Causal Synchrony
in
Distributed Processes

James P. Crutchfield
www.santafe.edu/∼chaos

21 May 2002

Abstract How can we infer the structure of a collective from observations? Can we find subcollectives? Is the collective smarter than its agents? I'll show how a theory of intrinsic computation can be adapted to answer these questions for dynamical networks. The key idea is that of causal synchrony, in which one infers the effective internal dynamics of agents and then estimates the agent-agent coordination in terms of that hidden dynamics. Applications to synchronization in neurobiological processes illustrate the uses and benefits of causal synchrony.

Joint work with Marcelo Camperi (USF) and Cosma and and Kristina Shalizi (SFI).

Innovation, sources

- Evolutionary:
  - Evolving Cellular Automata ['91-'97]: www.santafe.edu/∼evca
  - Epochal Evolution ['95-'98]: www.santafe.edu/∼evca
  - Evolutionary Dynamics Workshop ['98]: www.santafe.edu/∼jpc/e
  - Evolutionary Dynamics Book [2002]: www.santafe.edu/∼jpc

  Pattern Discovery: www.santafe.edu/∼cmg
  - Emergence ['94]
  - Learning and Adaptation

Networks?

When is a “system” a “network”?

Network $\leftrightarrow$ System architecture important

Three Kinds of Network Problem

1. The Forward Problem: Modeling Networks

2. The Reverse Problem: Design Networks to Function

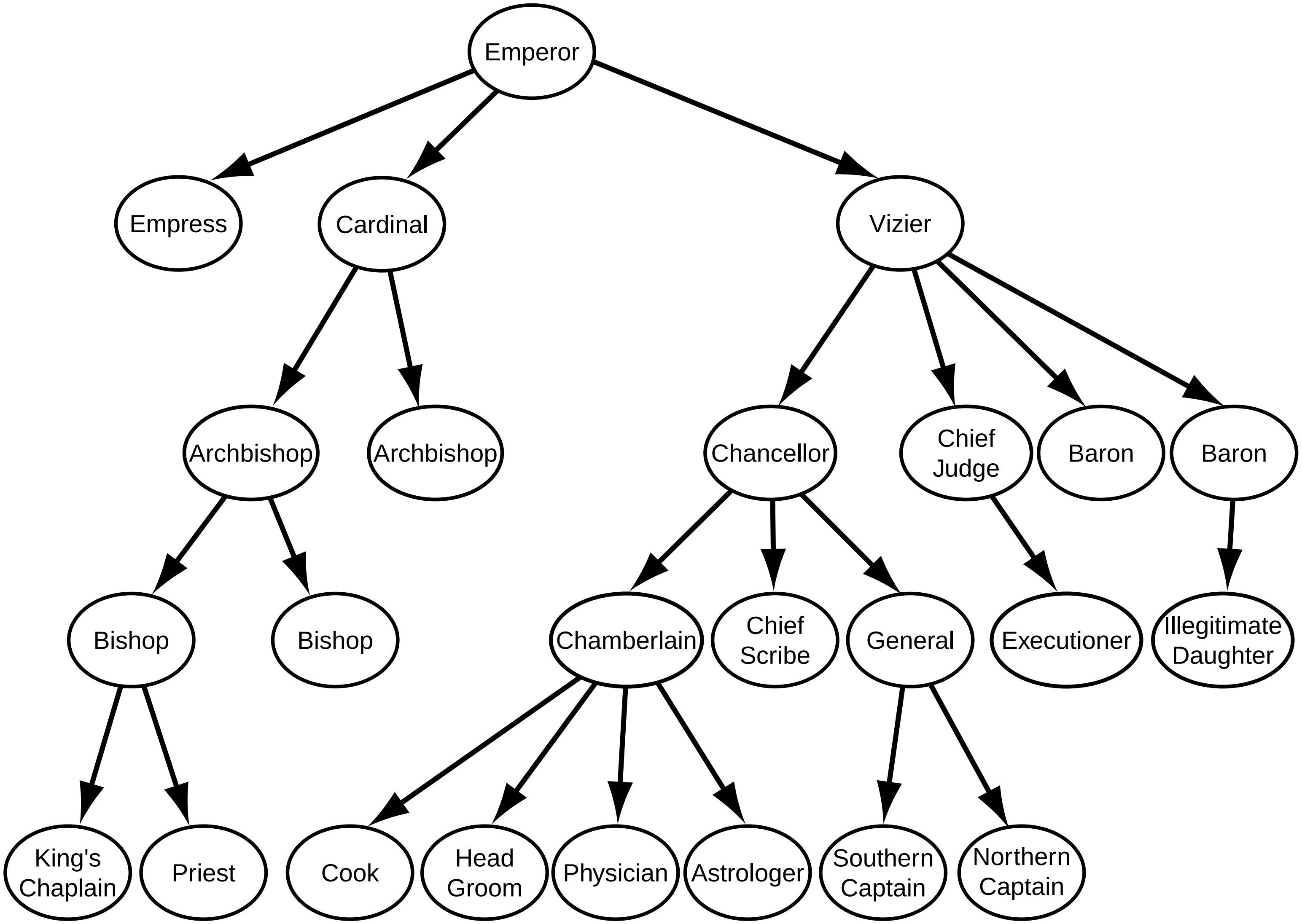
3. The Inverse Problem: Observations $\rightarrow$ Network

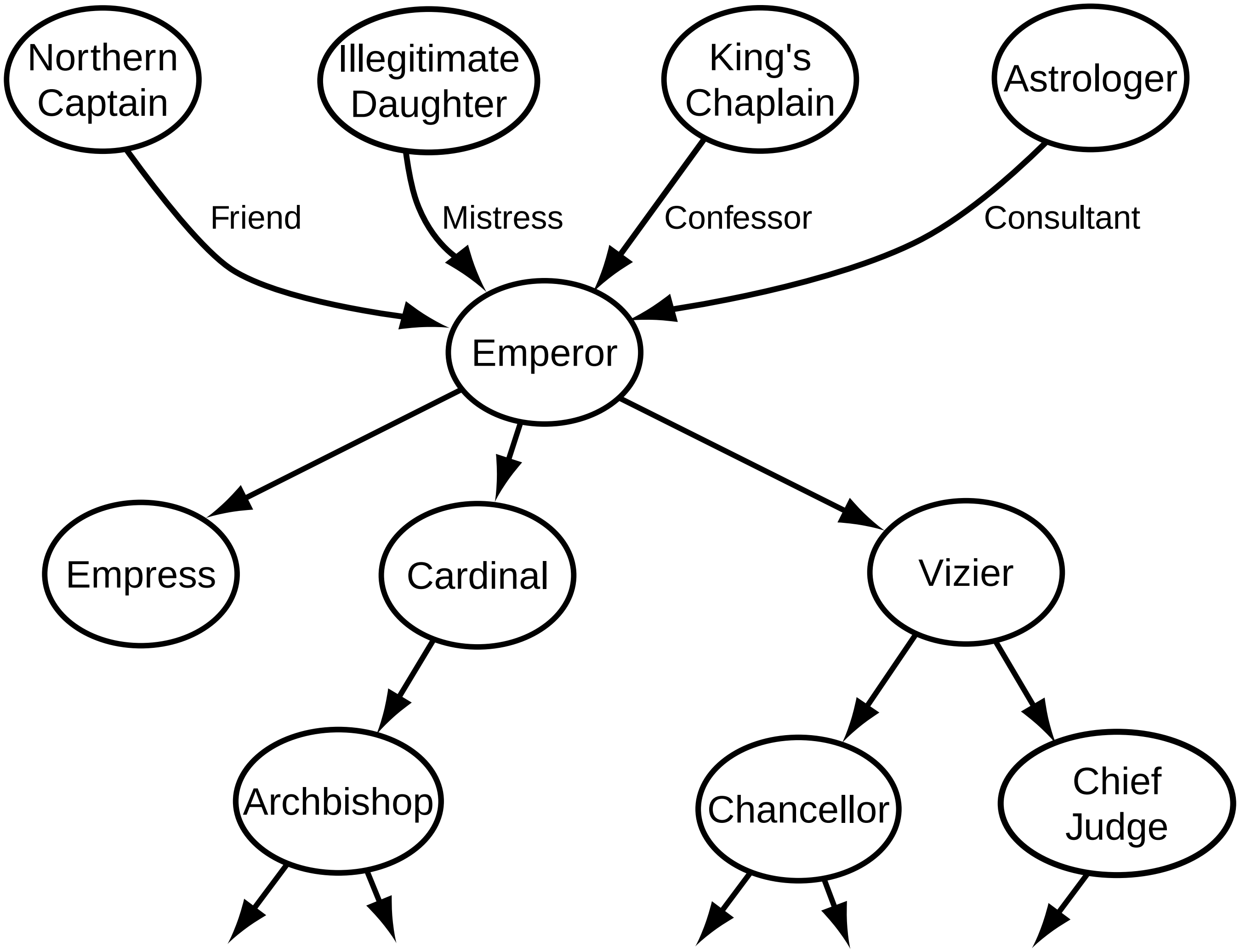
   - What is architecture?

   - What is information?

   - History

   - Design is not Behavior; Goals are not Functions


Emperor
Empress
Cardinal
Vizier
Archbishop
Archbishop
Chancellor
Chief Judge
Baron
Baron
Bishop
Bishop
Chamberlain
Chief Scribe
General
Executioner
Illegitimate Daughter
King's Chaplain
Priest
Cook
Head Groom
Physician
Astrologer
Southern Captain
Northern Captain


Northern Captain
Illegitimate Daughter
King's Chaplain
Astrologer
Friend
Mistress
Confessor
Consultant
Emperor
Empress
Cardinal
Vizier
Archbishop
Chancellor
Chief Judge

Synchrony and Distributed Information

When and how do the nodes in a network synchronize?

$\Rightarrow$ How do you measure if they're synchronized?

Problems with synchrony measures:

- Computationally expensive
- Restrictive assumptions: e.g., Cross-correlation $\Rightarrow$ linear dependence
- Need to know the nodes' inner dynamics: e.g., Fourier $\Rightarrow$ periodic
- Some combination of the above

Way out: Look at how information is shared between nodes

- Only want dynamically relevant information
- Must be able to extract it from observations
- Must not require serious assumptions about mechanisms

Computational Mechanics

Time series: $\ldots S_{-1} S_0 S_1 \ldots$; $\{S_i\}$, $i \in \mathbb{Z}$, $S_i \in \mathcal{A}$.
$\overleftarrow{S}_t$ = history up to t; $\overrightarrow{S}_t$ = future from t
Process: $\mathrm{P}(\ldots S_{-1} S_0 S_1 \ldots)$.

A Prediction = Distribution over futures $\mathrm{P}(\overrightarrow{S})$.
Predicting: Map η from history $\overleftarrow{s}$ to future distribution:

$$\eta(\overleftarrow{s}) = \mathrm{P}(\overrightarrow{S})$$

Prediction Method = A partition of the set $\{\overleftarrow{S}\}$ of histories
Effective States $\mathcal{R}$ = Partition + Future distributions
Optimal Prediction $\Rightarrow$ Find the best partition; How?
1. Maximize $I(\overrightarrow{S}; \mathcal{R})$
2. Then minimize $H[\mathcal{R}]$

Causal Equivalence

$$\overleftarrow{s} \sim \overleftarrow{s}' \Leftrightarrow \mathrm{P}(\overrightarrow{S} \mid \overleftarrow{S} = \overleftarrow{s}) = \mathrm{P}(\overrightarrow{S} \mid \overleftarrow{S} = \overleftarrow{s}')$$

$$\epsilon(\overleftarrow{s}) \equiv \left\{ \overleftarrow{s}' \middle| \mathrm{P}(\overrightarrow{S} \mid \overleftarrow{S} = \overleftarrow{s}) = \mathrm{P}(\overrightarrow{S} \mid \overleftarrow{S} = \overleftarrow{s}') \right\}$$

$$\text{Causal States}: \quad \mathcal{S} \equiv \epsilon(\overleftarrow{S})$$

All and only the distinctions that make a difference

Labeled transition probabilities:

$$T_{ij}^{(s)} \equiv \mathrm{P}(S_{t+1} = s, \mathcal{S}_{t+1} = j | \mathcal{S}_t = i)$$
$$= 0 \text{ if } j \neq g(i,s) \text{ (determinism)}$$

ϵ-Machine = Causal states + Transition probabilities

Properties

Optimal Prediction (Prescience): $\forall \eta$ (alternative states $\boldsymbol{\mathcal{R}}$)

$$I(\overrightarrow{S};\mathcal{R}) \ \leq \ I(\overrightarrow{S};\mathcal{S})$$

Prescience $\Leftrightarrow$ statistical sufficiency

Minimality: If $\boldsymbol{\mathcal{R}}$ prescient, then

$$H[\mathcal{R}] \ \geq \ H[\mathcal{S}]$$

Statistical complexity: $C_\mu = H[\mathcal{S}]$

Uniqueness: If $\boldsymbol{\mathcal{R}}$ prescient and minimal, then $\exists f$ such that

$$\begin{aligned} \mathcal{R} &= f(\mathcal{S}) \\ \mathcal{S} &= f^{-1}(\mathcal{R}) \end{aligned}$$

Remarks:

1. Markovian, but not only HMMs

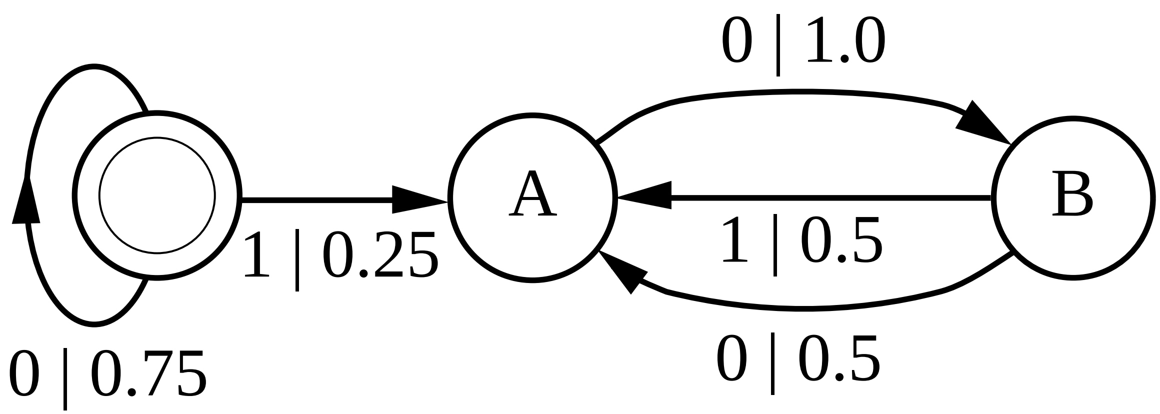
2. ϵ-Machine Reconstruction: Data/Theory $\rightarrow$ ϵ-machine
(i) Mathematically
(ii) Batch Algorithm
(iii) Online Algorithm

3. ϵ-Machine Filtering: Measurement series $\rightarrow$ causal-state series

$$\ldots s_{-2} s_{-1} s_0 s_1 s_2 \ldots \rightarrow \ldots \sigma_{-2} \sigma_{-1} \sigma_0 \sigma_1 \sigma_2 \ldots \ ,$$

where $s_i \in \mathcal{A}$ and $\sigma \in \boldsymbol{\mathcal{S}}$.

Example: Noisy Period-Two
("every other symbol a wildcard")



Difference between:

1. Observations: 01000010 $\sim 0\mathcal{A}0\mathcal{A}0\mathcal{A}0\mathcal{A}$
2. Causal States: ABABABAB.

Causal Synchrony Ψ in Network

Network G with N nodes, each with a time series S_t^i.
Do not know communication topology.

Reconstruct ϵ-machine for each node i: $\{M^i : i = 1, \ldots, N\}$
Filter time series for each node i: $S_t^i \to_{M^i} \mathcal{S}_t^i$
For each pair $(i, j) \in G$:

$$\Psi^{ij} = \frac{I(\mathcal{S}^i; \mathcal{S}^j)}{\min\left\{C_\mu^i, C_\mu^j\right\}}$$

$$\begin{aligned} I(\mathcal{S}^i; \mathcal{S}^j) &= \text{Error in treating } i \text{ and } j \text{ as independent} \\ &= \text{Amount of shared causal information} \\ \Psi^{ij} &= \text{Degree to which nodes share common causal state} \\ &= \text{Degree of information spread over network} \\ &\in [0, 1] \end{aligned}$$

Network Causal Synchrony $\Psi_G = \langle \Psi^{ij} \rangle_{(i,j) \in G}$

Example: Synchronized Clusters

$\Psi^{ij} = 1$ inside perfectly synchronized clusters

$\Psi^{ij} = 0$ across purely random (independent) nodes

Example: Cyclic Processes

Periodic in causal states: Can still be observationally random.

Periods $p, q,\ p \leq q$

$$
\begin{aligned}
r &= \text{lcm}(p, q) \\
I &= \log pq/r \\
\\
\Psi &= \frac{\log pq/r}{\log p} \\
\Psi &= 0, \text{ if p and q relatively prime}
\end{aligned}
$$

Example: Integrate and Fire Neurons

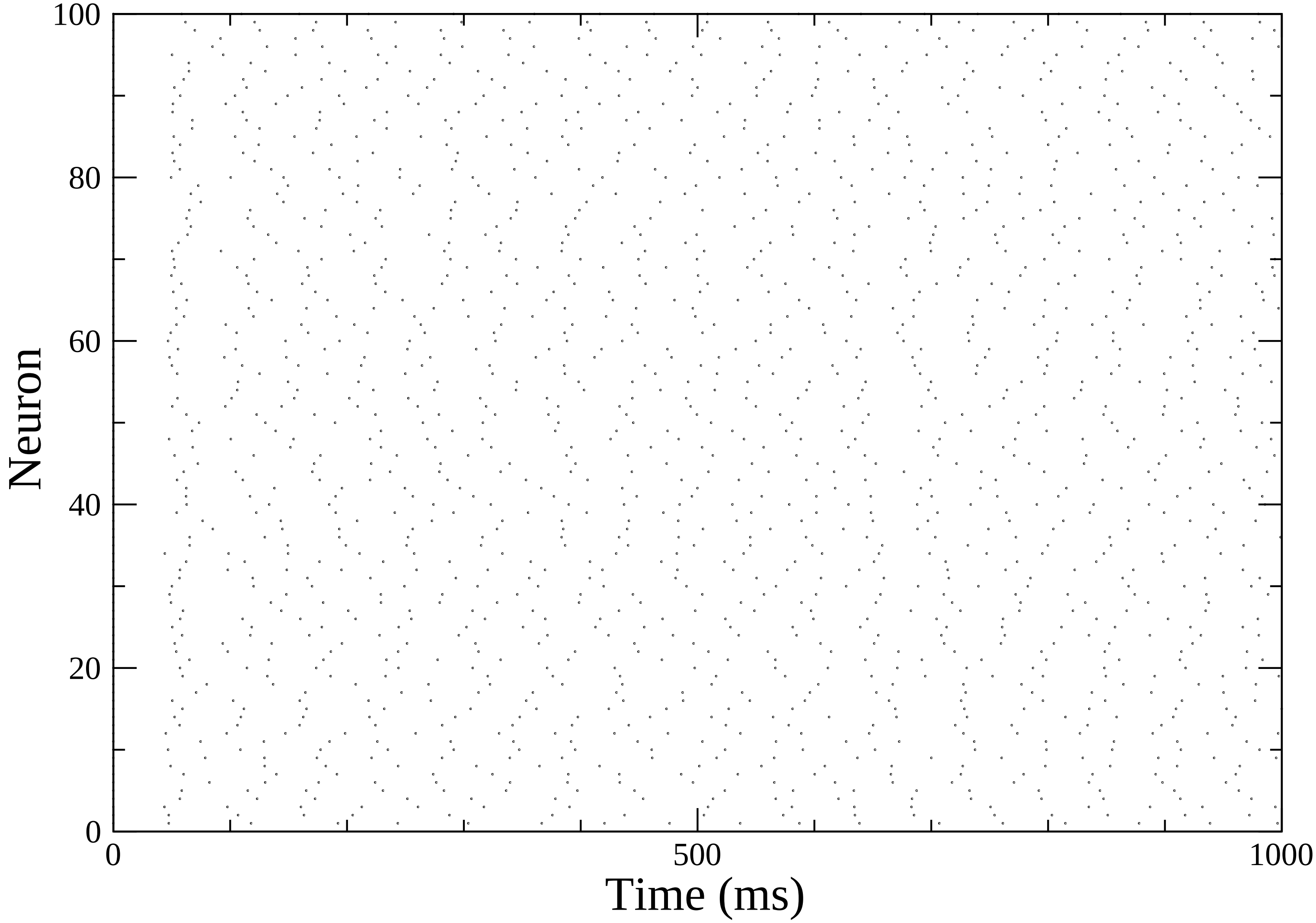
(Data due to M. Camperi)

100 integrate-and-fire neurons

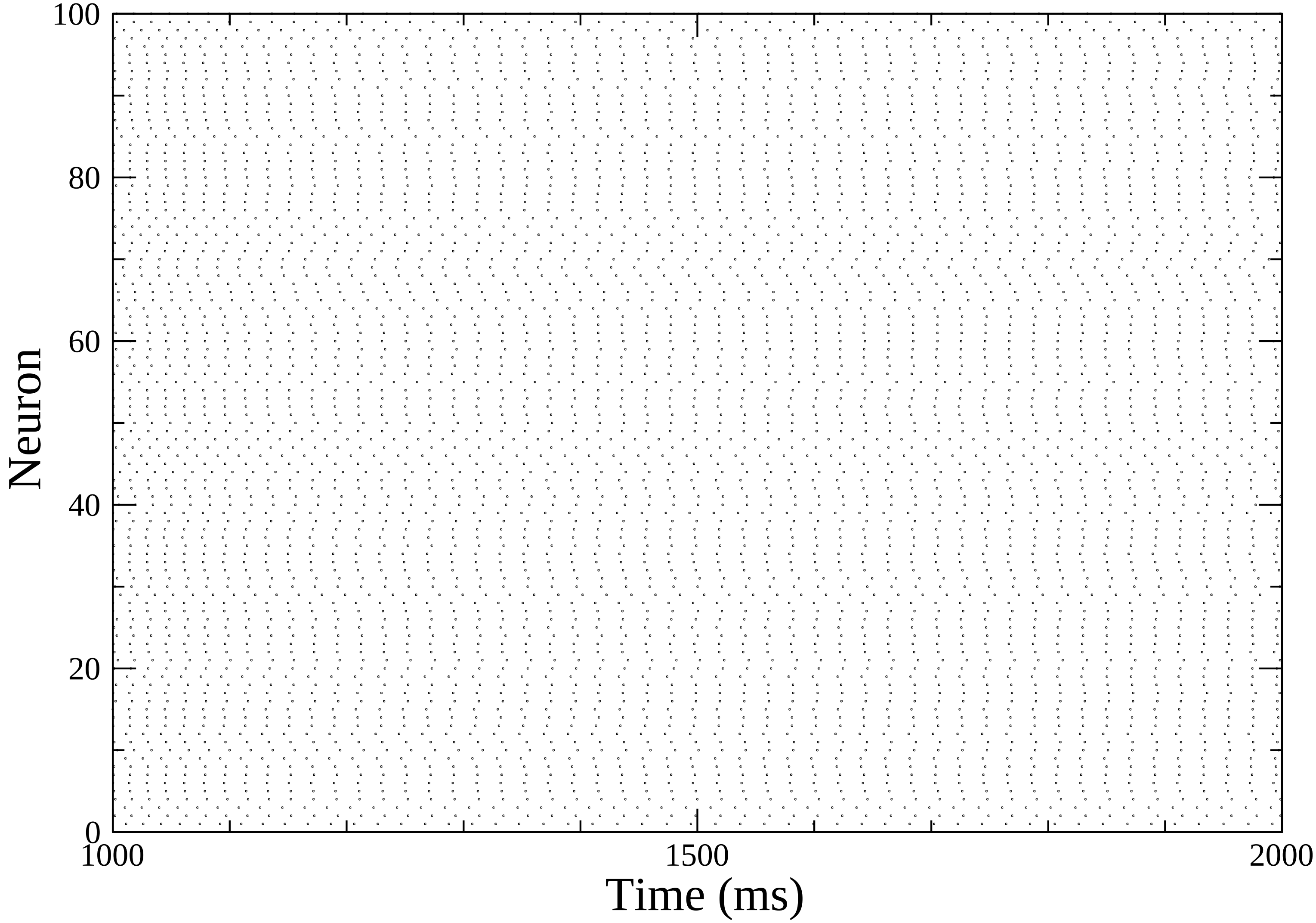
All connect to all

Interaction begins at 1 second

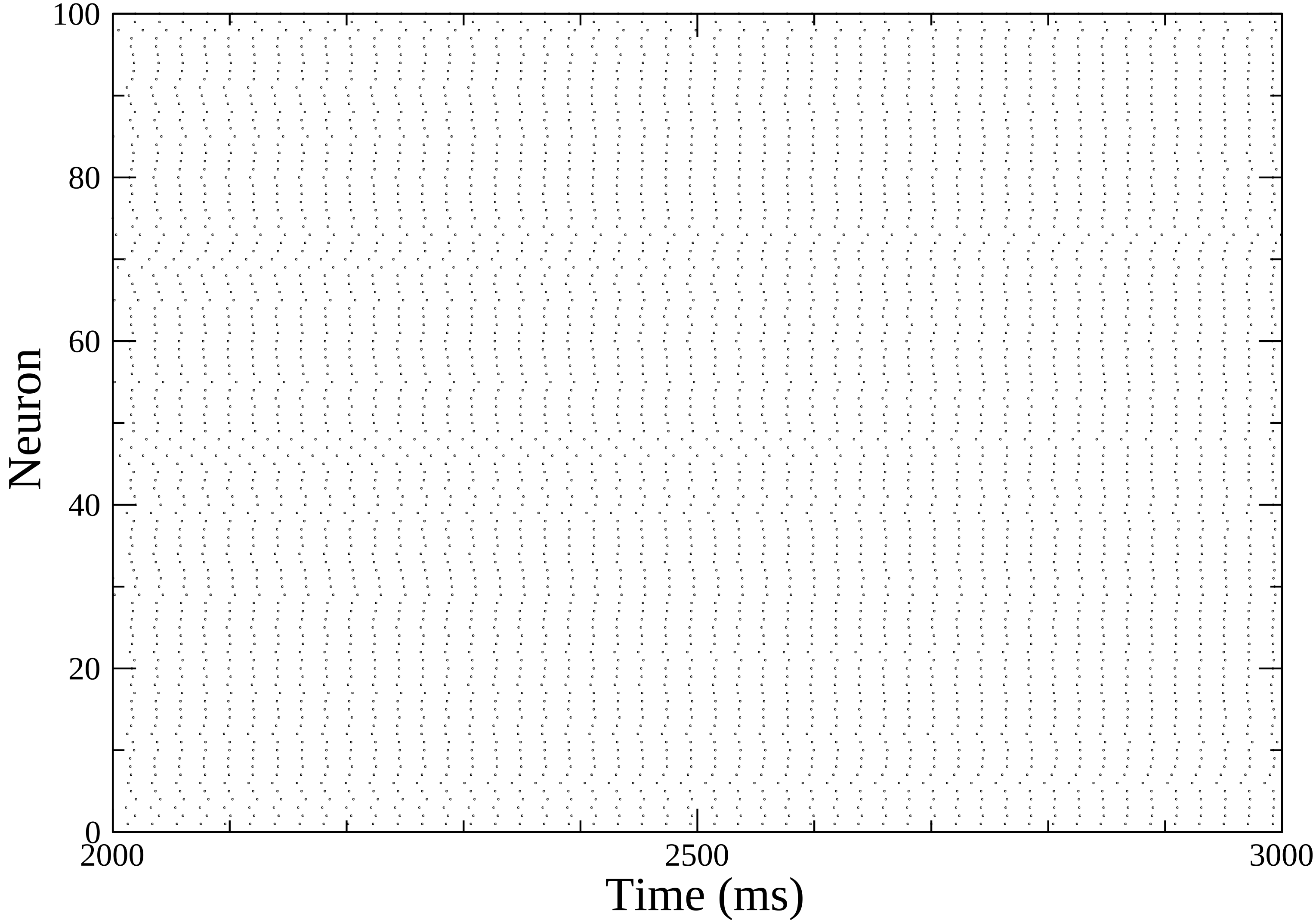
Spike Rastergram: 0 to 1000 ms

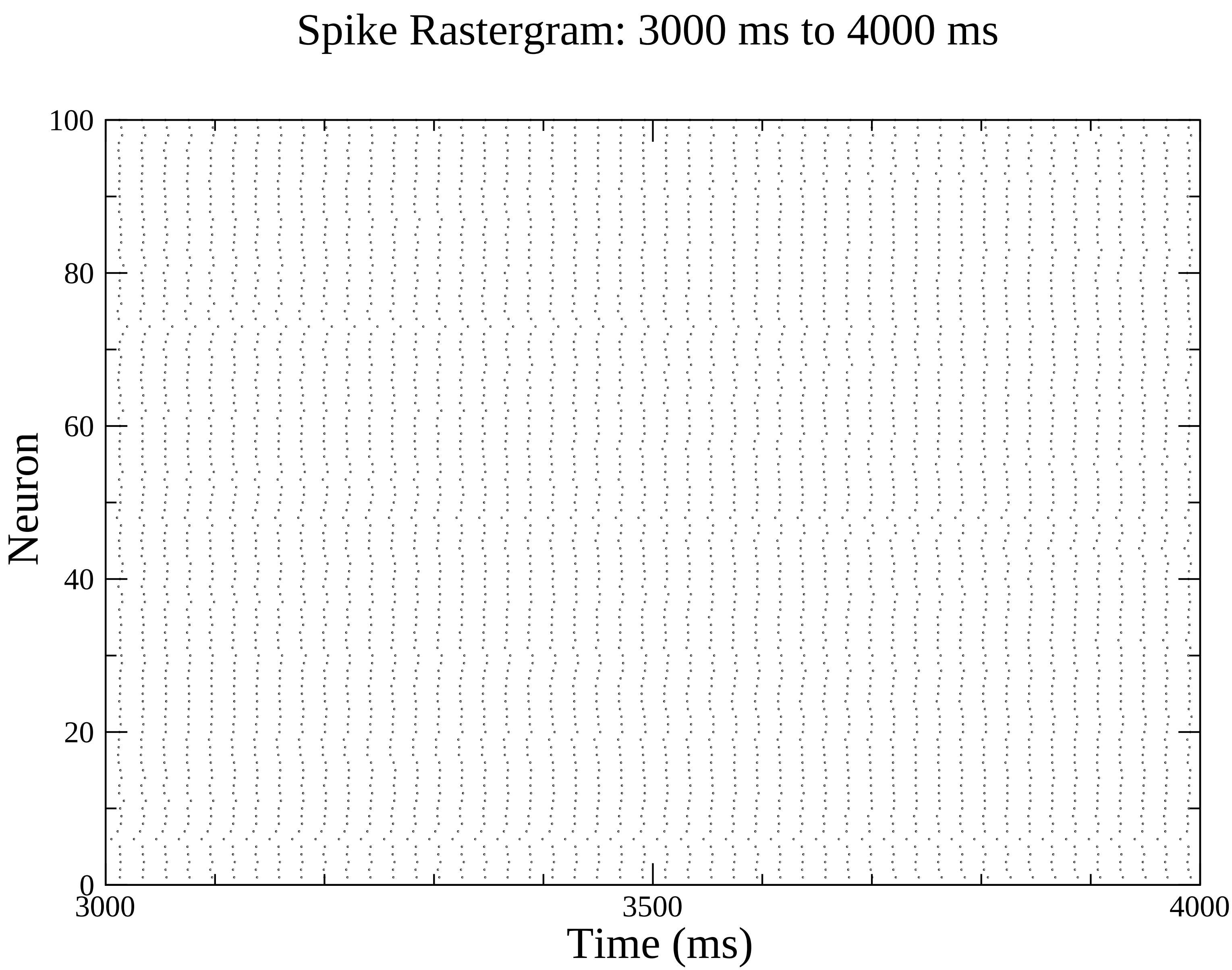


Spike Rastergram: 1000 ms to 2000 ms

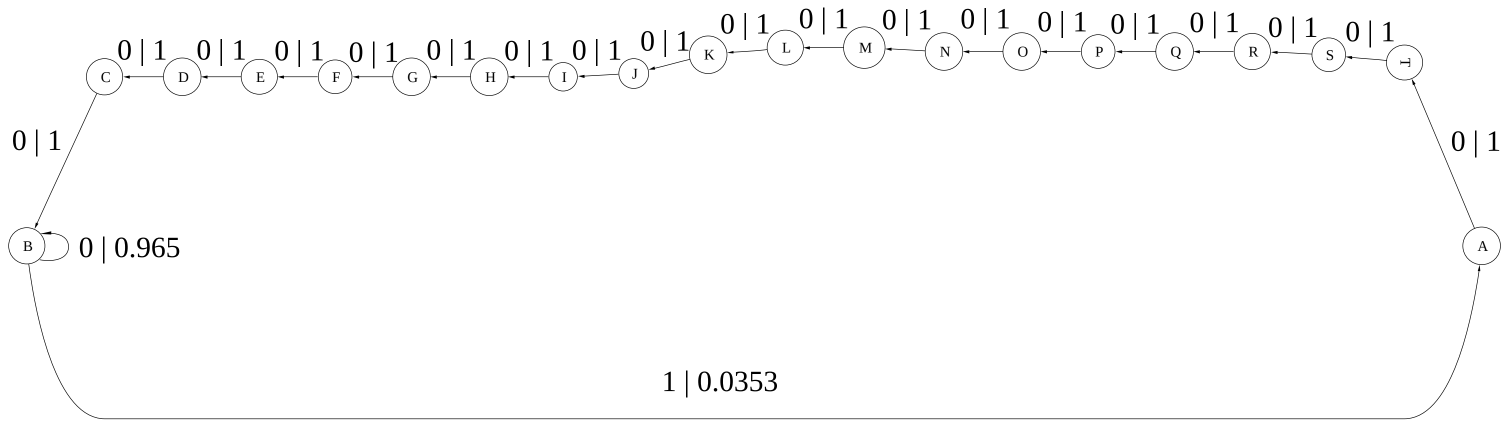


Spike Rastergram: 2000 ms to 3000 ms




Spike Rastergram: 3000 ms to 4000 ms
Neuron
Time (ms)
0
20
40
60
80
100
3000
3500
4000

Neuron ε-Machine, Reconstructed



Causal Synchrony in I&F Neural Network

t(s)	Ψ
0–1	0.073
1–2	0.353
2–3	0.395
3–4	0.434
4–5	0.439

Causal Architecture of Networks

Ψ_G too coarse; rather cluster by Ψ^{ij}

Hierarchical decomposition
— cf. Christopher Alexander, *Notes on the Synthesis of Form*

Causal drive: Conditional mutual information $I(\mathcal{S}^i_{t+1}; [\mathcal{S}^j_t, \mathcal{S}^k_t, \mathcal{S}^l_t])$

Stanislav Andreski, *Military Organization and Society*:
Coherence $\approx \Psi$
Subordination $\approx$ Change in distribution of actions given orders (relative entropy)