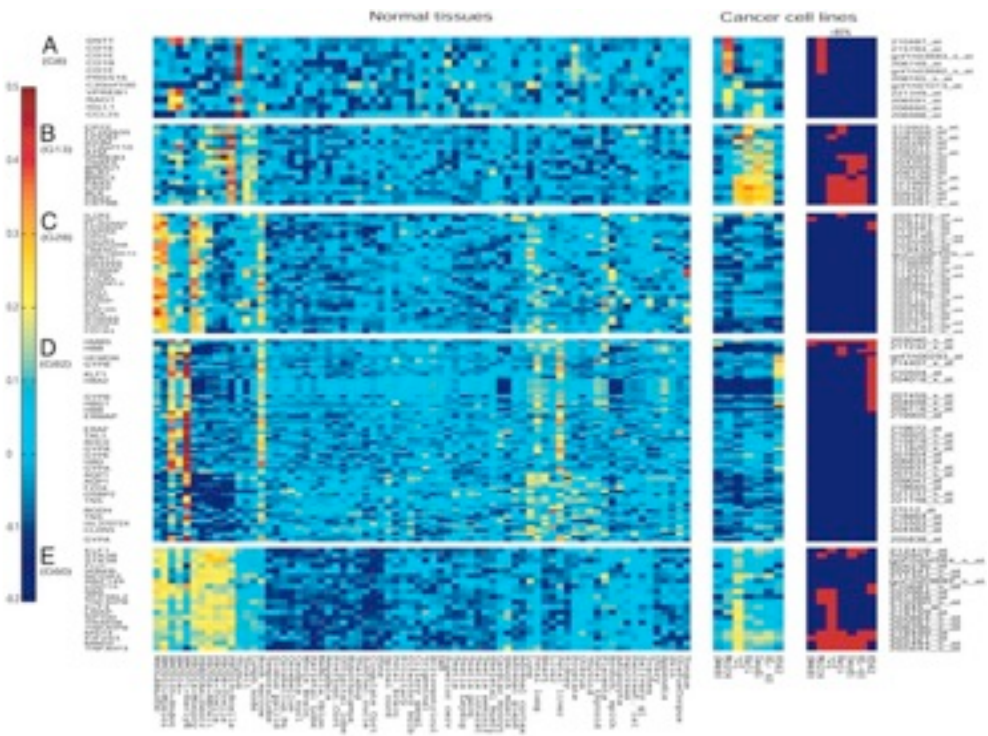
high-throughput data in biology

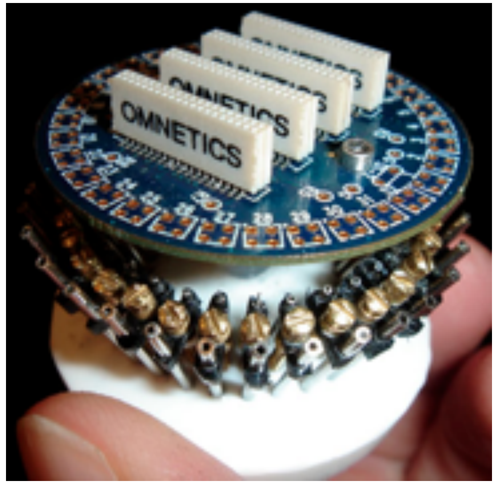




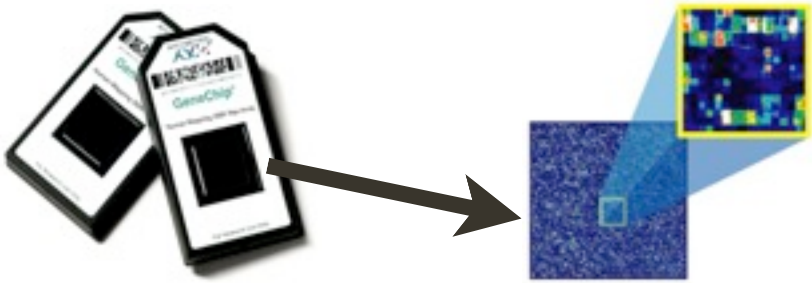
high-throughput data in biology

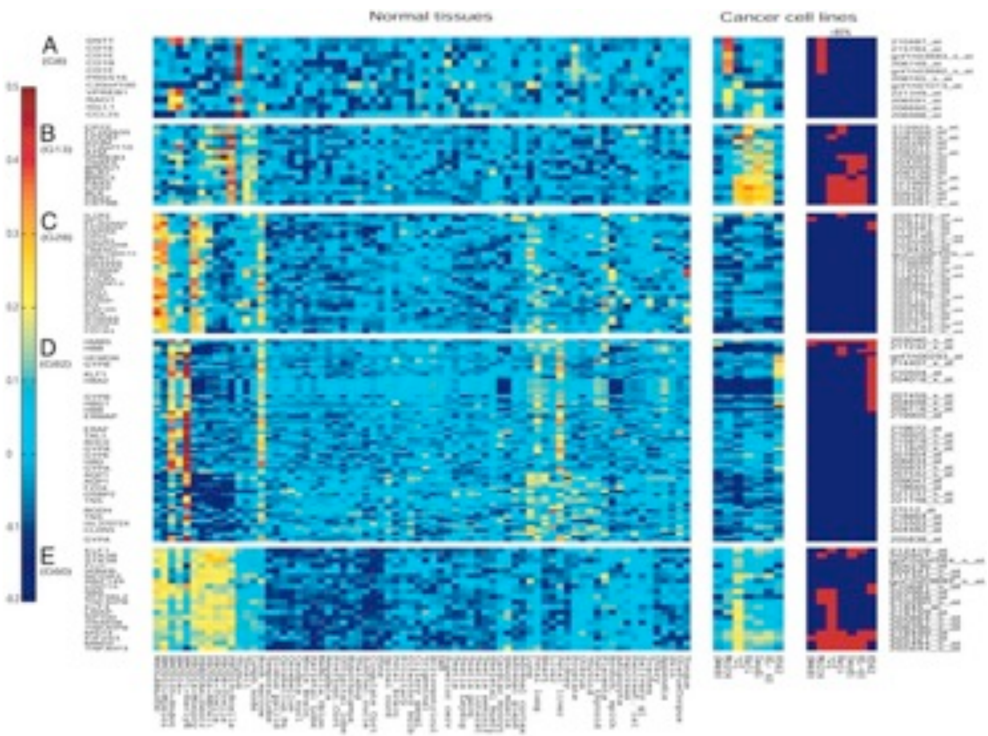


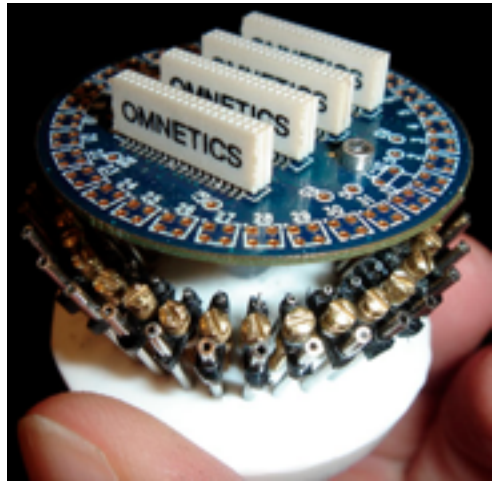


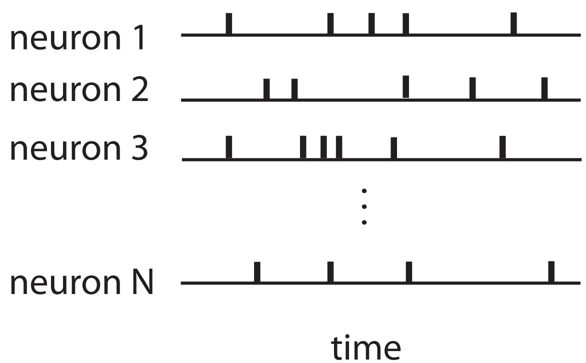


high-throughput data in biology









Goal

Goal

- make lower dimensional statistical descriptions of these data

Goal

- make lower dimensional statistical descriptions of these data
  - network reconstruction

Goal

- make lower dimensional statistical descriptions of these data
    - network reconstruction
    - generate synthetic data

Goal

- make lower dimensional statistical descriptions of these data
    - network reconstruction
    - generate synthetic data

Outline

Goal

- make lower dimensional statistical descriptions of these data
    - network reconstruction
    - generate synthetic data

Outline

- a simple model for this: equilibrium inverse Ising problem.

Goal

- make lower dimensional statistical descriptions of these data
    - network reconstruction
    - generate synthetic data

Outline

- a simple model for this: equilibrium inverse Ising problem.
- adding dynamics: kinetic Ising model.

Goal

- make lower dimensional statistical descriptions of these data
    - network reconstruction
    - generate synthetic data

Outline

- a simple model for this: equilibrium inverse Ising problem.
- adding dynamics: kinetic Ising model.
- problem of hidden nodes

equilibrium inverse Ising problem

$$\Pr(s_1, \ldots, s_N) = \frac{1}{Z} \exp\left[\sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j\right]$$

equilibrium inverse Ising problem

$$m_i = \langle s_i \rangle$$

$$C_{ij} = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle$$

$$\Pr(s_1, \ldots, s_N) = \frac{1}{Z} \exp \left[\sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j \right]$$

equilibrium inverse Ising problem

$$m_i = \langle s_i \rangle$$

$$C_{ij} = \langle s_i s_j \rangle - \langle s_i \rangle \langle s_j \rangle$$

find h_i and J_{ij} of

$$\Pr(s_1, \ldots, s_N) = \frac{1}{Z} \exp\left[\sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j\right]$$

Maximum-Likelihood approach

- the probability that the data is generated by the model at a given set of parameters (the likelihood)

- maximize the likelihood over the parameters.

- typically done iteratively

suppose we are given a set of L spin configuration

$$\boldsymbol{s^1} = (s_1^1, \ldots, s_N^1)$$
$$\boldsymbol{s^2} = (s_1^2, \ldots, s_N^2)$$

$$\boldsymbol{s^L} = (s_L^2, \ldots, s_N^L)$$

$$\Pr[\boldsymbol{s^1}, \cdots, \boldsymbol{s^L}] = \prod_l \frac{\exp\left[\sum_i h_i s_i^l + \sum_{i<j} J_{ij} s_i^l s_j^l\right]}{Z(h, J)}$$

$$\frac{1}{L}\frac{\partial \log \Pr[\boldsymbol{s^1}, \cdots, \boldsymbol{s^L}]}{\partial h_i} = \frac{1}{L}\sum_l s_i^l - \frac{\partial \log Z}{\partial h_i} = \langle s_i \rangle_{\text{data}} - \langle s_i \rangle_{\text{current h and J}}$$

$$\frac{1}{L}\frac{\partial \log \Pr[\boldsymbol{s^1}, \cdots, \boldsymbol{s^L}]}{\partial J_{ij}} = \frac{1}{L}\sum_l s_i^l s_j^l - \frac{\partial \log Z}{J_{ij}} = \langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{current h and J}}$$

- how to find h_i, J_{ij} for large N?

$$\delta h_i = \eta \Big[\langle s_i \rangle_{\text{data}} - \langle s_i \rangle_{\text{current h and J}} \Big]$$

$$\delta J_{ij} = \eta \Big[\langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{current h and J}} \Big]$$

- how to find h_i, J_{ij} for large N?

Exact method: Boltzmann learning

$$\delta h_i = \eta\Big[\langle s_i\rangle_{\rm data} - \langle s_i\rangle_{\rm current\ h\ and\ J}\Big]$$

$$\delta J_{ij} = \eta\Big[\langle s_i s_j\rangle_{\rm data} - \langle s_i s_j\rangle_{\rm current\ h\ and\ J}\Big]$$

- how to find h_i, J_{ij} for large N?

Exact method: Boltzmann learning

$$\delta h_i = \eta\Big[\langle s_i\rangle_{\rm data} - \langle s_i\rangle_{\rm current\ h\ and\ J}\Big]$$

$$\delta J_{ij} = \eta\Big[\langle s_i s_j\rangle_{\rm data} - \langle s_i s_j\rangle_{\rm current\ h\ and\ J}\Big]$$

Ackley, Hinton, Sejnowski 85

- how to find h_i, J_{ij} for large N?

Exact method: Boltzmann learning

$$\delta h_i = \eta \Big[\langle s_i \rangle_{\text{data}} - \langle s_i \rangle_{\text{current h and J}} \Big]$$

$$\delta J_{ij} = \eta \Big[\langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{current h and J}} \Big]$$

Ackley, Hinton, Sejnowski 85

requires long Monte Carlo runs to compute model statistics

- how to find h_i, J_{ij} for large N?

Exact method: Boltzmann learning

$$\delta h_i = \eta\big[\langle s_i\rangle_{\rm data} - \langle s_i\rangle_{\rm current\ h\ and\ J}\big]$$

$$\delta J_{ij} = \eta\big[\langle s_i s_j\rangle_{\rm data} - \langle s_i s_j\rangle_{\rm current\ h\ and\ J}\big]$$

Ackley, Hinton, Sejnowski 85

requires long Monte Carlo runs to compute model statistics

fast and reliable approximate methods exist

approximate learning

the forward problem can be solved using a number of ways

- mean-field approximations
- Bethe approximation
- iterative algorithms like Belief propagation

they give you mean magnetization as a function of J and h.

- combining these with susceptibility-response relation gives ways to solve the inverse problem

- naive mean-field

$$h_i = \tanh^{-1} m_i - \sum_j J_{ij} m_j \qquad C^{-1}_{ij} = \frac{\partial h_i}{\partial m_j} = \frac{\delta_{ij}}{1-m_i^2} - J_{ij}$$

- TAP

$$h_i = \tanh^{-1} m_i - \sum_j J_{ij} m_j + m_i \sum_j J_{ij}^2 (1-m_j^2)$$

$$C^{-1}_{ij} = -J_{ij} - 2J_{ij}^2 m_i m_j$$

Kappen & Rodriguez 98, Tanaka 98

Bethe approximation: Welling and Teh 2000, Ricci-Tersenghi 2012

Belief propagation: Mezard and Mora 2006

- independent-pairs
- independent-pairs+ minimal spanning tree
- high absolute magnetization expansion Roudi et al 09

$$m_i, m_j \to -1 \qquad J_{ij} = \frac{1}{4} \log \left[1 + \frac{C_{ij}}{(1+m_i)(1+m_j)} \right]$$

- Sessak-Monasson

$$J_{ij} = J_{ij}^{\mathrm{nMF}} - J_{ij}^{\mathrm{nMF,Pair}} + J_{ij}^{\mathrm{Pair}}$$

Sessak & Monasson 09

kinetic Ising model

synchronous discrete time

$$\Pr(\{s(t+1)\}|\{s(t)\}) = \prod_i \frac{\exp[s_i(t+1)h_i(t) + \sum_j J_{ij}s_i(t+1)s_j(t)]}{2\cosh[h_i(t) + \sum_j J_{ij}s_j(t)]}$$

asynchronous update

randomly pick a spin at a time

$$\Pr(s_i(t+\delta t)|\{s(t)\}) = \frac{\exp[s_i(t+\delta t)h_i(t) + \sum_j J_{ij}s_i(t+\delta t)s_j(t)]}{2\cosh[h_i(t) + \sum_j J_{ij}s_j(t)]}$$

exact learning for synchronous update

suppose we have observed R repeats each of length L

$$\boldsymbol{s}^r(t) = \{s_1^r(t), \cdots, s_N^r(t)\},\ r = 1 \ldots R.$$

suppose we have observed R repeats each of length L

$$\boldsymbol{s}^r(t) = \{s_1^r(t), \cdots, s_N^r(t)\},\ r = 1 \ldots R.$$

log likelihood of this data is

$$\mathcal{L}(\mathbf{h}, \mathsf{J}) = \sum_{t,r,i} \left[h_i s_i^r(t+1) + \sum_j J_{ij} s_i^r(t+1) s_j^r(t) - \log 2\cosh\left(h_i(t) + \sum_j J_{ij} s_j^r(t)\right)\right].$$

suppose we have observed R repeats each of length L

$$\boldsymbol{s}^r(t) = \{s_1^r(t), \cdots, s_N^r(t)\},\ r = 1 \ldots R.$$

log likelihood of this data is

$$\mathcal{L}(\mathbf{h}, \mathsf{J}) = \sum_{t,r,i} \left[\ h_i s_i^r(t+1) + \sum_j J_{ij} s_i^r(t+1) s_j^r(t) - \log 2\cosh(h_i(t) + \sum_j J_{ij} s_j^r(t))\right].$$

exact learning by maximizing the likelihood by gradient decent

$$\delta h_i = \eta_h \frac{\partial \mathcal{L}}{\partial h_i} \qquad \delta J_{ij} = \eta_J \frac{\partial \mathcal{L}}{\partial J_{ij}}$$

$$\delta h_i(t) = \eta_h \Big\{ \langle s_i(t+1) \rangle_r - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t))] \rangle_r] \Big\}$$

$$\delta J_{ij} = \eta_J \Big\{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \Big\}$$

$$\delta h_i(t) = \eta_h \{ \langle s_i(t+1) \rangle_r - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t))] \rangle_r] \}$$

$$\delta J_{ij} = \eta_J \{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \}$$

like (batch version) delta-rule for N independent perceptrons

Much faster than Boltzmann learning for the symmetric case because it doesn't need long Monte Carlo runs to evaluate the second term

$$\delta h_i(t) = \eta_h \{ \langle s_i(t+1) \rangle_r - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t))] \rangle_r] \}$$

$$\delta J_{ij} = \eta_J \{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \}$$

like (batch version) delta-rule for N independent perceptrons

Much faster than Boltzmann learning for the symmetric case because it doesn't need long Monte Carlo runs to evaluate the second term

Exact algorithm: mean square error ~ 1/L

Weak- coupling limit: $\left\langle \left(J_{ij}^{calculated} - J_{ij}^{true} \right)^2 \right\rangle = \frac{1}{\left(1 - m_i^2\right) L}$

forward mean-field theory

equilibrium case

Helmholtz free energy

$$-\beta F = \log \mathrm{Tr} \exp \left[\sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j \right]$$

equilibrium case

Helmholtz free energy

$$-\beta F = \log \mathrm{Tr} \exp \left[\sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j \right]$$

Legendre transform -> Gibbs free energy

$$-\beta \Gamma(\alpha) = \log \mathrm{Tr} \exp \left[\sum_i h_i(\alpha)(s_i - m_i) + \alpha \sum_{ij} J_{ij} s_i s_j \right] \qquad m_i = \langle s_i \rangle$$

equilibrium case

Helmholtz free energy

$$-\beta F = \log \mathrm{Tr} \exp \left[\sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j \right]$$

Legendre transform -> Gibbs free energy

$$-\beta \Gamma(\alpha) = \log \mathrm{Tr} \exp \left[\sum_i h_i(\alpha)(s_i - m_i) + \alpha \sum_{ij} J_{ij} s_i s_j \right] \qquad m_i = \langle s_i \rangle$$

$$= -\beta \Gamma(0) + \frac{\partial \Gamma}{\partial \alpha}\Big|_{\alpha=0} \alpha + \frac{1}{2} \frac{\partial^2 \Gamma}{\partial \alpha^2}\Big|_{\alpha=0} \alpha^2$$

equilibrium case

Helmholtz free energy

$$-\beta F = \log \mathrm{Tr} \exp \left[\sum_i h_i s_i + \sum_{ij} J_{ij} s_i s_j \right]$$

Legendre transform -> Gibbs free energy

$$-\beta\Gamma(\alpha) = \log \mathrm{Tr} \exp \left[\sum_i h_i(\alpha)(s_i - m_i) + \alpha \sum_{ij} J_{ij} s_i s_j \right] \qquad m_i = \langle s_i \rangle$$

$$= -\beta\Gamma(0) + \frac{\partial \Gamma}{\partial \alpha}\Big|_{\alpha=0} \alpha + \frac{1}{2} \frac{\partial^2 \Gamma}{\partial \alpha^2}\Big|_{\alpha=0} \alpha^2$$

$$\beta \frac{\partial \Gamma(1)}{\partial m_i} = h_i(1) \qquad \text{TAP}$$

Deriving dynamical naive MF and TAP equations

Deriving dynamical naive MF and TAP equations

$$Z[\psi, h] = \left\langle \exp\Big[\sum_{i,t} \psi_i(t) s_i(t)\Big] \right\rangle$$

Deriving dynamical naive MF and TAP equations

$$Z[\psi, h] = \left\langle \exp\Big[\sum_{i,t} \psi_i(t) s_i(t)\Big] \right\rangle$$

$$\langle s_i(t) \rangle = \lim_{\psi \to 0} \frac{\partial \log Z}{\partial \psi_i(t)}$$

Deriving dynamical naive MF and TAP equations

auxiliary field

$$Z[\psi, h] = \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle$$

$$\langle s_i(t) \rangle = \lim_{\psi \to 0} \frac{\partial \log Z}{\partial \psi_i(t)}$$

average over stochastic path

Deriving dynamical naive MF and TAP equations

auxiliary field

$$Z[\psi, h] = \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle$$

$$\langle s_i(t) \rangle = \lim_{\psi \to 0} \frac{\partial \log Z}{\partial \psi_i(t)}$$

average over stochastic path

$$Z[\psi, h] = \int D\boldsymbol{\theta} \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle \prod_{i,t} \delta\Big(\theta_i(t) - h_i(t) - \sum_j J_{ij} s_j(t)\Big)$$

$$= \int D\boldsymbol{\theta}\hat{\boldsymbol{\theta}} \left\langle \exp\left[i \sum_{i,t} \hat{\theta}_i(t) \{\theta_i(t) - h_i(t) - \sum_j J_{ij} s_j(t)\} + \sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle$$

Deriving dynamical naive MF and TAP equations

auxiliary field

$$Z[\psi, h] = \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle \qquad \langle s_i(t)\rangle = \lim_{\psi \to 0} \frac{\partial \log Z}{\partial \psi_i(t)}$$

average over stochastic path

$$Z[\psi, h] = \int D\boldsymbol{\theta} \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle \prod_{i,t} \delta\Big(\theta_i(t) - h_i(t) - \sum_j J_{ij} s_j(t)\Big)$$

$$= \int D\boldsymbol{\theta}\hat{\boldsymbol{\theta}} \left\langle \exp\left[i \sum_{i,t} \hat{\theta}_i(t)\{\theta_i(t) - h_i(t) - \sum_j J_{ij} s_j(t)\} + \sum_{i,t} \psi_i(t) s_i(t)\right] \right\rangle$$

$$\Gamma[\hat{m}, m] \equiv \log Z[\psi[\hat{m}, m], h[\hat{m}, m]] - \sum_{i,t} \psi_i[\hat{m}, m](t) m_i(t) + i \sum_{i,t} h_i[\hat{m}, m](t) \hat{m}_i(t),$$

Deriving dynamical naive MF and TAP equations

auxiliary field

$$Z[\psi, h] = \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right]\right\rangle \qquad \langle s_i(t)\rangle = \lim_{\psi\to 0} \frac{\partial \log Z}{\partial \psi_i(t)}$$

average over stochastic path

$$Z[\psi, h] = \int D\boldsymbol{\theta} \left\langle \exp\left[\sum_{i,t} \psi_i(t) s_i(t)\right]\right\rangle \prod_{i,t} \delta\Big(\theta_i(t) - h_i(t) - \sum_j J_{ij} s_j(t)\Big)$$

$$= \int D\boldsymbol{\theta}\hat{\boldsymbol{\theta}} \left\langle \exp\left[i \sum_{i,t} \hat{\theta}_i(t)\{\theta_i(t) - h_i(t) - \sum_j J_{ij} s_j(t)\} + \sum_{i,t} \psi_i(t) s_i(t)\right]\right\rangle$$

$$\Gamma[\hat{m}, m] \equiv \log Z[\psi[\hat{m}, m], h[\hat{m}, m]] - \sum_{i,t} \psi_i[\hat{m}, m](t) m_i(t) + i \sum_{i,t} h_i[\hat{m}, m](t) \hat{m}_i(t),$$

rescale $J_{ij} \longrightarrow \alpha J_{ij}$**, expand** Γ **around** $\alpha = 0$ **and then set** $\alpha = 1$

$$\frac{\partial \Gamma}{\partial \hat{m}_i(t)} = ih_i[\hat{m}, m](t)$$

synchronous update

first order in α

$$m_i(t+1) = \tanh\left[h_i(t) + \sum_j J_{ij} m_j(t)\right]$$

second order α

$$m_i(t+1) = \tanh\left[h_i(t) + \sum_j J_{ij} m_j(t) - m_i(t+1) \sum_j J_{ij}^2 (1 - m_j^2(t))\right]$$

asynchronous update

$$m_i(t) + \frac{dm_i(t)}{dt} = \tanh\left[h_i(t) + \sum_j J_{ij} m_j(t)\right]$$

$$m_i(t) + \frac{dm_i(t)}{dt} = \tanh\left[h_i(t) + \sum_j J_{ij} m_j(t) - \left(m_i(t) + \frac{dm_i(t)}{dt}\right) \sum_j J_{ij}^2 (1 - m_j^2(t))\right]$$

Roudi and Hertz 2011, JStat

forward mean-field theory

MF

$$m_i(t) = \tanh\left[h_i(t-1) + \sum_j J_{ij} m_j(t-1)\right]$$

TAP

$$m_i(t) = \tanh\left[h_i(t-1) + \sum_j J_{ij} m_j(t-1) - m_i(t) \sum_j J_{ij}^2 (1 - m_j(t-1)^2)\right]$$

exact MF for fully asymmetric couplings

$$m_i(t) = \int \frac{dx}{\sqrt{2\pi}} e^{-x^2/2} \tanh\left[h_i(t-1) + \sum_j J_{ij} m_j(t-1) + x\sqrt{\Delta_i(t-1)}\right]$$

$$\Delta_i(t) = \sum_j J_{ij}^2 (1 - m_j^2(t))$$

$$\delta J_{ij} = \eta_J \left\{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \right\}$$

Roudi and Hertz 2011, PRL

$$\delta J_{ij} = \eta_J \left\{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \right\}$$

after the learning is converged $\delta J_{ij} = 0$

$$\langle s_i(t+1) s_j(t) \rangle = \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle$$

Roudi and Hertz 2011, PRL

$$\delta J_{ij} = \eta_J \left\{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \right\}$$

after the learning is converged $\delta J_{ij} = 0$

$$\langle s_i(t+1) s_j(t) \rangle = \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle$$

$$S_i = m_i + \delta S_i$$
$$m_i = \langle s_i \rangle$$

Roudi and Hertz 2011, PRL

$$\delta J_{ij} = \eta_J \left\{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \right\}$$

after the learning is converged $\delta J_{ij} = 0$

$$\langle s_i(t+1) s_j(t) \rangle = \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle$$

$$S_i = m_i + \delta S_i$$

$$m_i = \langle s_i \rangle$$

expanding 1st order in δs and assuming $m_i = \tanh(h_i + \sum_j J_{ik}^{\mathrm{MF}} m_k)$

Roudi and Hertz 2011, PRL

$$\delta J_{ij} = \eta_J \Big\{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \Big\}$$

after the learning is converged $\delta J_{ij} = 0$

$$\langle s_i(t+1) s_j(t) \rangle = \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle$$

$$S_i = m_i + \delta S_i$$

$$m_i = \langle s_i \rangle$$

expanding 1st order in δs and assuming $m_i = \tanh(h_i + \sum_j J_{ik}^{\mathrm{MF}} m_k)$

$$\langle \delta s_i(t+1) \delta s_j(t) \rangle = (1 - m_i^2) \sum_k J_{ik}^{\mathrm{MF}} \langle \delta s_k(t) \delta s_j(t) \rangle.$$

Roudi and Hertz 2011, PRL

$$\delta J_{ij} = \eta_J \{ \langle s_i(t+1) s_j(t) \rangle - \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle \}$$

after the learning is converged $\delta J_{ij} = 0$

$$\langle s_i(t+1) s_j(t) \rangle = \langle \tanh[h_i(t) + \sum_k J_{ik} s_k(t)] s_j(t) \rangle$$

$S_i = m_i + \delta S_i$

$m_i = \langle s_i \rangle$

expanding 1st order in δs and assuming $m_i = \tanh(h_i + \sum_j J_{ik}^{\mathrm{MF}} m_k)$

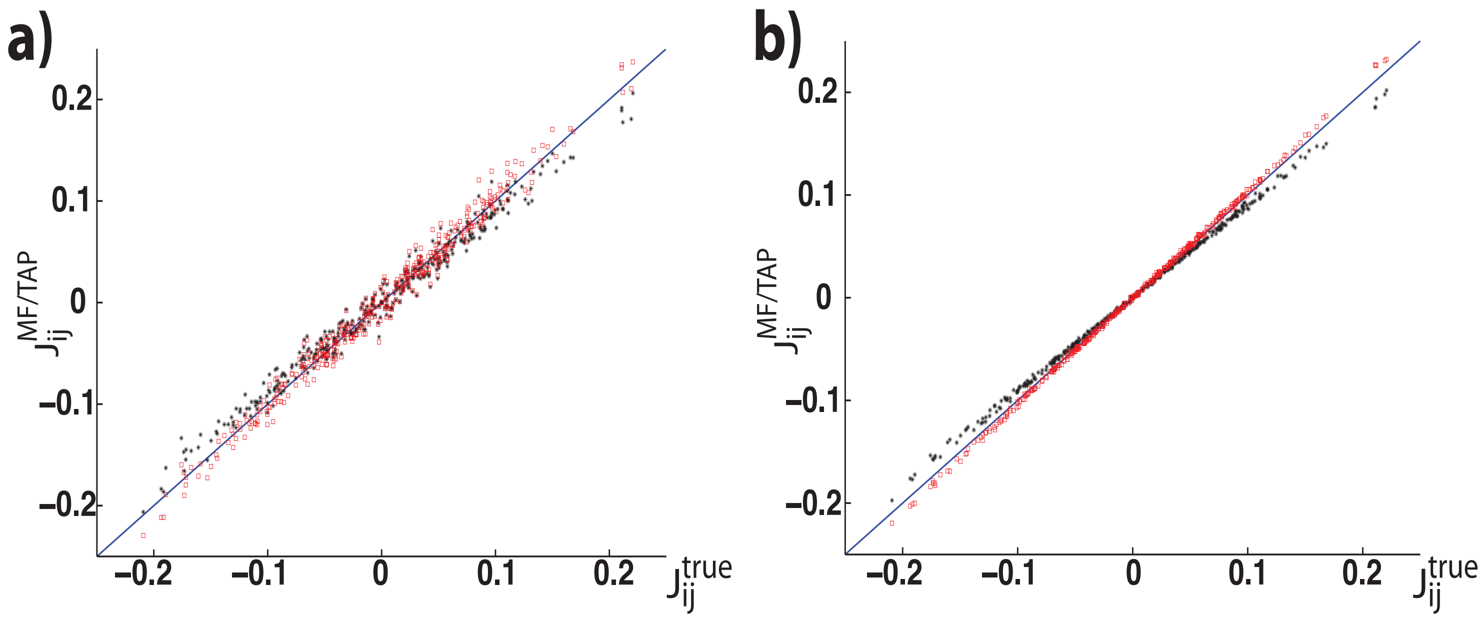
$$\langle \delta s_i(t+1) \delta s_j(t) \rangle = (1 - m_i^2) \sum_k J_{ik}^{\mathrm{MF}} \langle \delta s_k(t) \delta s_j(t) \rangle.$$

$$\mathsf{J}^{\mathrm{MF}} = \mathsf{A}^{-1} \mathsf{D} \mathsf{C}^{-1}$$

$$C_{ij} = \langle \delta s_i(t) \delta s_j(t) \rangle \qquad D_{ij} = \langle \delta s_i(t+1) \delta s_j(t) \rangle$$

Roudi and Hertz 2011, PRL

MF and TAP tested on data generated from a kinetic Ising model:



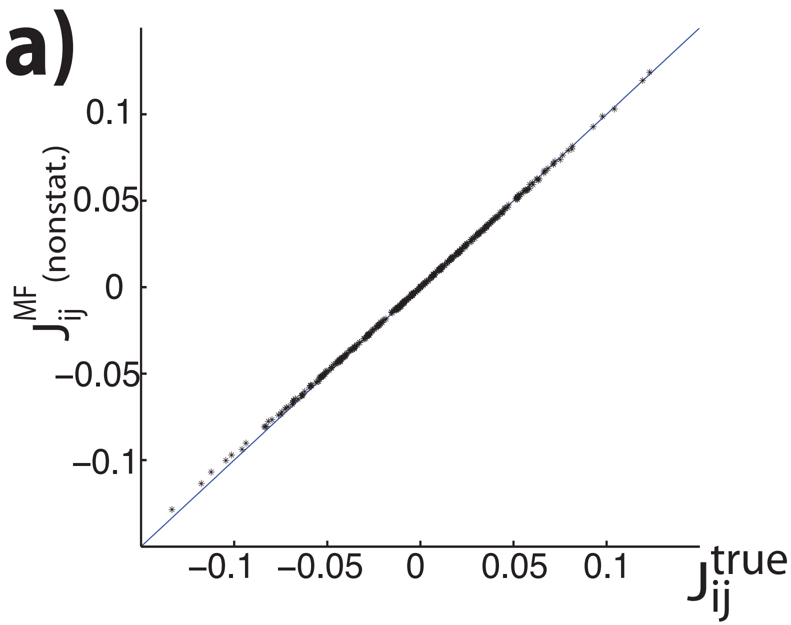
L = 10^4 L = 10^6

L time steps, generated by a model with random couplings:

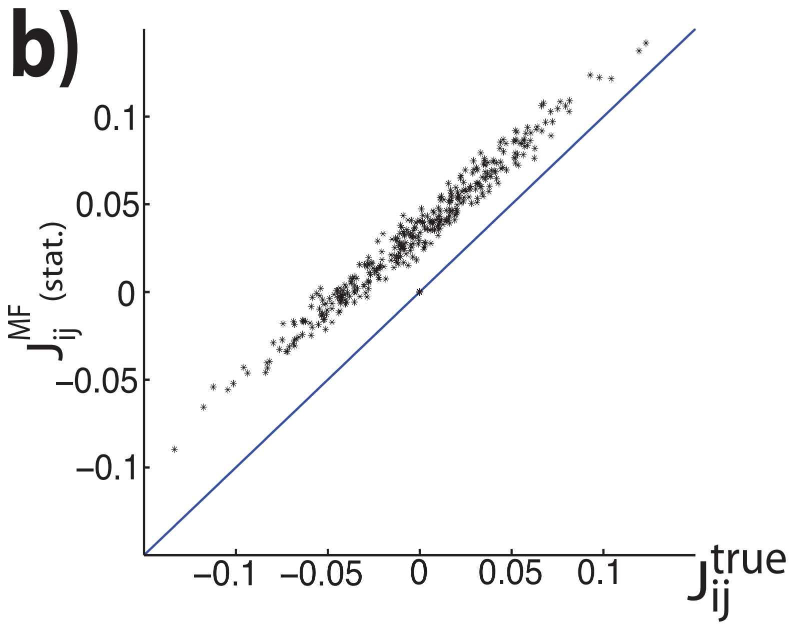
$$\langle J_{ij} \rangle = 0 \qquad \langle J_{ij}^2 \rangle = \frac{g^2}{N}$$

(asymmetric Sherrington-Kirkpatrick model)

sinusoidal field applied to all spins



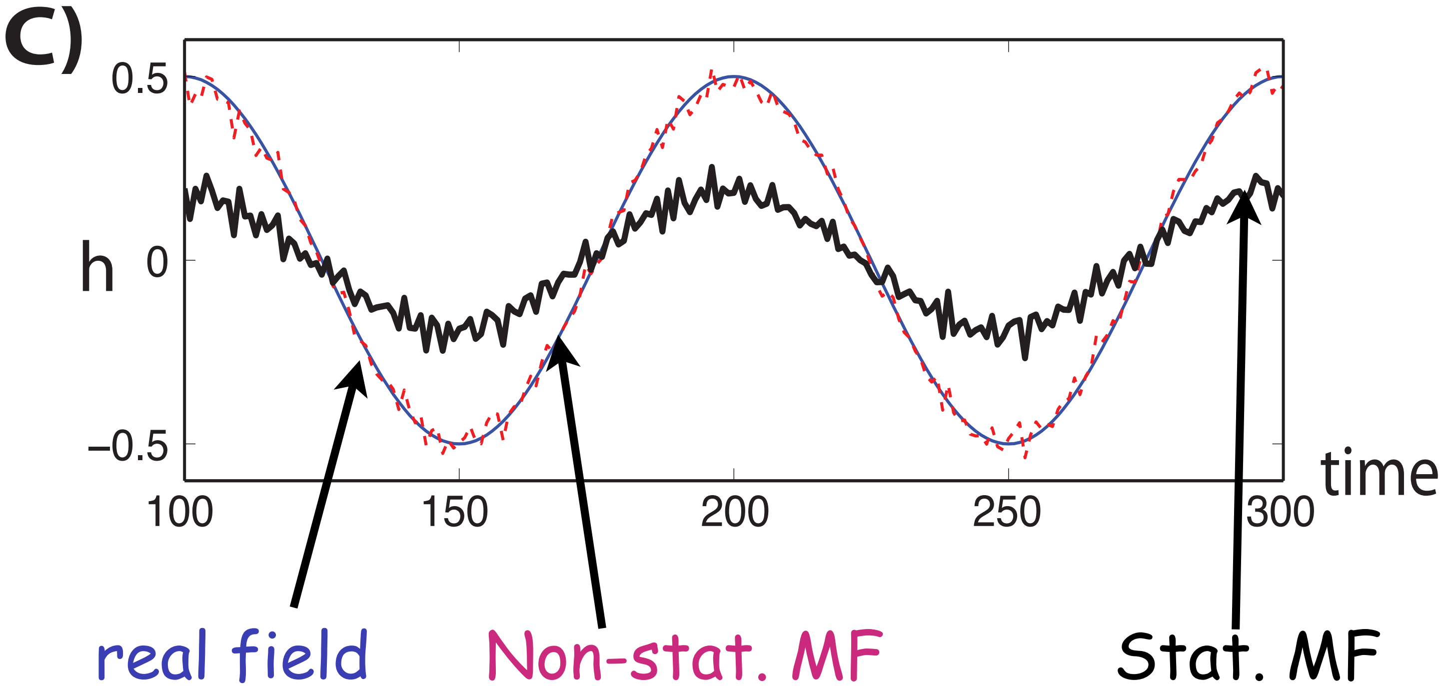
nonstationary MF inference
applied to nonstationary data



stationary MF inference
applied to nonstationary data

after we inferred the couplings, we can infer the fields

$$m_i(t+1) = \tanh[h_i(t) + \sum_j J_{ij}^{\mathrm{MF}} m_j(t)].$$



asynchronous update

randomly pick a spin at a time

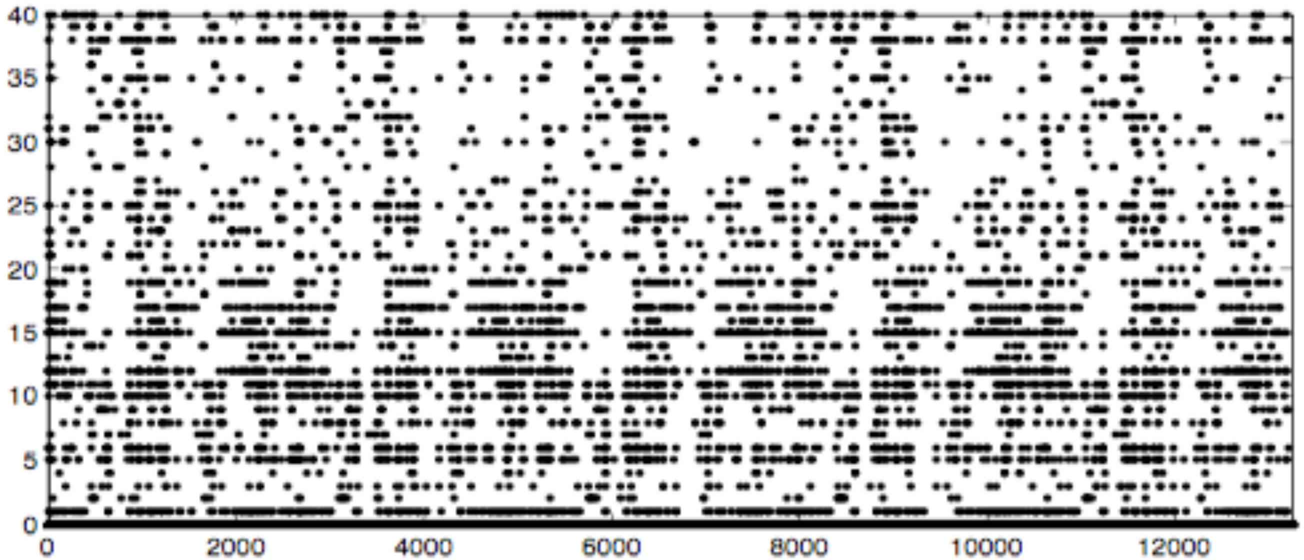
$$\Pr(s_i(t+\delta t)|\{s(t)\}) = \frac{\exp[s_i(t+\delta t)h_i(t) + \sum_j J_{ij} s_i(t+\delta t) s_j(t)]}{2\cosh[h_i(t) + \sum_j J_{ij} s_j(t)]}$$

exact ML learning is has some interesting depends on whether you know the update times or not.

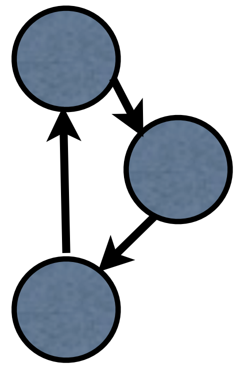
We can also marginalize the update times.
see Zeng, Alava, Aurell, Hertz, Roudi arXiv:1209.2401

MF learning can also done Zeng, Alava, Aurell, Mahmoudi PRE 2011

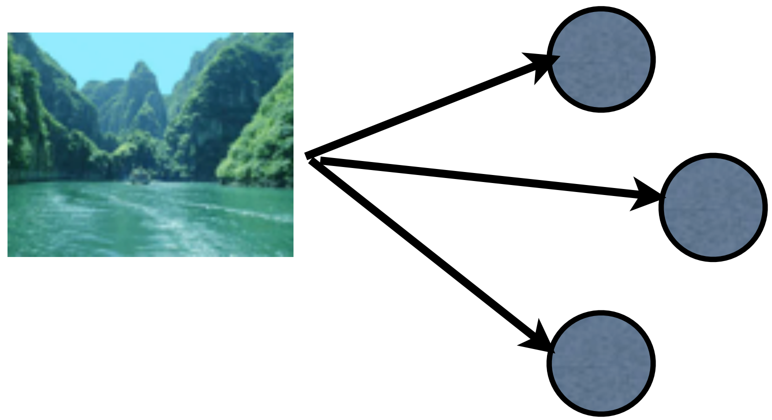
example application



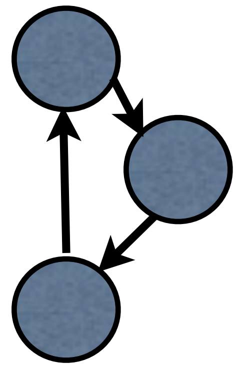
- genuine correlations between neurons may come from
  - internal connections between the recorded neurons



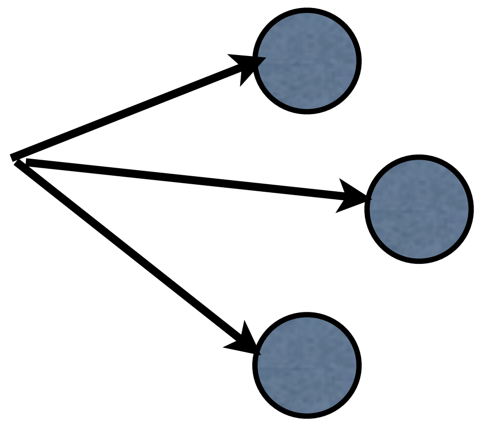
  - external common input



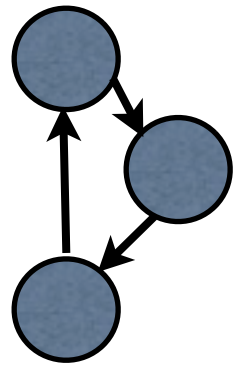
- genuine correlations between neurons may come from
    - internal connections between the recorded neurons



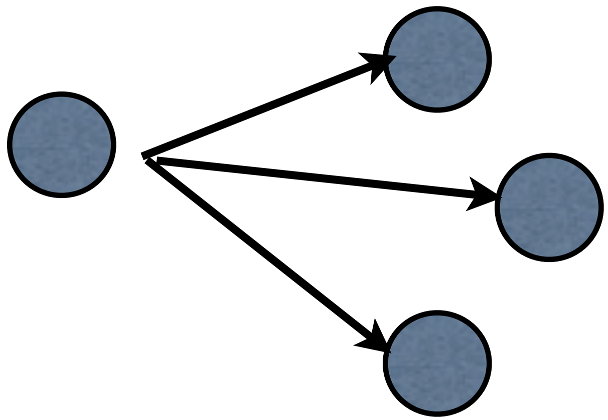
    - external common input



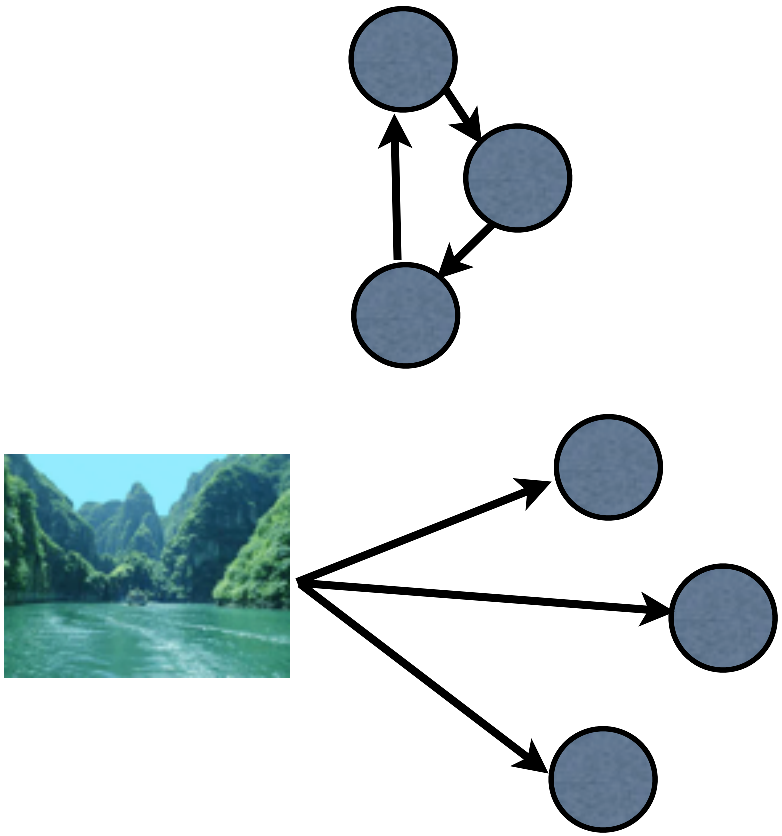
- genuine correlations between neurons may come from
    - internal connections between the recorded neurons



    - external common input

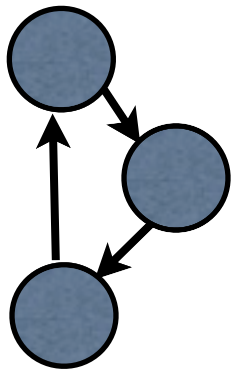


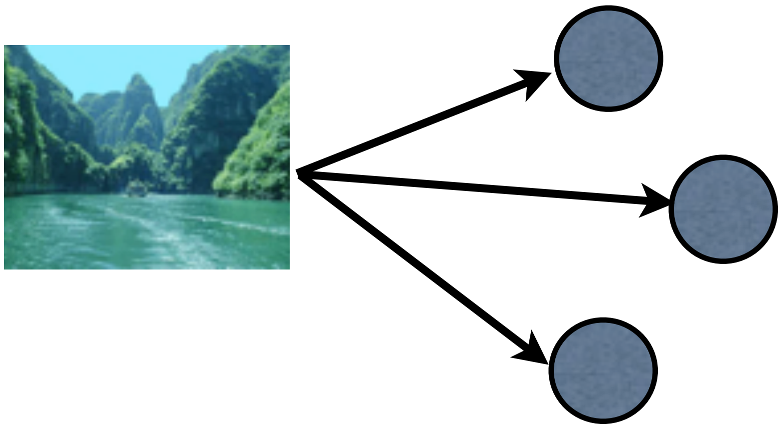
why do we care about these different explanations?



why do we care about these different explanations?

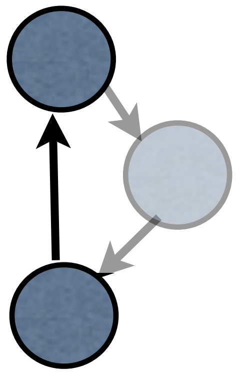
they lead to entirely different predictions about how the network responds to manipulations

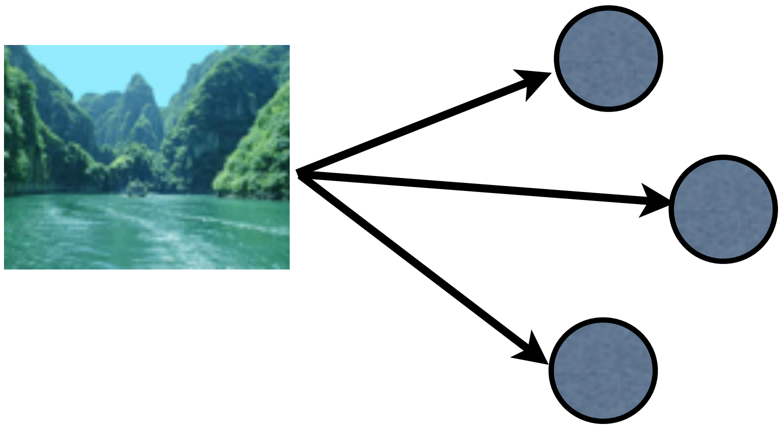




why do we care about these different explanations?

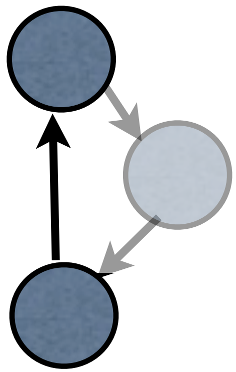
they lead to entirely different predictions about how the network responds to manipulations

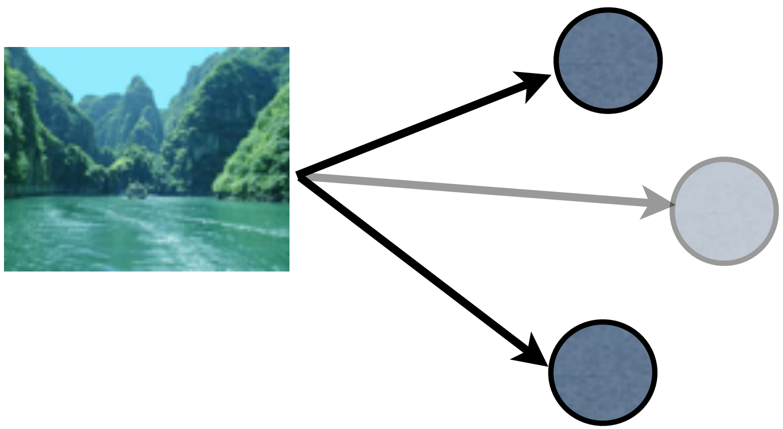




why do we care about these different explanations?

they lead to entirely different predictions about how the network responds to manipulations





salamander retinal data

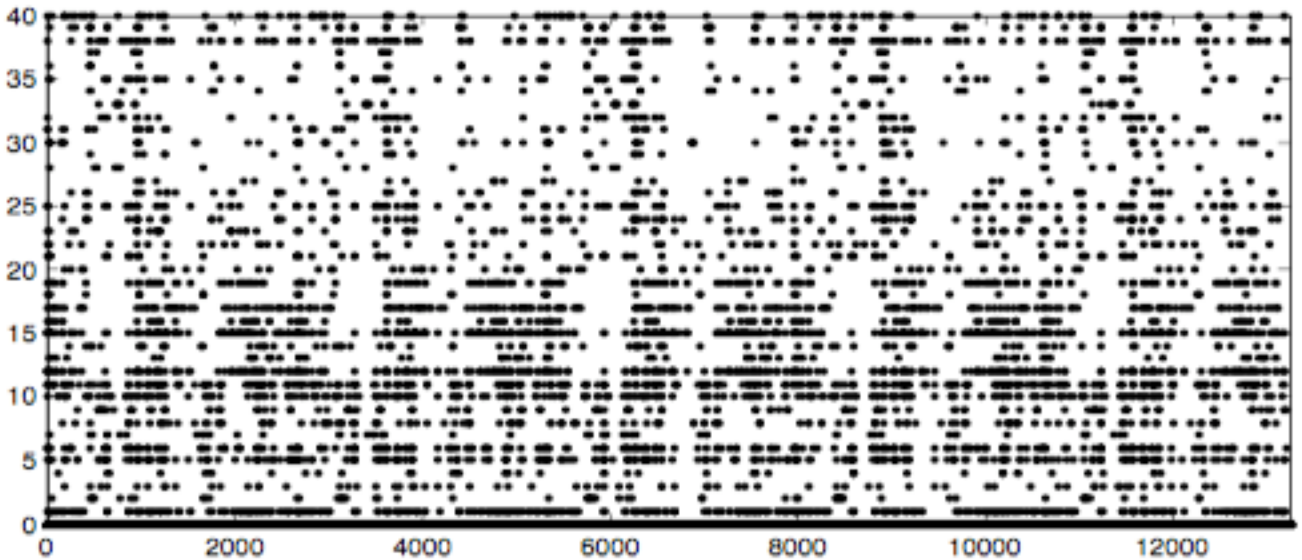
results based on Tyrcha, Roudi, Marsili and Hertz, Jstat 2013 in press

Electrode array in salamander retina (N = 40)
(courtesy of Michael Berry, Princeton Univ)

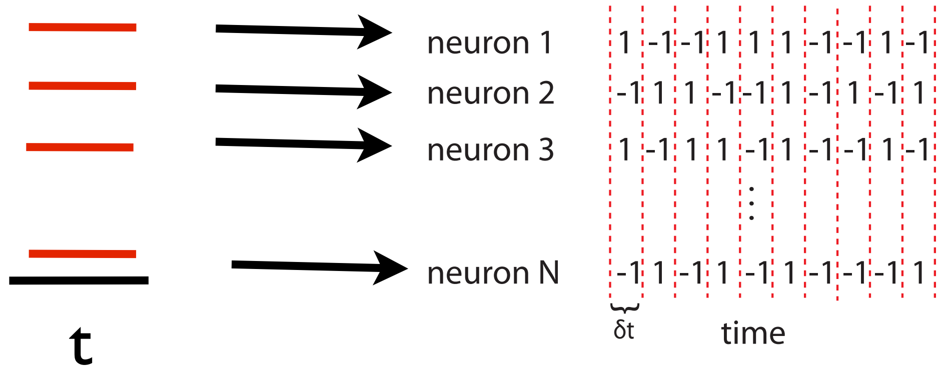
recording time: 3180 sec

Retina was shown a 26.5-s “movie clip” 120 times

(each movie = 2650 10-ms time bins)
(also tried 16⅔ ms, 20 ms time bins)
size of data matrix for 10-ms bins: **40** x **318000**

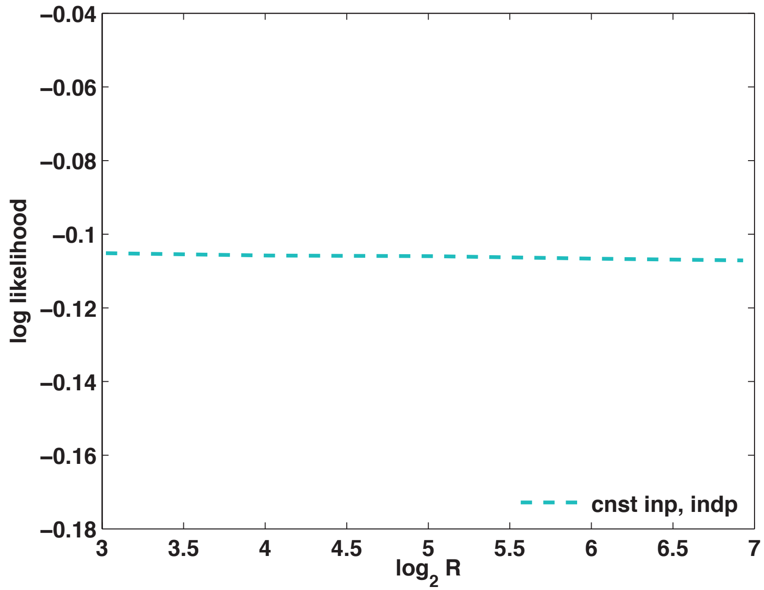


independent neuron model with constant input



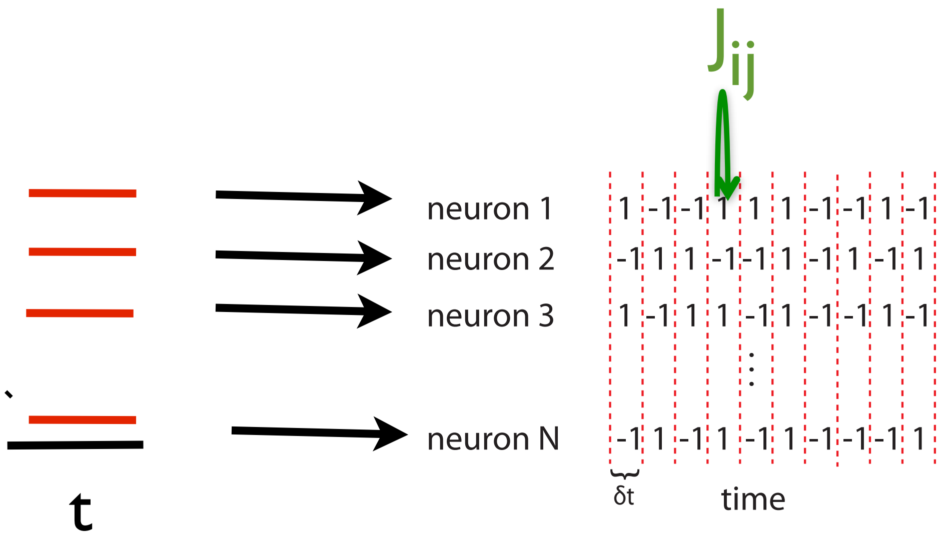
constant external input

probability that neuron *i* spikes at time *t+1*



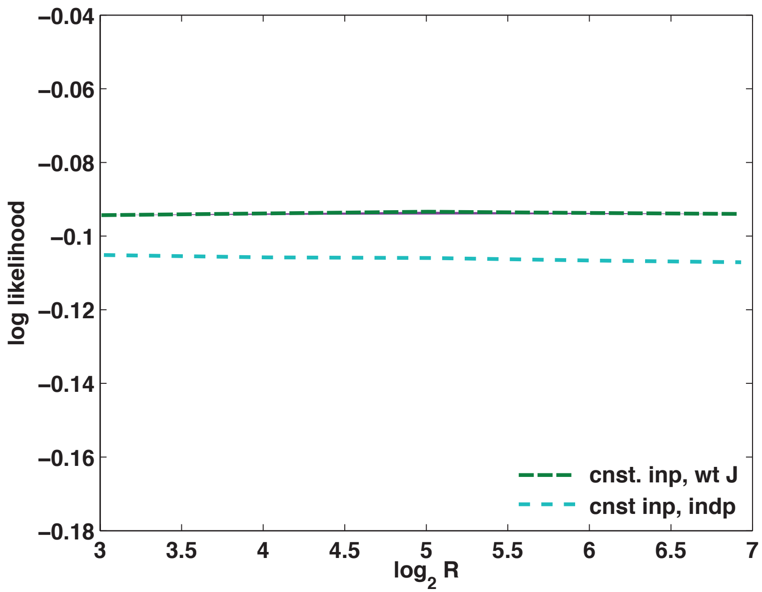
−0.04
−0.06
−0.08
−0.1
−0.12
−0.14
−0.16
−0.18
log likelihood
3
3.5
4
4.5
5
5.5
6
6.5
7
log2 R
cnst inp, indp

equilibrium Ising model



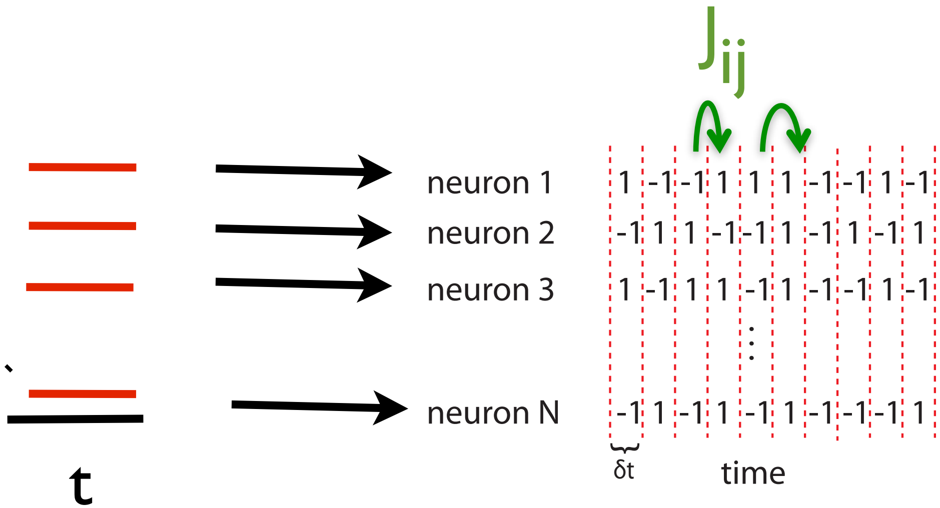
probability that neuron *i* spikes at time *t*

- constant external input
- total input from spiking neurons at time t



−0.04
−0.06
−0.08
−0.1
−0.12
−0.14
−0.16
−0.18
log likelihood
3
3.5
4
4.5
5
5.5
6
6.5
7
log2 R
cnst. inp, wt J
cnst inp, indp

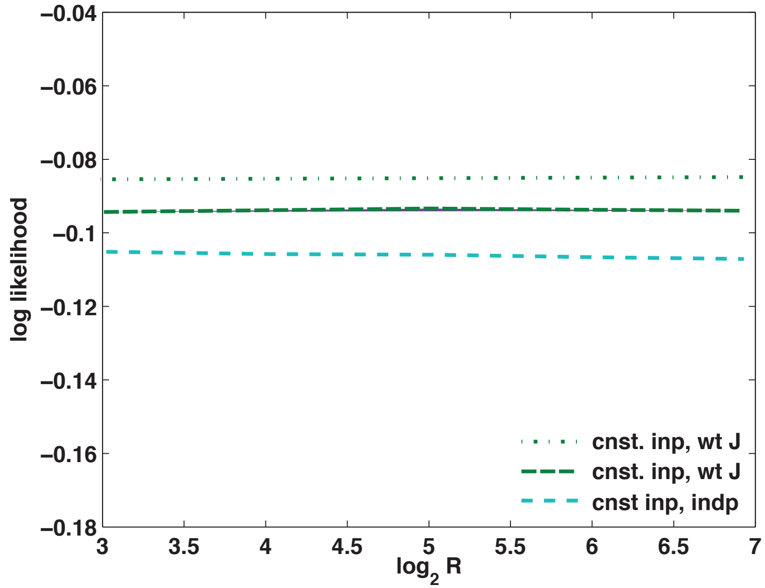
Kinetic Ising model constant input



probability that neuron *i* spikes at time *t+1*

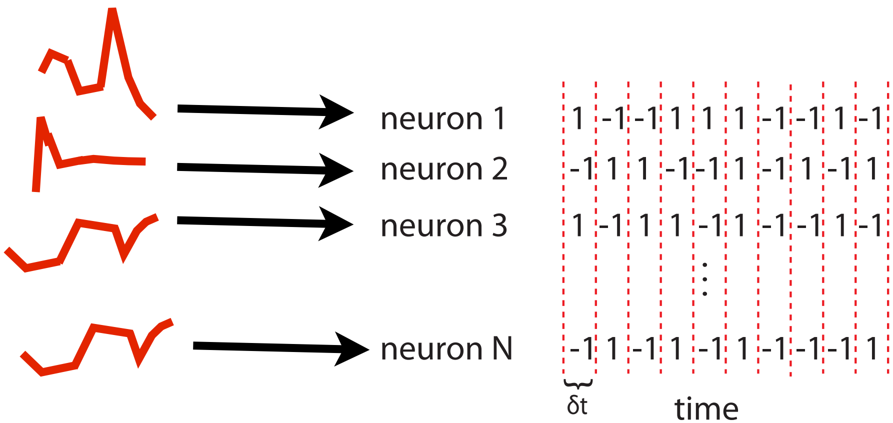
constant external input

total input from spiking neurons at time t



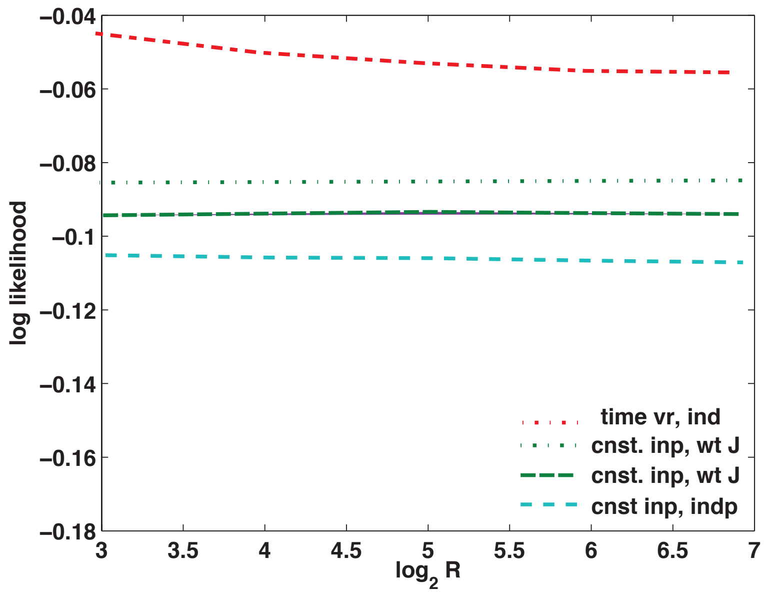
−0.04
−0.06
−0.08
−0.1
−0.12
−0.14
−0.16
−0.18
log likelihood
3
3.5
4
4.5
5
5.5
6
6.5
7
log2 R
cnst. inp, wt J
cnst. inp, wt J
cnst inp, indp

independent neuron model with time varying input

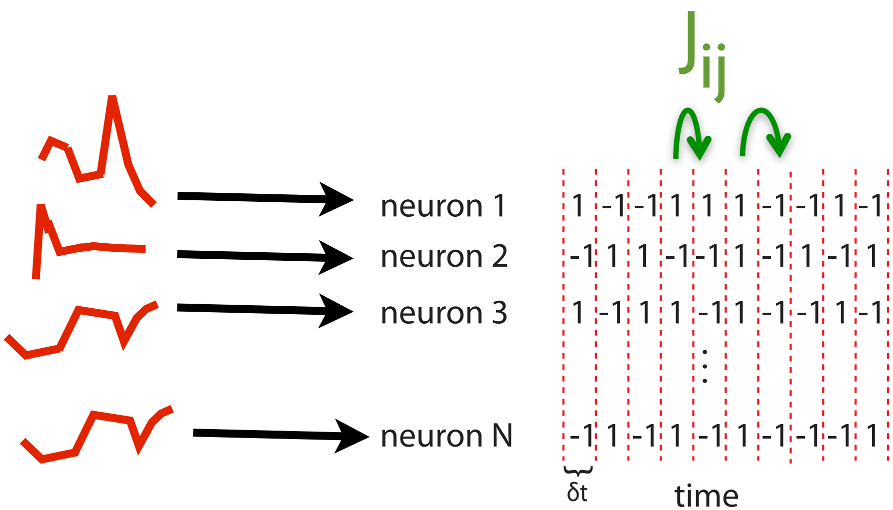


time varying external input

probability that neuron *i* spikes at time *t+1*


−0.04
−0.06
−0.08
−0.1
−0.12
−0.14
−0.16
−0.18
log likelihood
3
3.5
4
4.5
5
5.5
6
6.5
7
log2 R
time vr, ind
cnst. inp, wt J
cnst. inp, wt J
cnst inp, indp

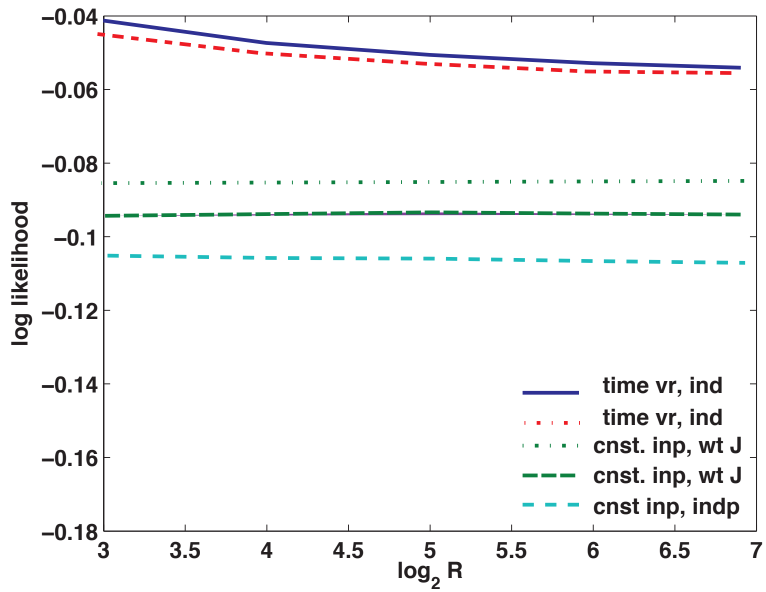
independent neuron model with time varying input



probability that neuron *i* spikes at time *t+1*

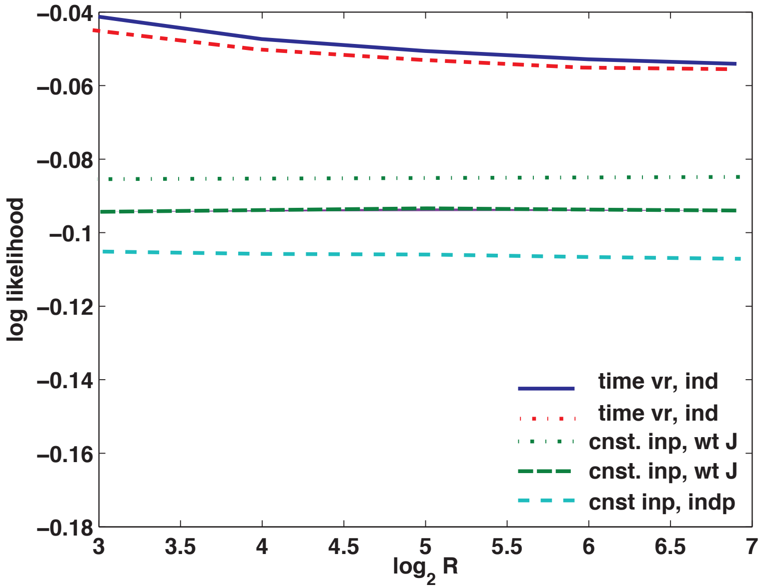
time varying external input

total input from spiking neurons at time t

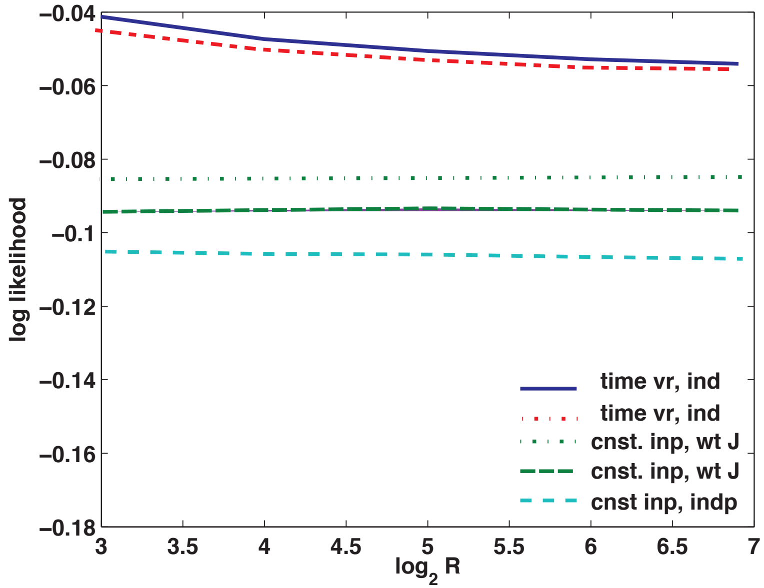


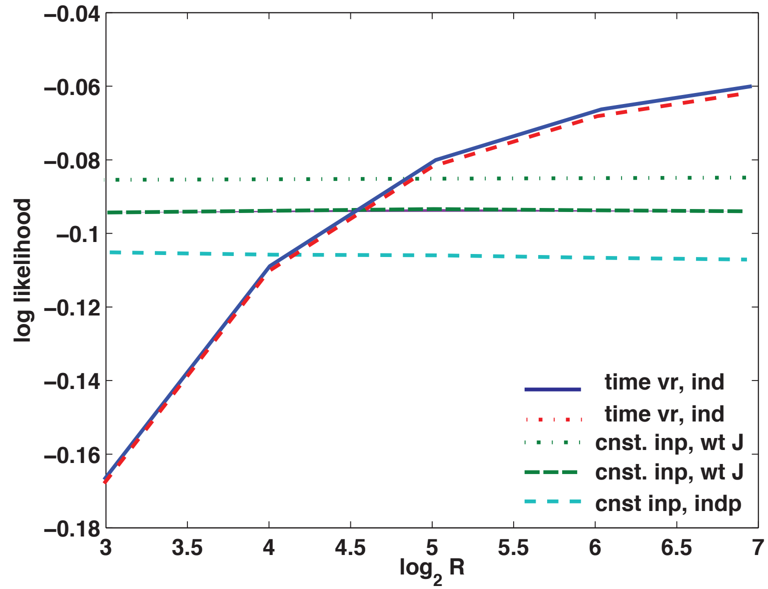
−0.04
−0.06
−0.08
−0.1
−0.12
−0.14
−0.16
−0.18
log likelihood
3
3.5
4
4.5
5
5.5
6
6.5
7
log2 R
time vr, ind
time vr, ind
cnst. inp, wt J
cnst. inp, wt J
cnst inp, indp

what about correcting for number of parameters?

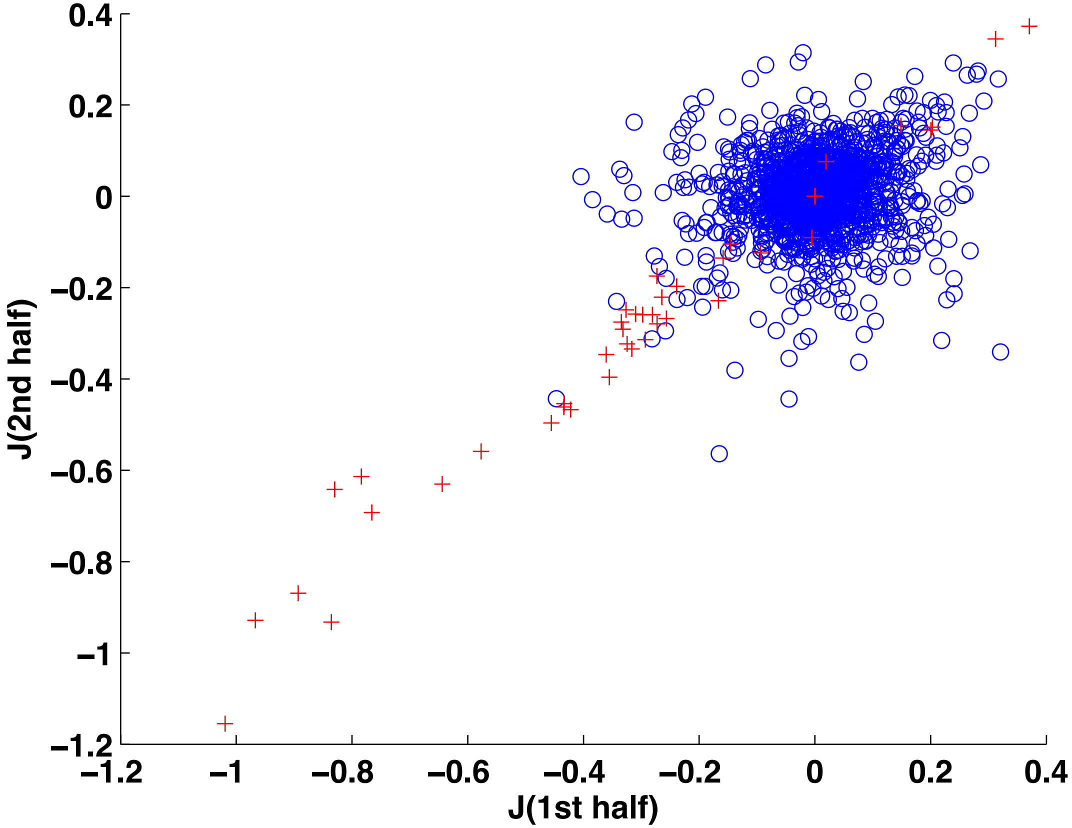


what about correcting for number of parameters?





another way of seeing the insignificant of couplings



- equilibrium Ising model and approximations for solving its inverse problem.
- kinetic Ising model and MF approximations for its inverse problem.
- can help us understand global activity in biological data.

- Bethe approximation for dynamics.
- hidden spins (Dunn and Roudi 2013, Tyrcha and Hertz 2013)
- going beyond one step memory
- continuous variables