1B – Parameter fitting & recovery

Klaus Wimmer

Centre de Recerca Matemàtica

Barcelona, Spain

kwimmer@crm.cat

BAMB! 2023 Summer School

The plan for the next 120 minutes

- Parameter estimation
  - Model fitting as an optimization problem
- Model validation (tutorial part 1)
  - Does the model capture the data?
- Parameter recovery (tutorial part 2)
  - Simulate, simulate, simulate!
- Parameter uncertainty (tutorial part 3)
  - Bootstrapping

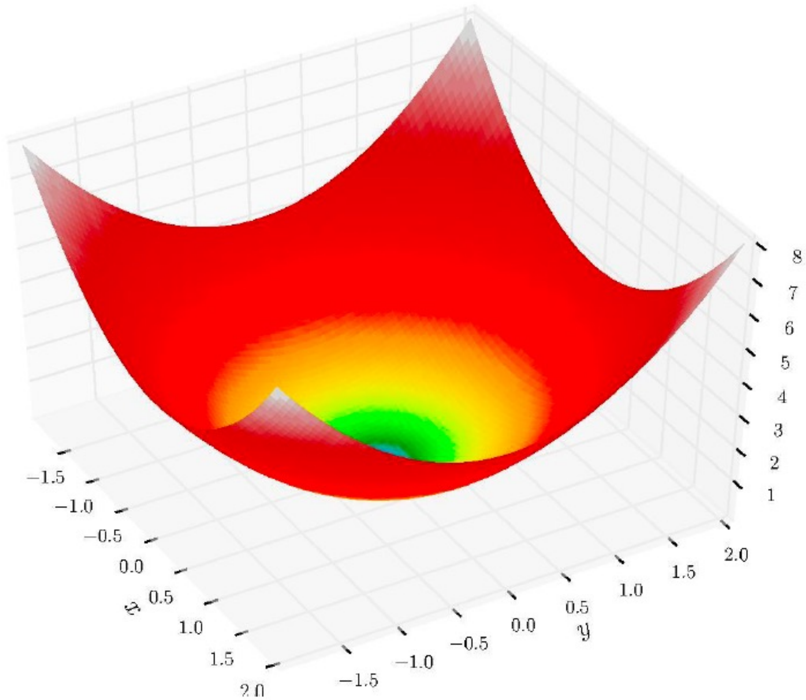
Fit the parameters

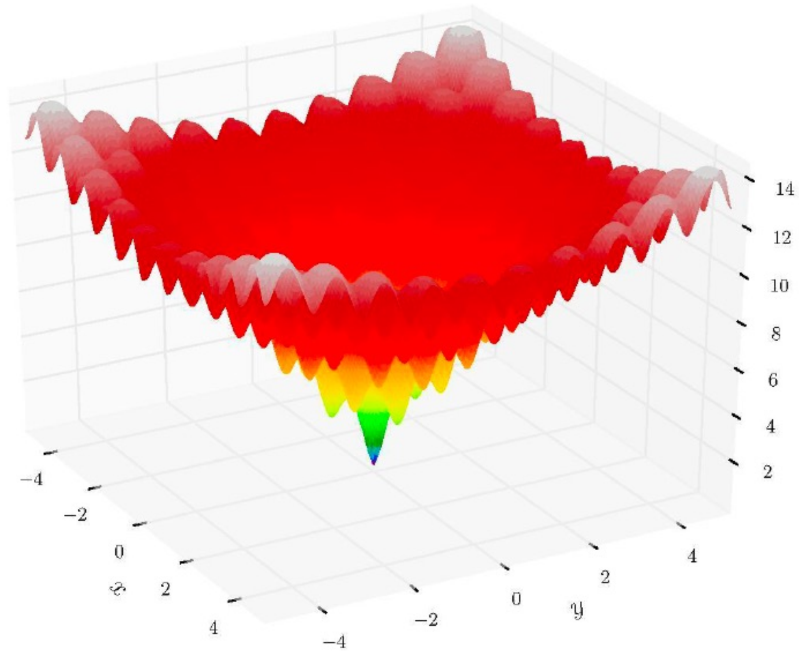
Maximum likelihood estimation: $$\hat{\boldsymbol{\theta}}_{ML} = \ arg\max_{\boldsymbol{\theta}} LL(\boldsymbol{\theta})$$

Model fitting via point estimation

- Goal: find x_{opt} = argmin f(x) as fast as possible
- Often f(x) is a black box; sometimes we can compute the gradient
- Solution: feed f(x) to an optimization algorithm

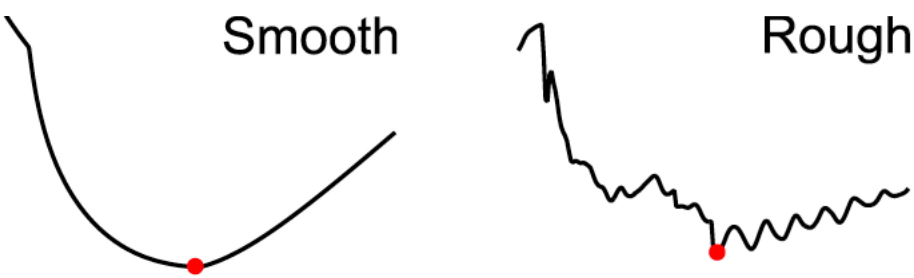
How hard can it be?





Optimization can be hard

- Multiple local minima or saddle points
- Expensive function evaluation (>> 1 s)
- Noisy function evaluation (stochastic problem)
- Rough landscape (numerical approximations, etc.)



Optimization algorithms

Gradient-based methods

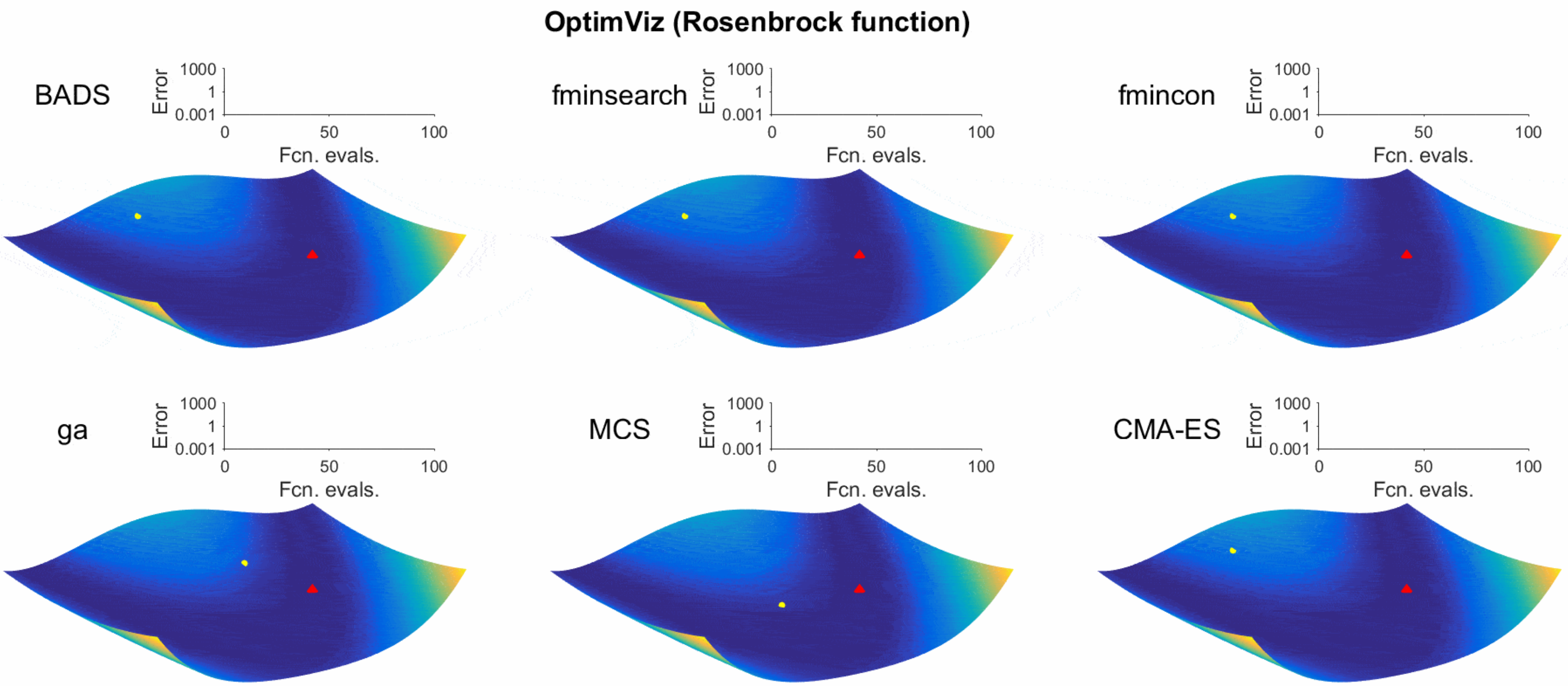
- Stochastic gradient descent (e.g., ADAM)
- Quasi-Newton methods (e.g., BFGS aka `fminunc`/`fmincon`)

Gradient-free methods

- Grid/random search
- Nelder-Mead (`fminsearch`)
- Pattern/direct search (`patternsearch`)
- Simulated annealing
- Genetic algorithms
- CMA-ES
- Bayesian optimization
- Bayesian Adaptive Direct Search (BADS; Acerbi & Ma, *NeurIPS* 2017)

See intro to optimization by Luigi Acerbi in BAMB! 2022:
https://github.com/lacerbi/bamb2022-model-fitting

Optimizers at work

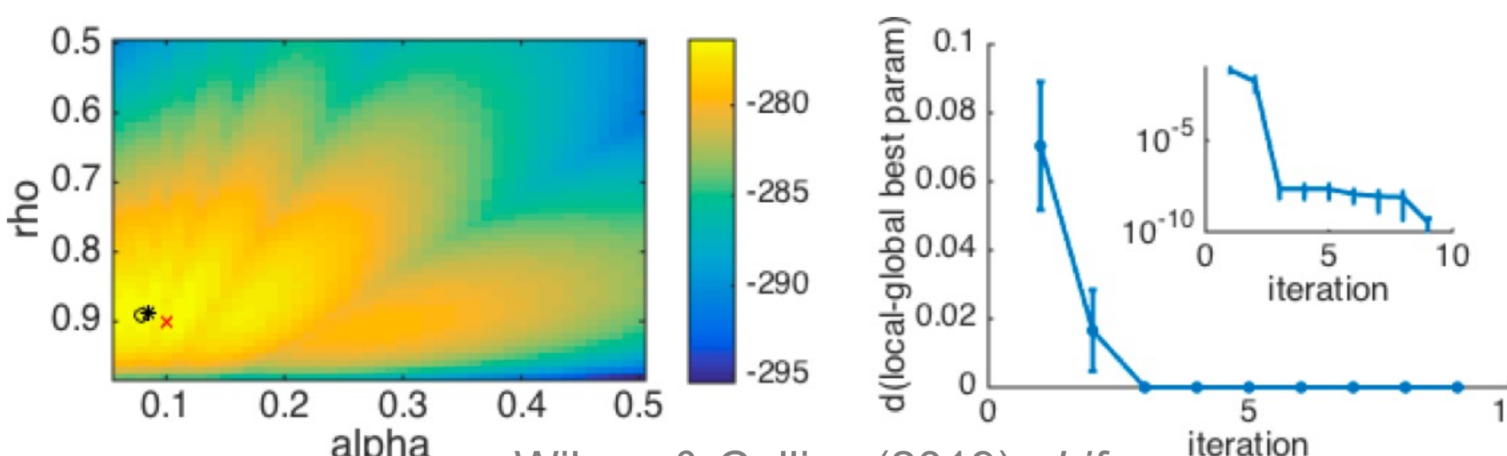


BADS: Acerbi and Ma (2017)

https://github.com/lacerbi/bads
https://github.com/lacerbi/optimviz

Tips and tricks

- Make sure that your log-likelihoods are finite (e.g. initial conditions)
- Carefully choose constraints on parameters: *hard* bounds and *plausible* bounds (avoid solutions at the bounds)
- Consider reparameterization (independence between parameters)
- Beware of local minima (always run fitting multiple times with random initial conditions)



Wilson & Collins (2019) *eLife*

The plan for the next 120 minutes

- Parameter estimation
  - Model fitting as an optimization problem
- **Model validation (tutorial part 1)**
  - **Does the model capture the data?**
- Parameter recovery (tutorial part 2)
  - Simulate, simulate, simulate!
- Parameter uncertainty (tutorial part 3)
  - Bootstrapping

Model validation

- Does the model capture the data in an absolute sense?
- Does the model get the essence of the behavior?

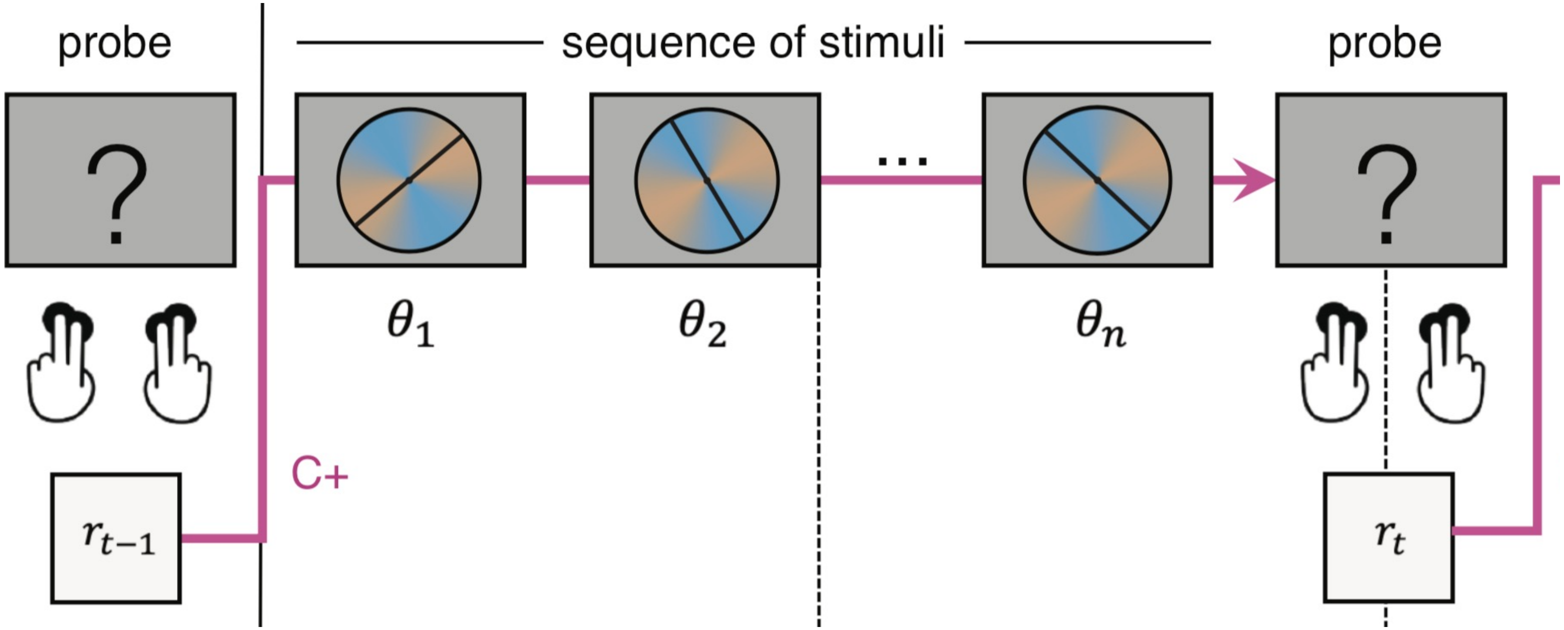
How to validate the model?

- Simulate the model with the fit parameter values and then analyze the simulated data in the same way that you analyzed the empirical data (model-independent analysis).

- Verify that all important behavioral effects are captured by the model

Example: model validation

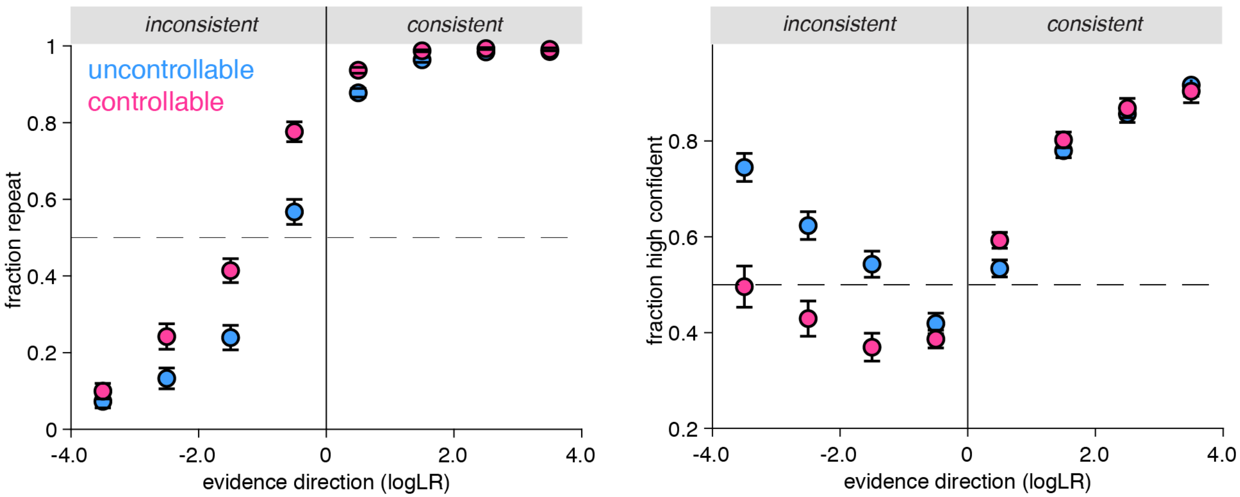
- Decision-making task under uncertainty
- Bayesian inference model plus different sources of noise



Rouault et al, *eLife* (2022)

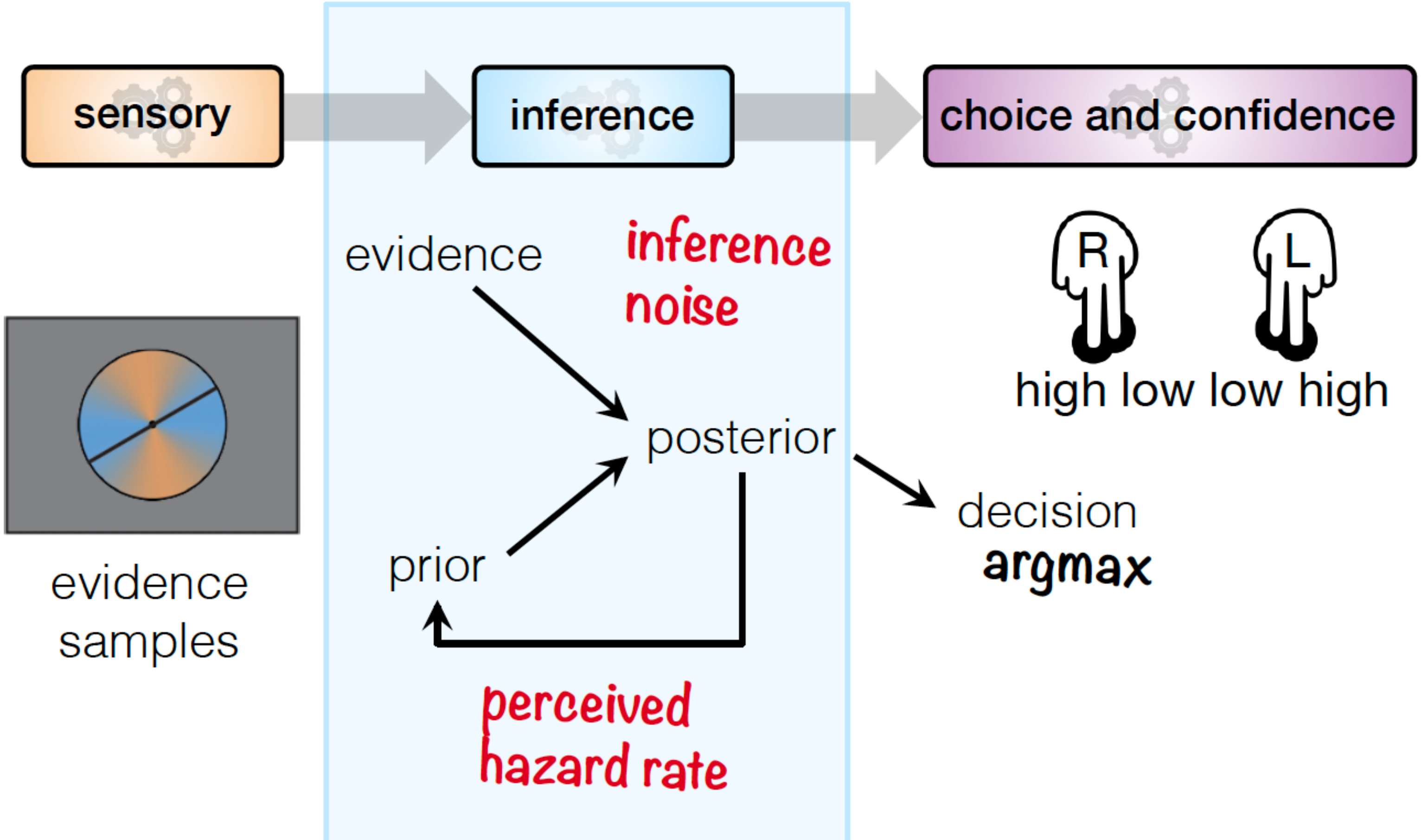
Example model validation

- Participants need more evidence to change their mind in controllable environments
- When they change their mind, they do so with reduced confidence



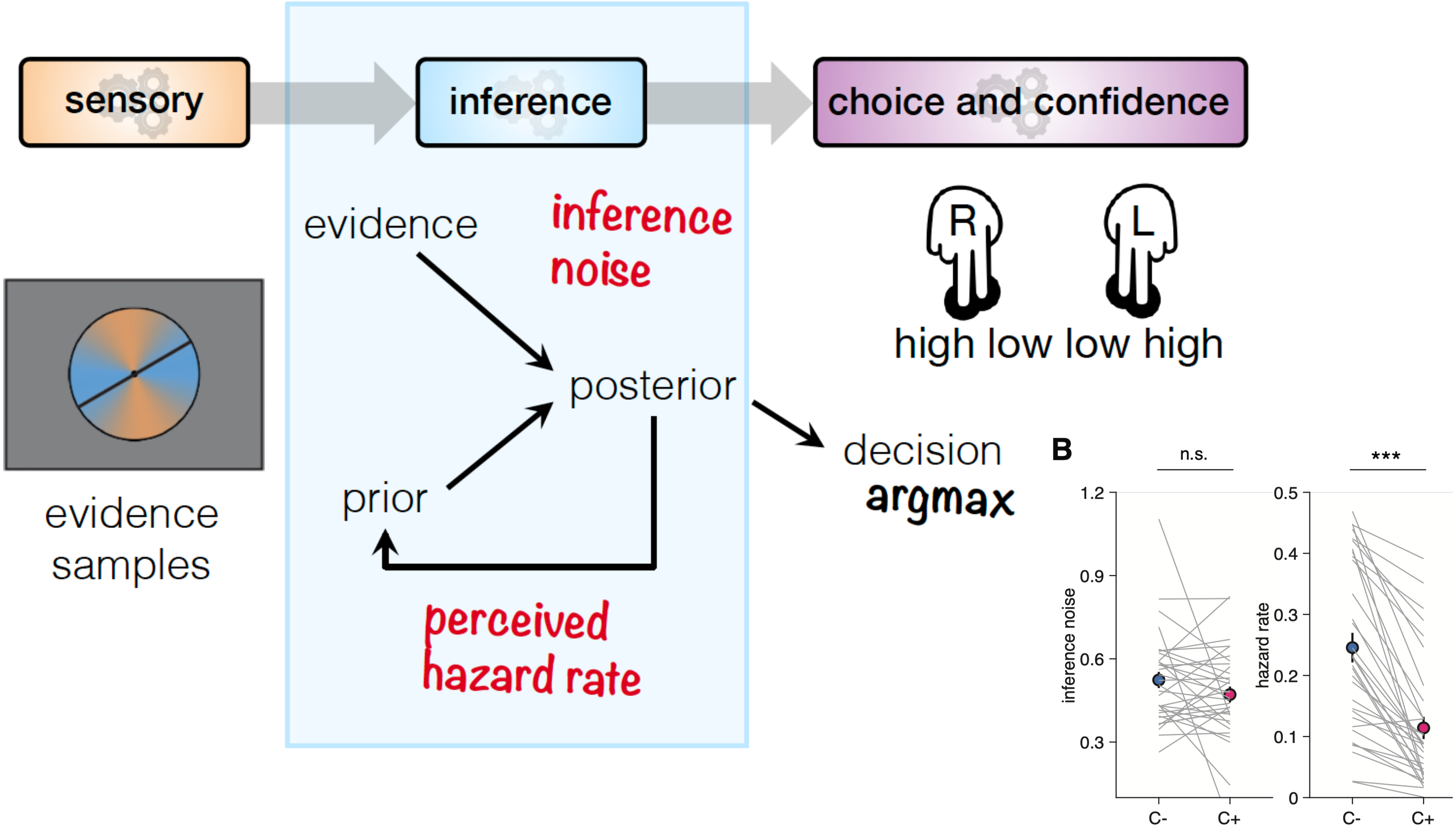
Rouault et al, *eLife* (2022)

Example model validation



Rouault et al, *eLife* (2022)

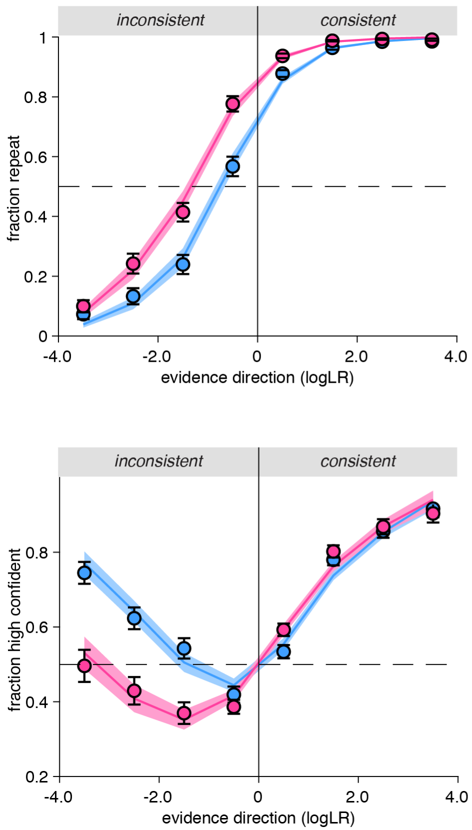
Example model validation



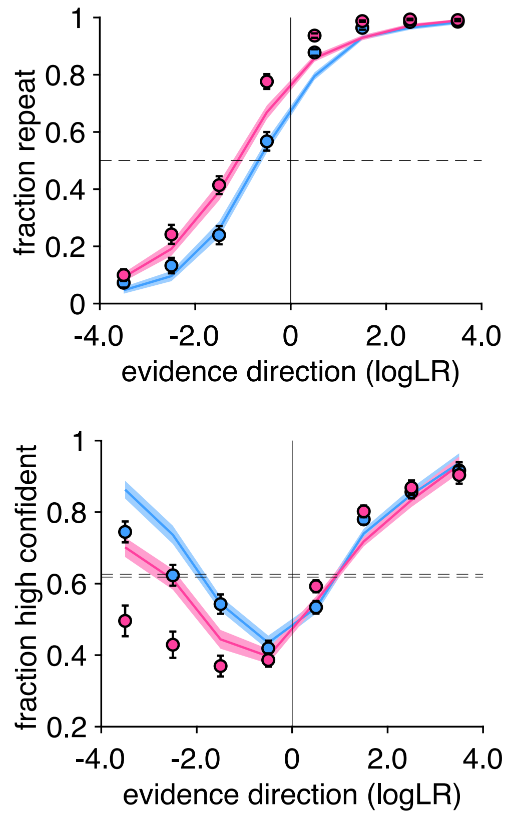
Rouault et al, *eLife* (2022)

Example model validation

Full model



Reduced model



💭 Can you recover/ reproduce all qualitative and quantitative results obtained from your actual data?

Rouault et al, *eLife* (2022)

The plan for the next 120 minutes

- Parameter estimation
  - Model fitting as an optimization problem
- Model validation (tutorial part 1)
  - Does the model capture the data?
- **Parameter recovery (tutorial part 2)**
  - **Simulate, simulate, simulate!**
- Parameter uncertainty (tutorial part 3)
  - Bootstrapping

Parameter recovery

Check whether the fitting procedure gives meaningful parameter values in the best case scenario, that is, when fitting fake data where the “true” parameter values are known.

🤷 Why do we care?

Palminteri, Wyart & Koechlin (2017) *TICS*; Wilson & Collins (2019) *eLife*

Parameter recovery

🤷‍♂️ Why do we care?

- ✓ We want to avoid a biased fitting procedure
- ✓ We want to avoid parameter roles overlapping with each other, capturing similar shares of the variance
- ✓ We want each parameter to be ***identifiable***
- ✓ ***Under these conditions we will be able to interpret the meaning of each of the parameters***
- ✓ And potentially to interpret their different value in different conditions or under different treatment groups

Palminteri, Wyart & Koechlin (2017) *TICS*; Wilson & Collins (2019) *eLife*

Parameter recovery

1. Simulate fake data with different values of parameters (e.g., inference noise, hazard rate)

When choosing the range of parameter values, make sure that you are covering the parameter regime of interest for your experiment!

- Either sweep across the whole parameter space
- Or focus on the space of interest regarding your best-fitting parameters

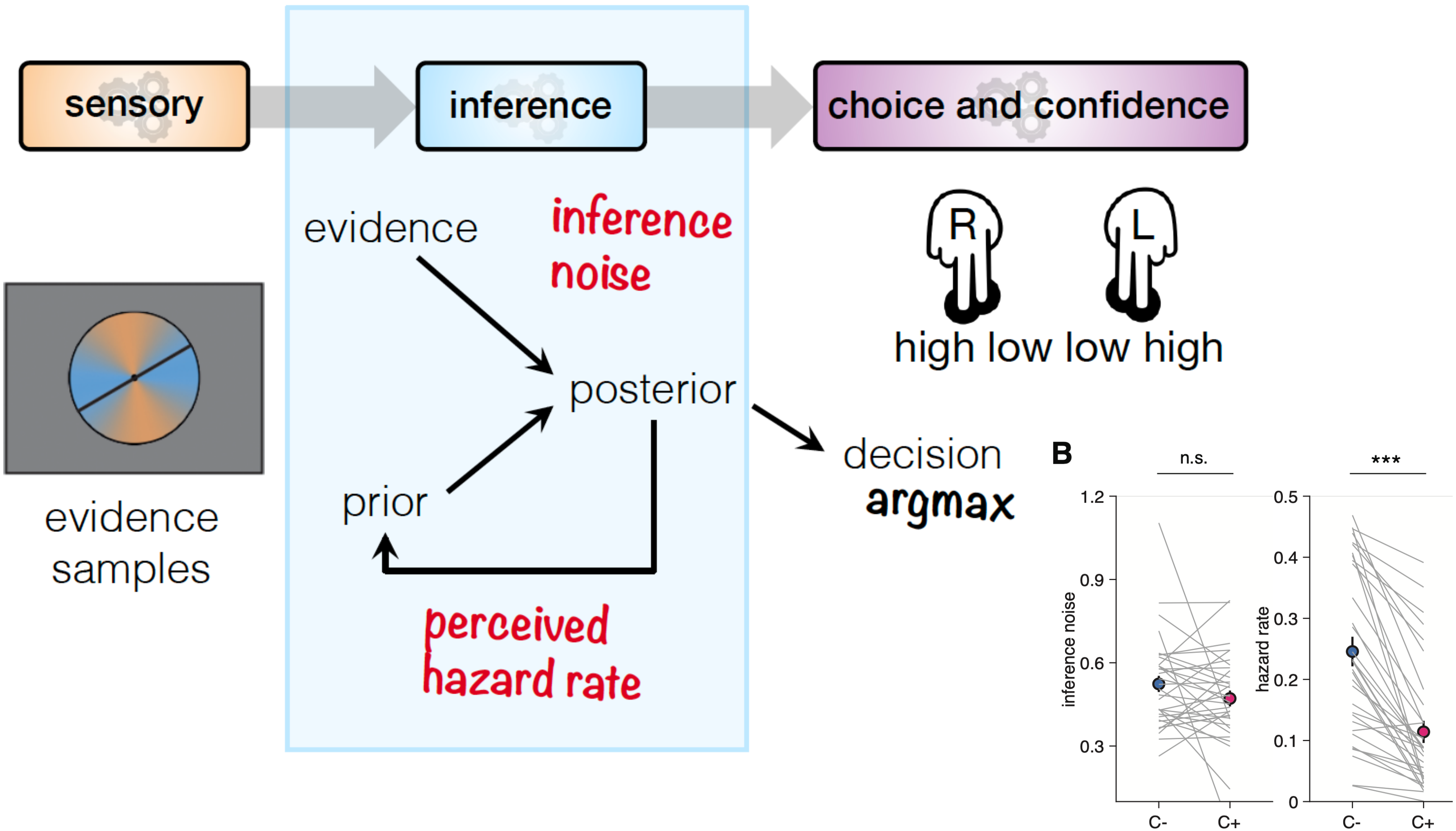
2. Estimate the values of the parameters from your simulated data

3. Compare the recovered parameters to their true values (should be highly correlated without bias)

Are there bugs in your code? Is the experiment underpowered?

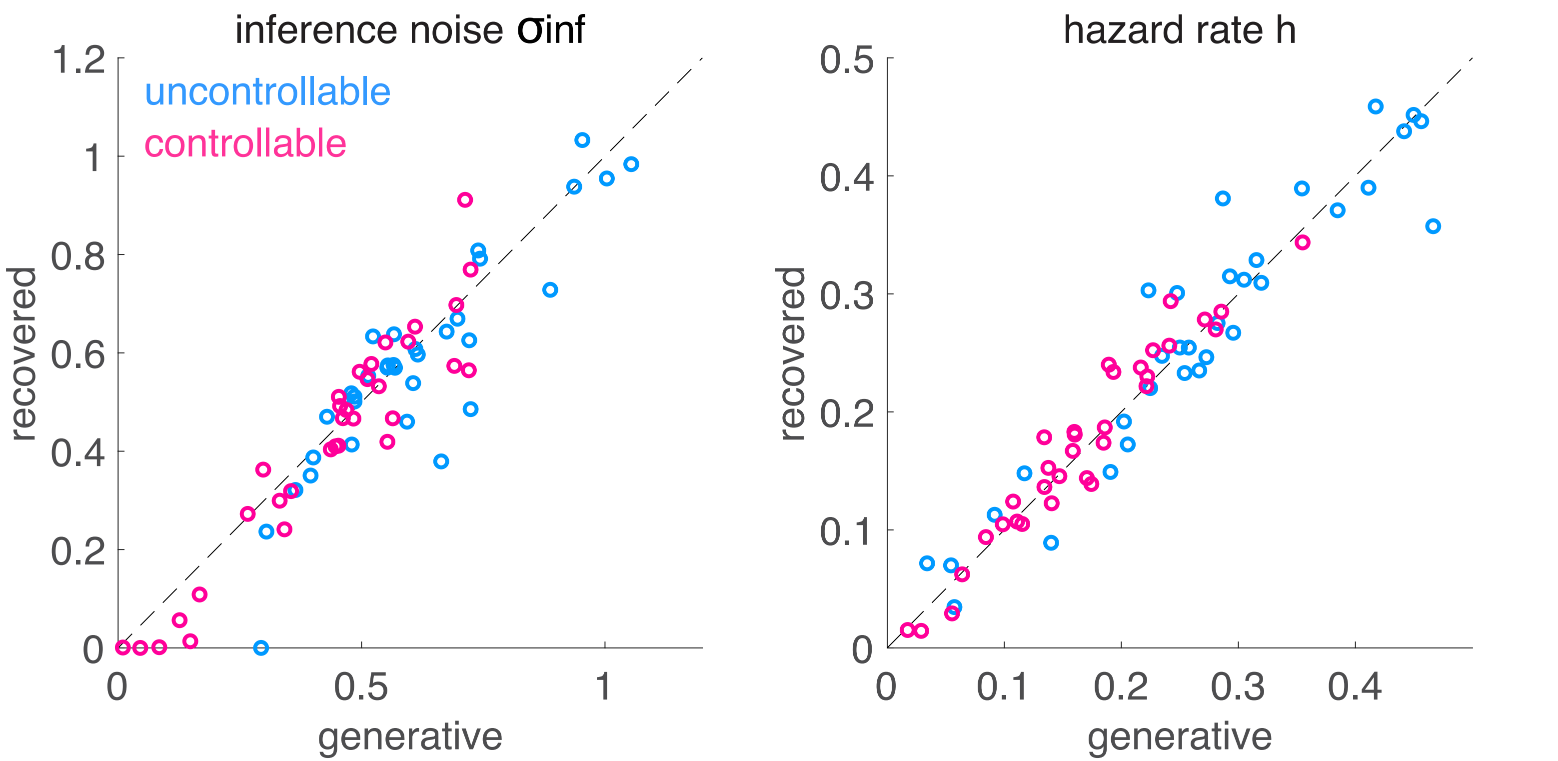
Palminteri, Wyart & Koechlin (2017) *TICS*; Wilson & Collins (2019) *eLife*

Example parameter recovery



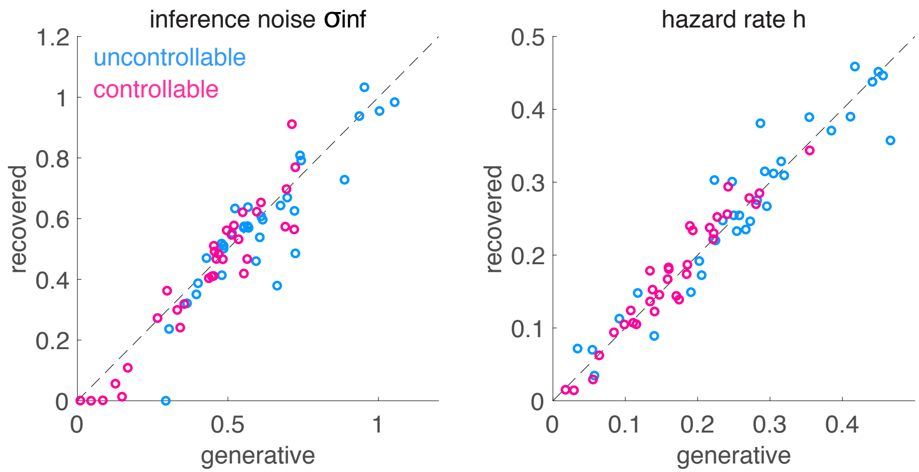
Rouault et al, *eLife* (2022)

Example parameter recovery



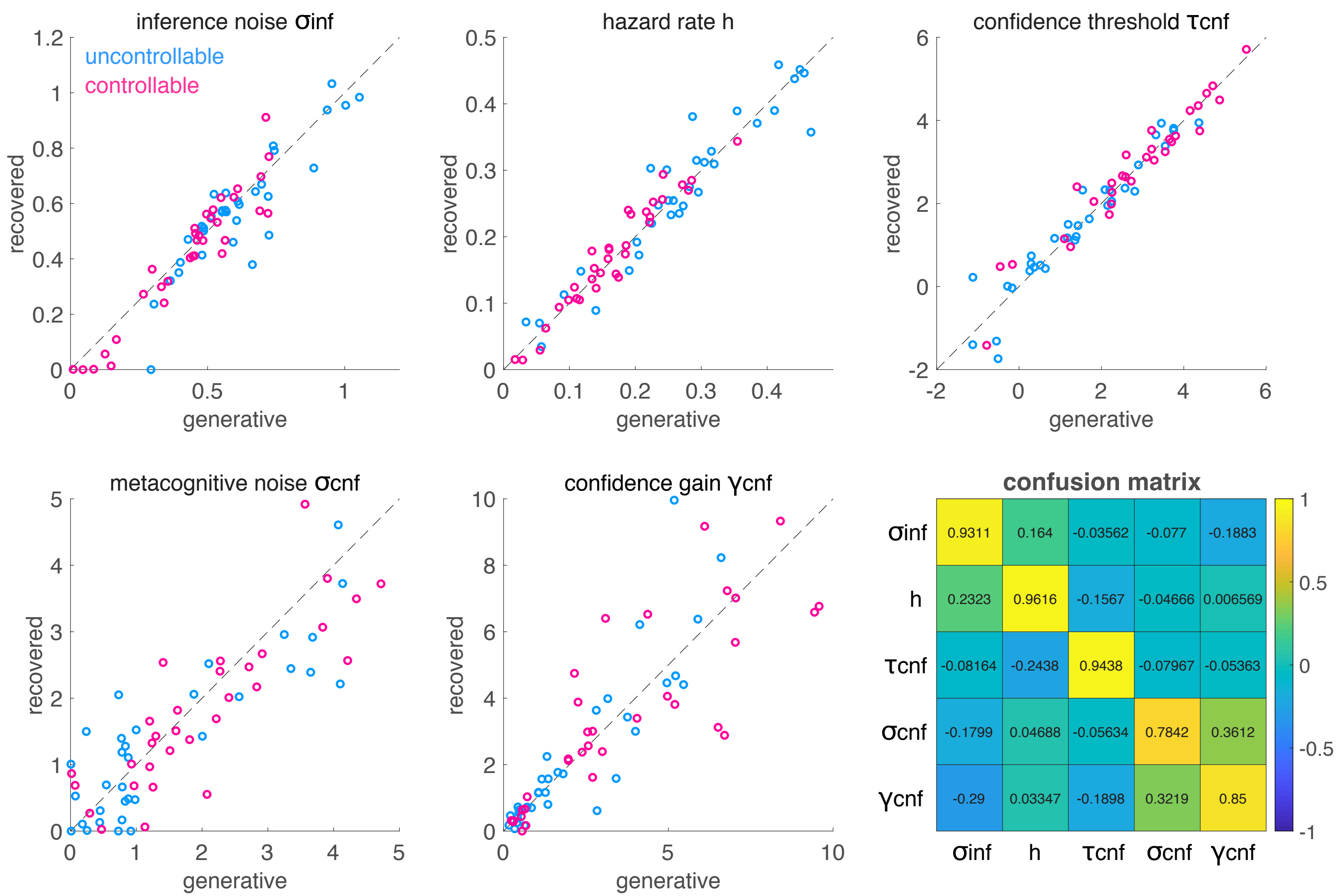
Rouault et al, *eLife* (2022)

Example parameter recovery



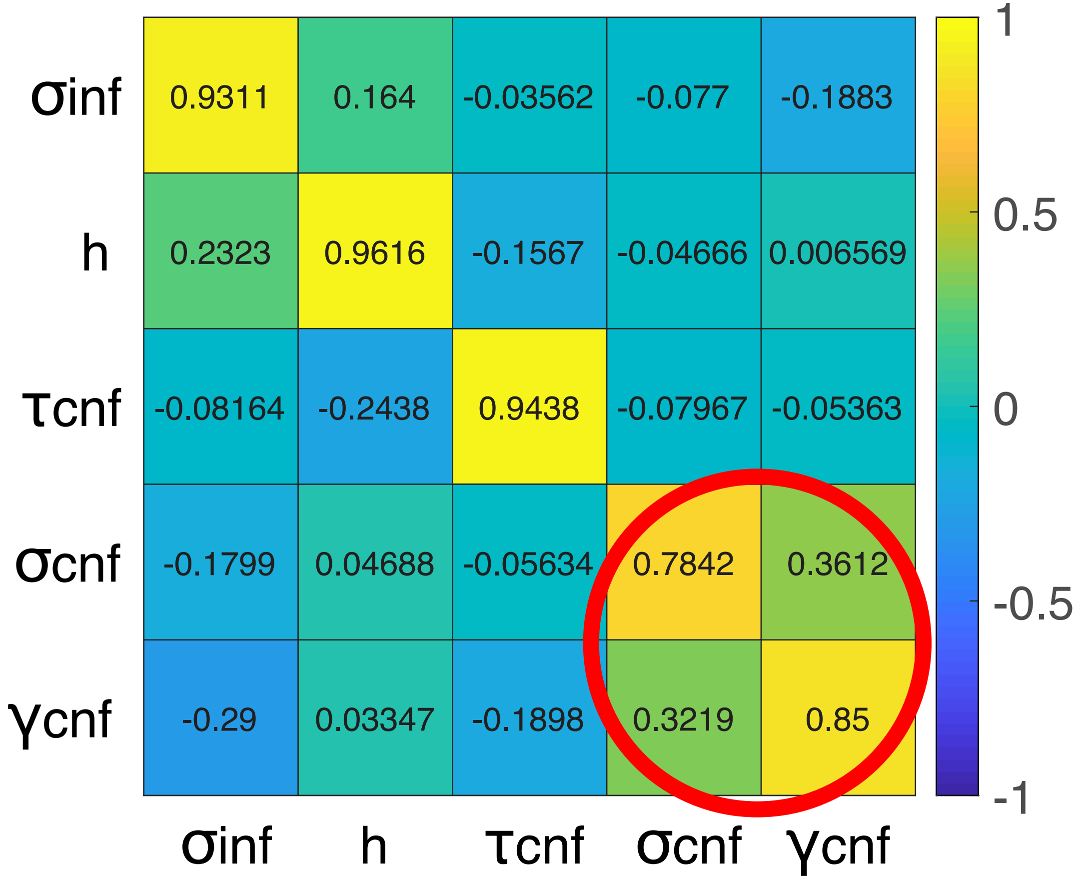
Rouault et al, *eLife* (2022)

Example parameter recovery



Rouault et al, *eLife* (2022)

Confusion matrix: parameter identifiability



Rouault et al, *eLife* (2022)

The plan for the next 120 minutes

- Parameter estimation
  - Model fitting as an optimization problem
- Model validation (tutorial part 1)
  - Does the model capture the data?
- Parameter recovery (tutorial part 2)
  - Simulate, simulate, simulate!
- **Parameter uncertainty (tutorial part 3)**
  - **Bootstrapping**

Uncertainty about parameter estimates

- Uncertainty around parameter estimates: how confident are you about your model fit?
- Confidence interval of a population parameter: mean and confidence levels indicating the probability that the interval contains the parameter
- Ideally, you would replicate your study several times and estimate the mean +/- margin of error
- Traditional statistical techniques are only asymptotically correct (large N)

Uncertainty about parameter estimates

- In practice, we don’t know **the underlying distribution** from which your parameter is drawn: hence the error distribution cannot be easily calculated

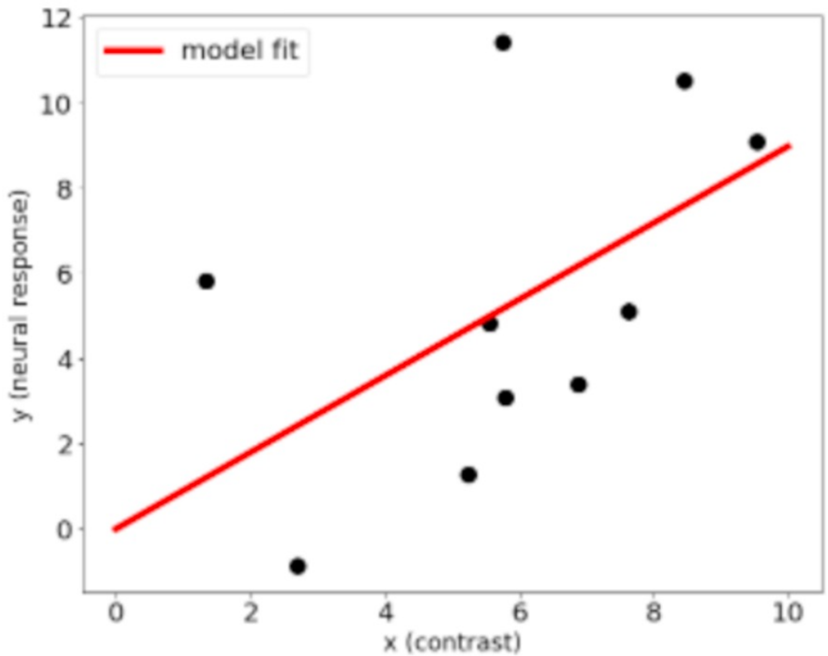
➢ **Bootstrapping** procedure, involves resampling your data

- Repeatedly draw independent samples from a data set, that is, with replacement (**resampling**)
- This creates a new data set, of same size as the initial one
- We make the assumption that your data set is representative, a good measure of the underlying distribution

Example: Bootstrapping procedure

- Uncertainty around parameter estimates: how confident are you about your model fit?

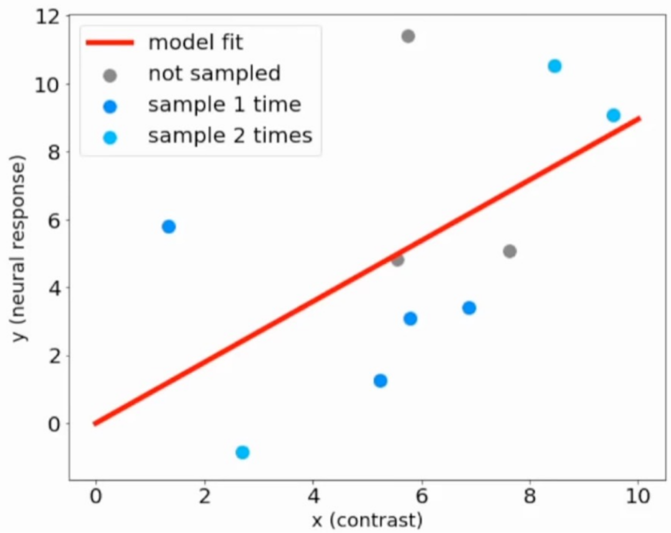
- Example: a model fit of neural response as a function of contrast

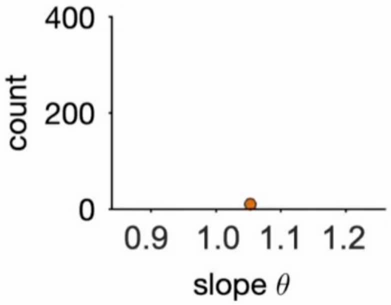


Anqi Wu, *neuromatch*

Bootstrapping procedure

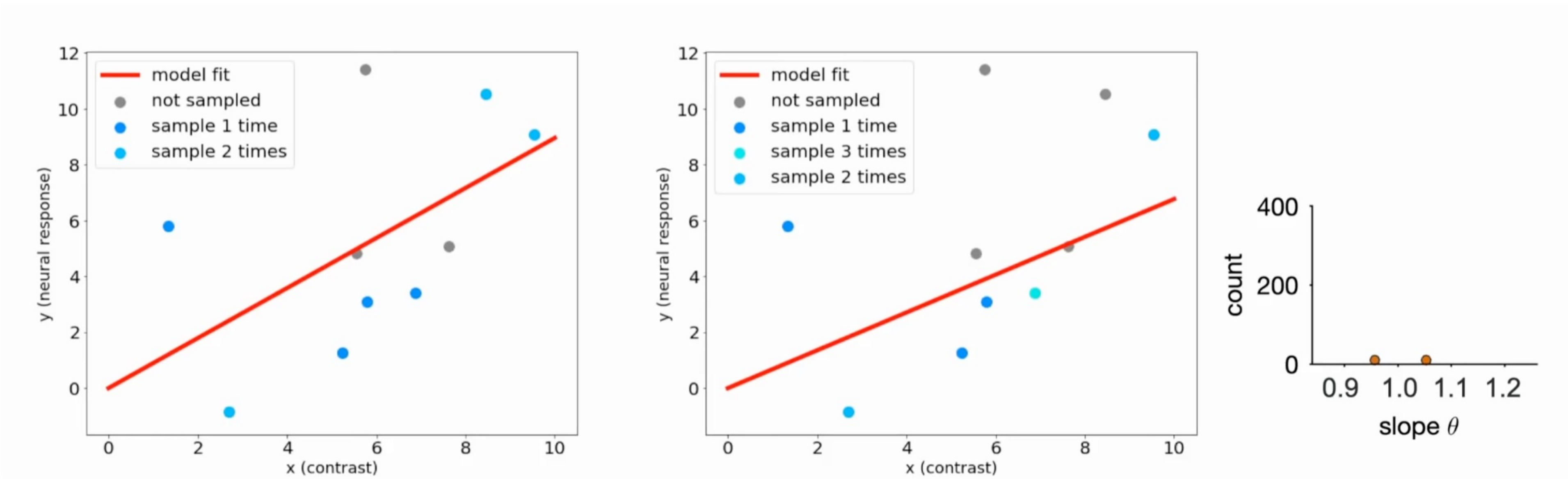
1. Resample from the observed dataset with replacement





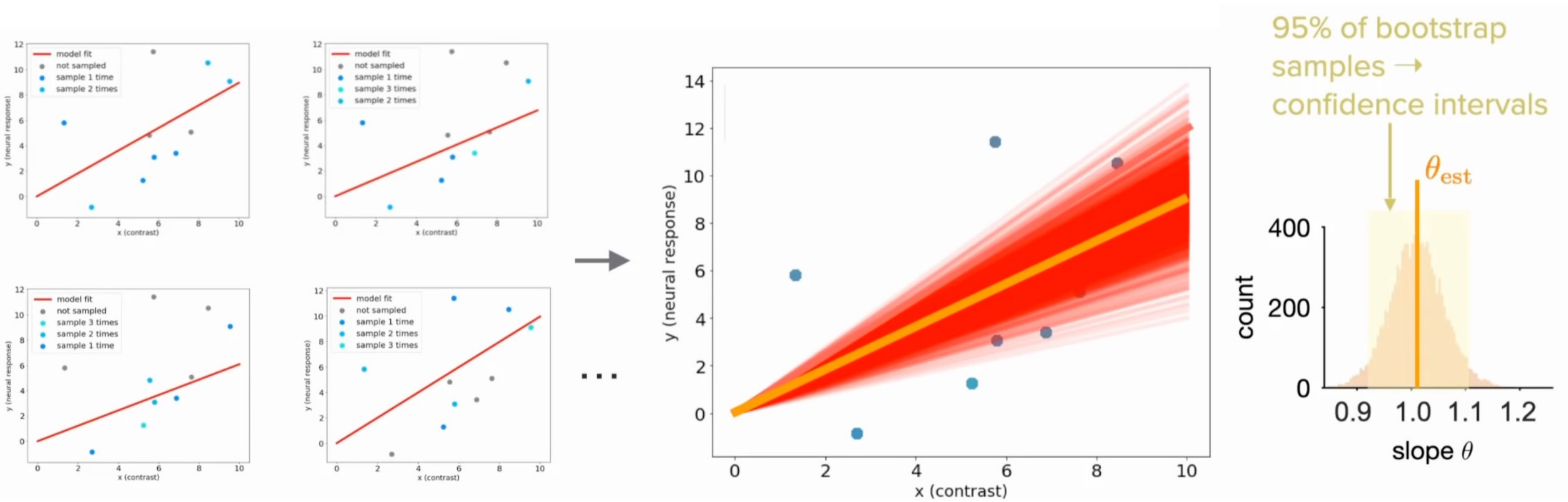
Bootstrapping procedure

1. Resample from the observed dataset with replacement



Bootstrapping procedure

1. Resample from the observed dataset with replacement
2. Collect all estimates into a distribution, and analyze the confidence intervals



The idea is to generate many new synthetic datasets from the initial true dataset by randomly sampling from it, then finding estimators for each one of these new datasets, and finally looking at the distribution of all these estimators to quantify our confidence.

Acknowledgements 🙏

- Marion Rouault, CNRS & Paris Brain Institute (BAMB! 2022)
- Luigi Acerbi (BAMB! 2022)
- Anqi Wu (Neuromatch)