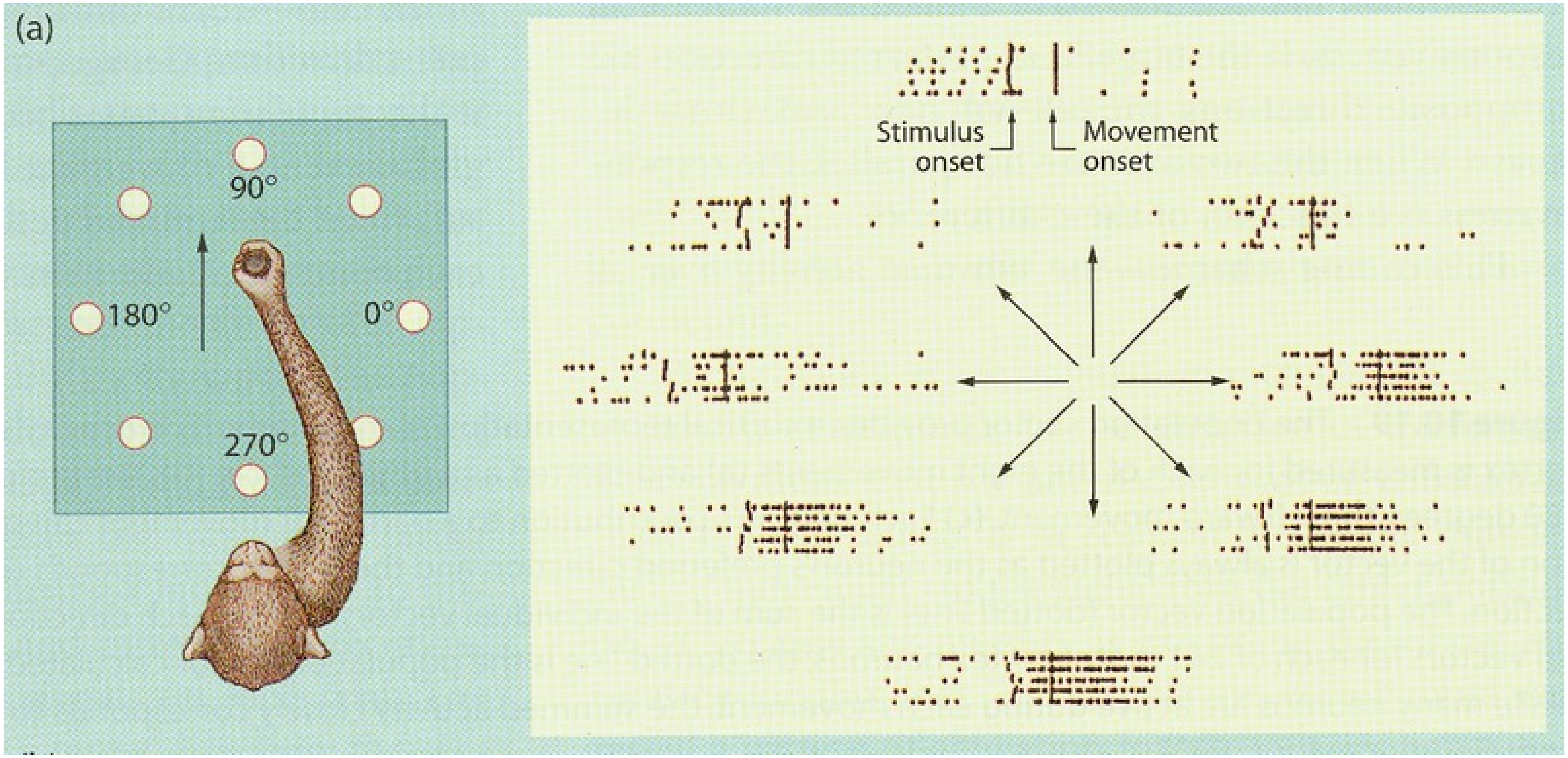
Tuning Curves
Population decoding

BME665/565

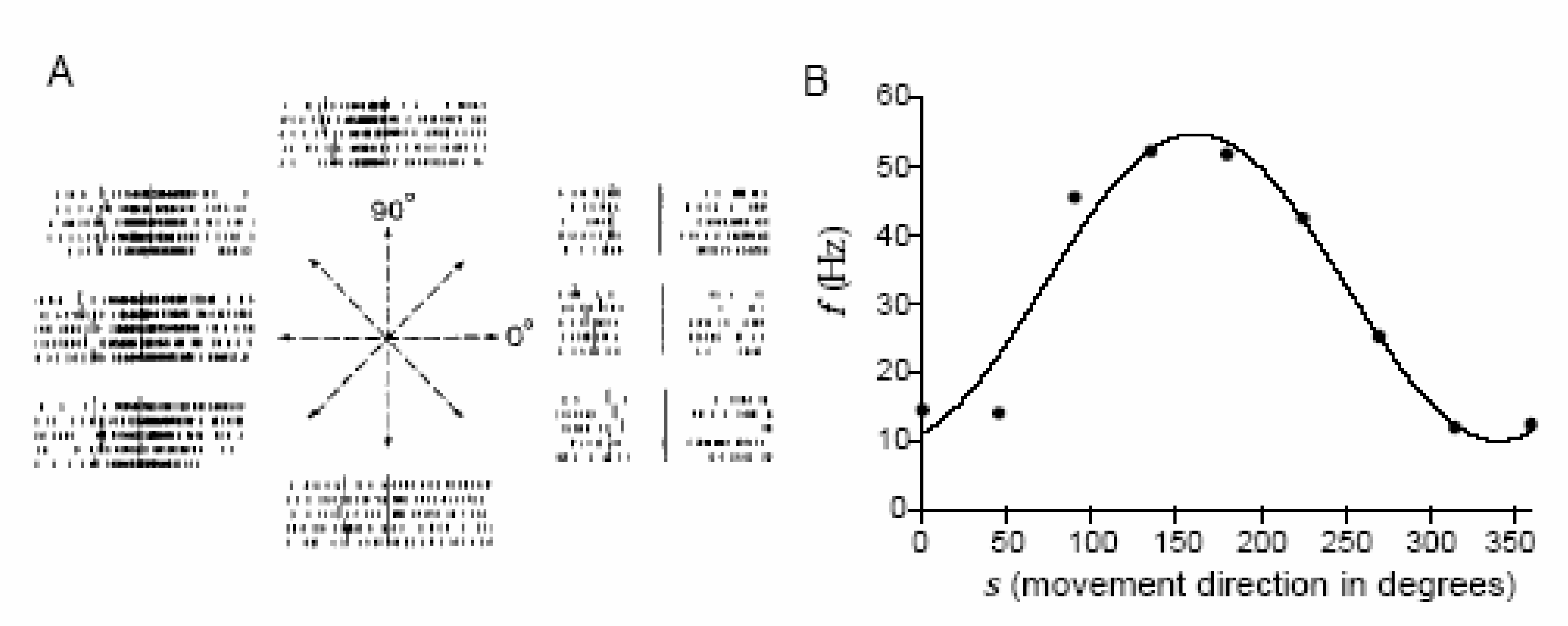
Population decoding



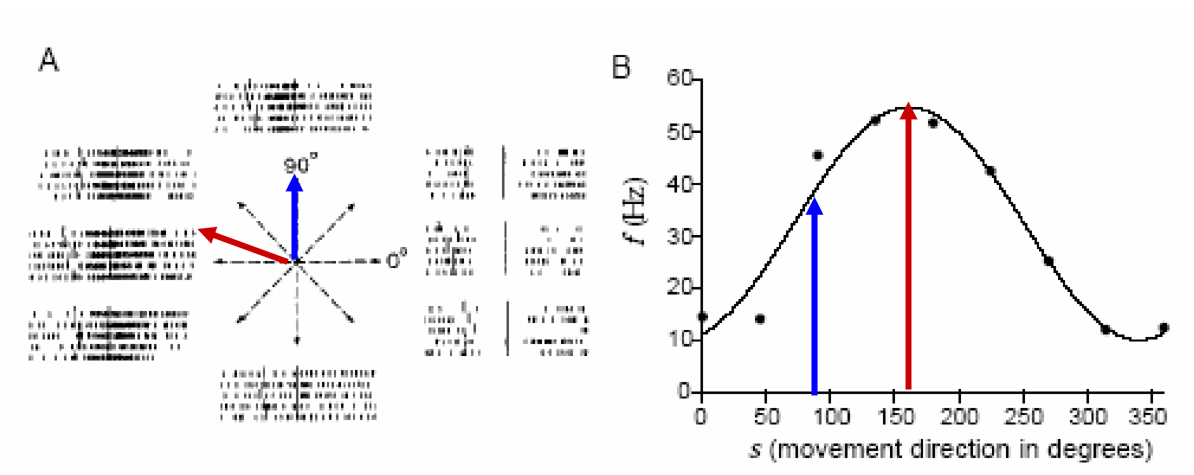
Directional sensitivity

- Motor neurons show a directional sensitivity
  - Data show a tuning curve of

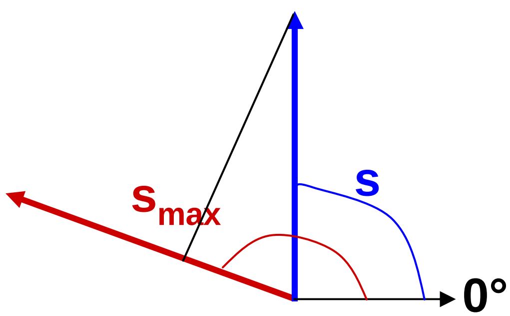
$$f(s) = r_0 + (r_{max} - r_0)\cos(s - s_{max})$$



Vector coding



$$stimulus\ \ vector = \vec{v}$$



$$preferred\ \ vector = \vec{c}_a$$

Response of neuron to non-preferred direction: projection of stimulus direction on to preferred direction for neuron:

$$\cos(s - s_{\max}) = [\vec{v} \cdot \vec{c}_a]_+$$

$$X \cdot Y = |X||Y|\cos\theta$$

Vector coding

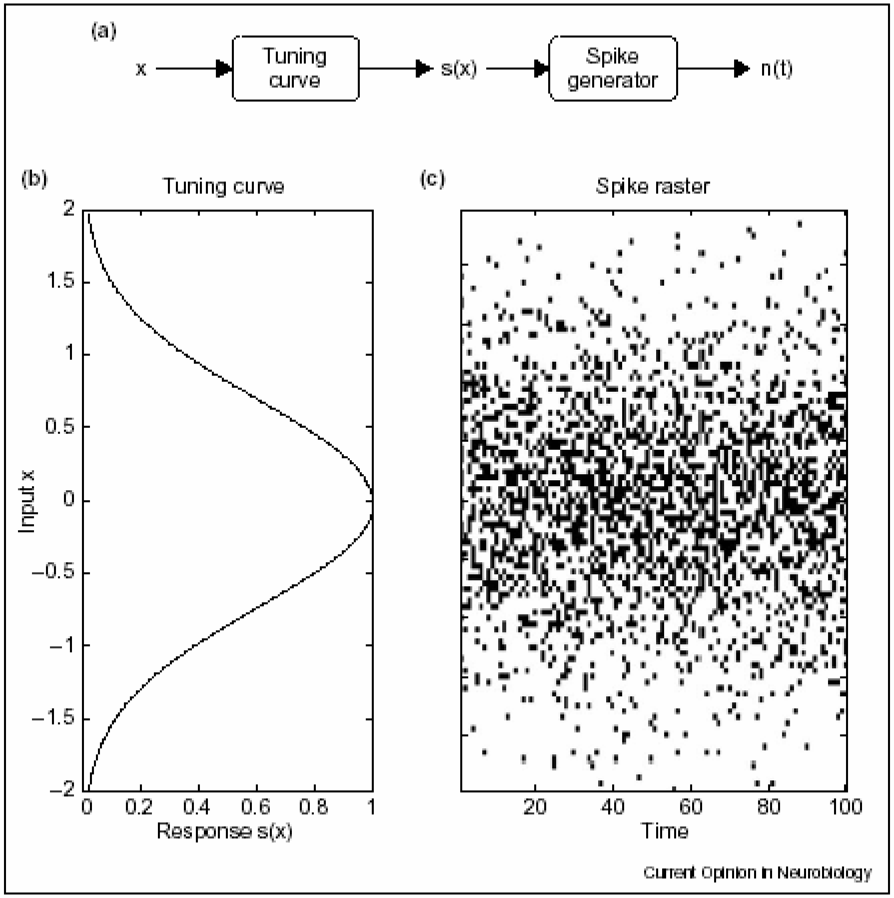
- If the average firing rate for the neuron is coded by

$$f(s) = r_0 + r_{\max}\cos(s - s_{\max})$$

- Then $\cos(s - s_{\max}) = \left[\vec{v} \cdot \vec{c}_a\right]$

- Can be written as: $\left(\dfrac{f(s) - r_0}{r_{\max}}\right)_a = \left[\vec{v} \cdot \vec{c}_a\right]$

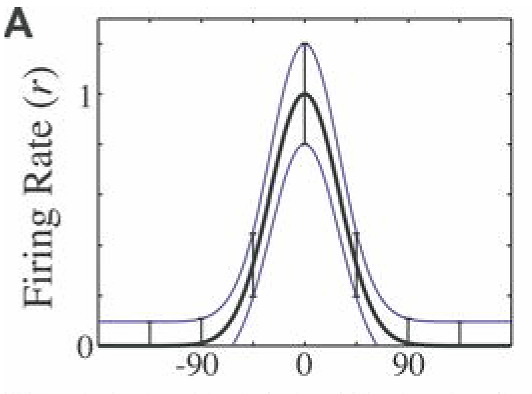
Tuning curves are not precise



Tuning curve gives the response of a single neuron as a stimulus-dependent firing rate

So, how does this code for precise position?

- Every neuron has a preferred direction, but…
  - Tuning curves are broad
  - Neuronal responses are noisy
  - How do we determine the precise direction that is encoded?



- How about the average response of a population of neurons?

Tovey et al., ***PLoS Biology,*** *4(4):* 639-646, 2006

Vector (de)coding

- If the single-neuron response tuning curve is represented as:

$$\left(\frac{f(s)-r_0}{r_{\max}}\right)_a = [\vec{v}\cdot\vec{c}_a]$$

- Then if we assume that for a given population of neurons the preferred directions point uniformly in all directions, then for a large number of neurons N

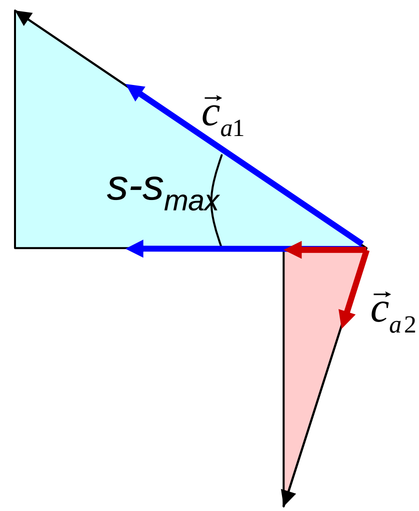
$$\vec{v}_{pop} = \sum_{a=1}^{N}\left(\frac{r-r_0}{r_{\max}}\right)_a \vec{c}_a$$

- Here, *r* is the instantaneous firing rate

Vector (de)coding

- The average population vector (over trials), given $\langle r \rangle = f(s)$ will be:

$$\langle \vec{v}_{pop} \rangle = \sum_{a=1}^{N} \left(\frac{f(s) - r_0}{r_{\max}} \right)_a \vec{c}_a = \sum_{a=1}^{N} (\vec{v} \cdot \vec{c}_a) \vec{c}_a$$



- Note that this assumes:
  - A large number of neurons
  - Preferred-direction vectors that point randomly in all directions with equal probability
  - Then, $\langle \vec{v}_{pop} \rangle$ approximates the actual stimulus direction $\vec{v}$

The neuron's view

$$P[s \mid r]$$ ⟶ How do we find an optimal *s*?

Conditional firing rate

$$P[s \mid r] = \frac{P[r \mid s]P[s]}{P[r]}$$

Probability of stimulus (the *prior*)

Probability of response *r*

Optimizing our estimate of s

- Using the homogeneous Poisson model to describe the variability of our firing rate
  - If the firing rate r_a is determined by counting n_a spikes in time T, and the average firing rate is

$$\langle r_a \rangle = f_a(s) \qquad \left(r_a = n_a / T \right)$$

  - Assume the neuron's firing statistics are described by a Poisson process – then r_a is distributed with mean of $f_a(s)$

  - The likelihood of this neuron firing in response to the stimulus s is $P[r_a \mid s]$:

$$P[r_a \mid s] = \frac{(f_a(s)T)^{n_a}}{(n_a)!} e^{-f_a(s)T}$$

Remember: Poisson firing

$$P_T(n) = \frac{(rT)^n}{n!} e^{-rT}$$

Optimizing s_{est} (cont.)

- If each neuron fires independently, the firing-rate probability for the population is

$$P[r \mid s] = \prod_{a=1}^{n} \frac{(f_a(s)T)^{r_a T}}{(r_a T)!} e^{-f_a(s)T}$$

- This is easier to work with as a *log likelihood*:

$$\log P[r \mid s] = T \sum_{a=1}^{N} r_a \log(f_a(s)) + T \sum_{a=1}^{N} r_a \log(T) - \sum_{a=1}^{N} f_a(s)T + \sum_{a=1}^{N} \log((r_a T)!)$$

Independent
of the stimulus

Optimizing s_{est} (cont.)

$$\log P[r \mid s] = T \sum_{a=1}^{N} r_a \log(f_a(s)) - \sum_{a=1}^{N} f_a(s) T$$

If all responses equally likely over the population, sums to constant

Clearly our tuning curve should match our data – however, often it is acceptable to take a tuning curve that is Gaussian:

$$f_a(s) = r_{\max} \exp\left(-\frac{1}{2} \left(\frac{s - s_a}{\sigma_a} \right)^2 \right)$$

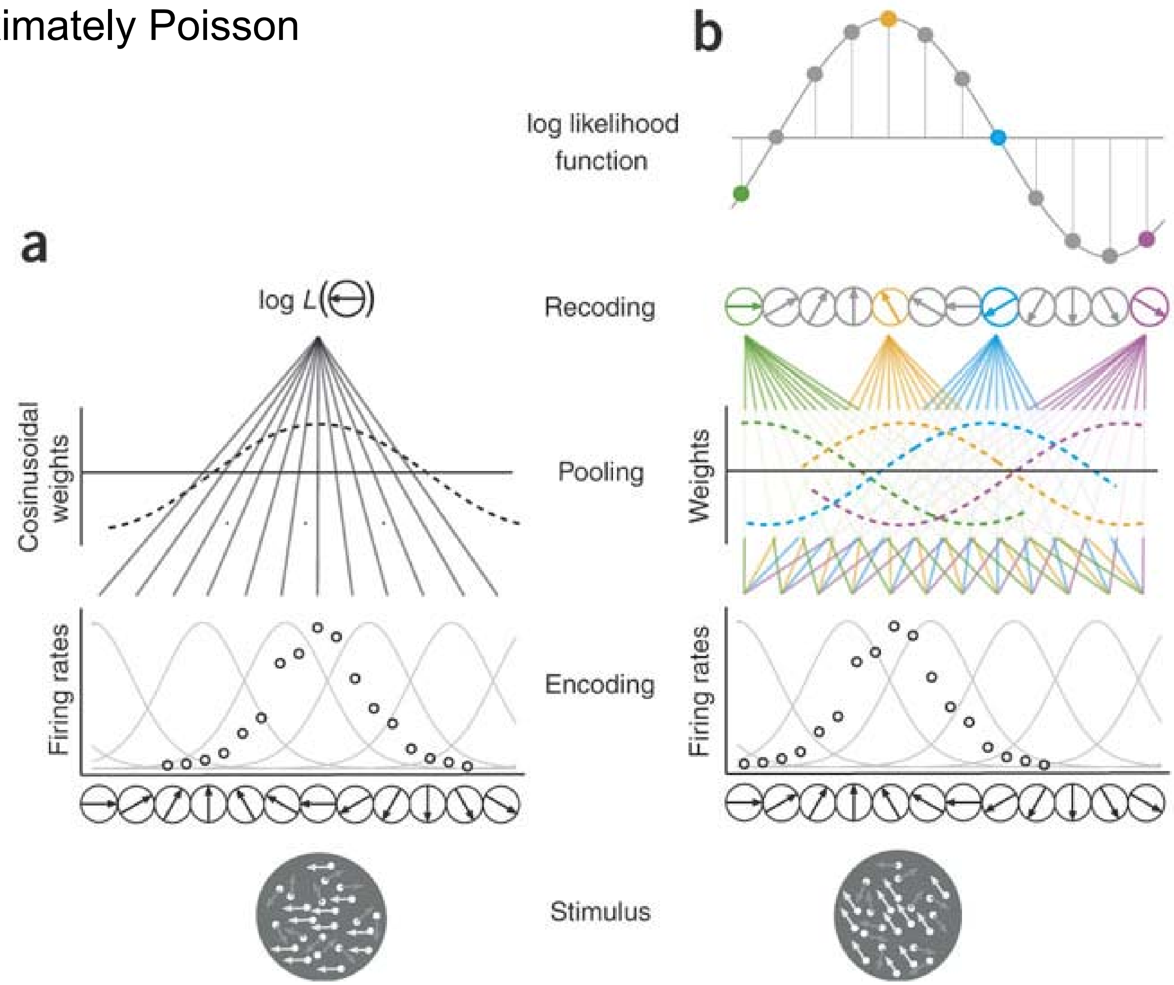
In which case $\sum_{a=1}^{N} f_a(s)$ will be independent of s

Does not assume any particular tuning curve
Does assume Poisson firing
Assumes neurons are statistically independent

Example: directionally tuned neurons in MT

Bell-shaped tuning functions
Kappa determines the tuning bandwidth
Cells increase their firing rate for favored directions roughly in proportion to coherence
Their firing statistics are approximately Poisson

$$\log P[r \mid s] = \kappa \sum_{a=1}^{N} n_i \cos(s - s_i)$$



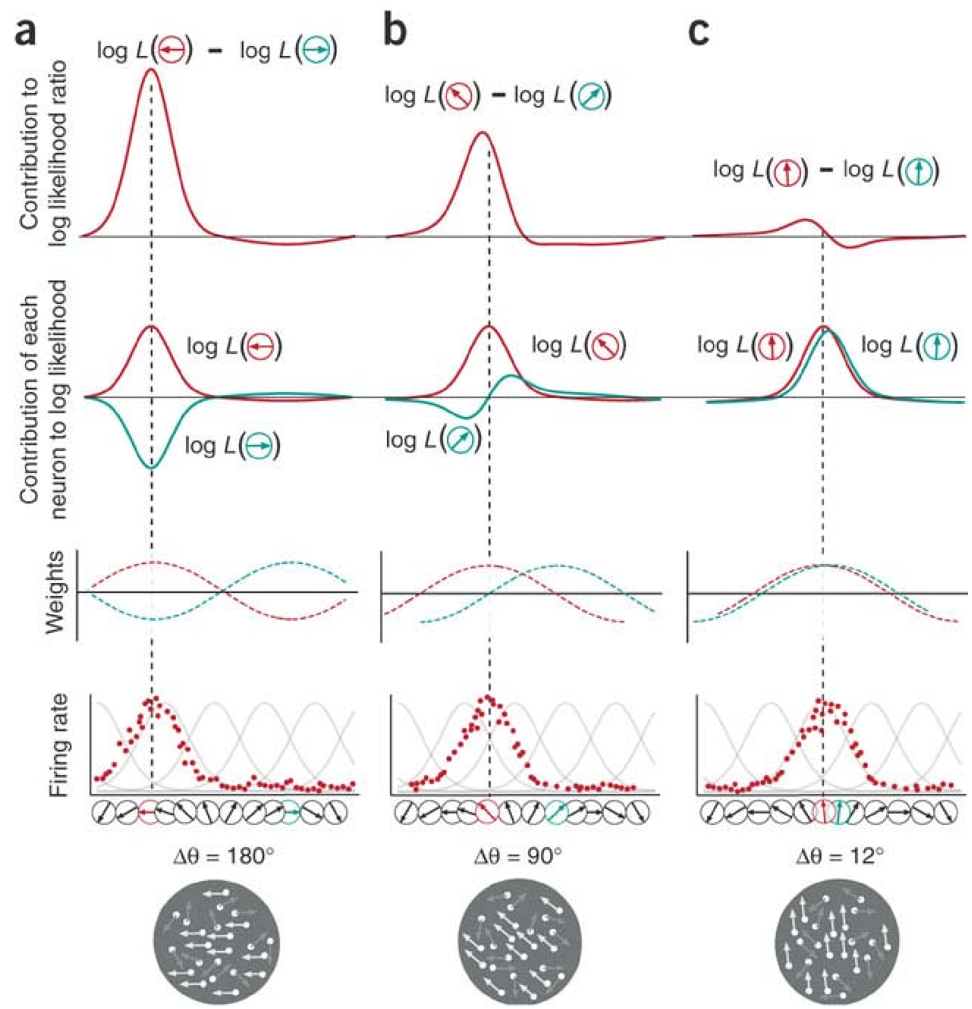
Jazayeri et al., ***Nature Neuroscience*** 9(5), 690-696 (2006)

Discrimination behavior:
Contributions of individual neurons to Log Likelihood

$$\log P[r \mid s_1] - \log P[r \mid s_2]$$

$$= \kappa \sum_{a=1}^{N} n_i \left(\cos(s_1 - s_i) - \cos(s_2 - s_i)\right)$$

Neurons with similar weights in each of the log likelihoods cancel and do not contribute strongly to the discrimination, whereas neurons with more dissimilar weights in the two log likelihoods have a stronger influence on the model's discrimination behavior.



Jazayeri et al., ***Nature Neuroscience*** 9(5), 690-696 (2006)

The neuron's view

$$P[s \mid r]$$ → How do we find an optimal *s*?

Conditional firing rate → $P[r \mid s]$

Probability of stimulus (the *prior*) → $P[s]$

Probability of response *r* → $P[r]$

$$P[s \mid r] = \frac{P[r \mid s]P[s]}{P[r]}$$

Optimizing s_{est} (cont.)

- If we further assume that *P[s]* is independent of s, then we can find a stimulus value s_{ML} that maximizes our conditional probability stimulus:

$$P[s \mid r] = \frac{P[r \mid s]P[s]}{P[r]}$$

- The Maximum Likelihood estimated stimulus maximizes:

$$\log P[s_{ML} \mid r] = \log P[r \mid s] + \log P[s] - \log P[r]$$

Optimizing s_{est} (cont.)

- Remember:

$$\log P[r \mid s] = T\sum_{a=1}^{N} r_a \log(f_a(s)) - \sum_{a=1}^{N} f_a(s)T$$

- So,

$$\log P[s_{ML} \mid r] = \log P[r \mid s] + \log P[s] - \log P[r]$$
$$= T\sum_{a=1}^{N} r_a \log(f_a(s)) + \log P[s] - \log P[r]$$

- To find the optimum with respect to the stimulus, we set the derivative with respect to *s* to zero

Optimizing s_{est} (cont.)

When is this true?

$$\frac{d}{ds}\left(T\sum_{a=1}^{N} r_a \log\left(f_a(s)\right) + \log P[s] - \log P[r]\right) = 0$$

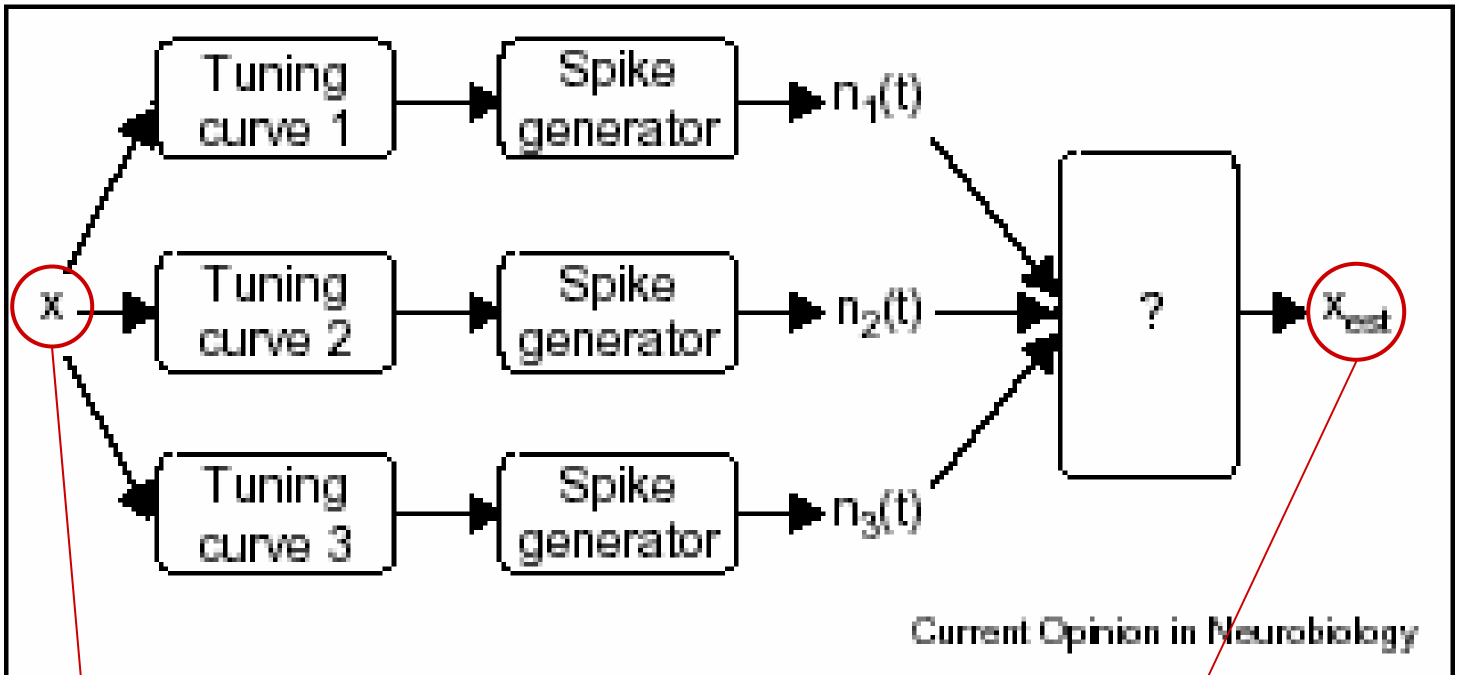
- If *P[s]* is independent of *s*, then setting the derivative with respect to *s* to 0 gives us:

$$\frac{d}{ds}\left(T\sum_{a=1}^{N} r_a \log\left(f_a(s)\right)\right) = 0$$

$$\sum_{a=1}^{N} r_a \frac{f_a'(s_{ML})}{f_a(s_{ML})} = 0$$

- For our Gaussian tuning curve $f_a(s) = r_{\max} \exp\left(-\frac{1}{2}\left(\frac{s - s_a}{\sigma_a}\right)^2\right)$

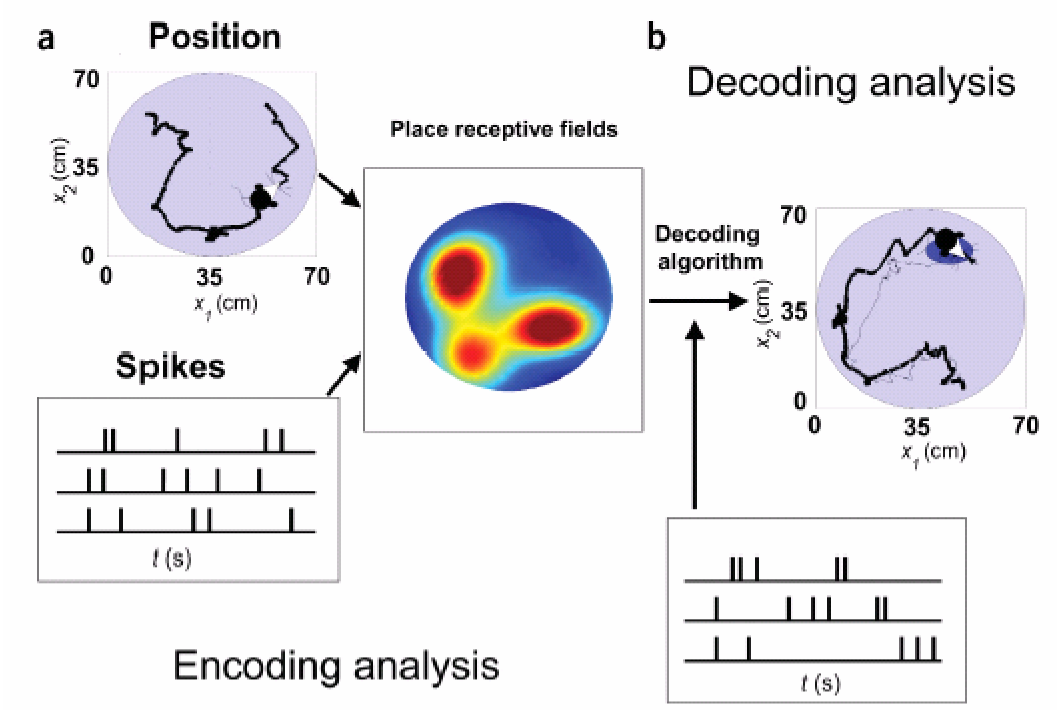
$$s_{ML} = \frac{\sum r_a s_a / \sigma_a^2}{\sum r_a / \sigma_a^2}$$



$s = actual\ \ stimulus$

$$s_{ML} = \frac{\sum r_a s_a / \sigma_a^2}{\sum r_a / \sigma_a^2}$$

Example: Hippocampal place cells



Brown et al., ***Nature Neuroscience*** 7, 456 - 461 (2004)

What if *P[s]* is not independent of *s*?

- Use the same approach, with some estimate of the distribution of the probability of the prior (i.e. the stimulus)

$$T\sum_{a=1}^{N} r_a \log\left(f_a(s)\right) + \log P\left[s\right] - \log P\left[r\right]$$

- By setting:

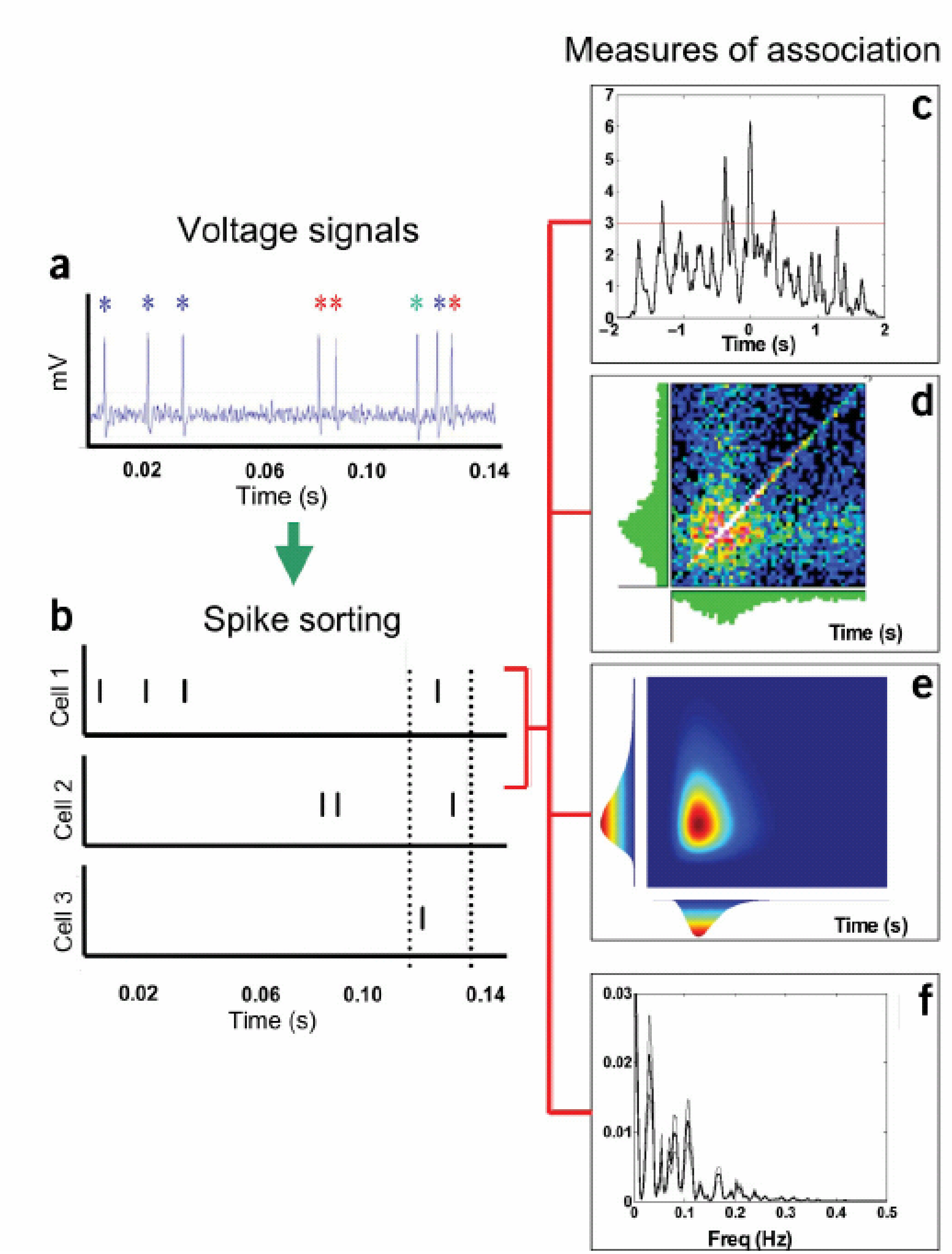
$$\frac{d}{ds}\left(T\sum_{a=1}^{N} r_a \log\left(f_a(s)\right) + \log(P\left[s\right])\right) = 0$$

Maximum a posteriori (MAP) inference

- Using our friend the Gaussian to represent the distribution of the prior (with mean s_{prior} and variance σ_{prior}) we can again solve for s_{MAP}:

$$T\sum_{a=1}^{N} r_a \frac{f_a'(s_{MAP})}{f_a(s_{MAP})} + \frac{P_a'[s_{MAP}]}{P_a[s_{MAP}]} = 0$$

$$s_{MAP} = \frac{T\sum r_a s_a / \sigma_a^2 + s_{prior} / \sigma_{prior}^2}{T\sum r_a / \sigma_a^2 + 1/\sigma_{prior}^2}$$



Cross-correlogram

Joint PSTH

Maximum likelihood estimation

Cross-coherence

Brown et al., ***Nature Neuroscience*** 7, 456 - 461 (2004)

How good is our estimate?

- We can evaluate how good our estimate is using two measures:
  - Bias = difference between the average estimated response (over trials) and the true stimulus

$$b_{est}(s) = \langle s_{est} \rangle - s$$

  - Variance = how much the estimate varies about its mean value

$$\sigma_{est}^2(s) = \left\langle \left(s_{est} - \langle s_{est} \rangle \right)^2 \right\rangle$$