Spike sorting: What is it? Why do we need it? Where does it come from? How is it done? How to interpret it? III. Improving sorting quality through stochastic modeling of spike trains.

Christophe Pouzat

Mathématiques Appliquées à Paris 5 (MAP5)

Université Paris-Descartes and CNRS UMR 8145

christophe.pouzat@parisdescartes.fr

Thursday November 27, 2014

Where are we ?

What makes data "difficult"

An interlude to keep you motivated

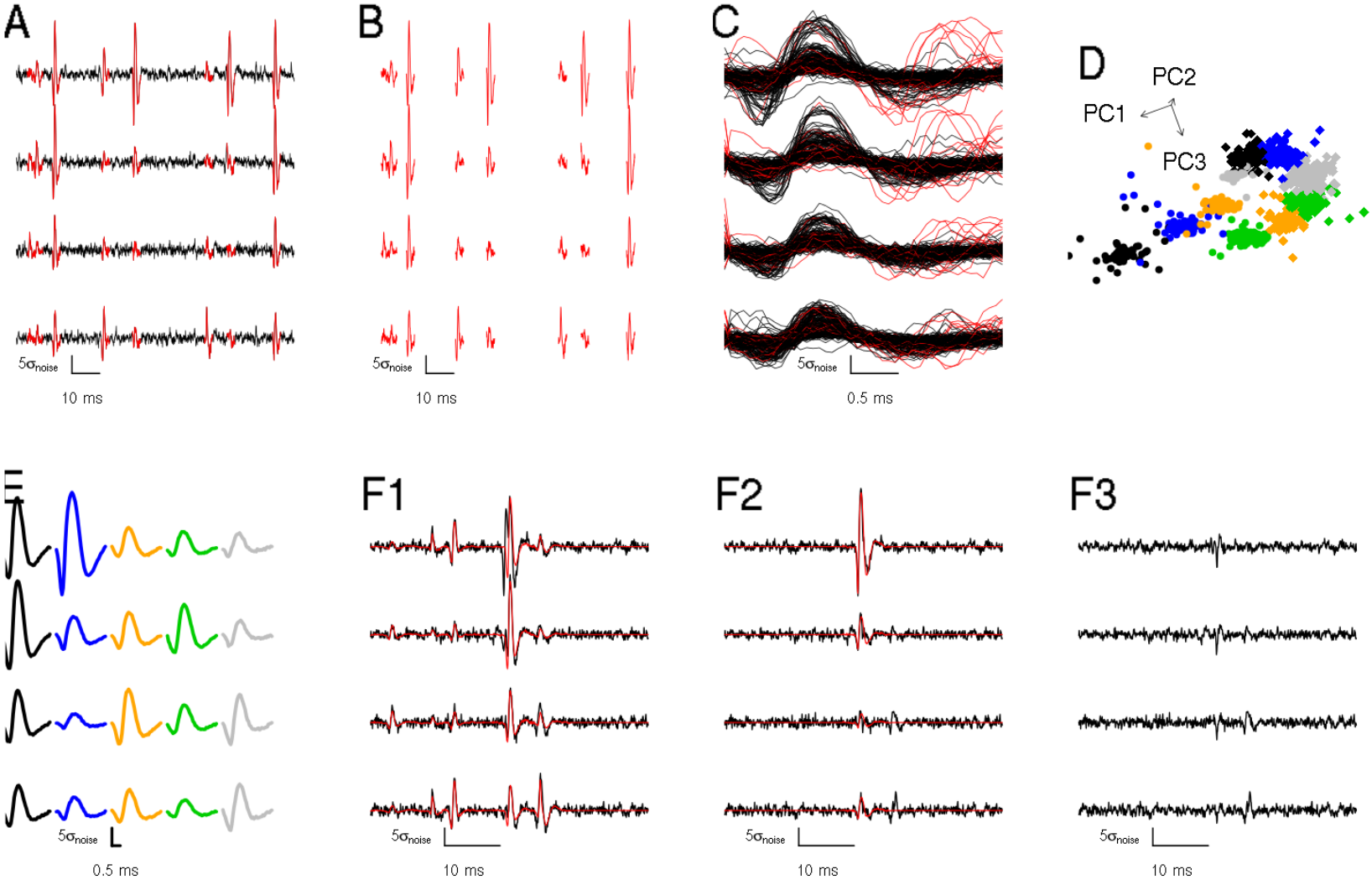
A more realistic model and its problems

A analogy with statistical physics

Back to our simulated data

Application to real data

Yesterday's summary



Spike amplitude dynamics (1)

We did not deal with that

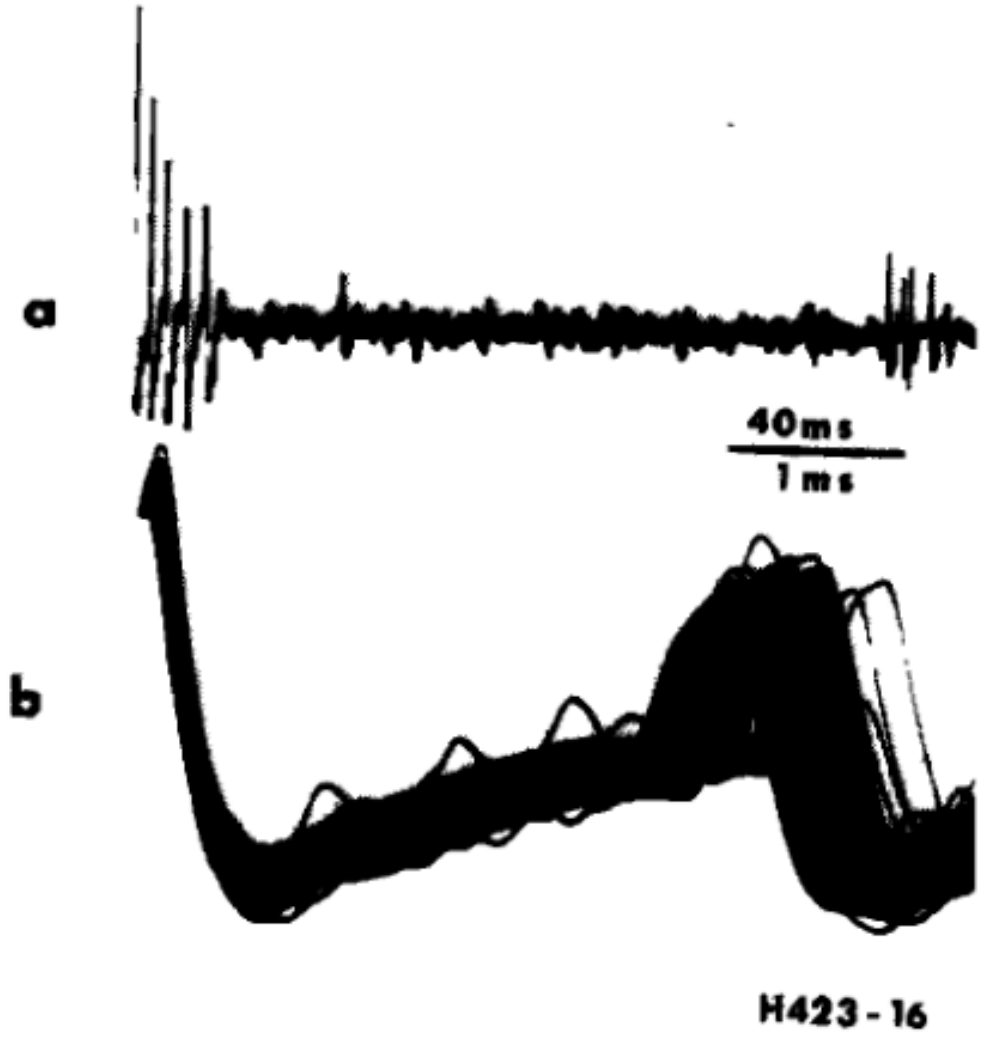


Fig. 2 of Calvin (1972) *Electroenceph clin Neurophysiol* **34**:94.

Spike amplitude dynamics (2)

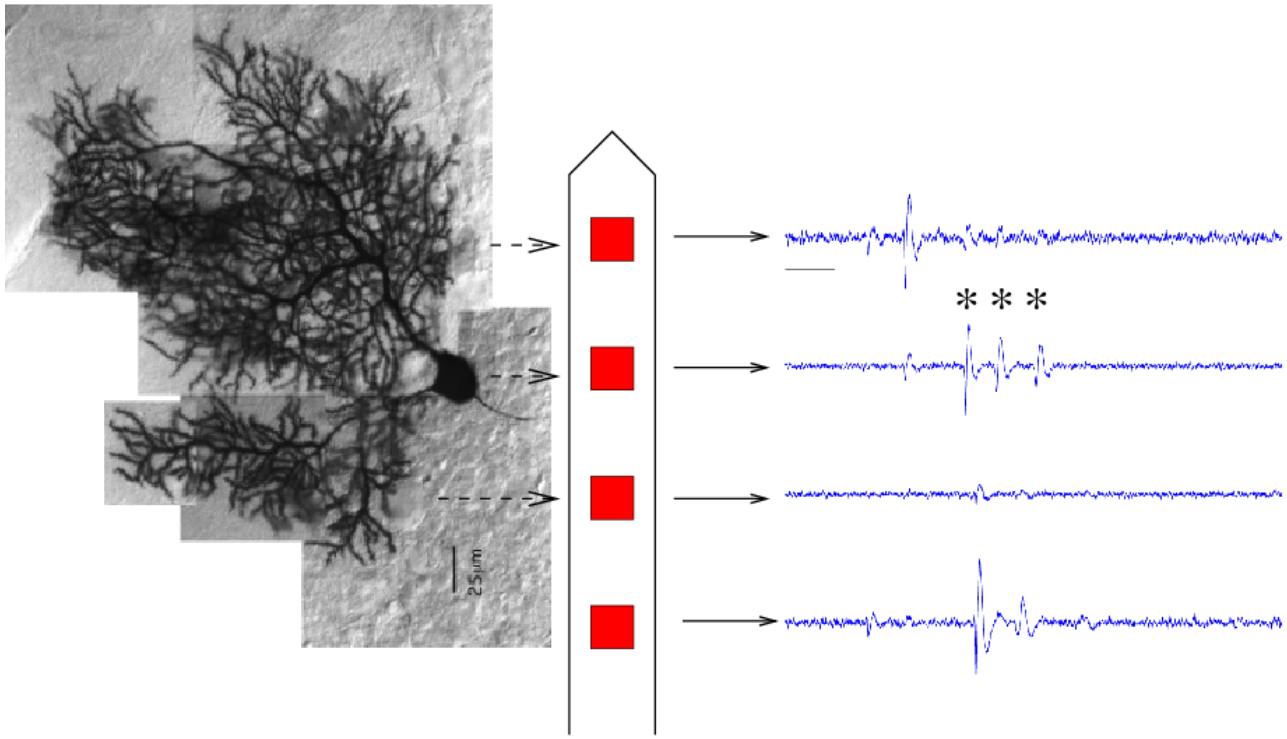
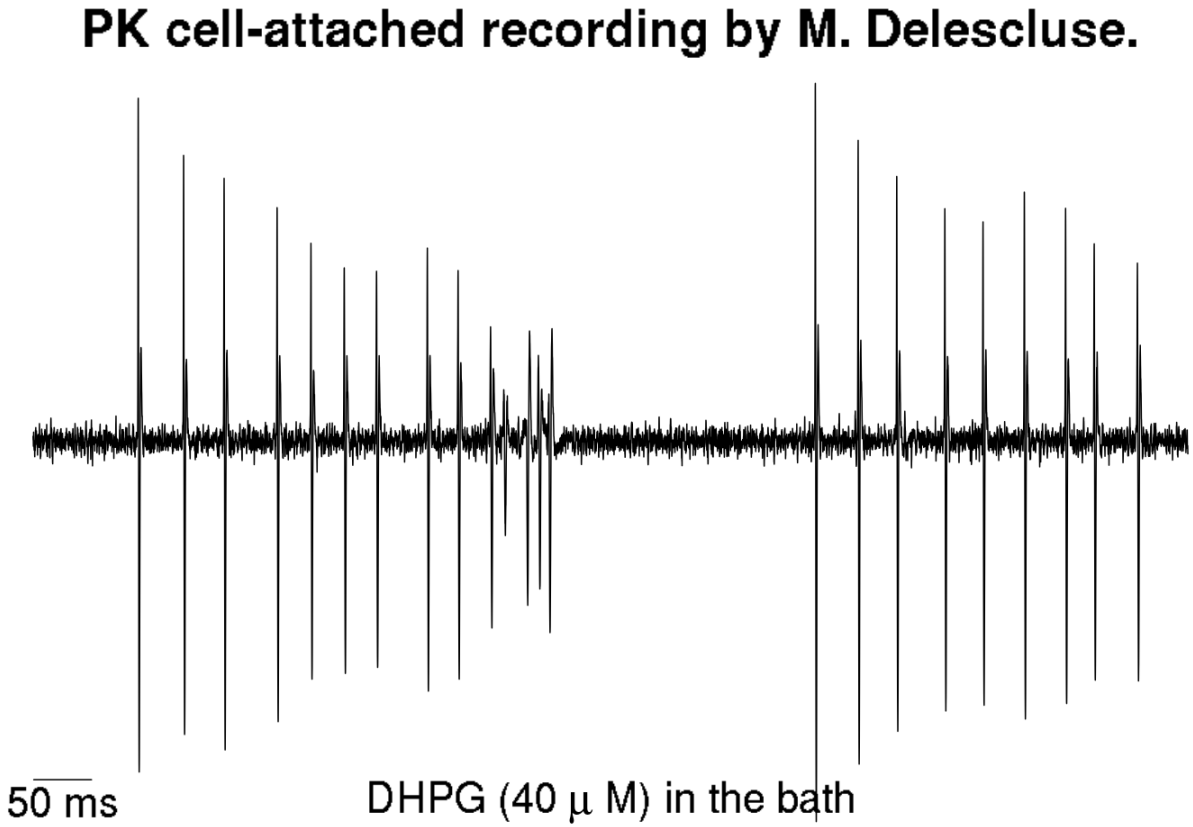


Fig. 3 of Pouzat (2005) *Technique(s) for Spike Sorting*. In *Methods and Models in Neurophysics. Les Houches 2003.* Elsevier.

Spike amplitude dynamics (3)



We try a naive "exponential relaxation model" for these data:

$$\mathbf{a}\left(\mathit{isi}\right) = \mathbf{p} \cdot \left(1 - \delta \cdot \exp\left(-\lambda \cdot \mathit{isi}\right)\right) \ .$$

The exponential relaxation in only an approximation!

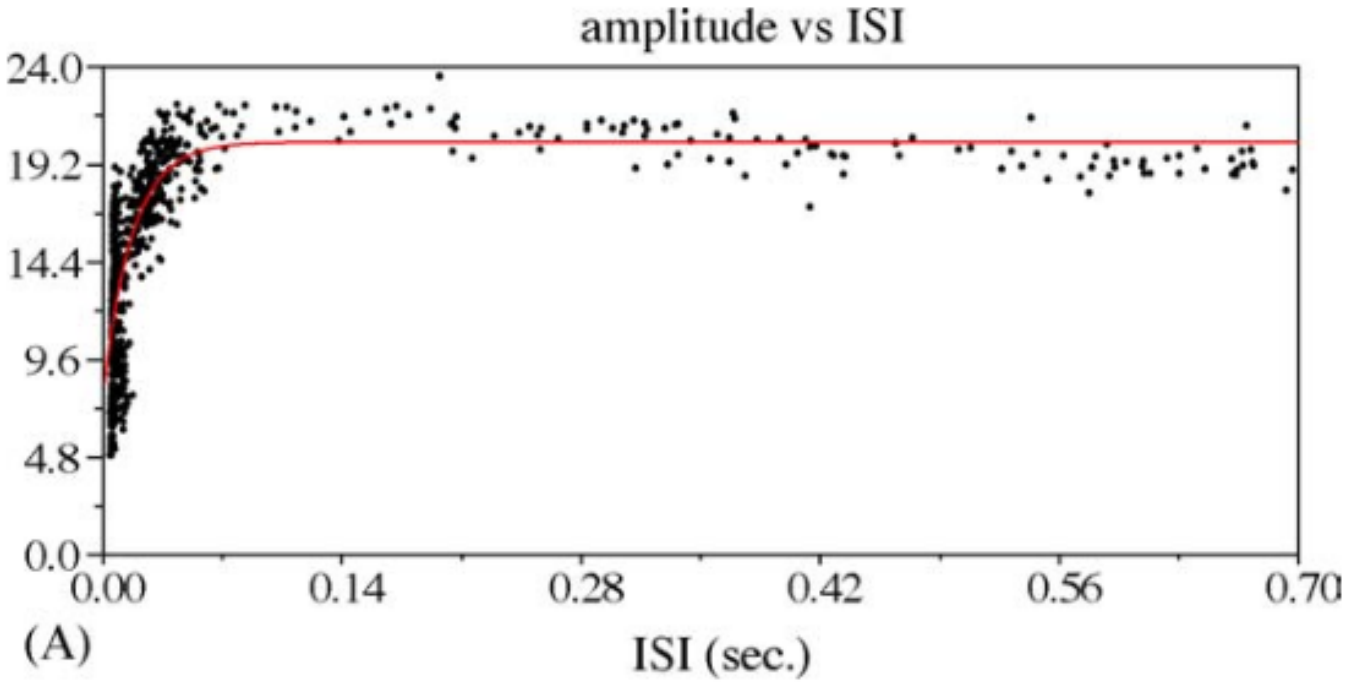


Fig. 3A of Delescluse and Pouzat (2006) *J Neurosci Methods* **150**:16.

Neuronal discharges are not Poisson

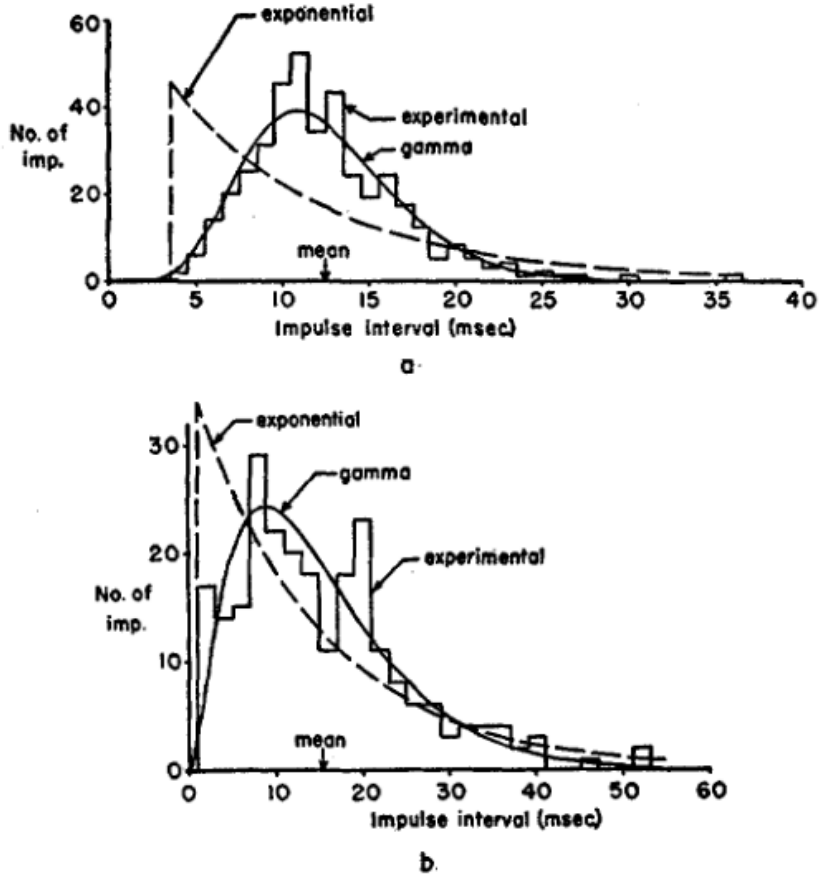


FIG. 7. Two distributions of impulse intervals from different ganglion cells. (*a*) shows unit 3 (see Table I), and (*b*), unit 4. Two theoretical curves, the exponential and the gamma distributions, are shown. Only the gamma gives a satisfactory fit.

Fig. 7 of Kuffler, Fitzhugh and Barlow (1957) *J Gen Physiol* **40**:683. Evidence for serial correlation of the intervals was found (p 687). Hagiwara (1950, 1954) made similar reports.

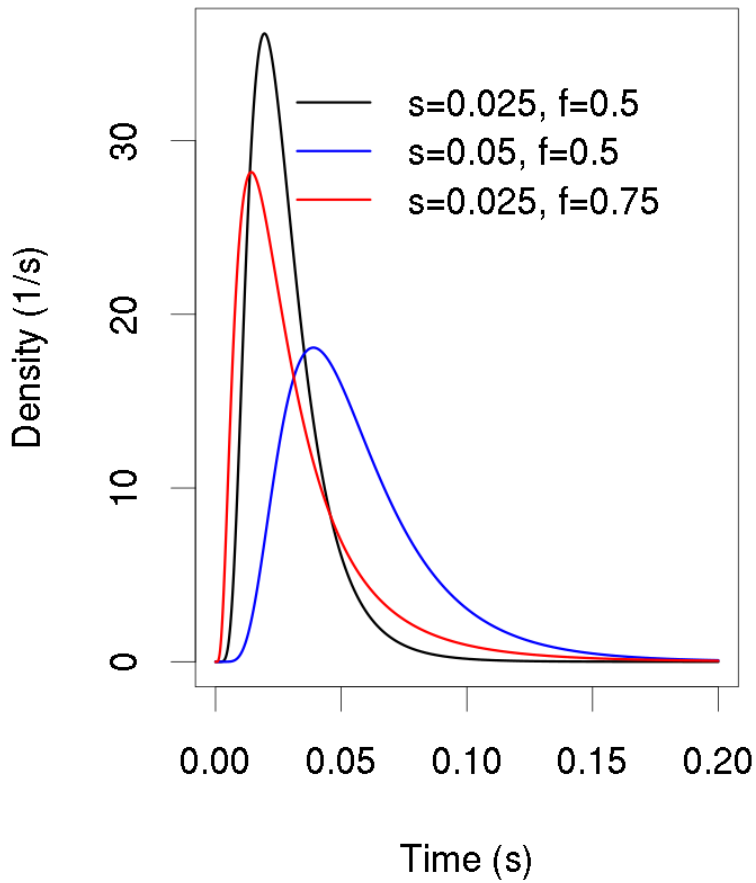
Log-normal density

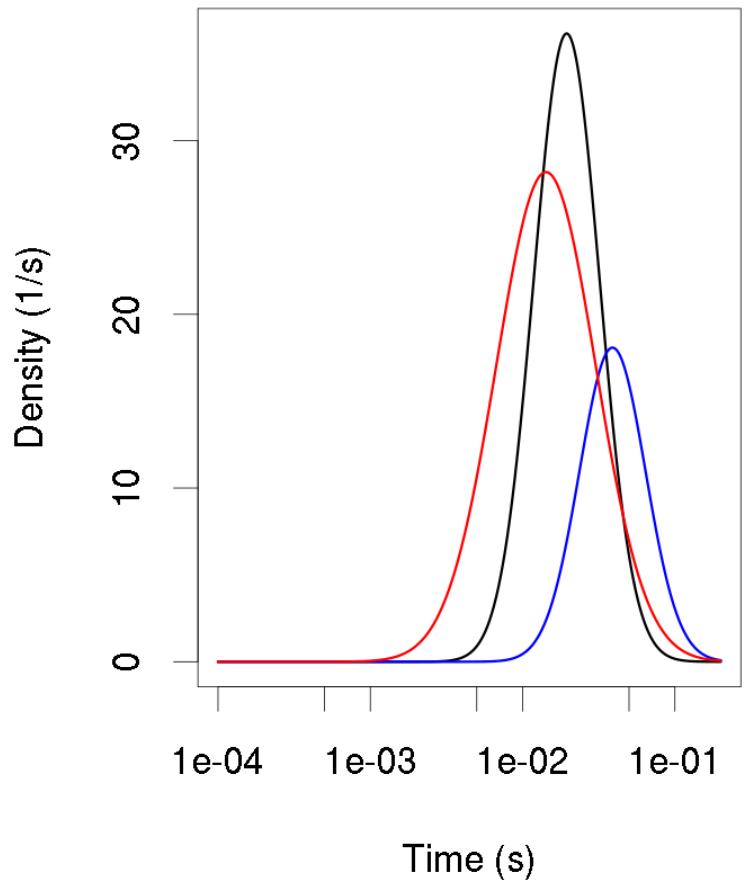
Empirical ISI densities are better described by a log-normal density than by a Poisson density:

$$\pi_{ISI}\left(ISI = isi\right) = \frac{1}{isi\, f\, \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\ln isi - \ln s}{f}\right)^2\right]$$

where, s is a scale parameter (measured in sec) and f is a dimensionless shape parameter.

Log-normal density: exemples



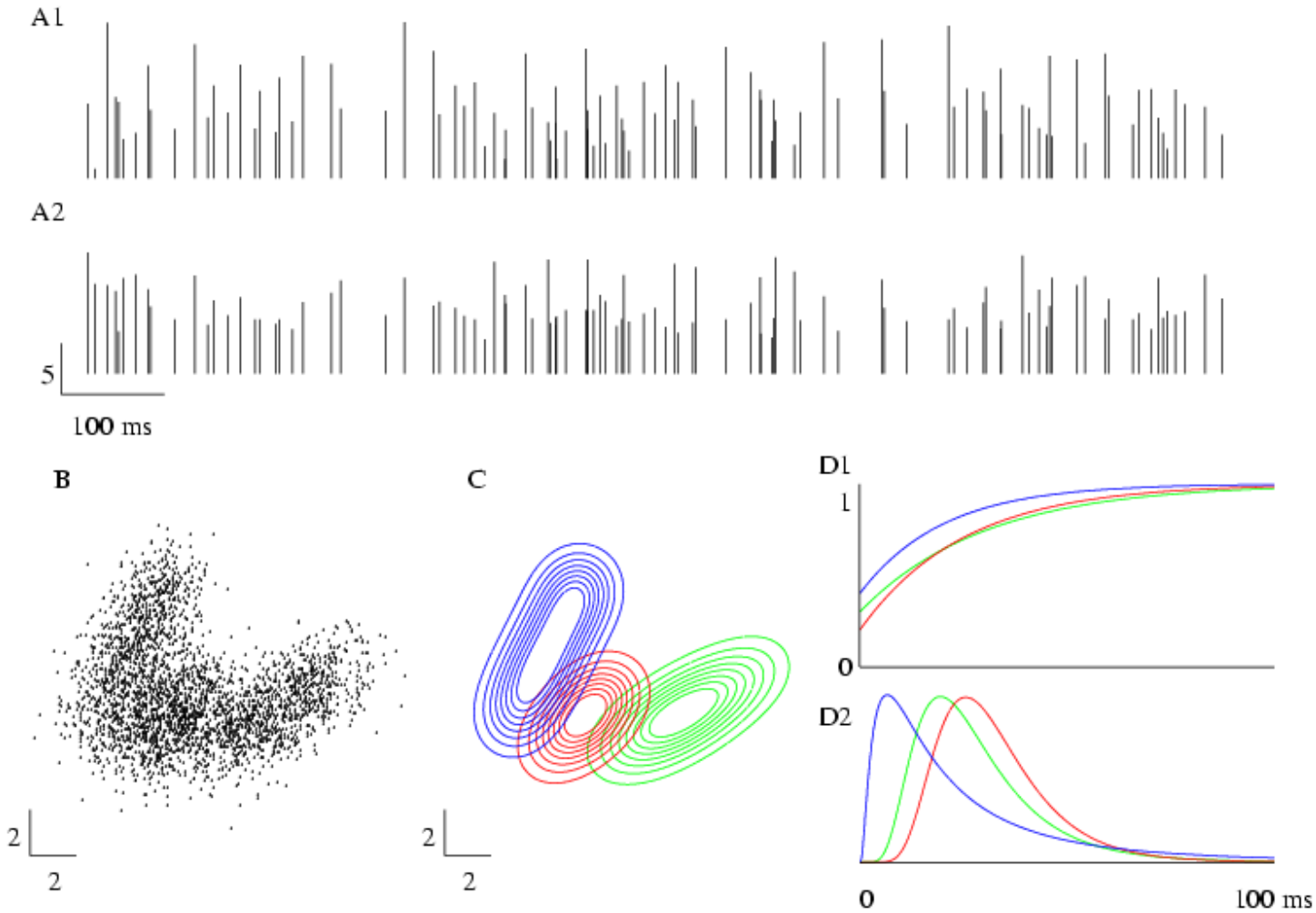


What do we want?

- Recognizing that the inter spike interval (ISI) distribution, and more generally the neuron's stochastic intensity, provides potentially useful information for spike sorting, we would like a data generation model including this information.
- We would also like a data generation model giving room for a description of the spikes amplitude dynamics during bursts.

Where are we ?

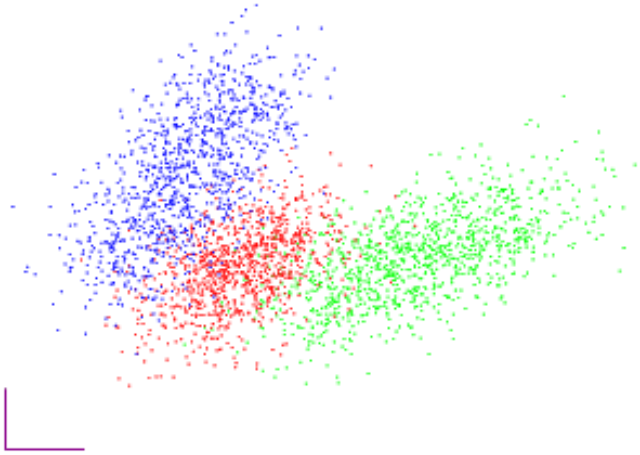
Some simulated data



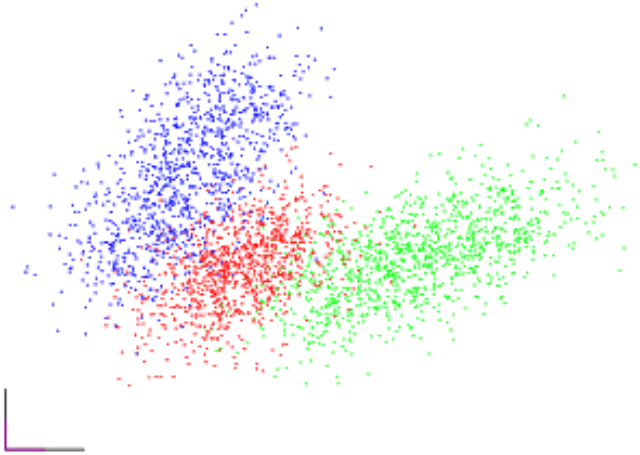
Simulated data with 3 neurones on 2 recording sites.

The results we will get

Actual configuration



Most likely configuration



Errors

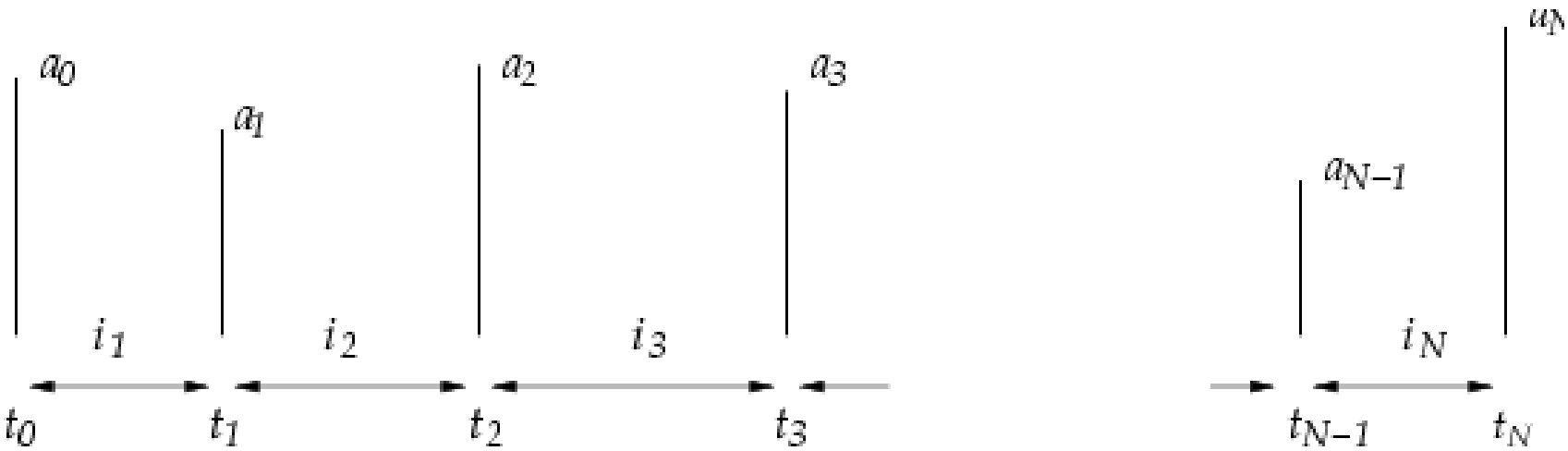
Where are we ?

First "improved" model

We will assume that the neuron discharge independently and that:

- Individual discharges are well approximated by a renewal process with a log-normal density.
- The spike amplitude depends on the elapsed time since the last spike (of the same neuron); this dependence is moreover well approximated by an exponential relaxation.
- The recording noise is white and Gaussian and is independent of the spikes.

Likelihood for single neuron data (1)



We have:

$$\pi\left(D \mid \mathbf{p}, \delta, \lambda, s, f\right) = \prod_{j=1}^{N} \pi_{isi}\left(i_j \mid s, f\right) \cdot \pi_{amp}\left(\mathbf{a}_j \mid i_j, \mathbf{p}, \delta, \lambda\right),$$

where

$$\pi_{amp}\left(\mathbf{a}_j \mid i_j, \mathbf{p}, \delta, \lambda\right) = \frac{1}{(2\pi)^{\frac{n_s}{2}}} \cdot \mathrm{e}^{-\frac{1}{2}\left\|\mathbf{a}_j - \mathbf{p}\cdot\left(1-\delta\cdot\exp\left(-\lambda\cdot i_j\right)\right)\right\|^2}.$$

Likelihood for single neuron data (2)

The log-likelihood can be written as the sum of two terms:

$$\mathcal{L}(D \mid \mathbf{p}, \delta, \lambda, s, f) = \mathcal{L}_{isi}(D \mid s, f) + \mathcal{L}_{amp}(D \mid \mathbf{p}, \delta, \lambda)$$

where:

$$\mathcal{L}_{isi}(D \mid s, f) = -N \cdot \ln f - \sum_{j=1}^{N} \left\{ \ln i_j + \frac{1}{2} \left[\frac{\ln\left(\frac{i_j}{s}\right)}{f} \right]^2 \right\} + Cst$$

and:

$$\mathcal{L}_{amp}(D \mid \mathbf{p}, \delta, \lambda) = -\frac{1}{2} \sum_{j=1}^{N} \|\mathbf{a}_j - \mathbf{p} \cdot (1 - \delta \cdot \exp(-\lambda \cdot i_j))\|^2 + Cst\,.$$

Configuration

We will use Θ for our model parameters vector, that is, for a model with K neurons:

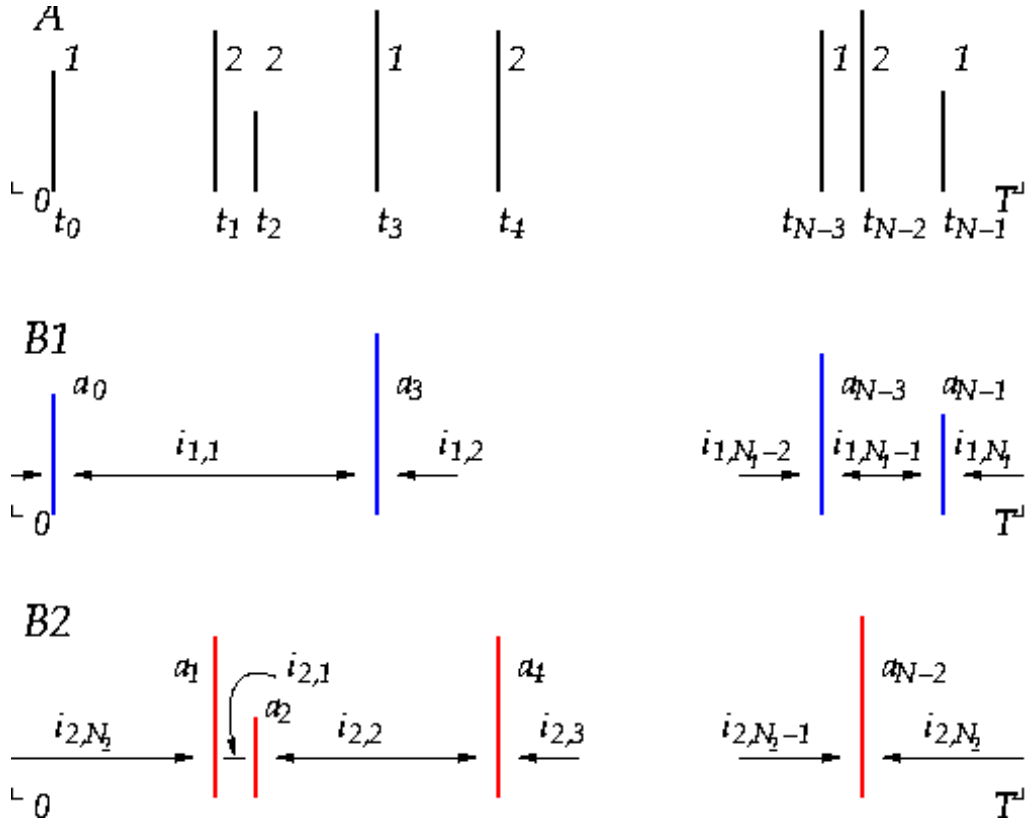
$$\Theta = (\mathbf{P}_1, \Delta_1, \Lambda_1, S_1, F_1, \ldots, \mathbf{P}_K, \Delta_K, \Lambda_K, S_K, F_K) \ .$$

We will formalize our *a priori* ignorance of the origin of each spike by associating to each spike, j, a label $L_j \in \{1, \ldots, K\}$.
$l_j = 3$ means that event j was generated by neuron 3.
We will use C for the configuration, that is, the random variable:

$$C = (L_1, \ldots, L_N)^T \ .$$

With this formalism, spike sorting amounts to estimating the configuration.

Likelihood for data from several neurons



If the configuration realization is known, the likelihood computation is easily done:

$$\pi\left(D \mid c, \theta\right) = \prod_{q=1}^{K} \prod_{j=1}^{N_q} \pi_{isi}\left(i_{q,j} \mid s_q, f_q\right) \cdot \pi_{amp}\left(\mathbf{a}_{q,j} \mid i_{q,j}, \mathbf{p}_q, \delta_q, \lambda_q\right) .$$

Bayesian formalism (1)

- Being somewhat opportunist, we are going to formalize our ignorance on the configuration c and the model parameters θ by viewing them as realizations of the random variables C and Θ.
- We then have a triplet of random variables $\mathcal{D}$ (whose realization is D), C and Θ.
- We have, by definition of the conditional probability:

$$\pi(D, c, \theta) = \pi(c, \theta \mid D)\, \pi(D) = \pi(D \mid c, \theta)\, \pi(c, \theta)\,.$$

- Rearranging the last two members we get (Bayes rule or Bayes "theorem"):

$$\pi(c, \theta \mid D) = \frac{\pi(D \mid c, \theta)\, \pi(c, \theta)}{\pi(D)}\,.$$

Bayesian formalism (2)

- In the literature, our $\pi(c, \theta)$ is called the *a priori* density and is written $\pi_{prior}(c, \theta)$; that's what we know about the parameters and configuration before observing the data.
- Our $\pi(c, \theta \mid D)$ is called the *a posteriori* density and is written $\pi_{posterior}(c, \theta \mid D)$; that's what we know about the parameters and configuration after observing the data.
- Bayes rule tells us how to update our knowledge once the data have been observed.
- We know how to compute the "likelihood" $\pi(D \mid c, \theta)$.

Bayesian formalism (3)

- We will write $\pi_{prior}(c, \theta) = \pi_{prior}(c)\pi_{prior}(\theta)$ and take $\pi_{prior}(c)$ uniform on $\mathcal{C} = \{1, \ldots, K\}^N$ (notice that I'm cheating, I can know N only after seeing the data) and take

$$\pi_{prior}(\theta) = \prod_{q=1}^{K} \pi_{prior}(s_q)\pi_{prior}(f_q)\pi_{prior}(\mathbf{p}_q)\pi_{prior}(\delta_q)\pi_{prior}(\lambda_q)\,,$$

  and each of the terms is going to be the density of a uniform distribution over a large domain.
- The real problem is the denominator:

$$\pi(D) = \sum_{c \in \mathcal{C}} \int_{\theta} \pi(D, c, \theta) d\theta = \sum_{c \in \mathcal{C}} \int_{\theta} \pi(D \mid c, \theta)\pi_{prior}(c, \theta) d\theta\,,$$

  since it involves a summation over K^N configurations and K is of the order of 10 and N of the order of 1000!

Bayesian formalism (4)

- Assuming we find a way to circumvent the "combinatorial explosion" (K^N) problem and to get an estimator, $\hat{\pi}(c, \theta \mid D)$, of:

$$\pi_{posterior}(c, \theta \mid D) = \frac{\pi(D \mid c, \theta)\, \pi(c, \theta)}{\pi(D)} \,.$$

- The estimated posterior configuration probability would be:

$$\hat{\pi}_{posterior}(c \mid D) = \int_{\theta} \hat{\pi}(c, \theta \mid D) d\theta \,.$$

- The spike sorting results could then be summarized by the *Maximum A Posteriori* (MAP) estimator:

$$\hat{c} = \arg\min_{c \in \mathcal{C}} \hat{\pi}_{posterior}(c \mid D) \,.$$

Bayesian formalism (5)

- Clearly, a better use of the results would be for any statistics $T(C)$ (think of the cross-correlogram between two spike trains) to use:

$$\widehat{T(C)} = \sum_{c \in \mathcal{C}} T(c) \hat{\pi}_{posterior}(c \mid D)\,,$$

  instead of: $T(\hat{c})$.

- But to do all that we have to deal with the combinatorial explosion!
- In such situations, a "good strategy" is too look at other domains to see if other people met a similar problem and found a solution...

Where are we ?

A analogy with statistical physics (1)

- Luckily for us, Physicists got a problem similar to ours at the turn of the 1950s and found a solution.
- Let us write:

$$E\left(c, \theta \mid D\right) = -\log\left[\pi\left(D \mid c, \theta\right) \cdot \pi_{prior}\left(c, \theta\right)\right] ,$$

  notice that we know how to compute $E\left(c, \theta \mid D\right)$.
- Then:

$$\pi_{posterior}(c, \theta \mid D) = \frac{\exp\left[-\beta E\left(c, \theta \mid D\right)\right]}{\pi(D)} ,$$

  where $\beta = 1$, β is the inverse temperature for Physicists who write $Z(\beta) = \pi(D)$ and call it the partition function.
- Written in this way, $\pi_{posterior}(c, \theta \mid D)$, is nothing more than a Gibbs distribution.

A analogy with statistical physics (2)

- Physicists don't know how to compute the partition function $Z(\beta)$ in many cases of interest.
- But they need to compute quantities like:

$$\sum_{c \in \mathcal{C}} \int_{\theta} T(c, \theta) \pi_{posterior}(c, \theta \mid D) d\theta \, .$$

- In 1953, Metropolis, Rosenbluth, Rosenbluth, Teller and Teller published "Equation of State Calculations by Fast Computing Machines" in *Journal of Chemical Physics*, proposing an algorithm for estimating this kind of integral without knowing the partition function.

A analogy with statistical physics (3)

- In 1971, Hastings published "Monte Carlo Sampling Methods Using Markov Chains and Their Applications" in *Biometrika* proving that the previous algorithm was indeed doing what it was suppose to do.
- This algorithm is know called the Metropolis-Hastings Algorithm.

Metropolis-Hastings in a simple setting (1)

- We will generate a collection of states $\{x^{(1)}, x^{(2)}, \ldots, x^{(R)}\}$ according to a target distribution $\pi(x)$ (the x stand here for our previous c, θ).
- To that end we use a Markov chain which asymptotically reaches the stationary distribution $\pi(x)$.
- A Markov chain is uniquely defined by its transition matrix $\mathrm{P}(X^{(k+1)} = x' \mid X^{(k)} = x)$ and its initial state distribution.

Metropolis-Hastings in a simple setting (2)

- A Markov chain has a unique stationary distribution when the following two conditions are met:
  1. A stationary distribution exists: a *sufficient* condition for that is the detailed balance that implies

$$\pi(x)\mathrm{P}(x' \mid x) = \pi(x')\mathrm{P}(x \mid x')\,.$$

  2. The stationary distribution is unique if the chain is aperiodic (the system does not return to the same state at fixed intervals) and positive recurrent (the expected number of steps for returning to the same state is finite).
- Checking the detailed balance requires a knowledge of $\pi(x)$ up to a normalizing constant.
- Constructing a transition matrix satisfying the detailed balance for a given target distribution also requires the knowledge of that distribution up to a normalizing constant.

Metropolis-Hastings in a simple setting (3)

- The "trick" is to construct the transition matrix elements in two sub-steps: a proposal, $g(x' \mid x)$, and an acceptance-rejection, $A(x' \mid x)$, to get $\mathrm{P}(x' \mid x) = g(x' \mid x)\, A(x' \mid x)$.
- The detailed balance is satisfied if:

$$\pi(x) g(x' \mid x) A(x' \mid x) = \pi(x') g(x \mid x') A(x \mid x')$$

that is if

$$\frac{A(x' \mid x)}{A(x \mid x')} = \frac{\pi(x') g(x \mid x')}{\pi(x) g(x' \mid x)} \, .$$

- The Metropolis choice is:

$$A(x' \mid x) = \min \left(1, \frac{\pi(x') g(x \mid x')}{\pi(x) g(x' \mid x)} \right) \, .$$

Metropolis-Hastings in a simple setting (4)

- The Metropolis choice for $A(x' \mid x)$ requires only the knowledge of the target distribution up to a normalizing constant.
- The proposal matrix $g(x' \mid x)$ is chosen to ensure positive recurrence and aperiodicity.
- If g is positive recurrent and aperiodic and if $g(x' \mid x) = 0$ implies $g(x \mid x') = 0$ then we can show that $\mathrm{P}(x' \mid x)$ is positive recurrent and aperiodic.
- We then have a "recipe" to modify "a positive recurrent and aperiodic transition matrix" in order to have the stationary distribution of our choice while just knowing it up to a normalizing constant.

A first demo

Show the demo on a toy example.

Where are we ?

What makes data "difficult"

An interlude to keep you motivated

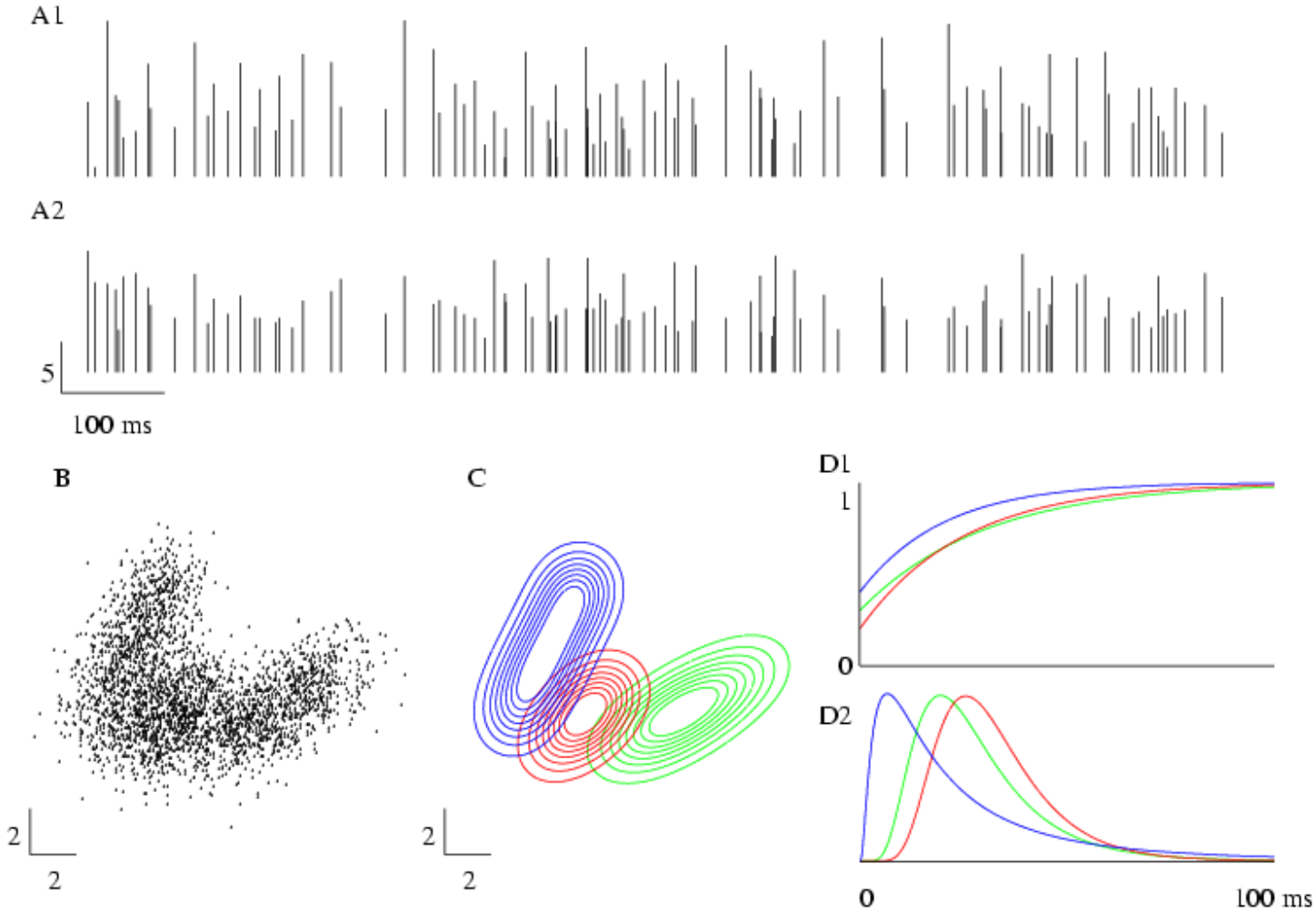
A more realistic model and its problems

A analogy with statistical physics

Back to our simulated data

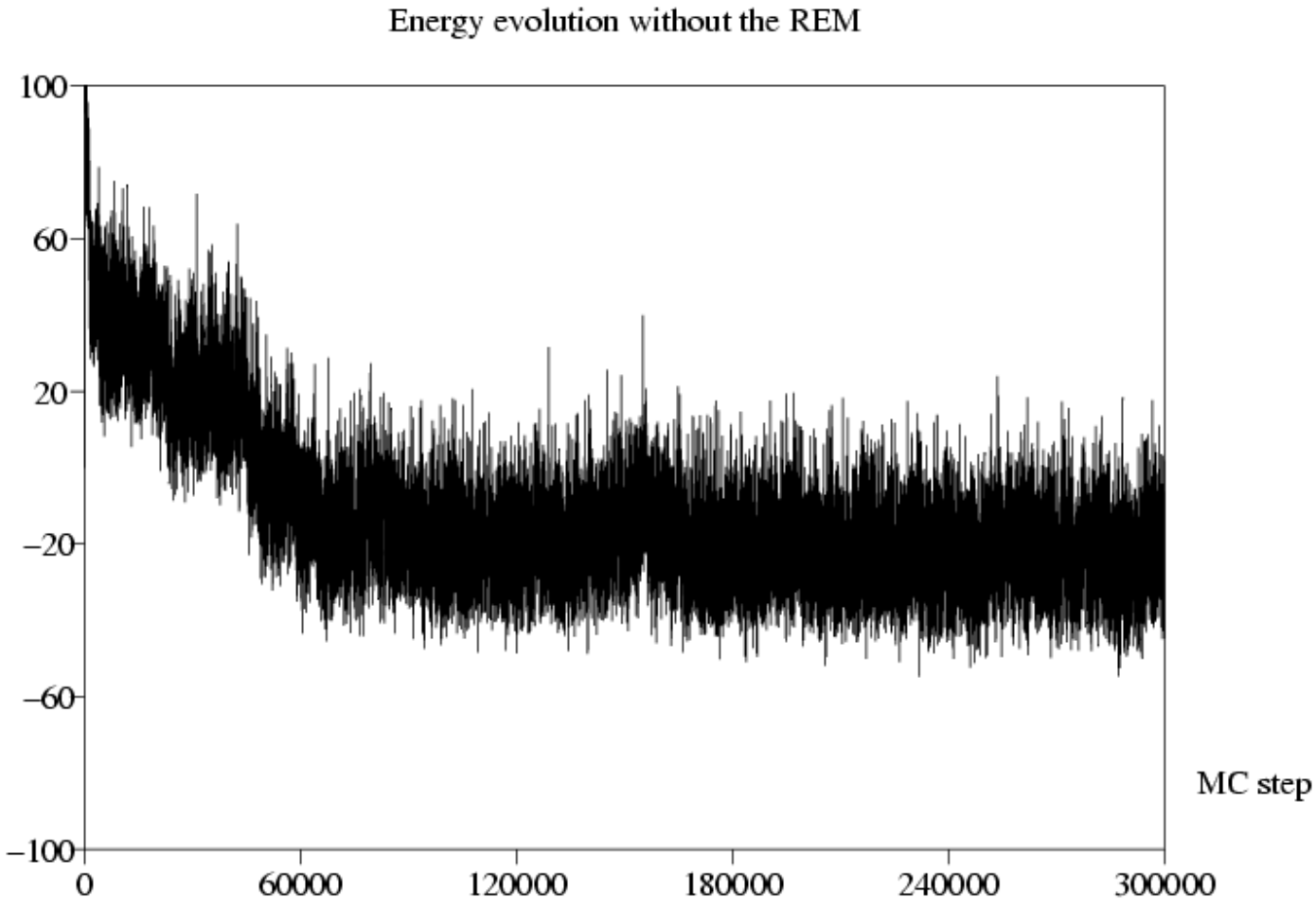
Application to real data

Back to our simulated data



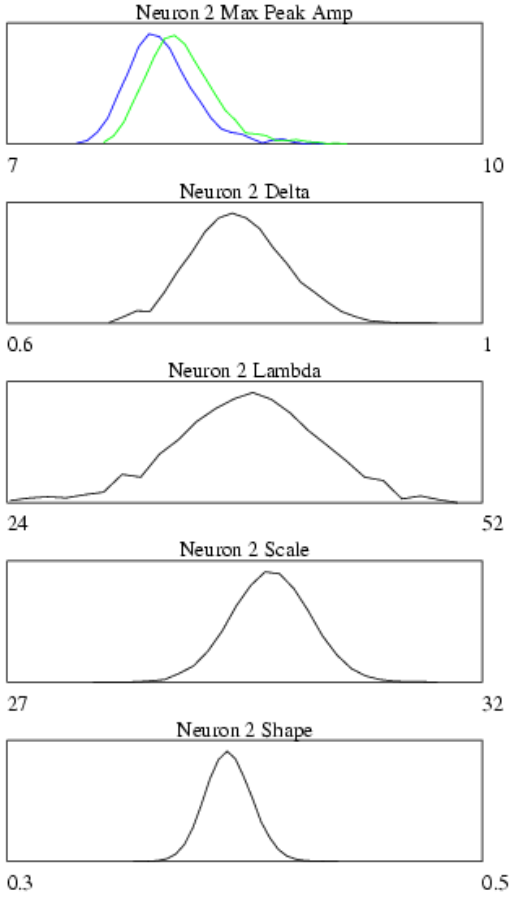
Simulated data with 3 neurones on 2 recording sites.

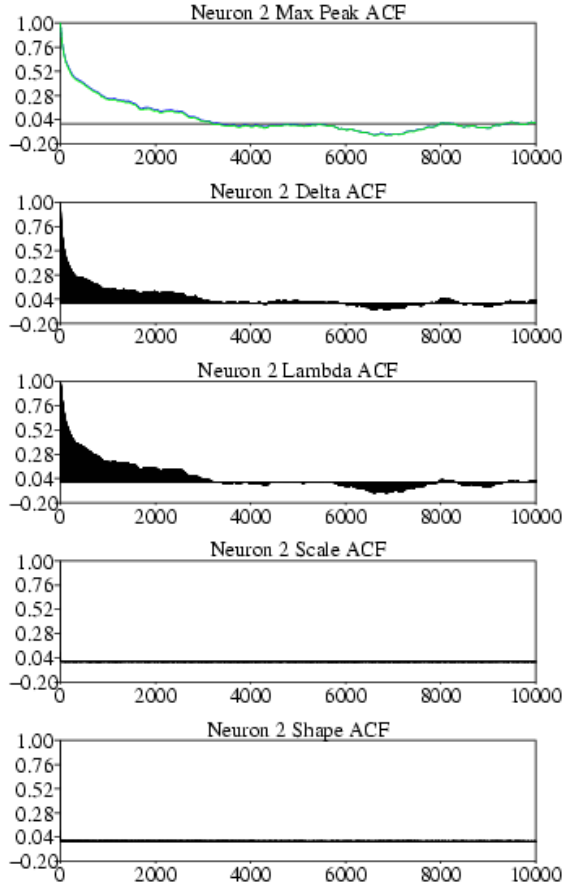
Energy evolution



Energy evolution during 3×10^5 MC steps.

Posterior densities

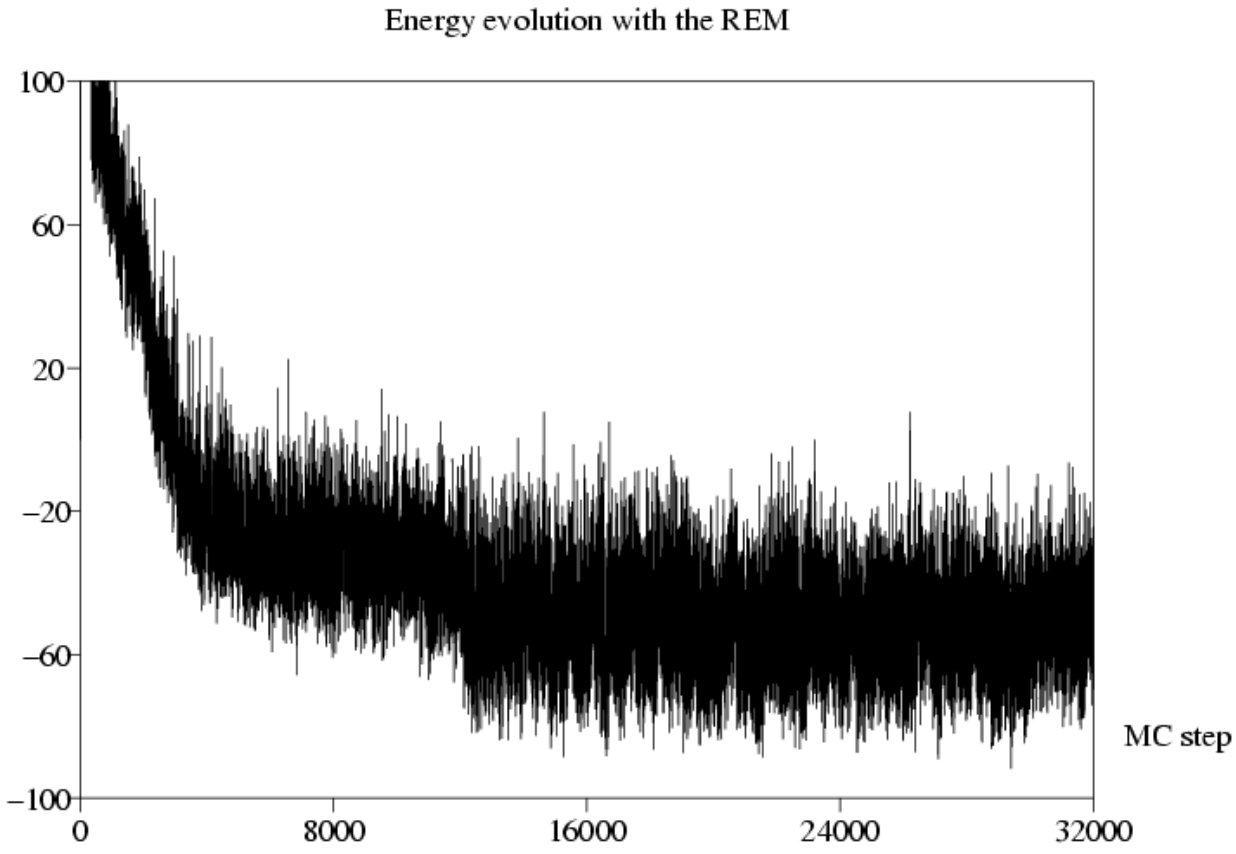




Replica Exchange Method / Parallel Tempering

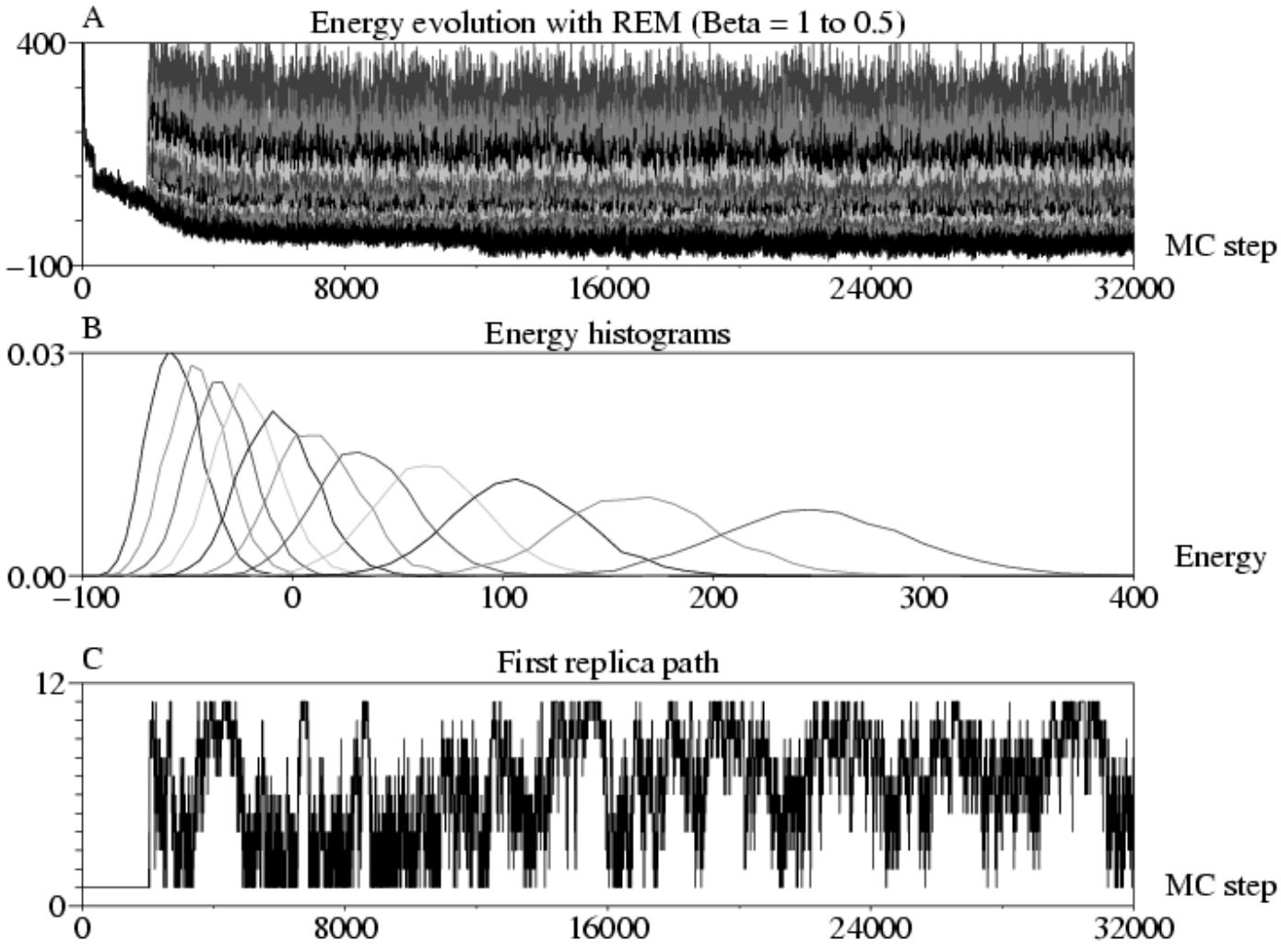
REM / Tempering demo...

Energy evolution with REM

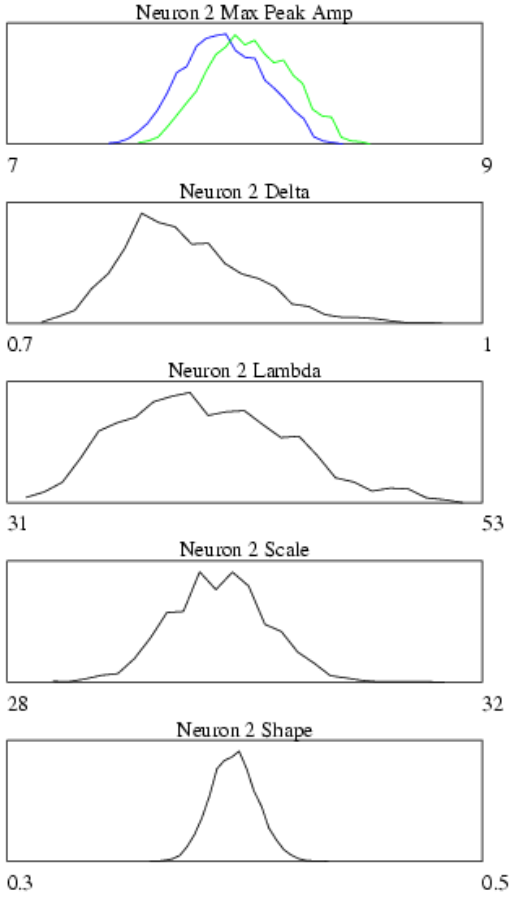


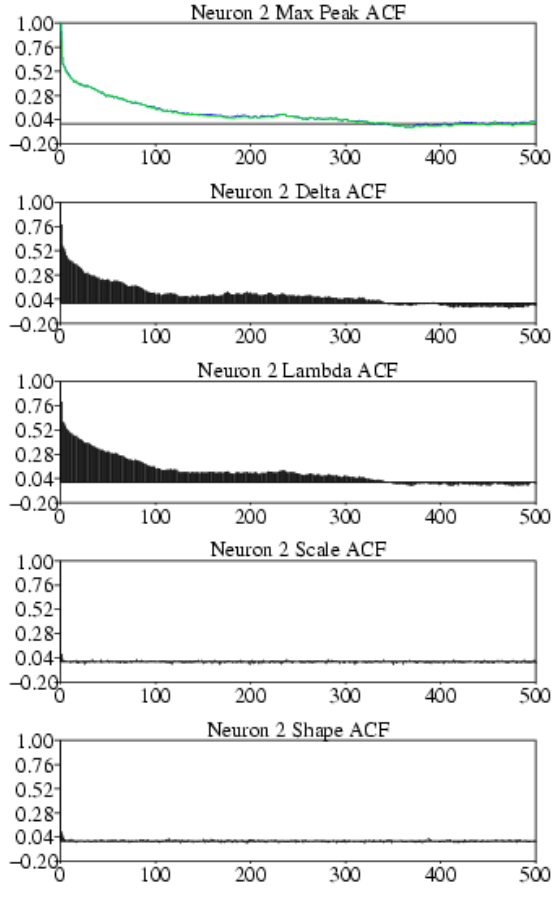
Energy evolution during 32×10^4 MC steps, REM "turned on" after 10^4 steps.

Closer look at REM

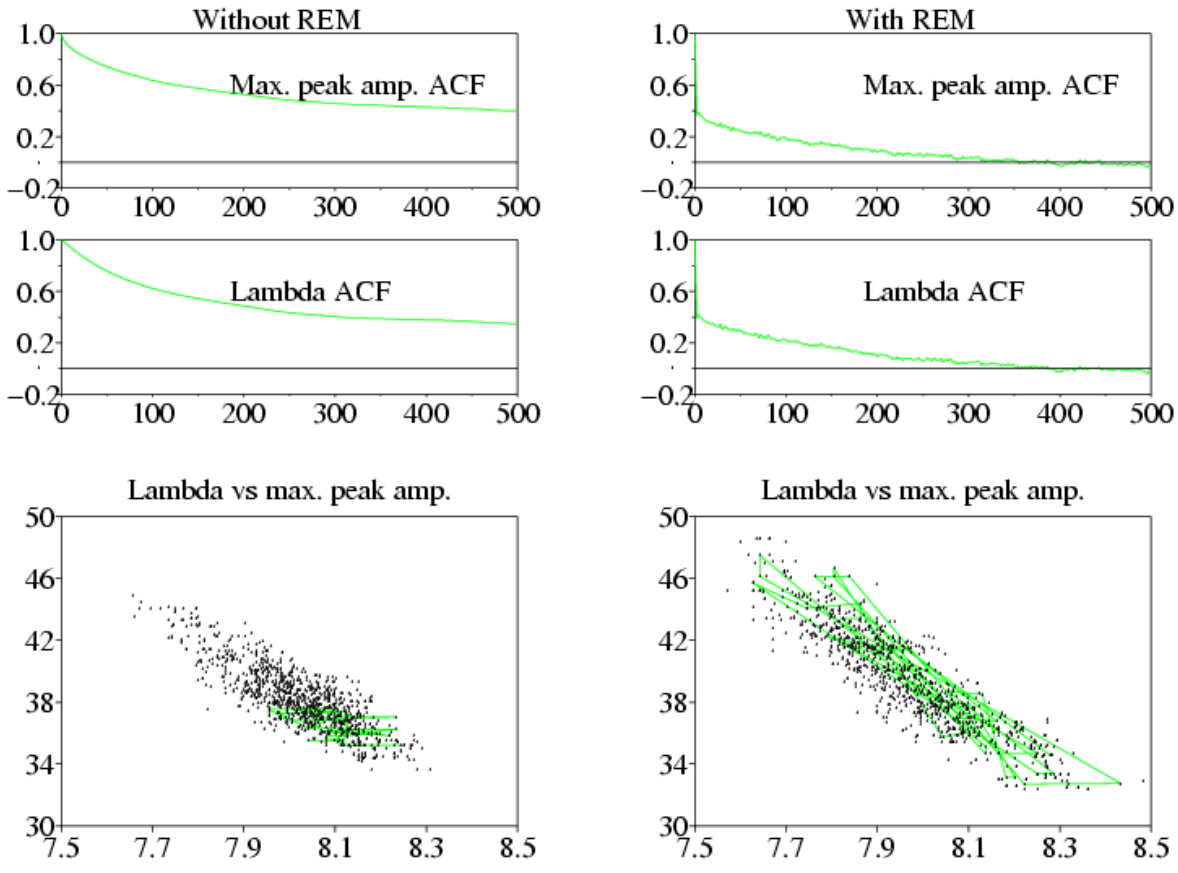


Posterior densities

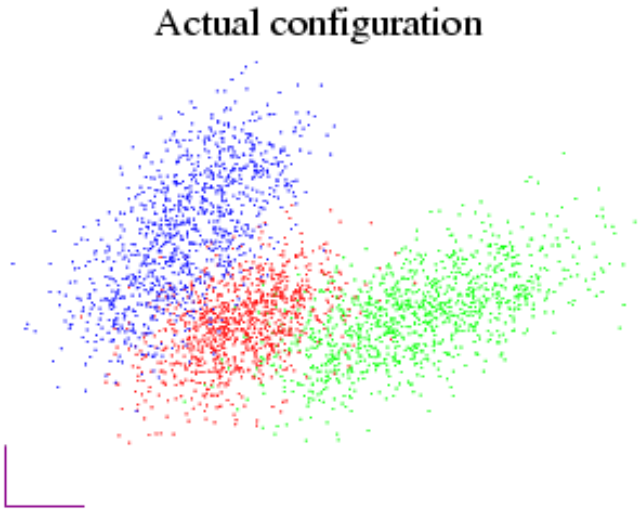


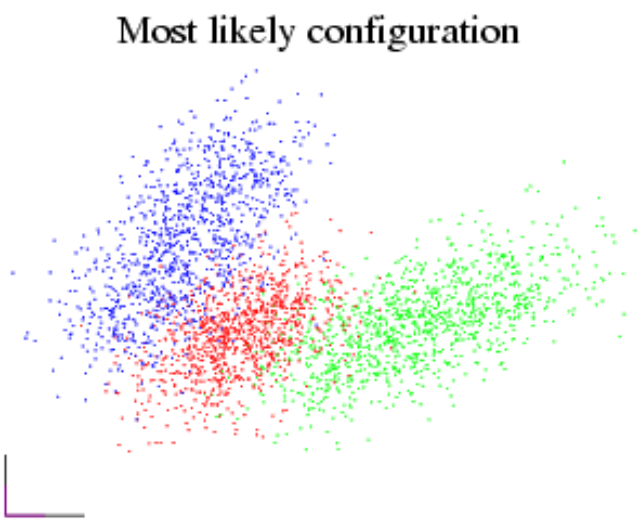


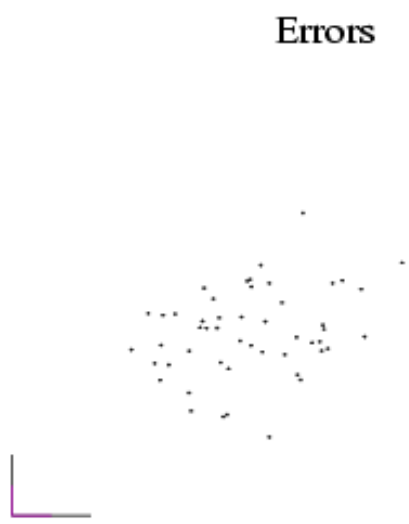
REM dynamics



Most likely configuration







Where are we ?

What makes data "difficult"

An interlude to keep you motivated

A more realistic model and its problems

A analogy with statistical physics

Back to our simulated data

Application to real data

A Hidden-Markov model for actual discharges

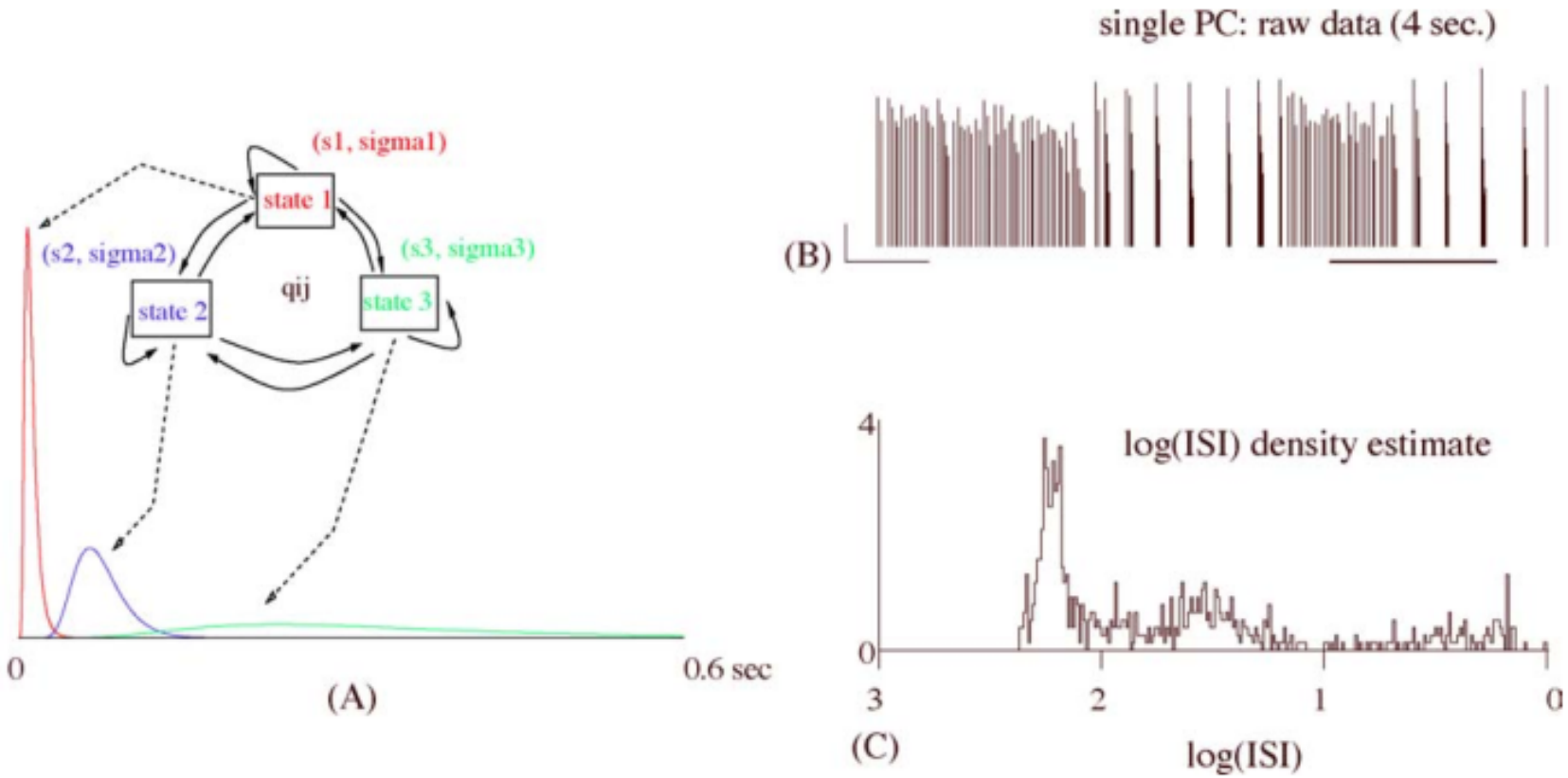


Fig. 1 of Delescluse and Pouzat (2006) *J Neurosci Methods* **150**:16.

MCMC algorithm applied to a *single* neuron discharge

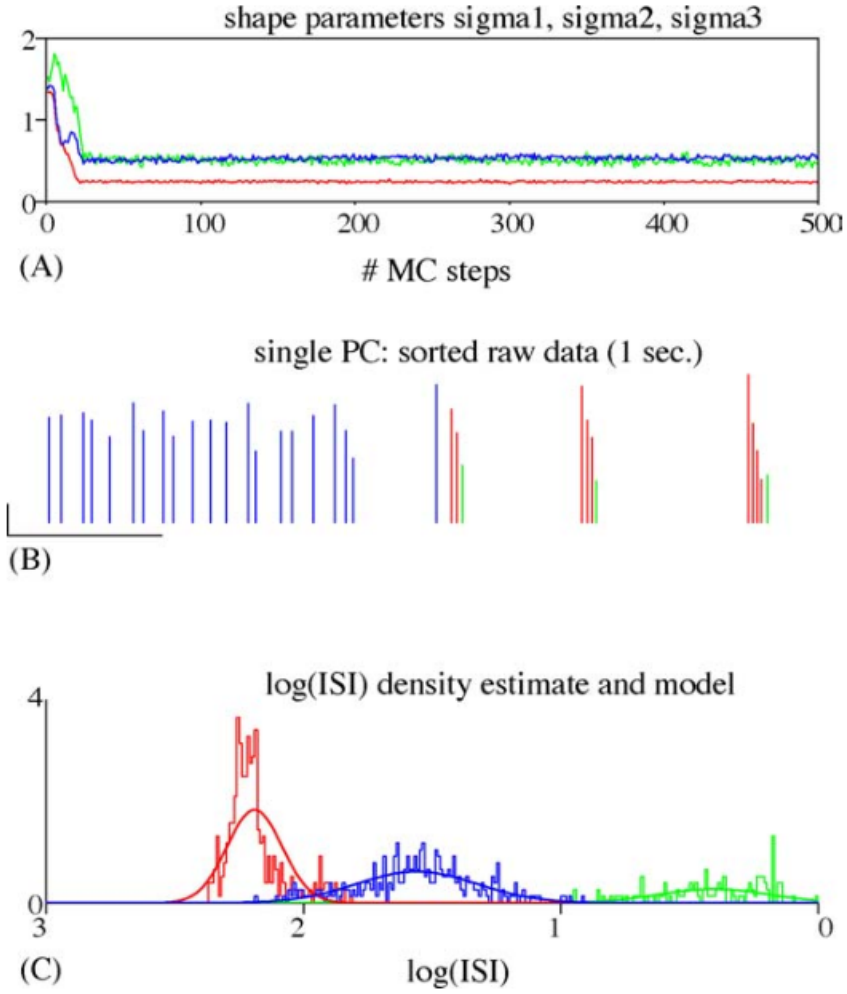


Fig. 2 of Delescluse and Pouzat (2006).

Sorting with a reference

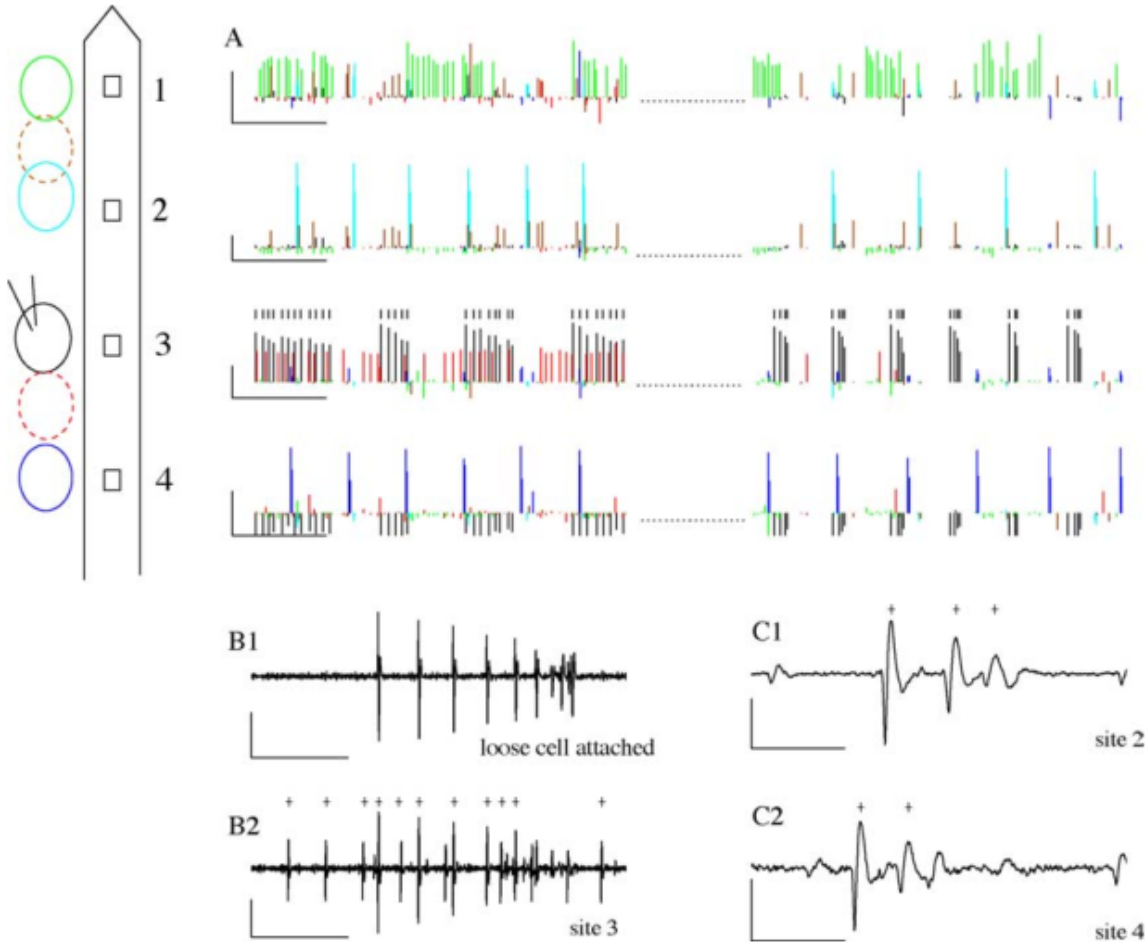


Fig. 4 of Delescluse and Pouzat (2006).

Bibliography

- Hammersley, J.M. and Handscomb, D.C. (1964) *Monte Carlo Methods*. Methuen. Must read! Buy it second hand or find it on the web.
- Liu, Jun S. (2001) *Monte Carlo Strategies in Scientific Computing*. Springer. Modern equivalent of the previous one, lots of applications in Biology.
- MacKay, David JC (2003) *Information theory, inference, and learning algorithms*. CUP. Clear and fun! Available online: `http://www.inference.phy.cam.ac.uk/mackay/itila/`.
- Neal, Radford M (1993) *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto. I learned MCMC with that. Available online: `http://www.cs.toronto.edu/~radford/papers-online.html`.

That's all for today!

I want to thank:

- Antonio Galvès for inviting me to give these lectures.
- João Alexandre Peschanski and Simone Harnik for taking care of the transmission of these lectures.
- You guys for listening!