AN ASTRONOMER'S INTRODUCTION TO GAUSSIAN PROCESSES

Dan Foreman-Mackey
CCPP@NYU // github.com/dfm // @exoplaneteer // dfm.io

github.com/dfm/gp

gaussianprocess.org/gpml

Rasmussen & Williams

I write code for good & astrophysics.

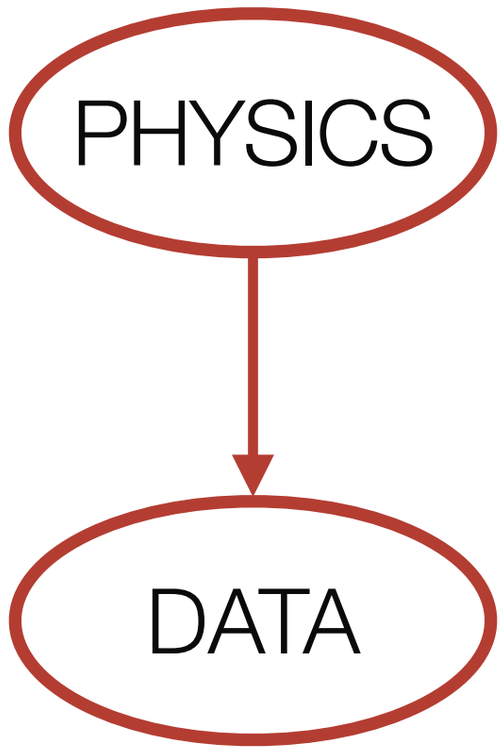
I (probably) do data science.

not data science.

Photo credit **James Silvester**
silvesterphoto.tumblr.com

data science.

Data Science

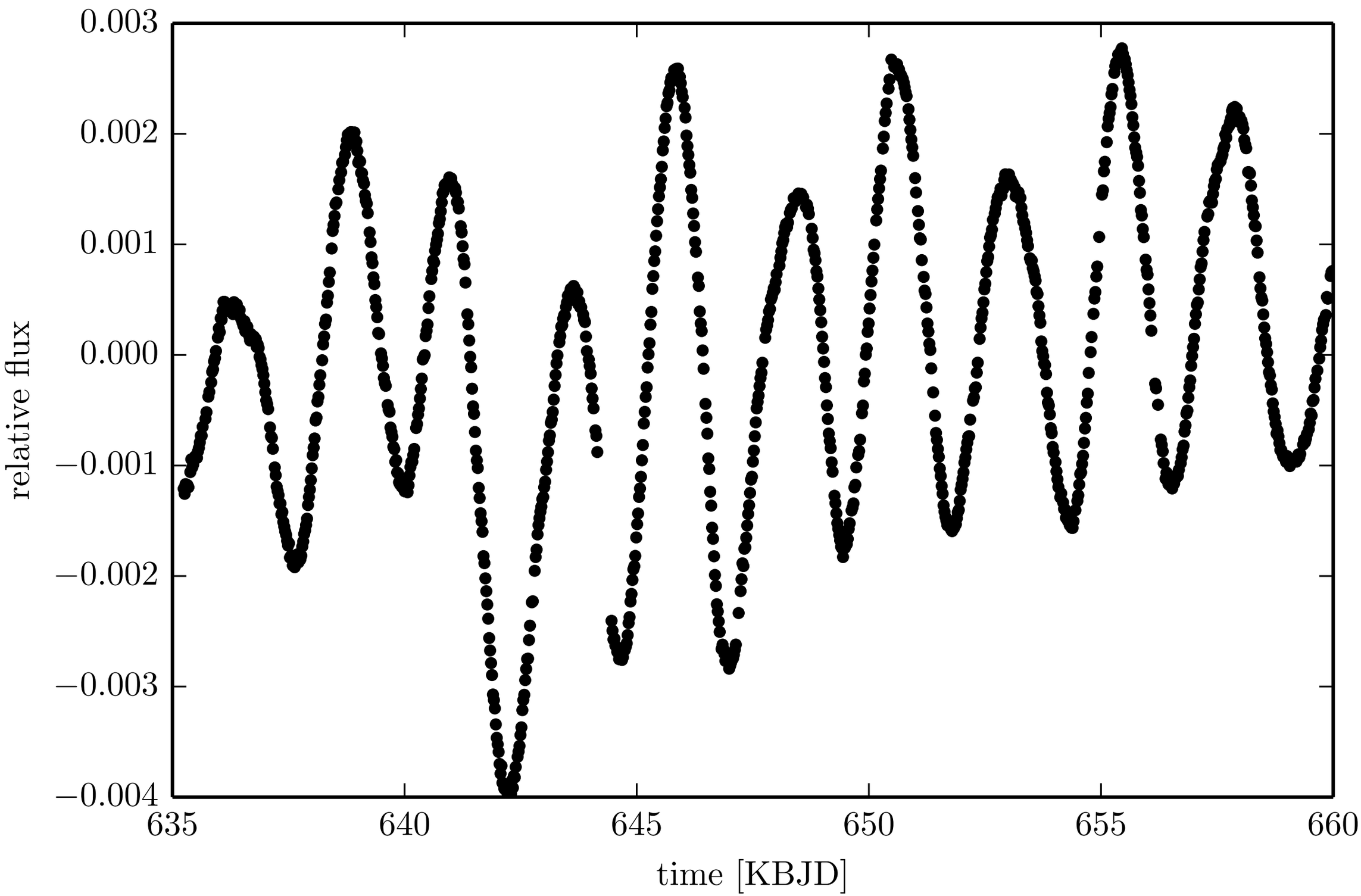


$$p(\text{data} \,|\, \text{physics})$$

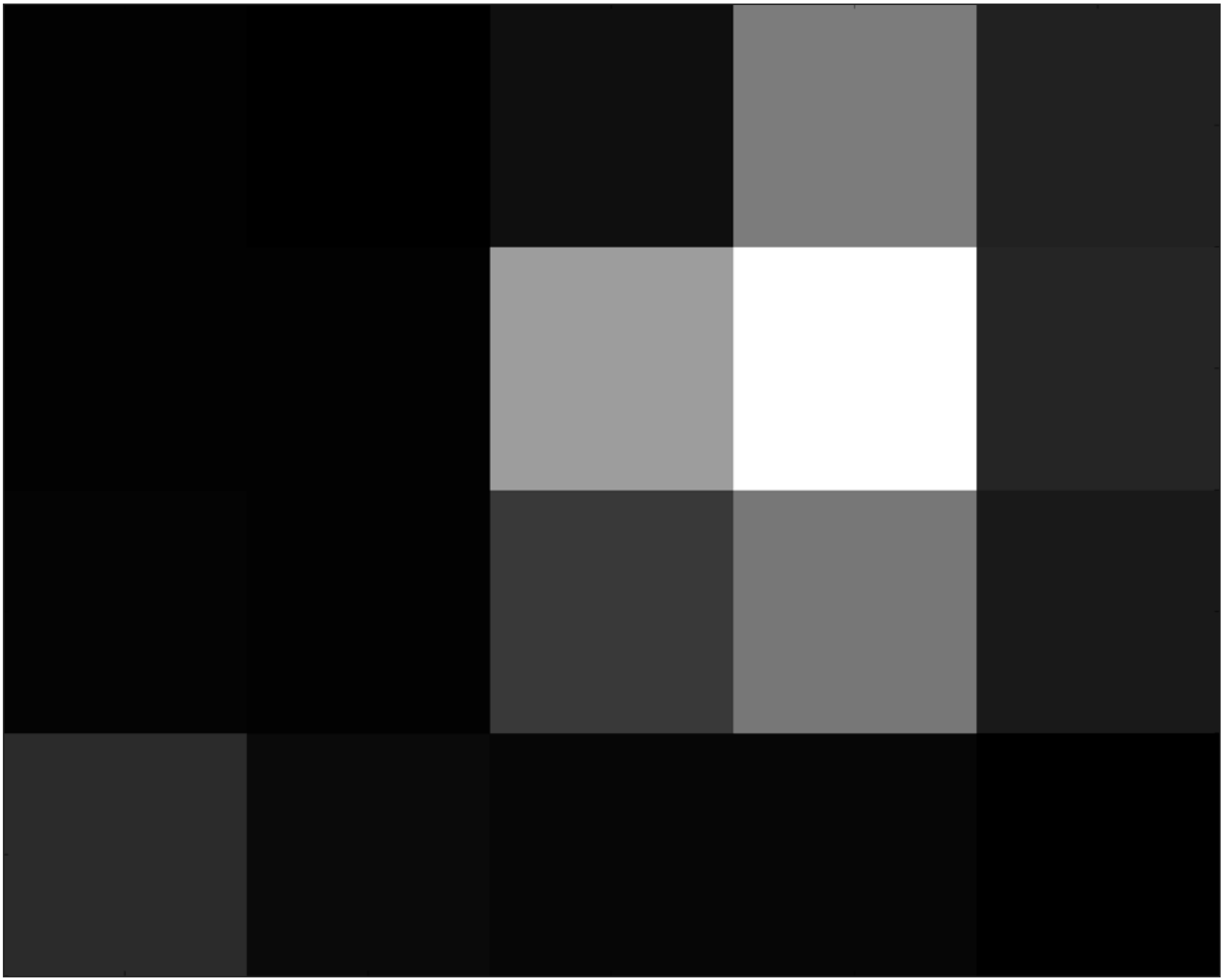
I work with Kepler data.

I get really passionate about

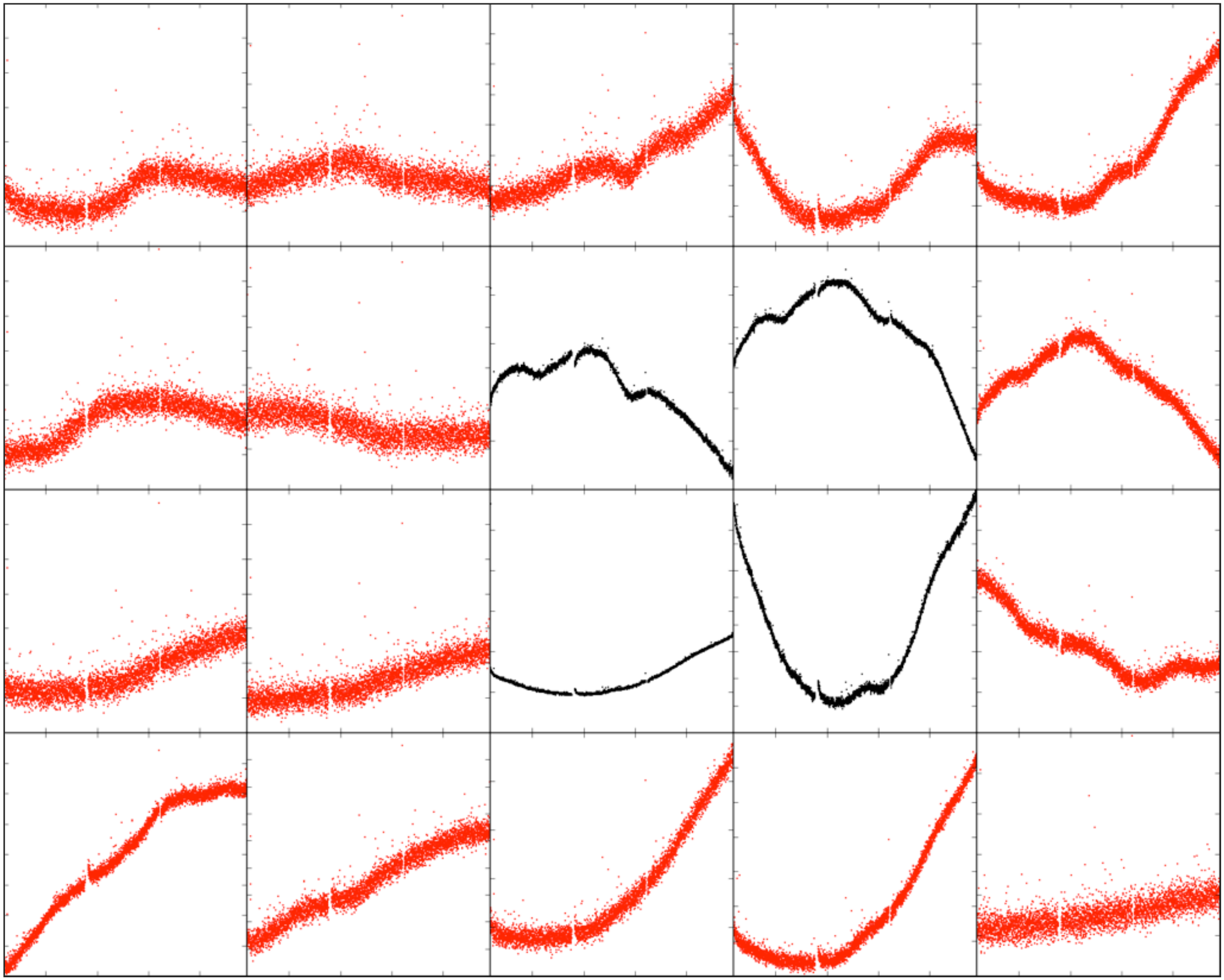
NOISE


0.003
0.002
0.001
0.000
−0.001
−0.002
−0.003
−0.004
relative flux
635
640
645
650
655
660
time [KBJD]

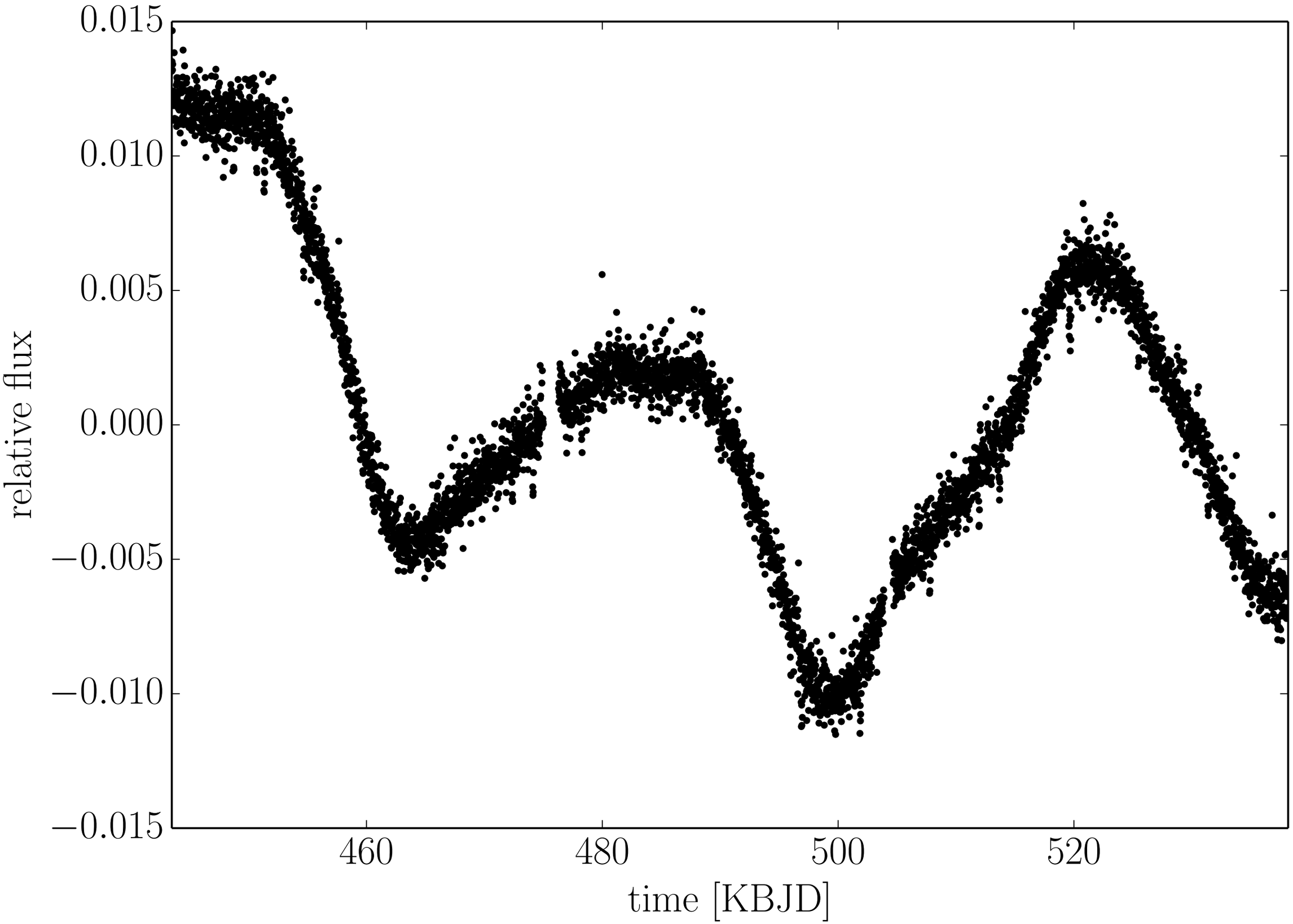
KIC 3223000



Kepler 32



Kepler 32


0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
relative flux
460
480
500
520
time [KBJD]

Kepler 32

HACKS

ALL HACKS

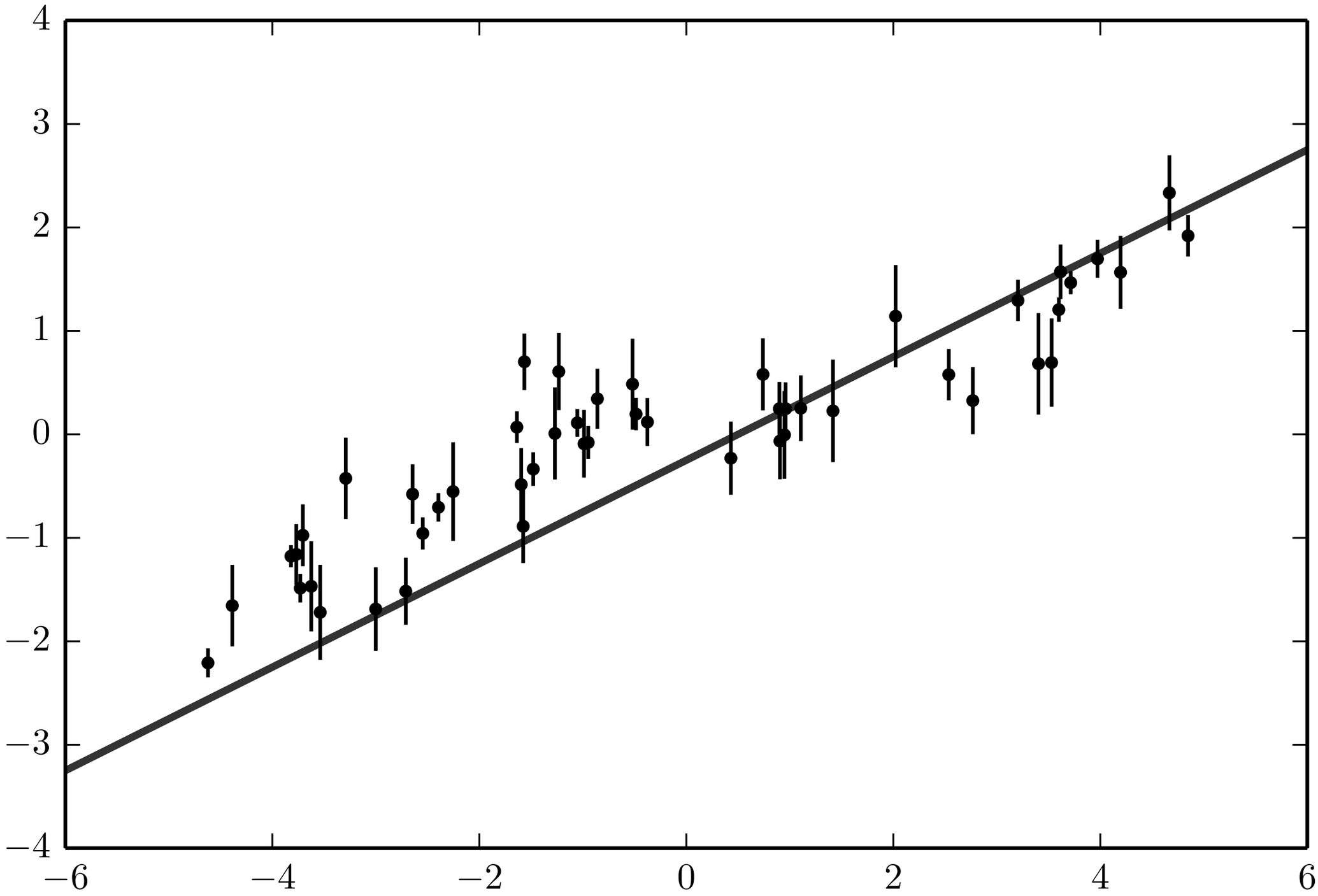
and I'm ***righteous***!

Why not model all the things?

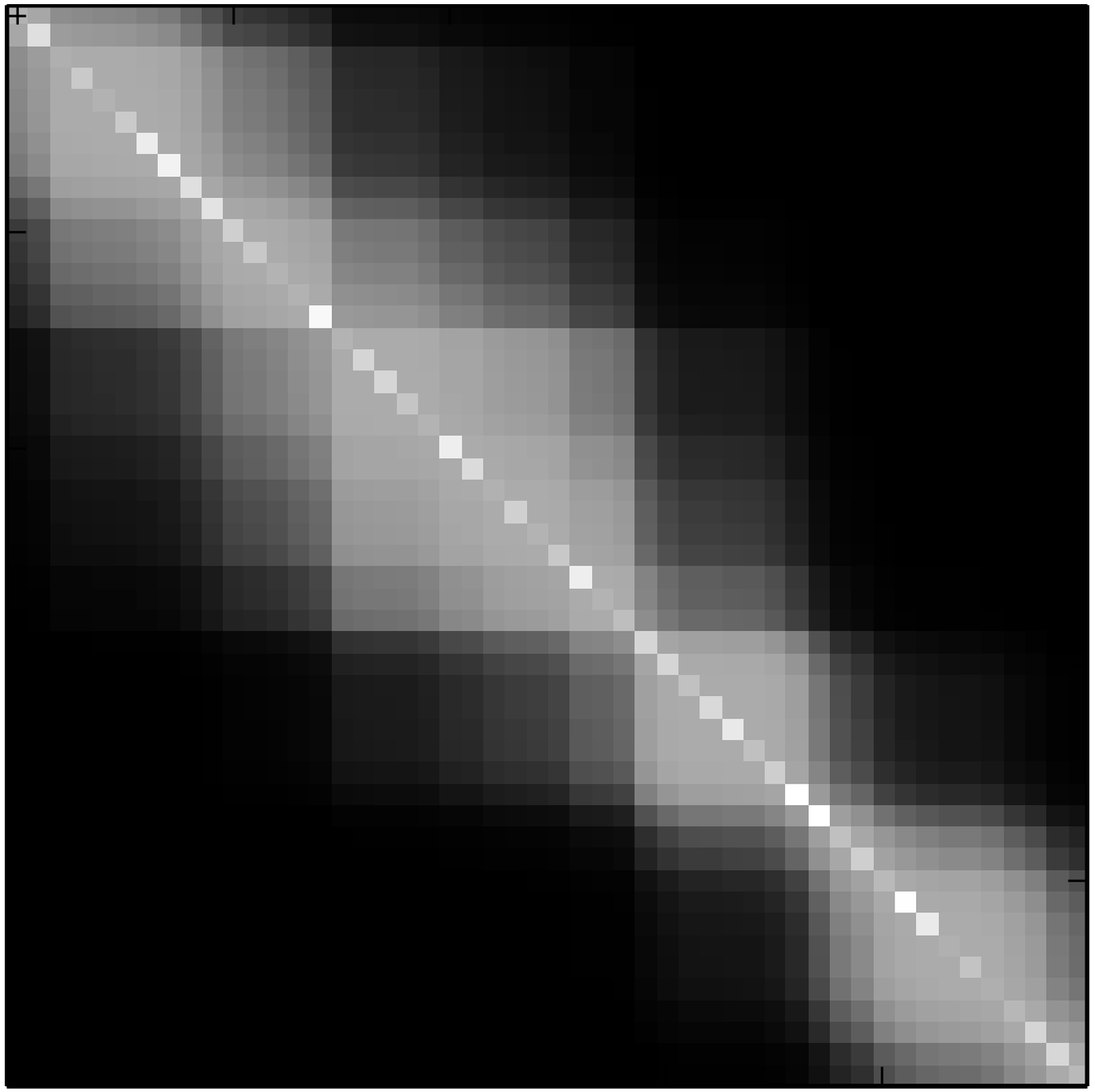
with, for example, a Gaussian Process

The power of correlated noise.

$$y = m\,x + b$$



The true covariance of the observations.



Let's assume that the noise is…

independent

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\theta}) = -\frac{1}{2} \sum_{n=1}^{N} \left[\frac{[y_n - f_{\boldsymbol{\theta}}(x_n)]^2}{\sigma_n^2} + \log 2\pi\sigma_n^2 \right]$$

Gaussian with known variance

Or equivalently…

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta}) = -\frac{1}{2}\, \boldsymbol{r}^{\mathrm{T}}\, \boldsymbol{C}^{-1}\, \boldsymbol{r} - \frac{1}{2} \log\det \boldsymbol{C} - \frac{N}{2} \log 2\,\pi$$

Or equivalently…

data covariance

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta}) = -\frac{1}{2}\, \boldsymbol{r}^{\mathrm{T}}\, \boldsymbol{C}^{-1}\, \boldsymbol{r} - \frac{1}{2} \log \det \boldsymbol{C} - \frac{N}{2} \log 2\,\pi$$

Or equivalently…

data covariance

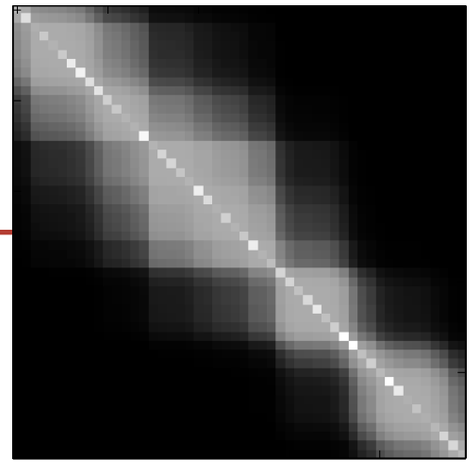
$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta}) = -\frac{1}{2}\, \boldsymbol{r}^{\mathrm{T}}\, \boldsymbol{C}^{-1}\, \boldsymbol{r} - \frac{1}{2} \log\det \boldsymbol{C} - \frac{N}{2} \log 2\,\pi$$

residual vector

$$\boldsymbol{r} = \left[\; y_1 - f_{\boldsymbol{\theta}}(x_1) \quad y_2 - f_{\boldsymbol{\theta}}(x_2) \quad \cdots \quad y_n - f_{\boldsymbol{\theta}}(x_n) \;\right]^{\mathrm{T}}$$

Or equivalently…

data covariance



$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta}) = -\frac{1}{2}\, \boldsymbol{r}^{\mathrm{T}}\, \boldsymbol{C}^{-1}\, \boldsymbol{r} - \frac{1}{2} \log\det \boldsymbol{C} - \frac{N}{2} \log 2\,\pi$$

residual vector

$$\boldsymbol{r} = \left[\; y_1 - f_{\boldsymbol{\theta}}(x_1) \quad y_2 - f_{\boldsymbol{\theta}}(x_2) \quad \cdots \quad y_n - f_{\boldsymbol{\theta}}(x_n) \;\right]^{\mathrm{T}}$$

Linear least-squares.

$$\begin{bmatrix} m \\ b \end{bmatrix} = \mathbf{S}\,\mathbf{A}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{y} \qquad\qquad \mathbf{S} = \left[\mathbf{A}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{A}\right]^{-1}$$

maximum likelihood &
in this case only mean of posterior

posterior covariance

$$\mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Linear least-squares.

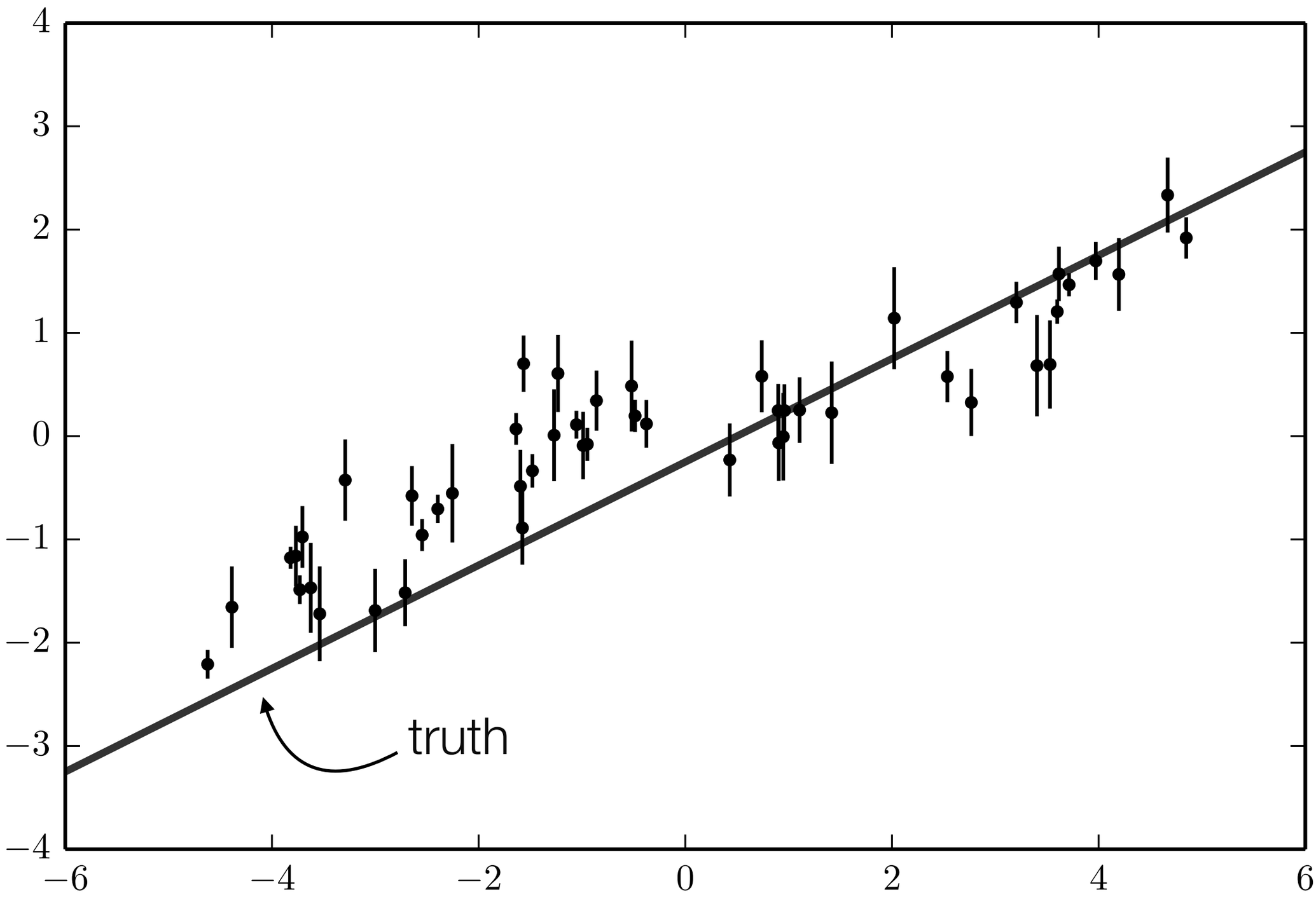
$$\begin{bmatrix} m \\ b \end{bmatrix} = \mathbf{S}\,\mathbf{A}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{y} \qquad\qquad \mathbf{S} = \left[\mathbf{A}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{A}\right]^{-1}$$

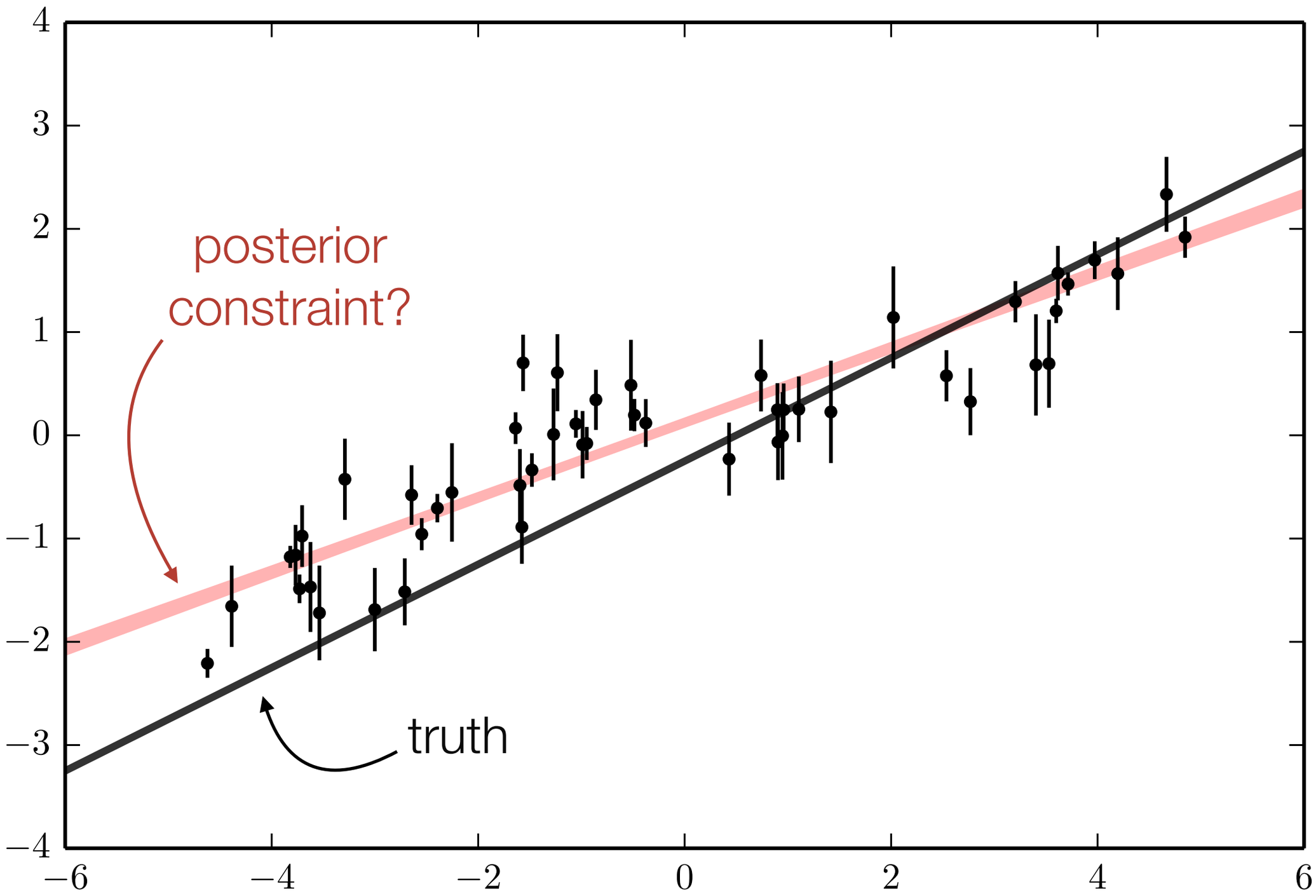
maximum likelihood &
in this case only mean of posterior

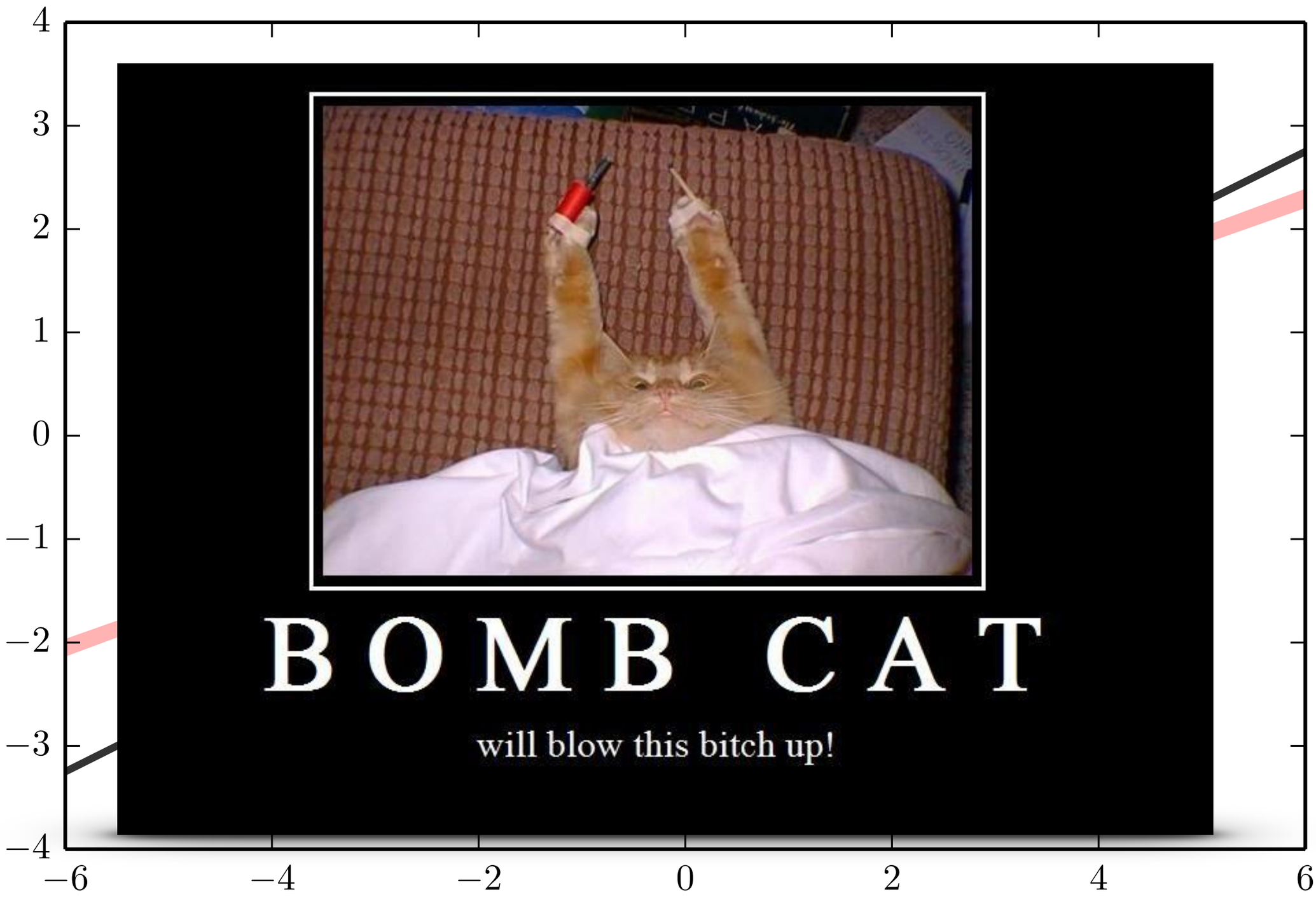
posterior covariance

assuming uniform priors

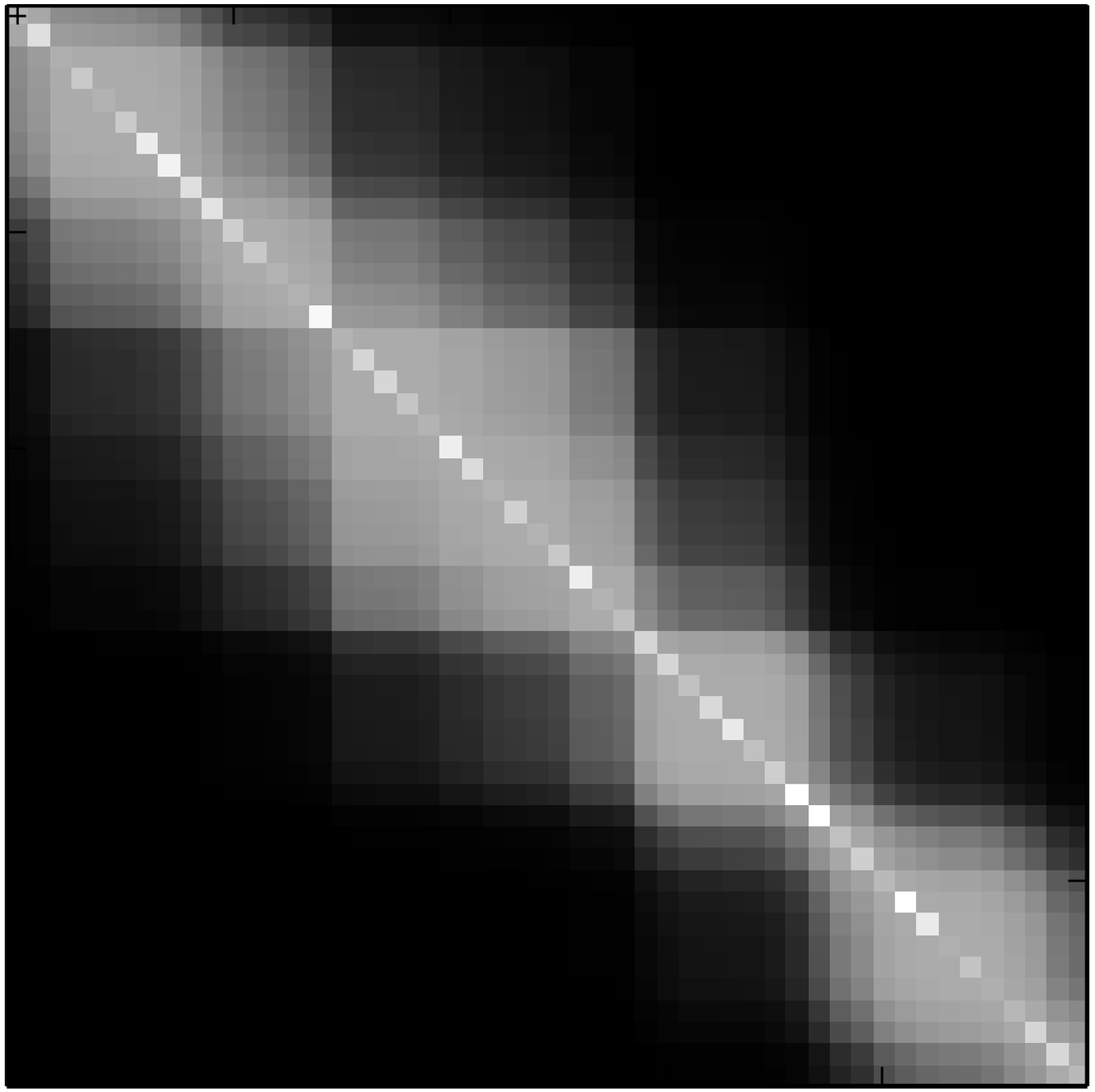
$$\mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$


4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6
truth

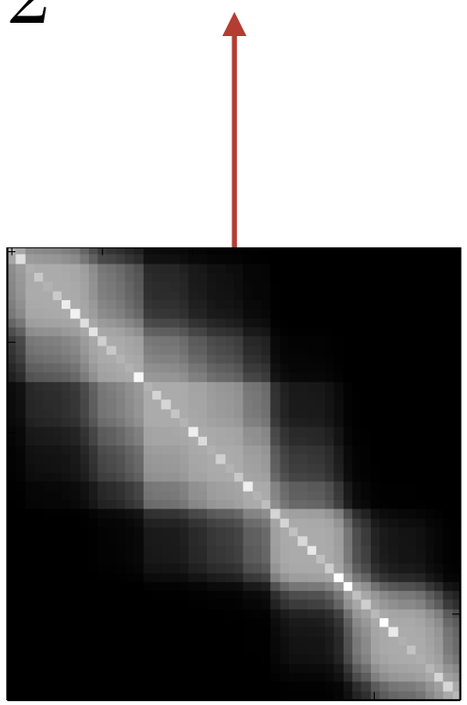

posterior
constraint?
truth
4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6


BOMB CAT
will blow this bitch up!
4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6

But we know the true covariance matrix.



$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta}) = -\frac{1}{2}\, \boldsymbol{r}^{\mathrm{T}}\, \boldsymbol{C}^{-1}\, \boldsymbol{r} - \frac{1}{2} \log\det \boldsymbol{C} - \frac{N}{2} \log 2\,\pi$$



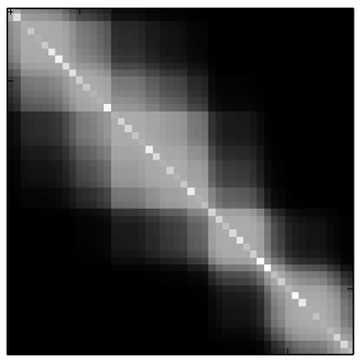
Linear least-squares.

$$\begin{bmatrix} m \\ b \end{bmatrix} = \mathbf{S}\,\mathbf{A}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{y} \qquad\qquad \mathbf{S} = \left[\mathbf{A}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\mathbf{A}\right]^{-1}$$

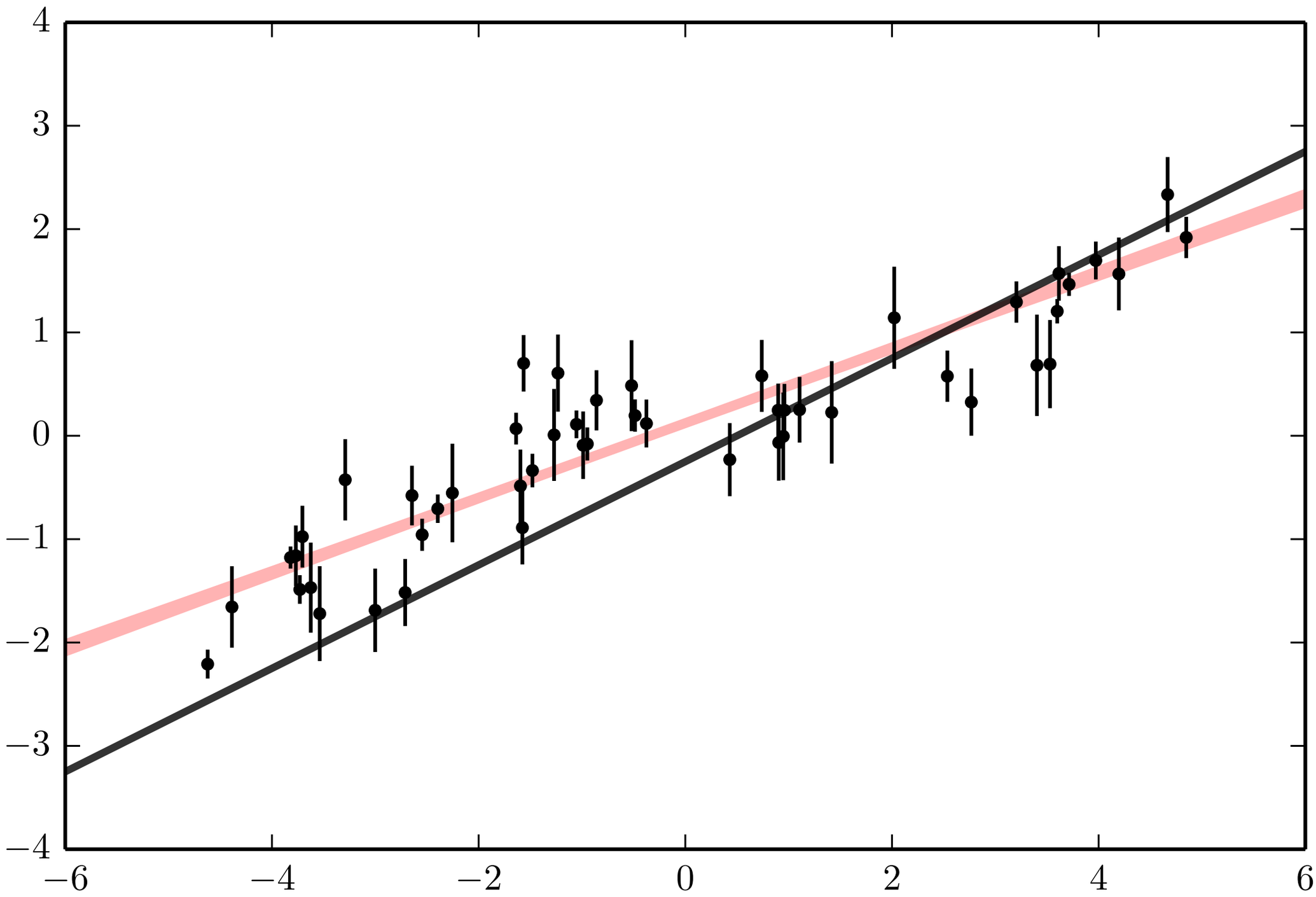
maximum likelihood &
in this case only mean of posterior

posterior covariance

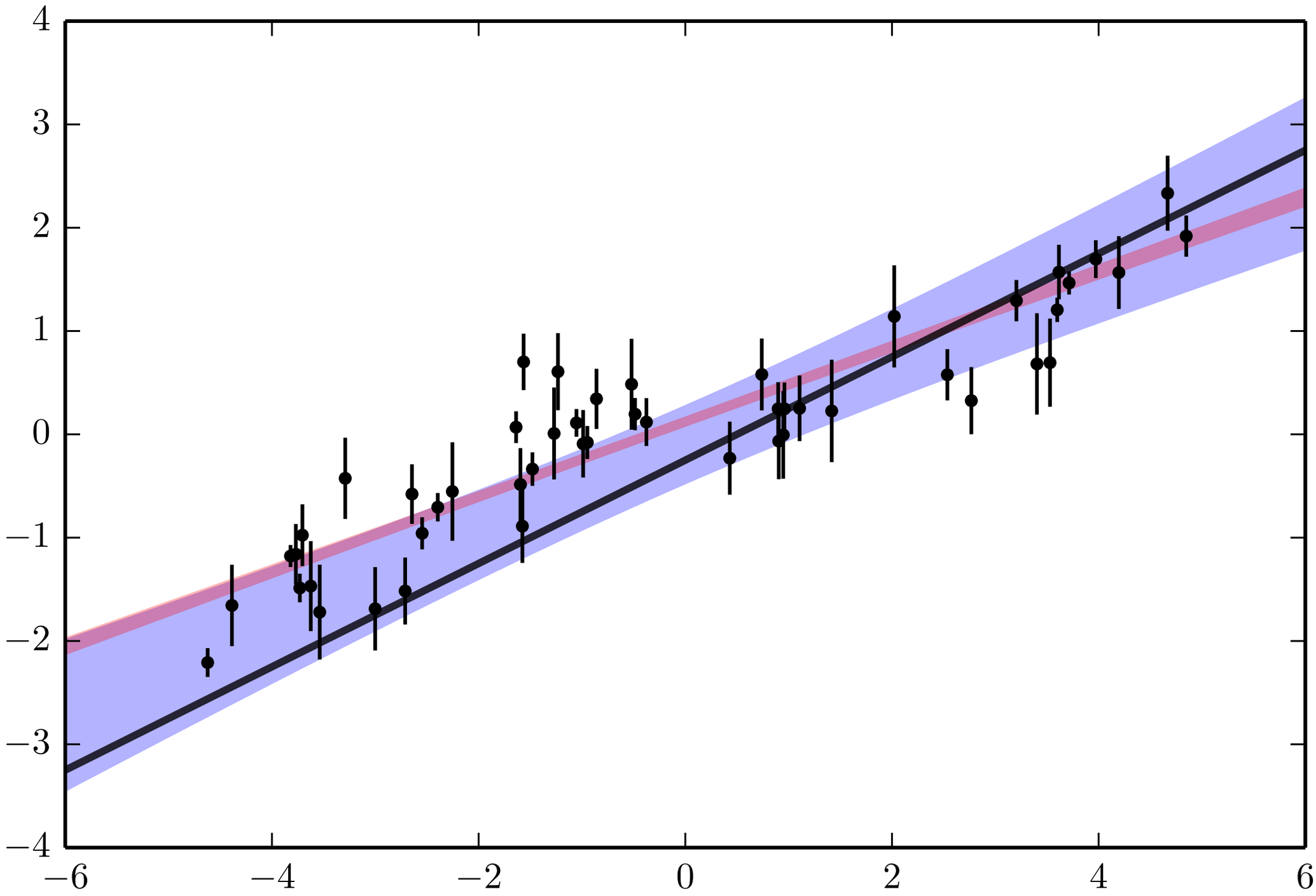
$$\mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} \\ \\ \\ \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$



Before.



After.

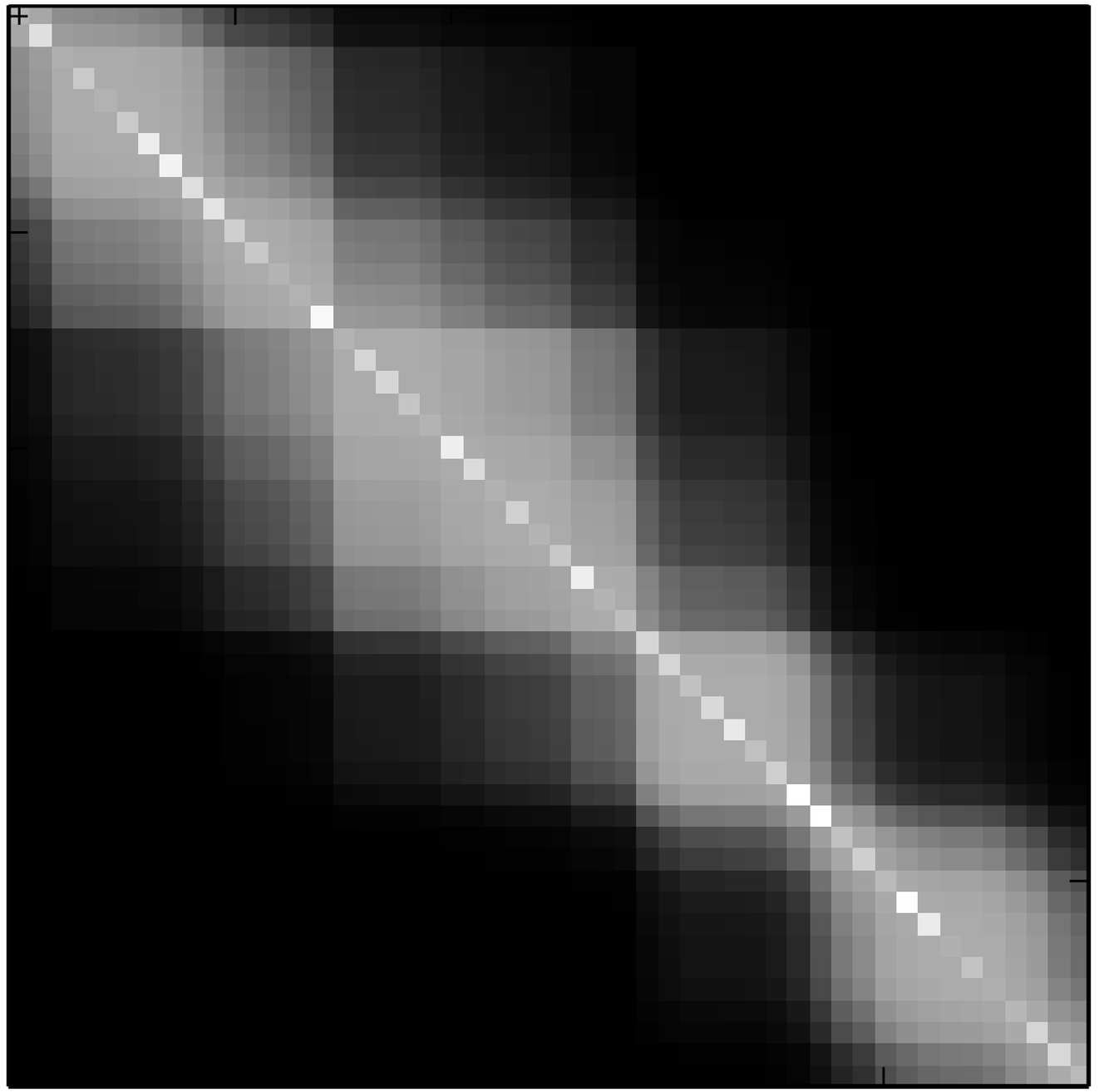


the responsible scientist.

So… we're finished, right?

In The Real World™, we never know the noise.

Just gotta model it!



Model it!

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta},\, \boldsymbol{\alpha}) = -\frac{1}{2}\, \boldsymbol{r}^{\mathrm{T}}\, \boldsymbol{K}_{\boldsymbol{\alpha}}^{-1}\, \boldsymbol{r} - \frac{1}{2}\, \log\det \boldsymbol{K}_{\boldsymbol{\alpha}} + C$$

where

$$K_{ij} = \sigma_i^2\, \delta_{ij} + k_{\boldsymbol{\alpha}}(x_i,\, x_j)$$

for example

$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = a^2\, \exp\left(-\frac{[x_i - x_j]^2}{2\, l^2}\right)$$

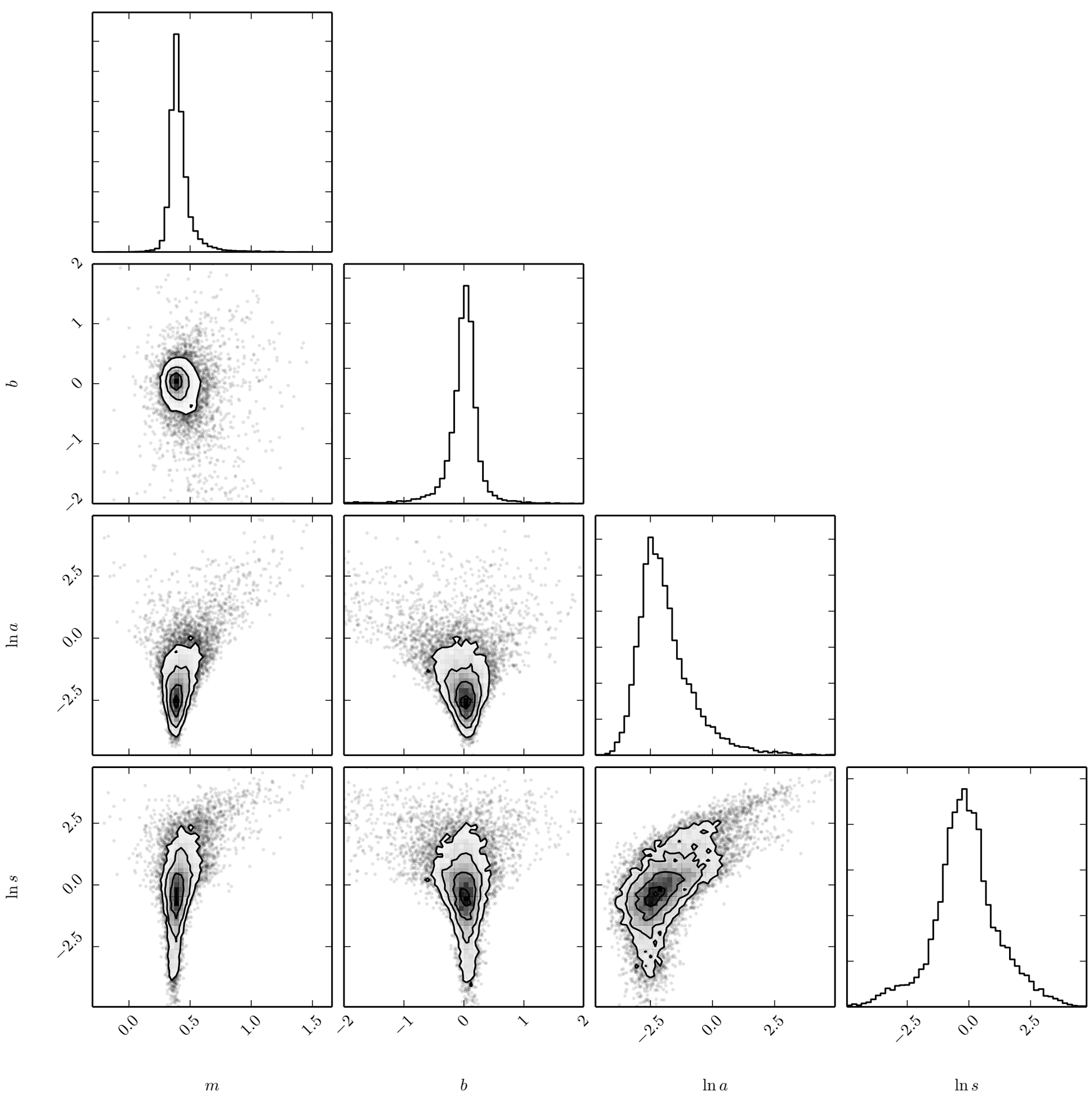
drop-in replacement for your current log-likelihood function!

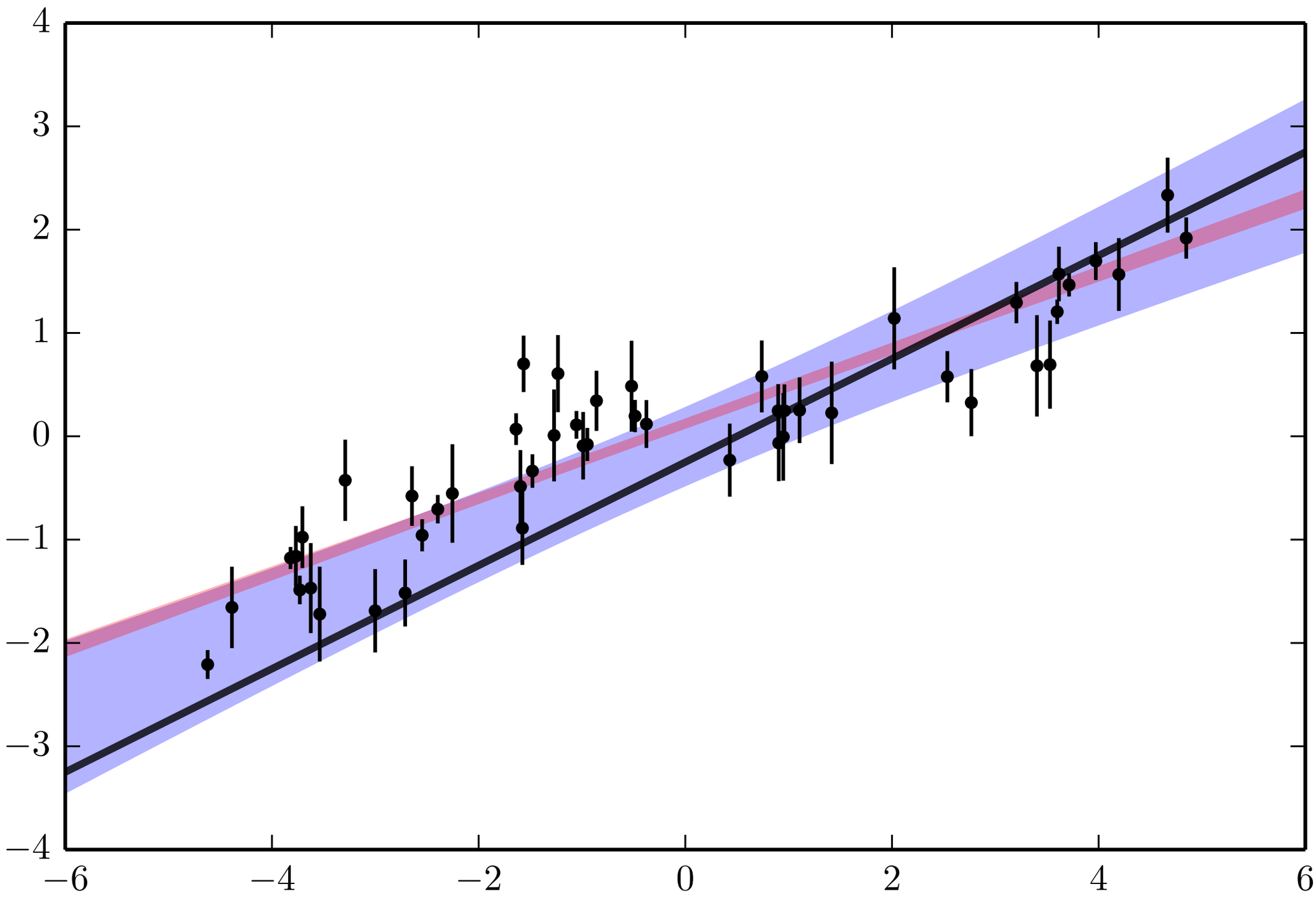
emcee

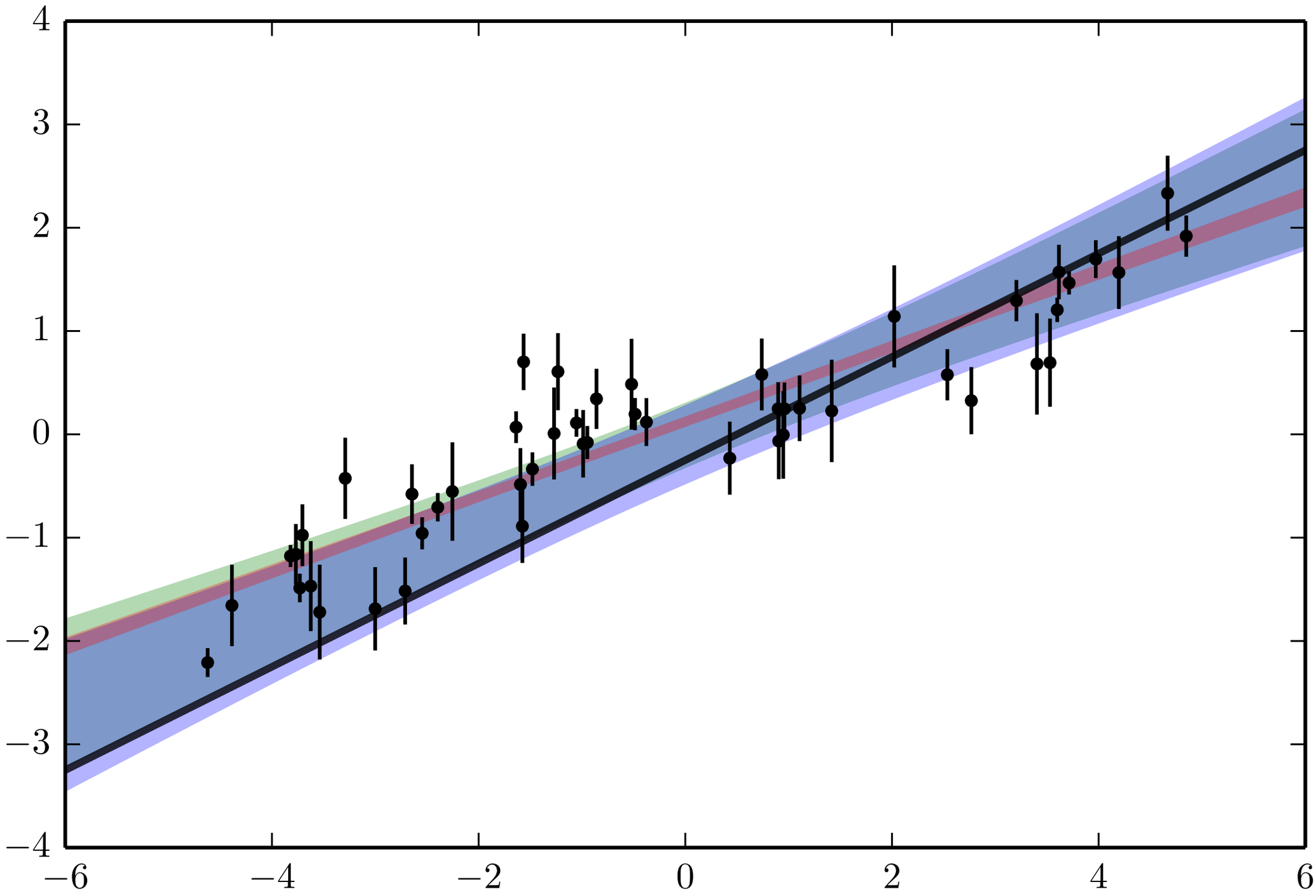
the MCMC Hammer

arxiv.org/abs/1202.3665

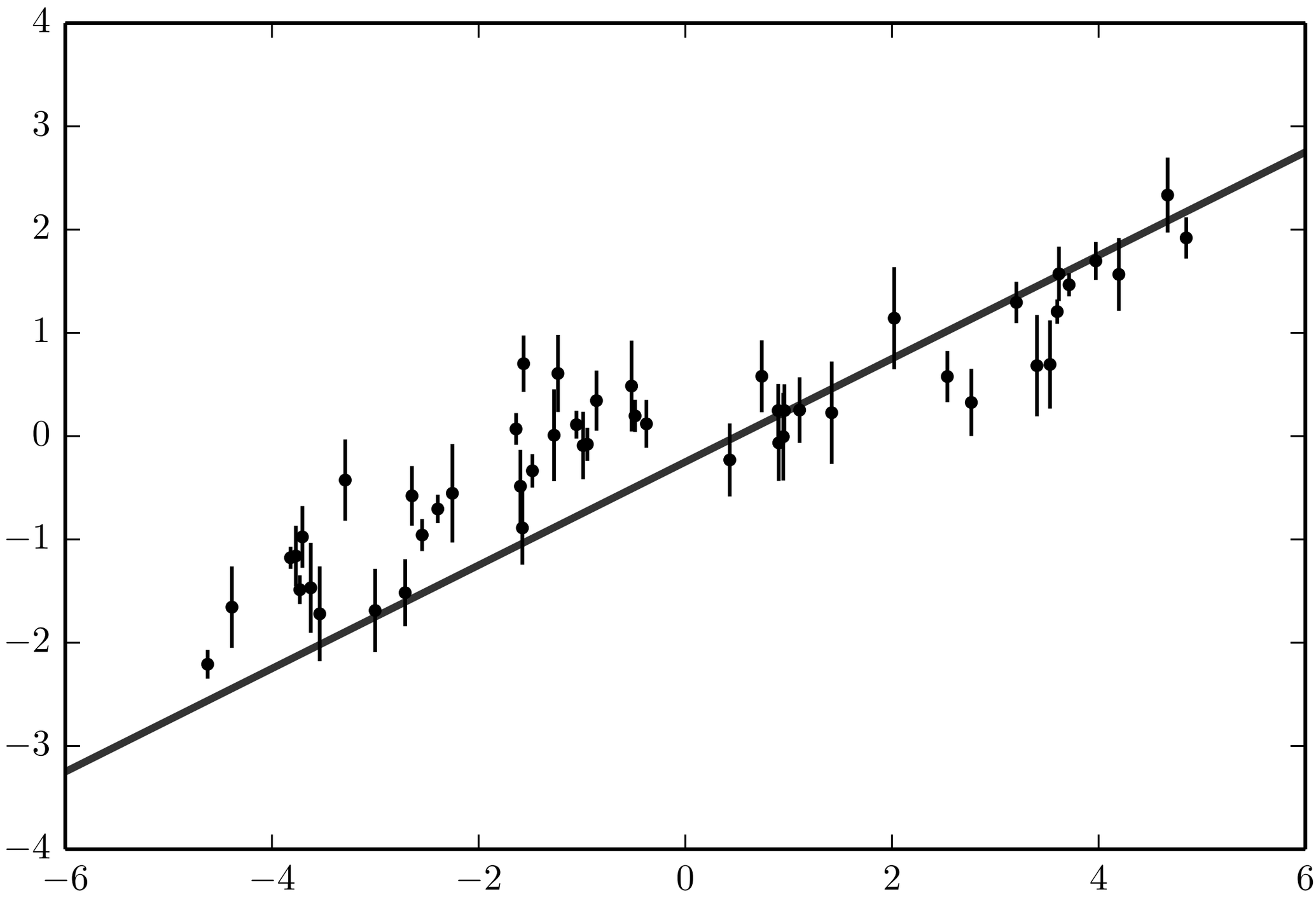
dan.iel.fm/emcee

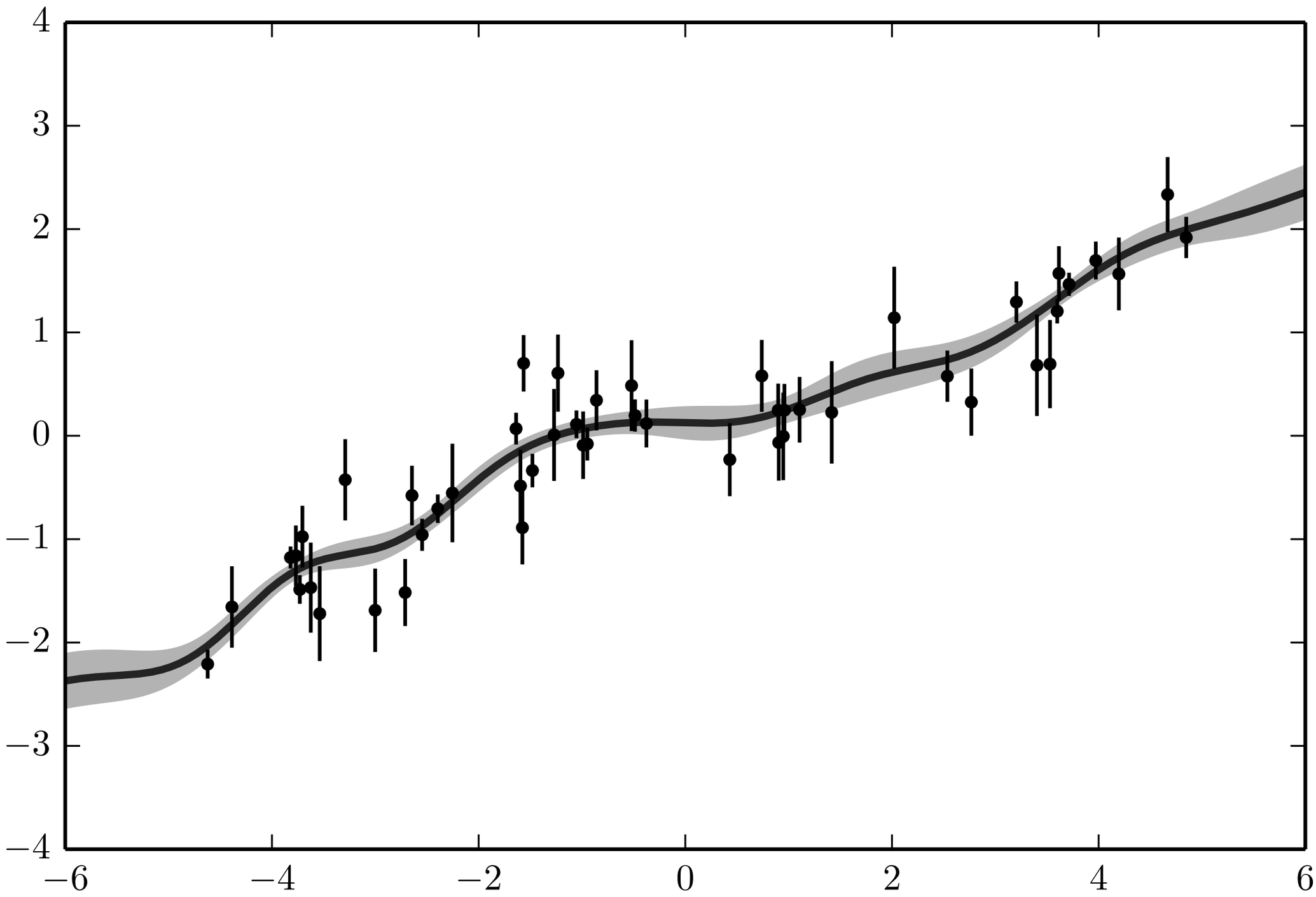

b
2
1
0
−1
−2
ln a
2.5
0.0
−2.5
ln s
2.5
0.0
−2.5
0.0
0.5
1.0
1.5
m
−2
−1
0
1
2
b
−2.5
0.0
2.5
ln a
−2.5
0.0
2.5
ln s


4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6


4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6

Prediction?


4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6


4
3
2
1
0
−1
−2
−3
−4
−6
−4
−2
0
2
4
6

2

The formal Gaussian process.

The model.

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta},\, \boldsymbol{\alpha}) = -\frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right]^{\mathrm{T}} K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})^{-1}\left[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right] - \frac{1}{2}\log\det K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma}) - \frac{N}{2}\log 2\,\pi$$

where

$$\left[K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right]_{ij} = {\sigma_i}^2\,\delta_{ij} + k_{\boldsymbol{\alpha}}(x_i,\, x_j)$$

drop-in replacement for your current log-likelihood function!

The model.

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{f_\theta}(\boldsymbol{x}),\, K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right)$$

where

$$\left[K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right]_{ij} = {\sigma_i}^2\, \delta_{ij} + k_{\boldsymbol{\alpha}}(x_i,\, x_j)$$

drop-in replacement for your current log-likelihood function!

the data are drawn from one

HUGE*

Gaussian

* the dimension is the number of data points.

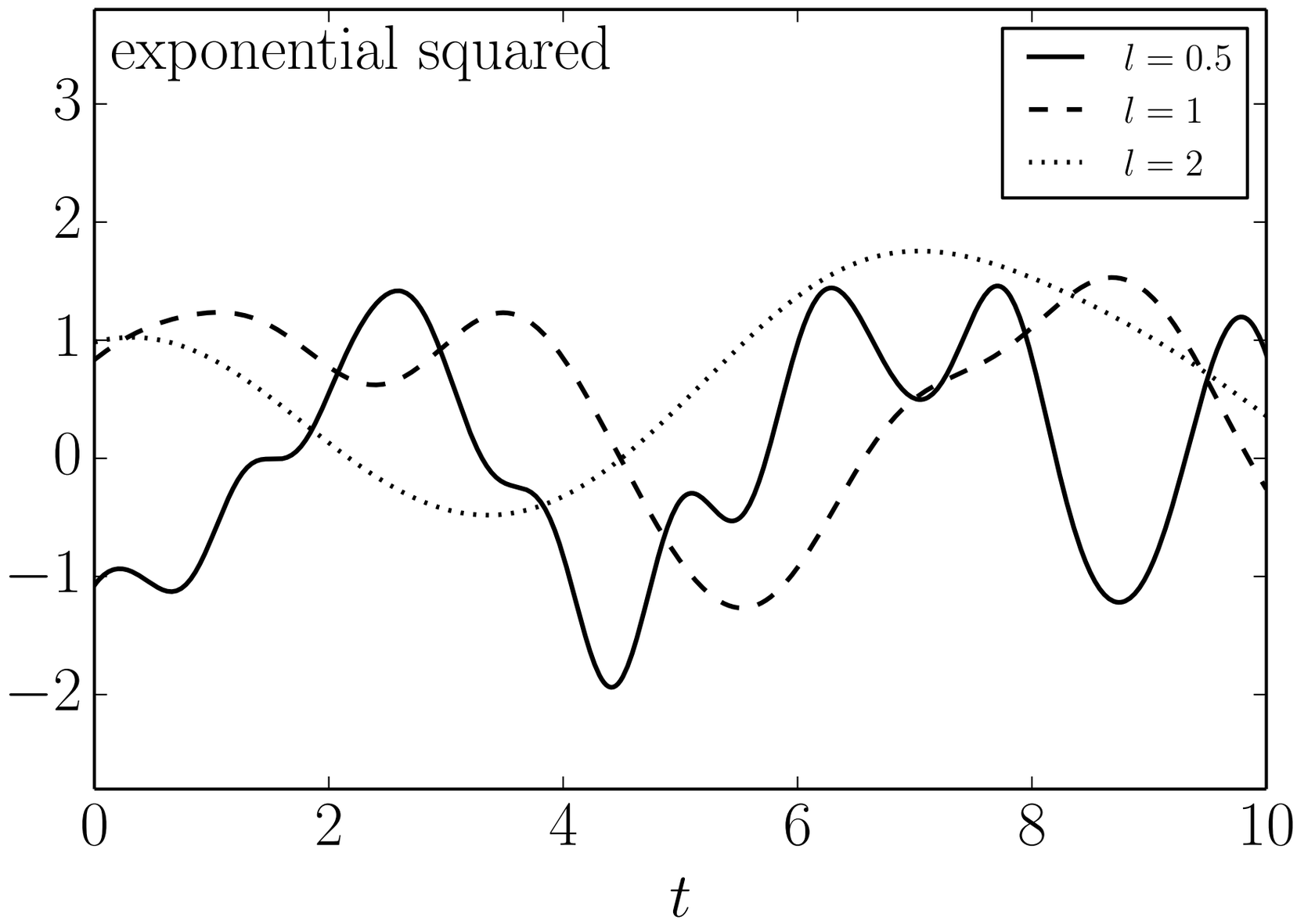
A generative model

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{f_\theta}(\boldsymbol{x}),\, K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right)$$

a probability distribution for *y* values

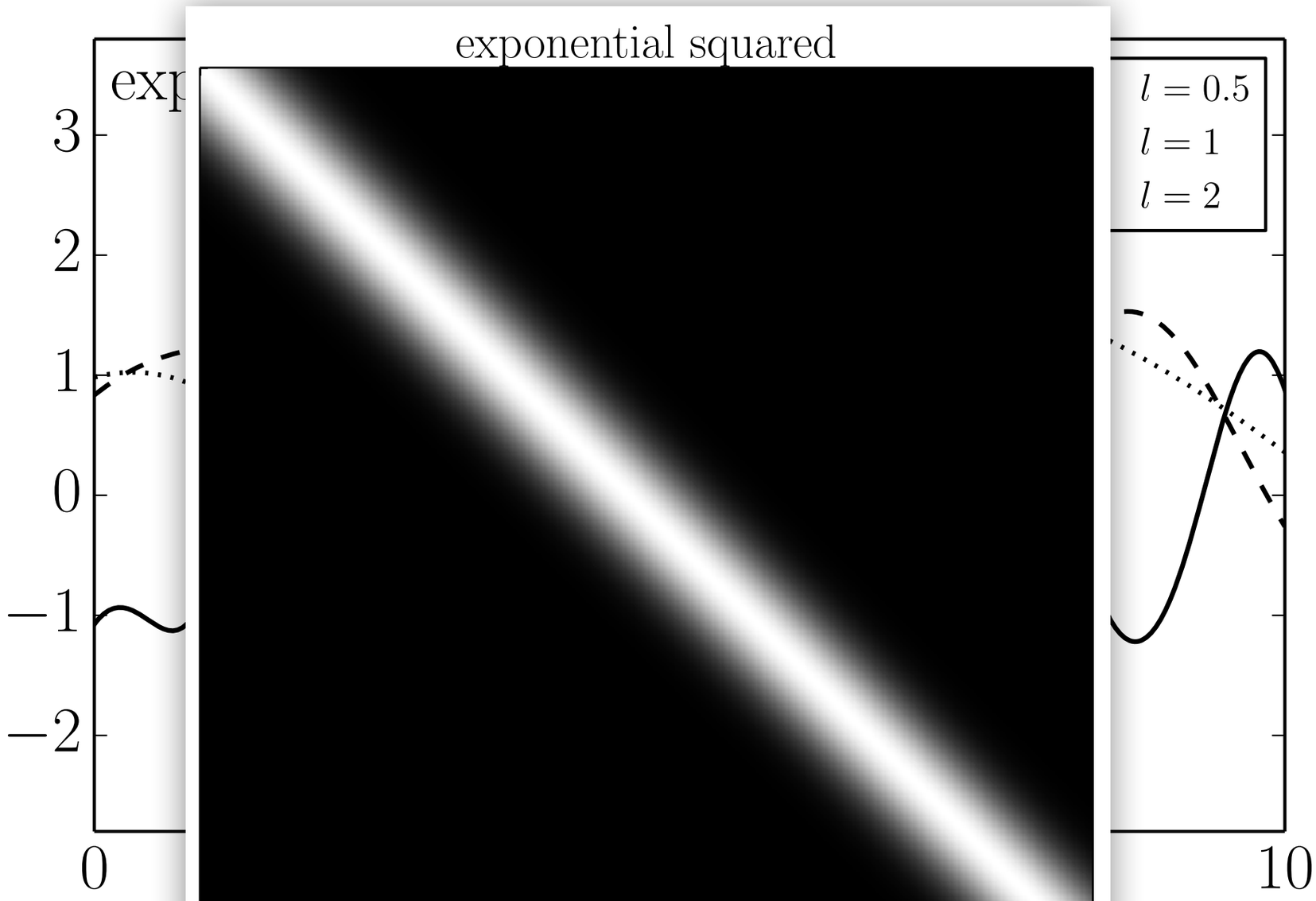
“Likelihood” samples.

$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \exp\left(-\frac{[x_i - x_j]^2}{2\,\ell^2}\right)$$



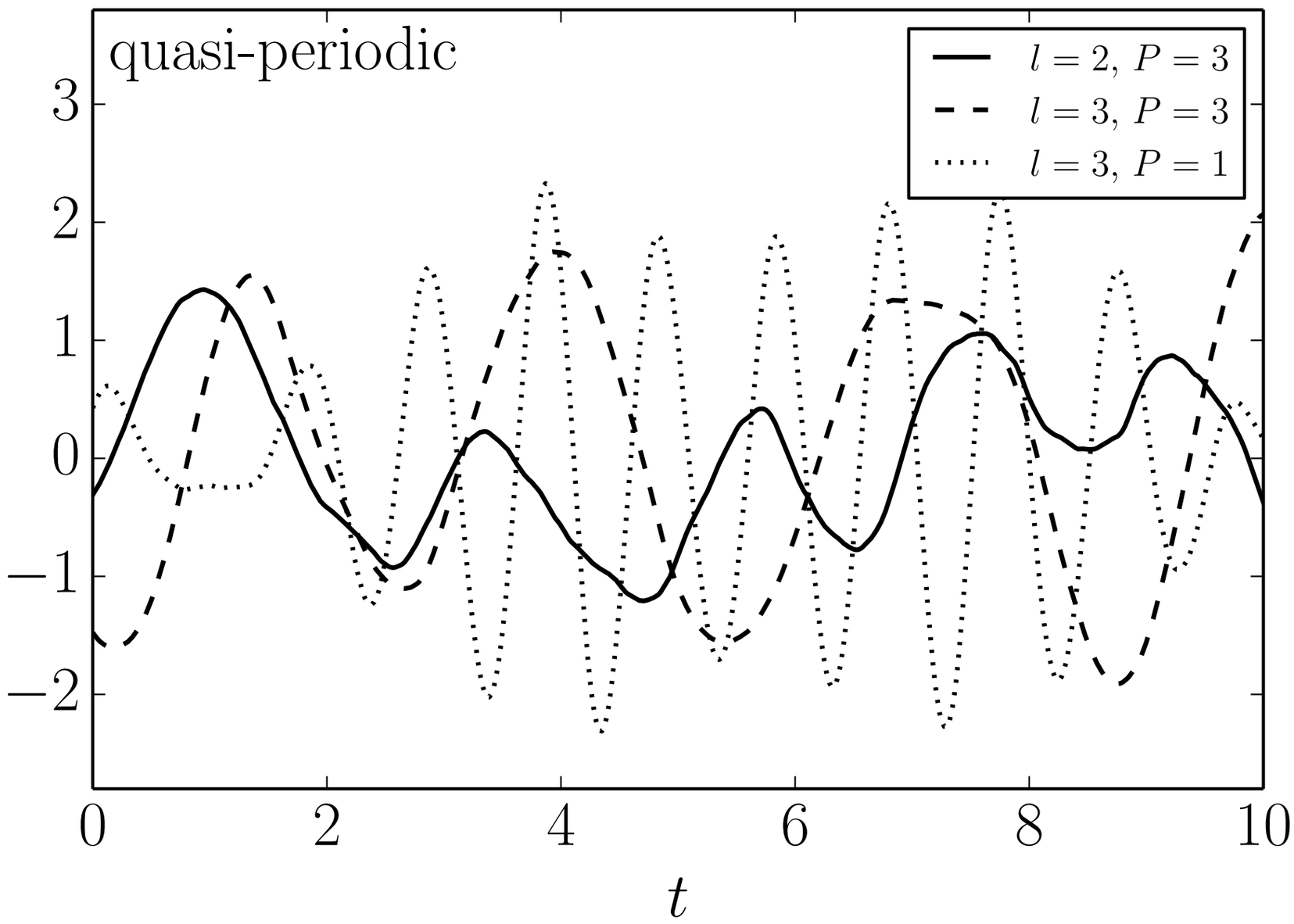
“Likelihood” samples.

$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \exp\left(-\frac{[x_i - x_j]^2}{2\,\ell^2}\right)$$



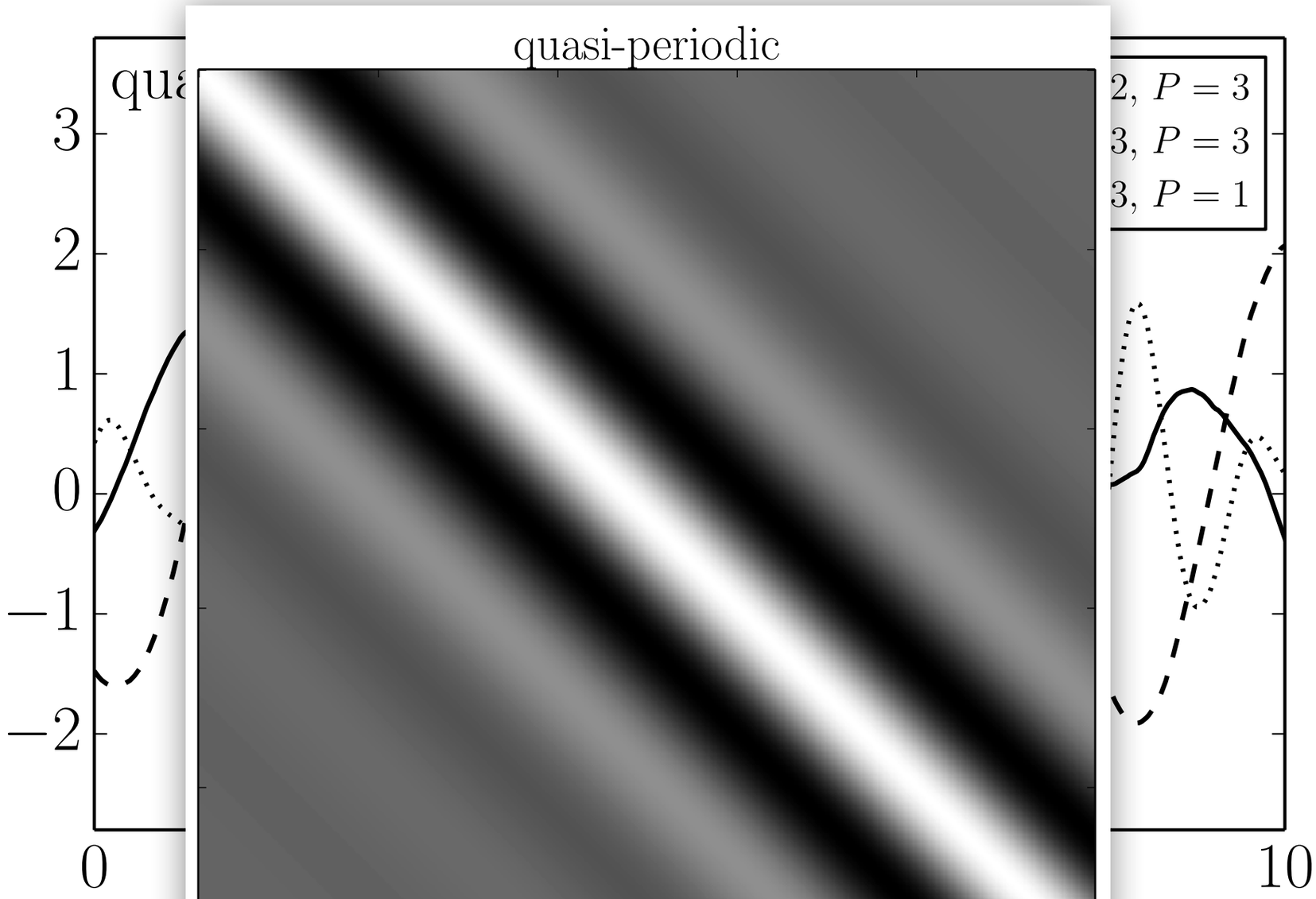
“Likelihood” samples.

$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \left[1 + \frac{\sqrt{3}\,|x_i - x_j|}{\ell}\right] \exp\left(-\frac{|x_i - x_j|}{\ell}\right) \cos\left(\frac{2\,\pi\,|x_i - x_j|}{P}\right)$$



“Likelihood” samples.

$$k_{\boldsymbol{\alpha}}(x_i,\, x_j) = \left[1 + \frac{\sqrt{3}\,|x_i - x_j|}{\ell}\right] \exp\left(-\frac{|x_i - x_j|}{\ell}\right) \cos\left(\frac{2\,\pi\,|x_i - x_j|}{P}\right)$$



The conditional distribution

$$\boldsymbol{y} \sim \mathcal{N}\left(\boldsymbol{f_\theta}(\boldsymbol{x}),\, K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})\right)$$

$$\downarrow$$

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}_\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f_\theta}(\boldsymbol{x}) \\ \boldsymbol{f_\theta}(\boldsymbol{x}_\star) \end{bmatrix}, \begin{bmatrix} K_{\boldsymbol{\alpha},\boldsymbol{x},\boldsymbol{x}} & K_{\boldsymbol{\alpha},\boldsymbol{x},\star} \\ K_{\boldsymbol{\alpha},\star,\boldsymbol{x}} & K_{\boldsymbol{\alpha},\star,\star} \end{bmatrix}\right)$$

$$\downarrow$$

$$\boldsymbol{y}_\star \,|\, \boldsymbol{y} \sim \mathcal{N}\left(K_{\boldsymbol{\alpha},\star,\boldsymbol{x}}\, K^{-1}_{\boldsymbol{\alpha},\boldsymbol{x},\boldsymbol{x}}\, [\boldsymbol{y} - \boldsymbol{f_\theta}(\boldsymbol{x})] + \boldsymbol{f_\theta}(\boldsymbol{x}_\star),\right.$$
$$\left. K_{\boldsymbol{\alpha},\star,\star} - K_{\boldsymbol{\alpha},\star,\boldsymbol{x}}\, K^{-1}_{\boldsymbol{\alpha},\boldsymbol{x},\boldsymbol{x}}\, K_{\boldsymbol{\alpha},\boldsymbol{x},\star}\right)$$

just see Rasmussen & Williams (Chapter 2)

What's the catch?

Kepler

=

Big Data

(by some definition)

Note: I hate myself for this slide too…

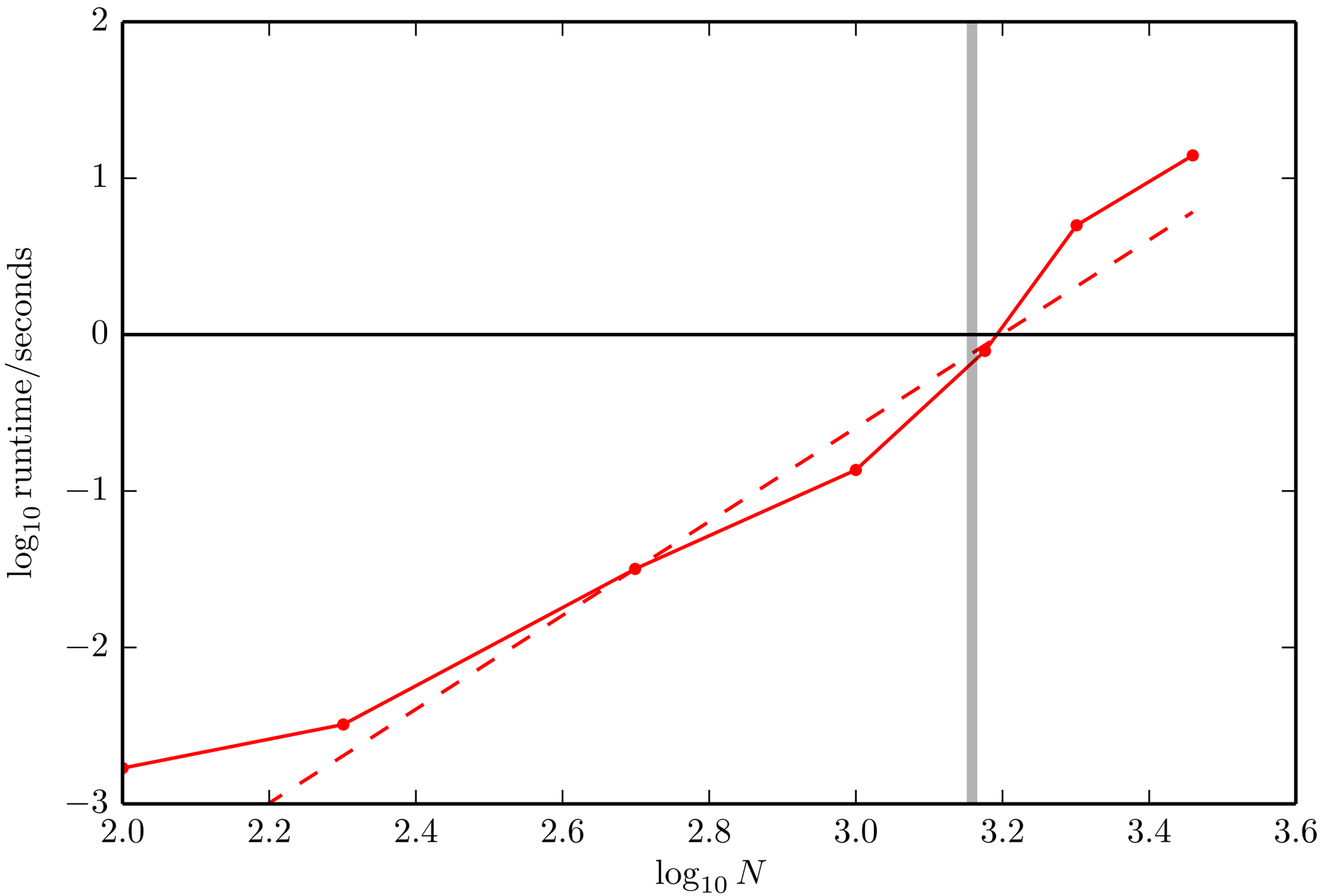
Computational complexity.

$$\log p(\boldsymbol{y} \,|\, \boldsymbol{x},\, \boldsymbol{\sigma},\, \boldsymbol{\theta},\, \boldsymbol{\alpha}) = -\frac{1}{2}\left[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right]^{\mathrm{T}} K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma})^{-1}\left[\boldsymbol{y} - \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})\right] - \frac{1}{2}\log\det K_{\boldsymbol{\alpha}}(\boldsymbol{x},\, \boldsymbol{\sigma}) - \frac{N}{2}\log 2\,\pi$$

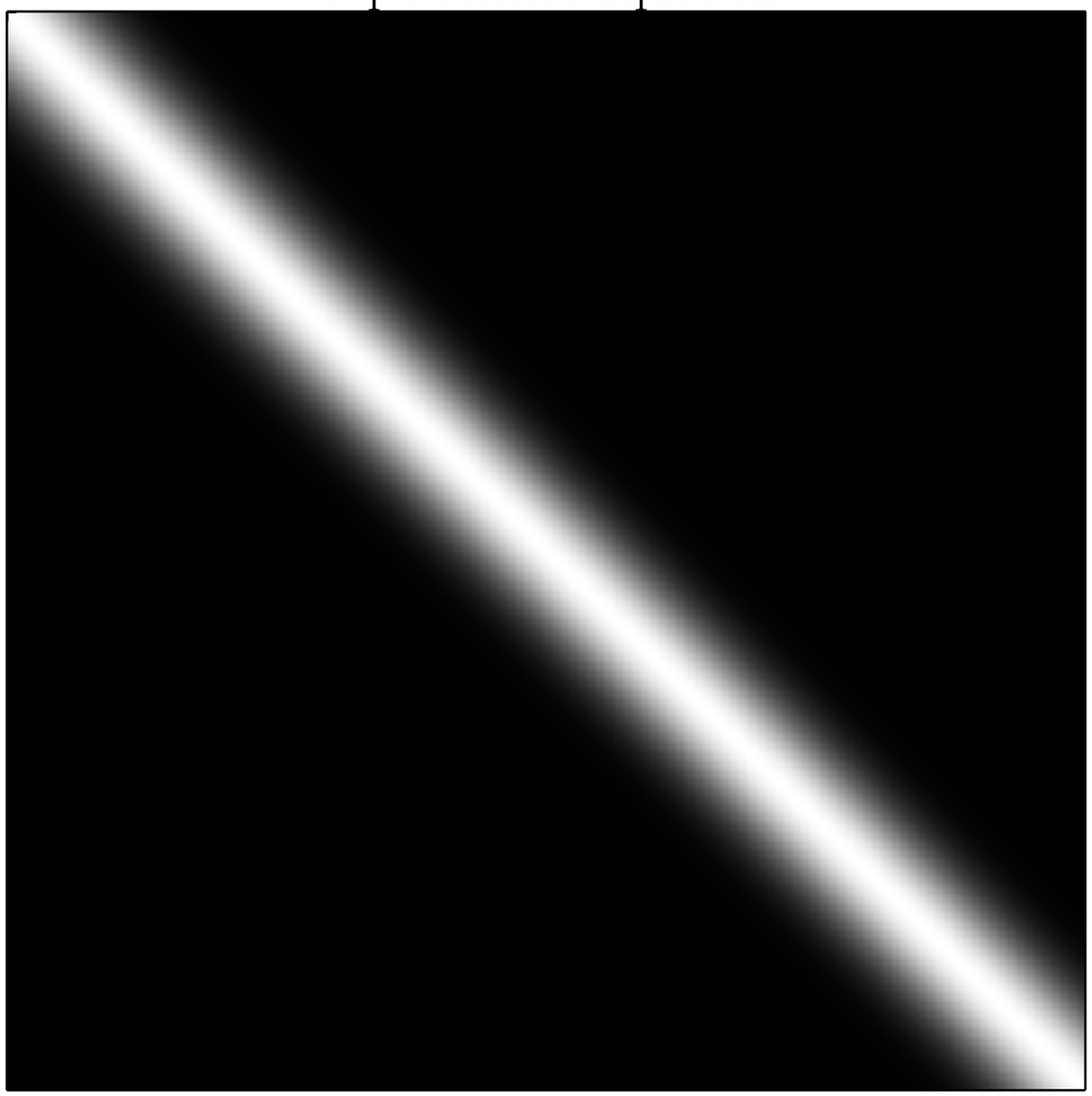
compute factorization // evaluate log-det // apply inverse

naïvely: $\mathcal{O}(N^3)$

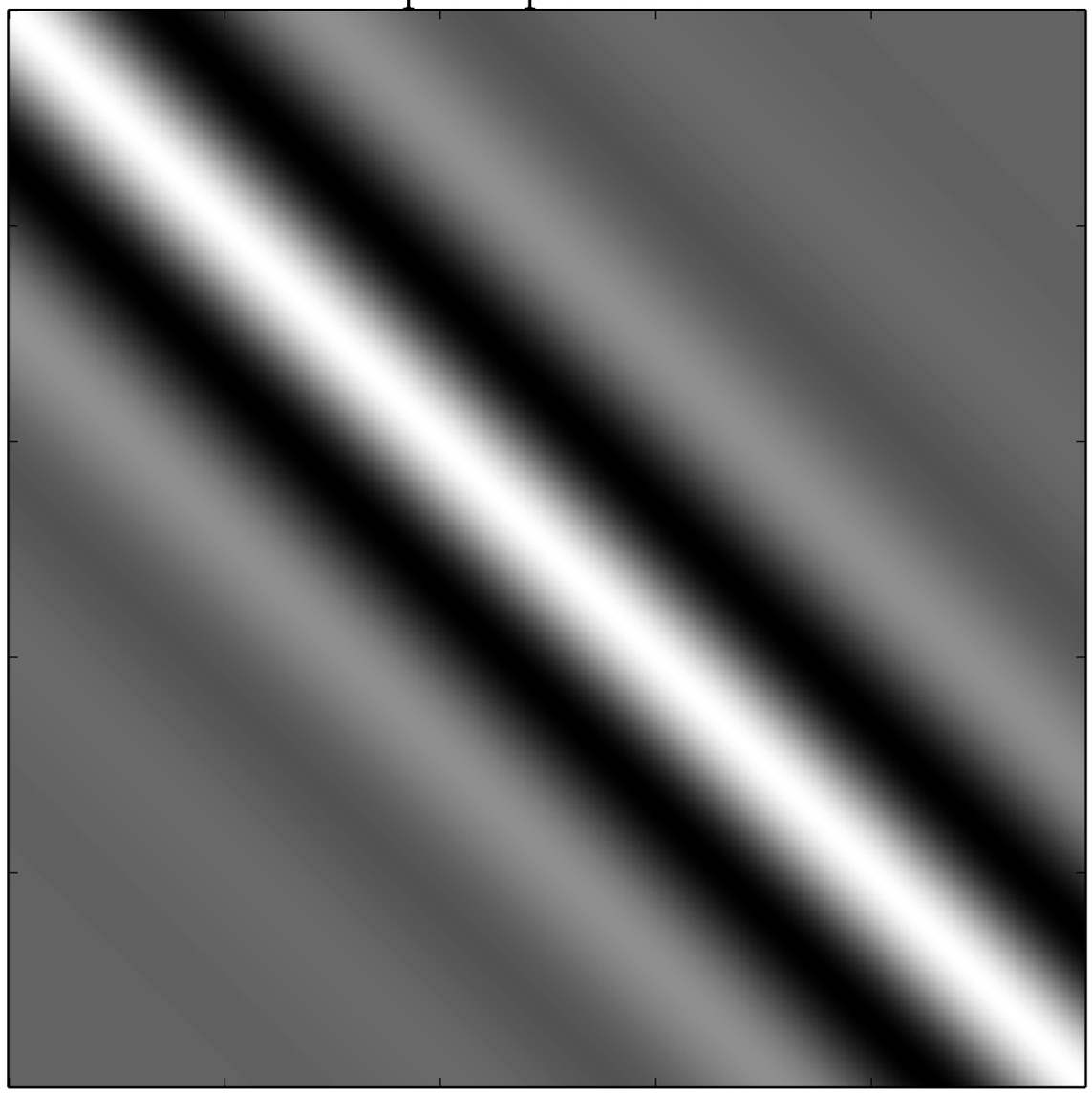
```python
import numpy as np
from scipy.linalg import cho_factor, cho_solve

def simple_gp_lnlike(x, y, yerr, a, s):
    r = x[:, None] - x[None, :]
    C = np.diag(yerr**2) + a*np.exp(-0.5*r**2/(s*s))
    factor, flag = cho_factor(C)
    logdet = 2*np.sum(np.log(np.diag(factor)))
    return -0.5 * (np.dot(y, cho_solve((factor, flag), y))
                   + logdet + len(x)*np.log(2*np.pi))
```


2
1
0
−1
−2
−3
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
$\log_{10}$ runtime/seconds
$\log_{10} N$

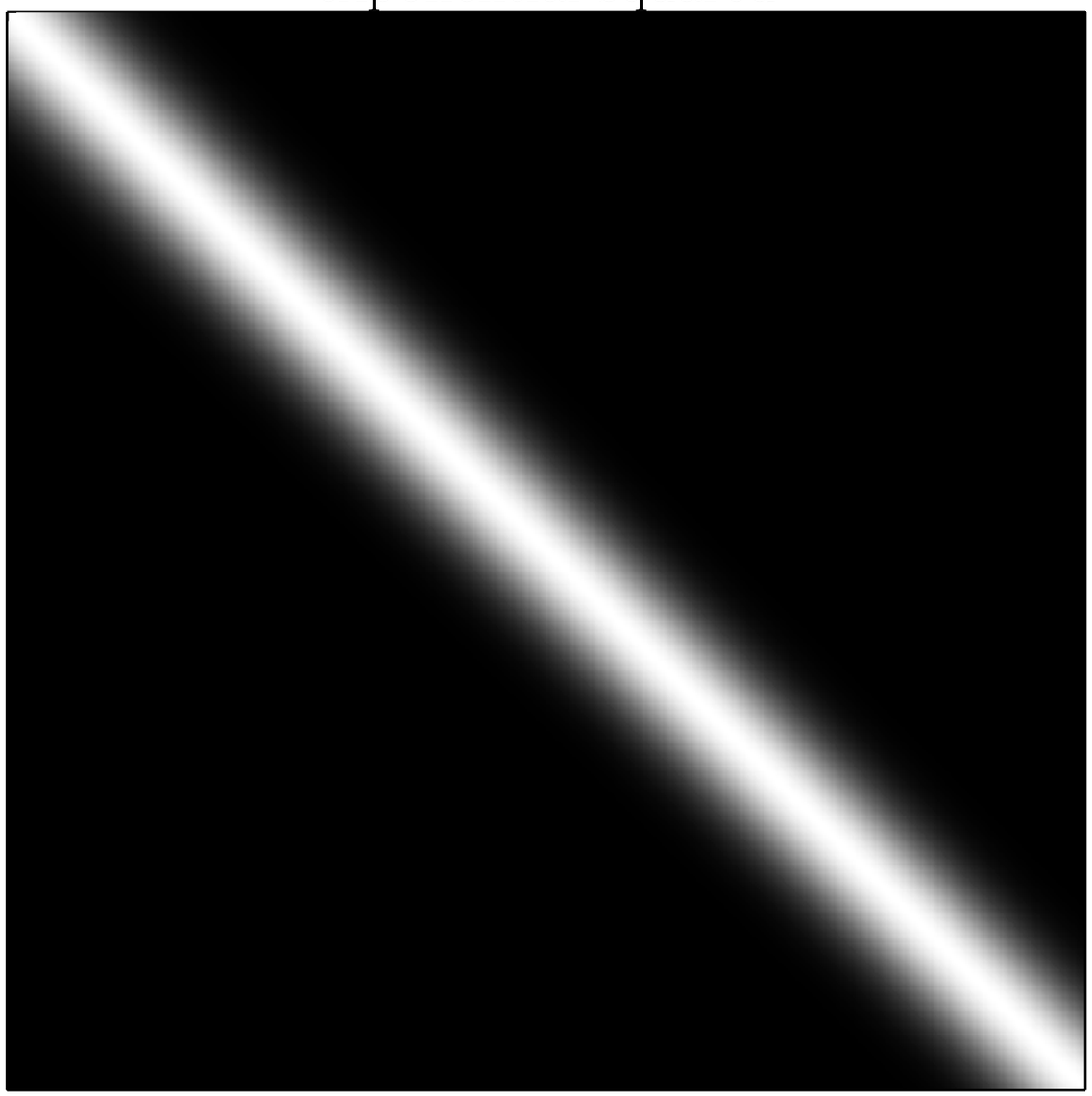
exponential squared



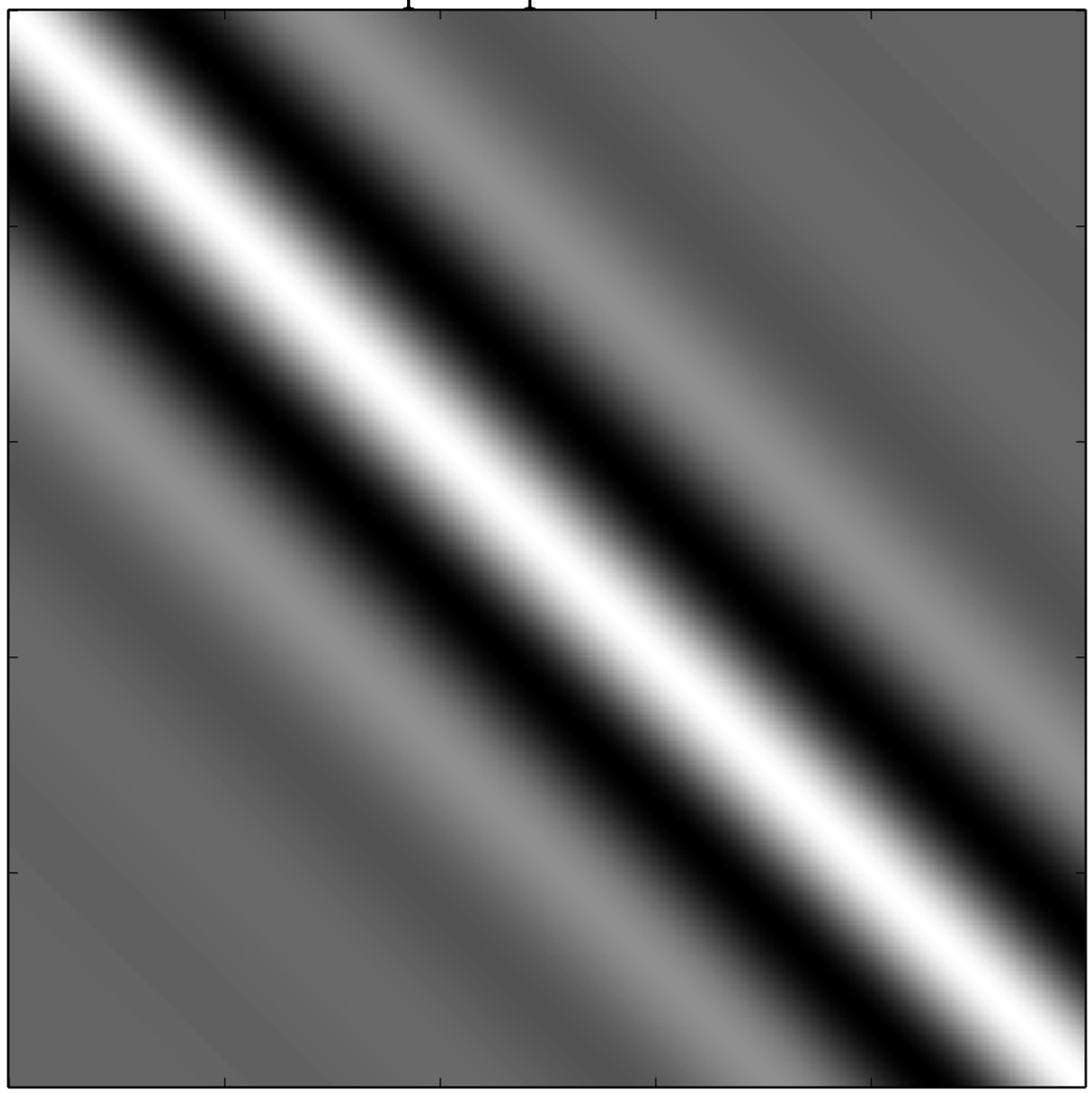
quasi-periodic



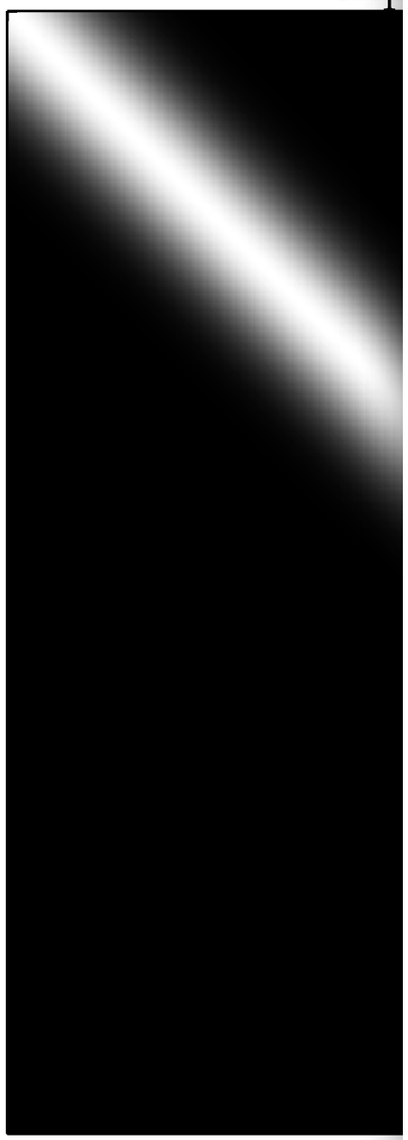
exponential squared

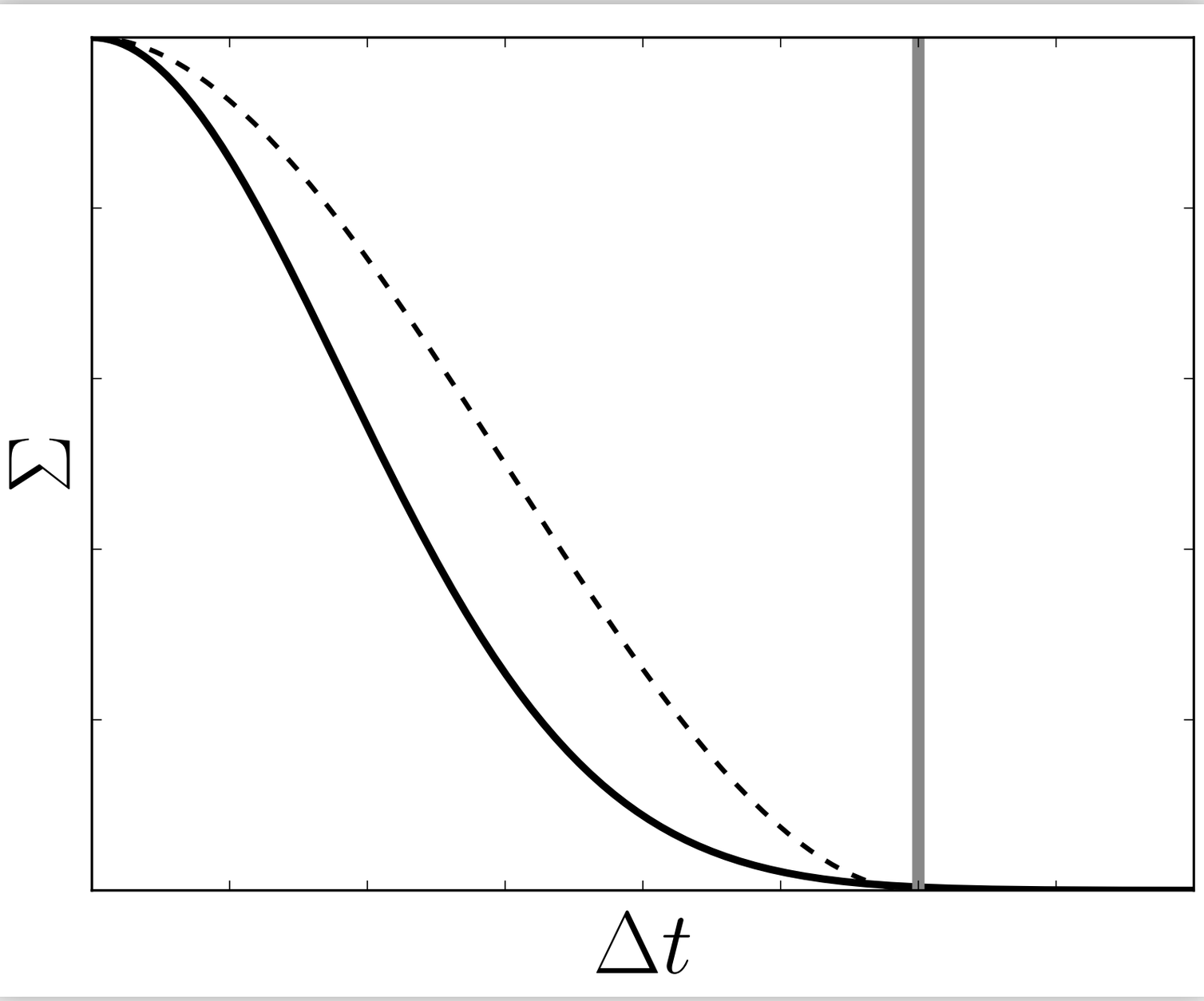


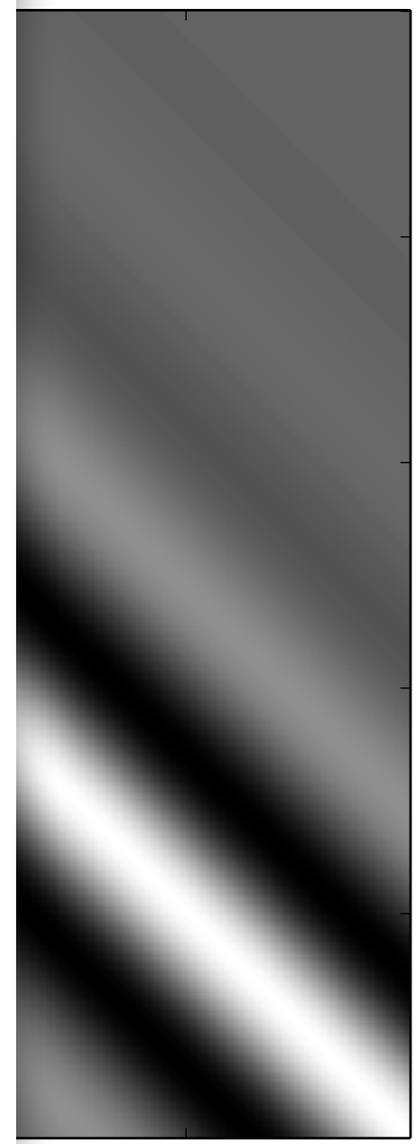
quasi-periodic



exp



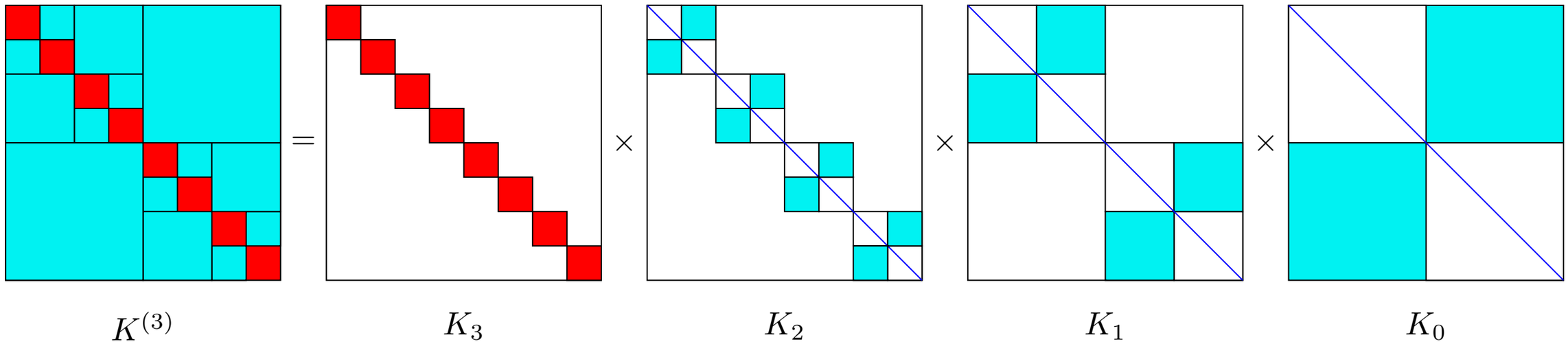




“

Aren’t kernel matrices Hierarchical Off-Diagonal Low-Rank?

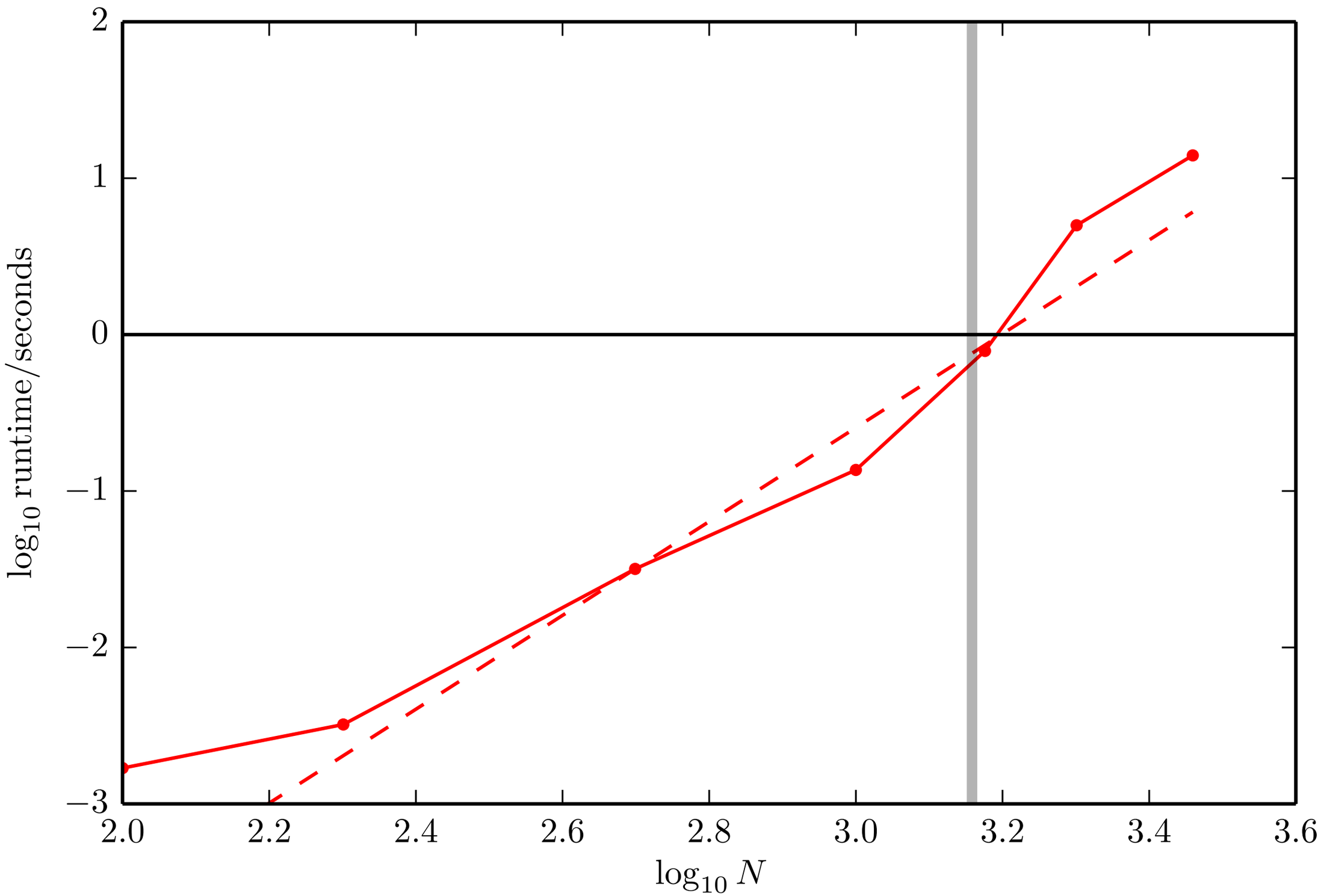
— no astronomer ever

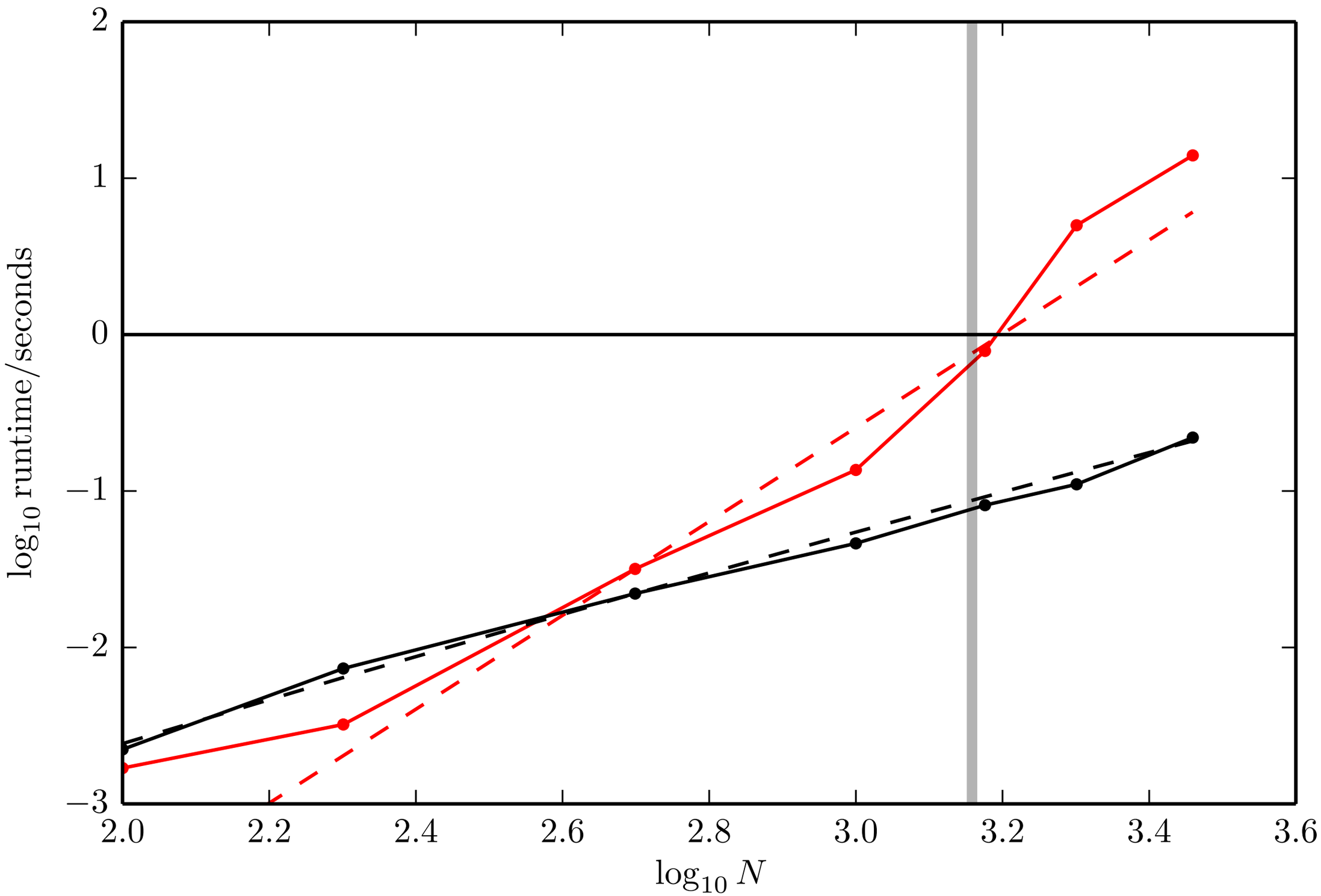


Ambikasaran, DFM, *et al.* (arXiv:1403.6015)

github.com/dfm/george

```python
import numpy as np
from george import GaussianProcess, kernels

def george_lnlike(x, y, yerr, a, s):
    kernel = a * kernels.RBFKernel(s)
    gp = GaussianProcess(kernel)
    gp.compute(x, yerr)
    return gp.lnlikelihood(y)
```


2
1
0
−1
−2
−3
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
$\log_{10}$ runtime/seconds
$\log_{10} N$


2
1
0
−1
−2
−3
2.0
2.2
2.4
2.6
2.8
3.0
3.2
3.4
3.6
$\log_{10}$ runtime/seconds
$\log_{10} N$

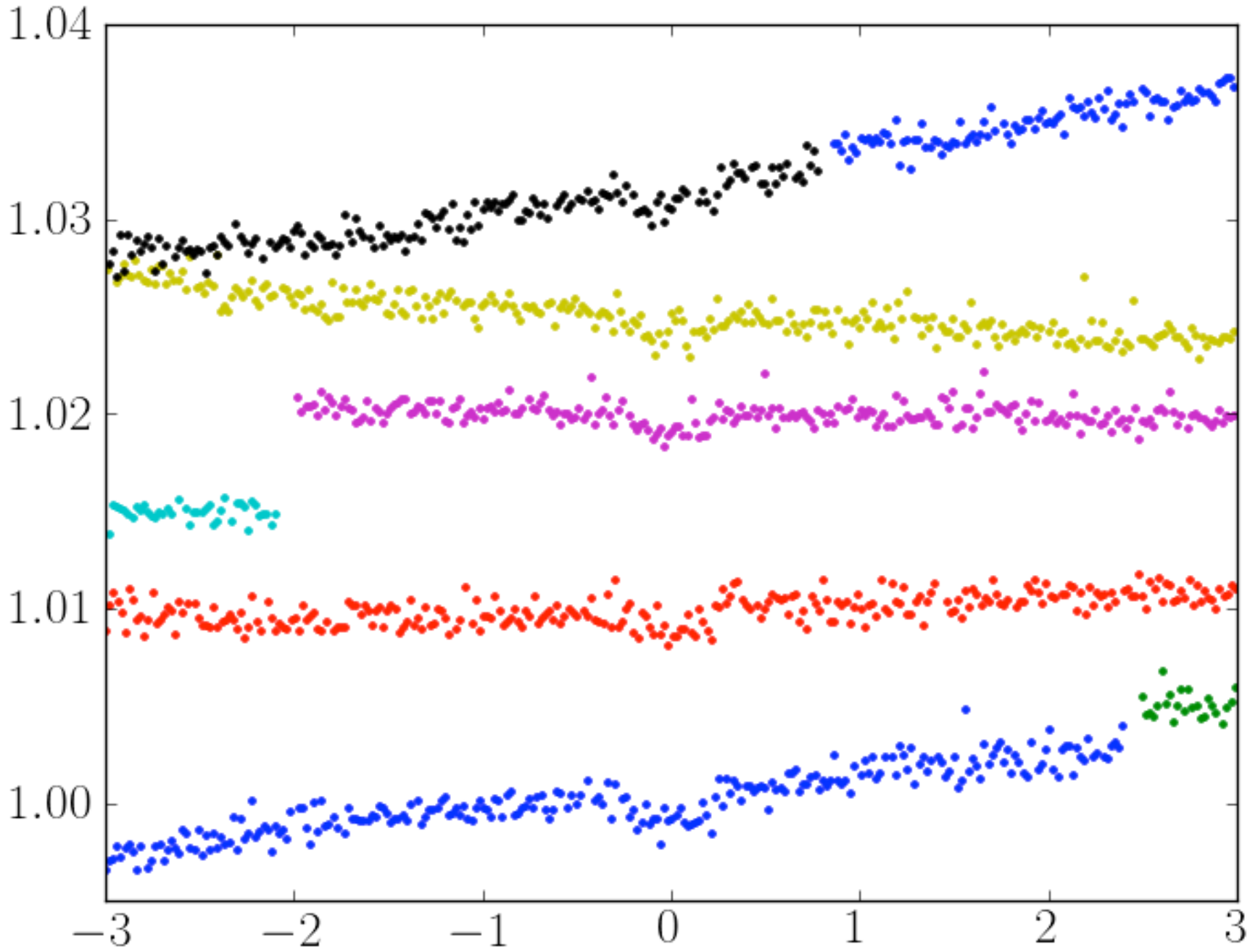
and short cadence data?

one month of data in 4 seconds

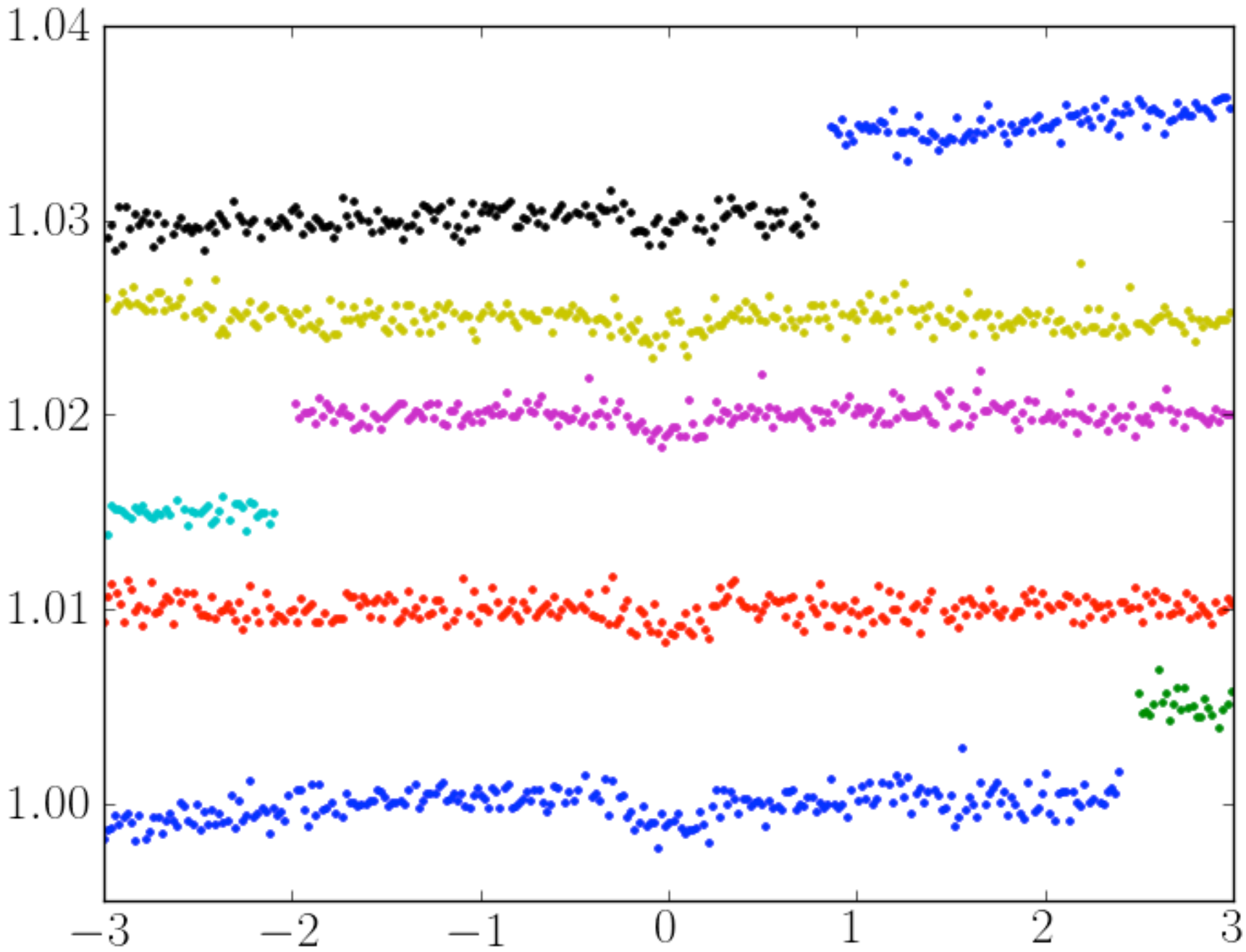
3

Applications to Kepler data.

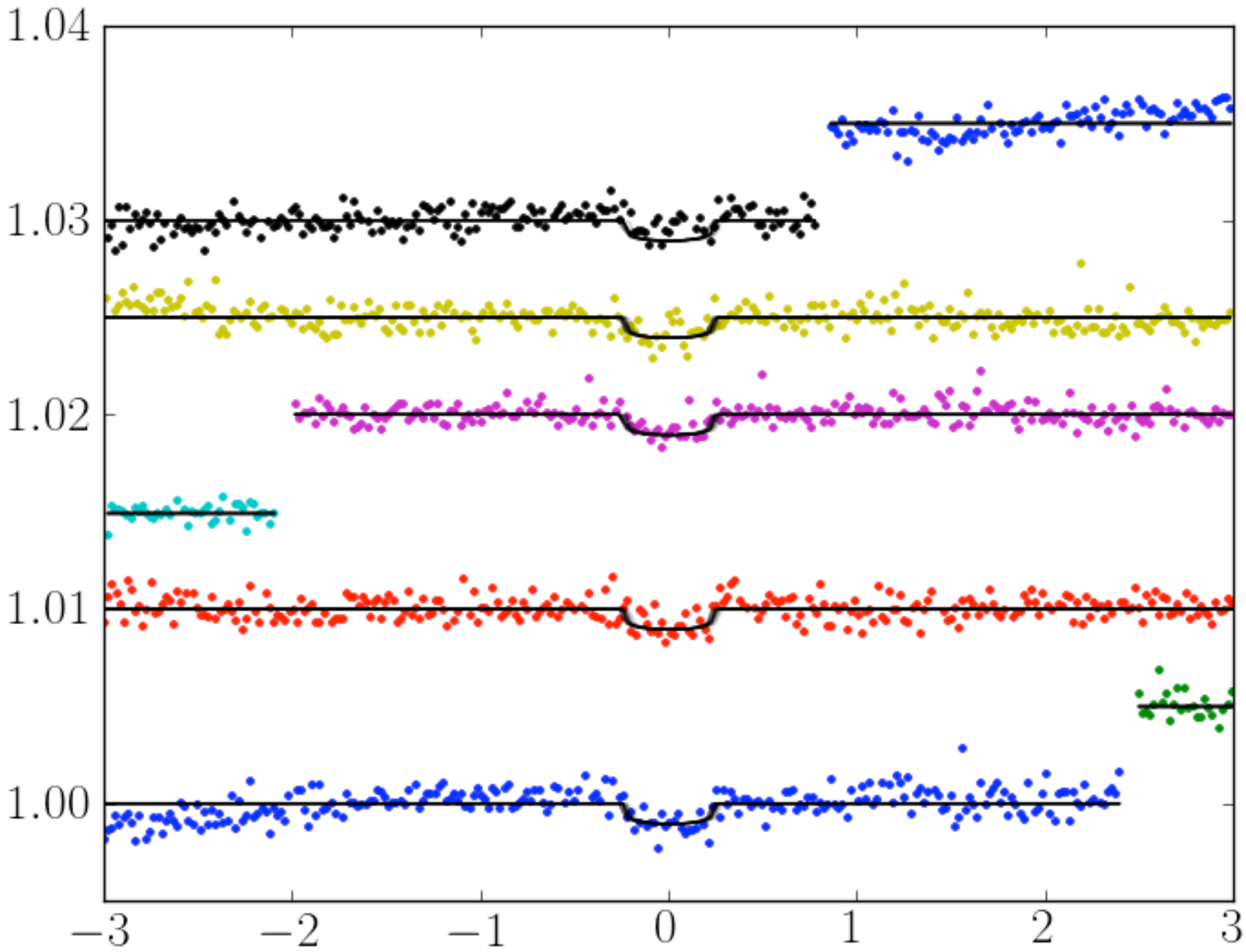
Parameter Recovery



KIC 2301306 + injection

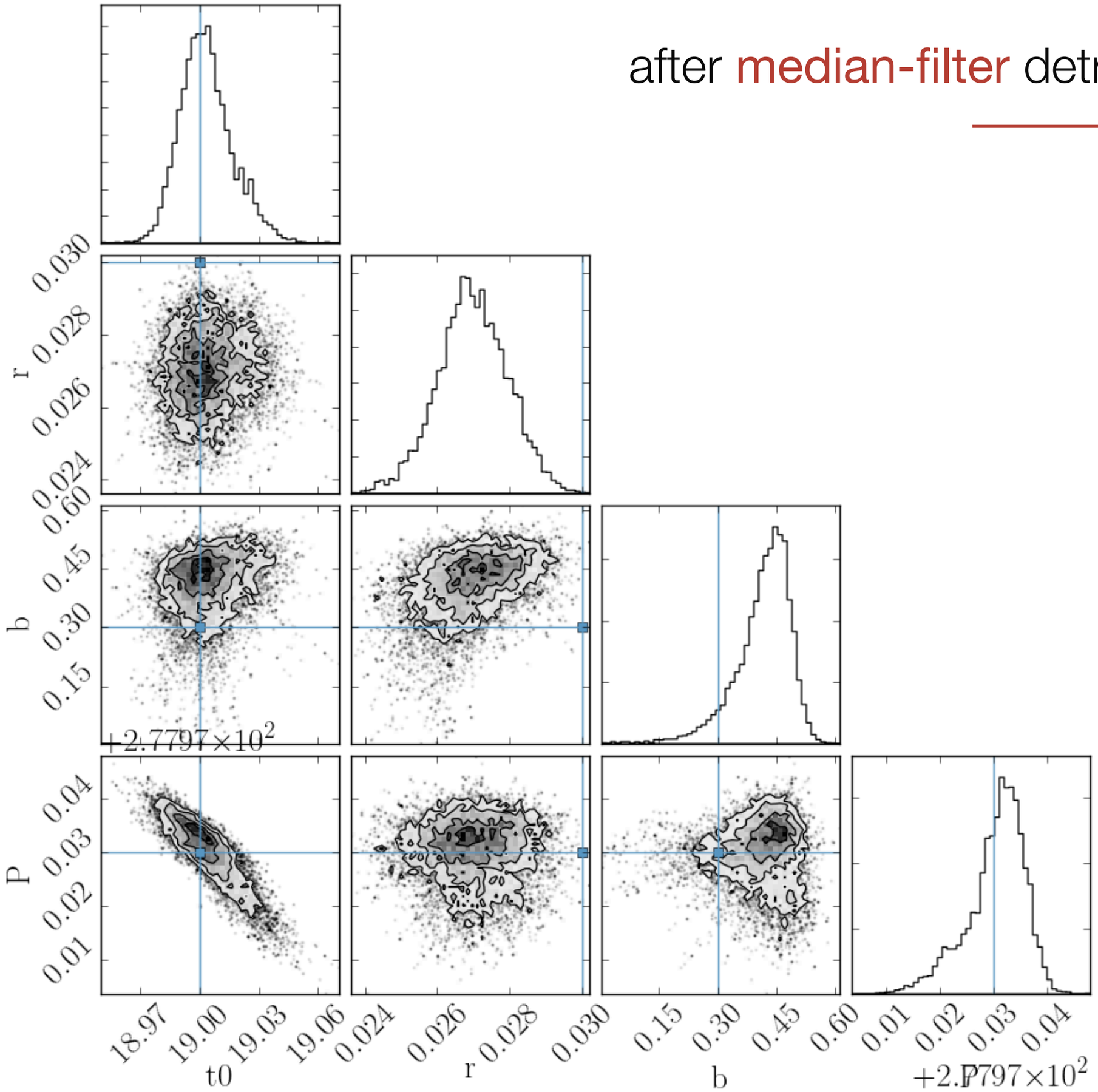


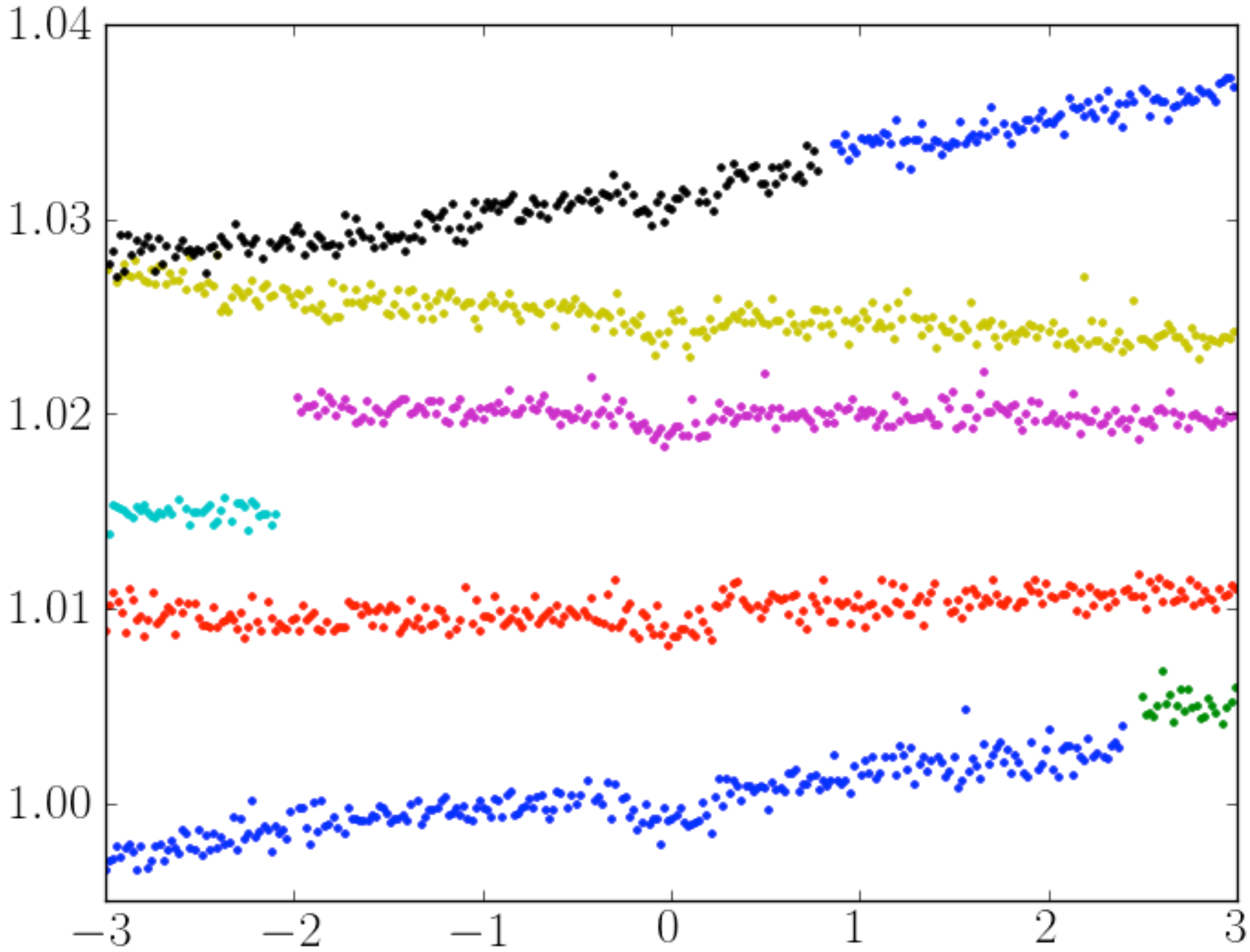
KIC 2301306 + injection



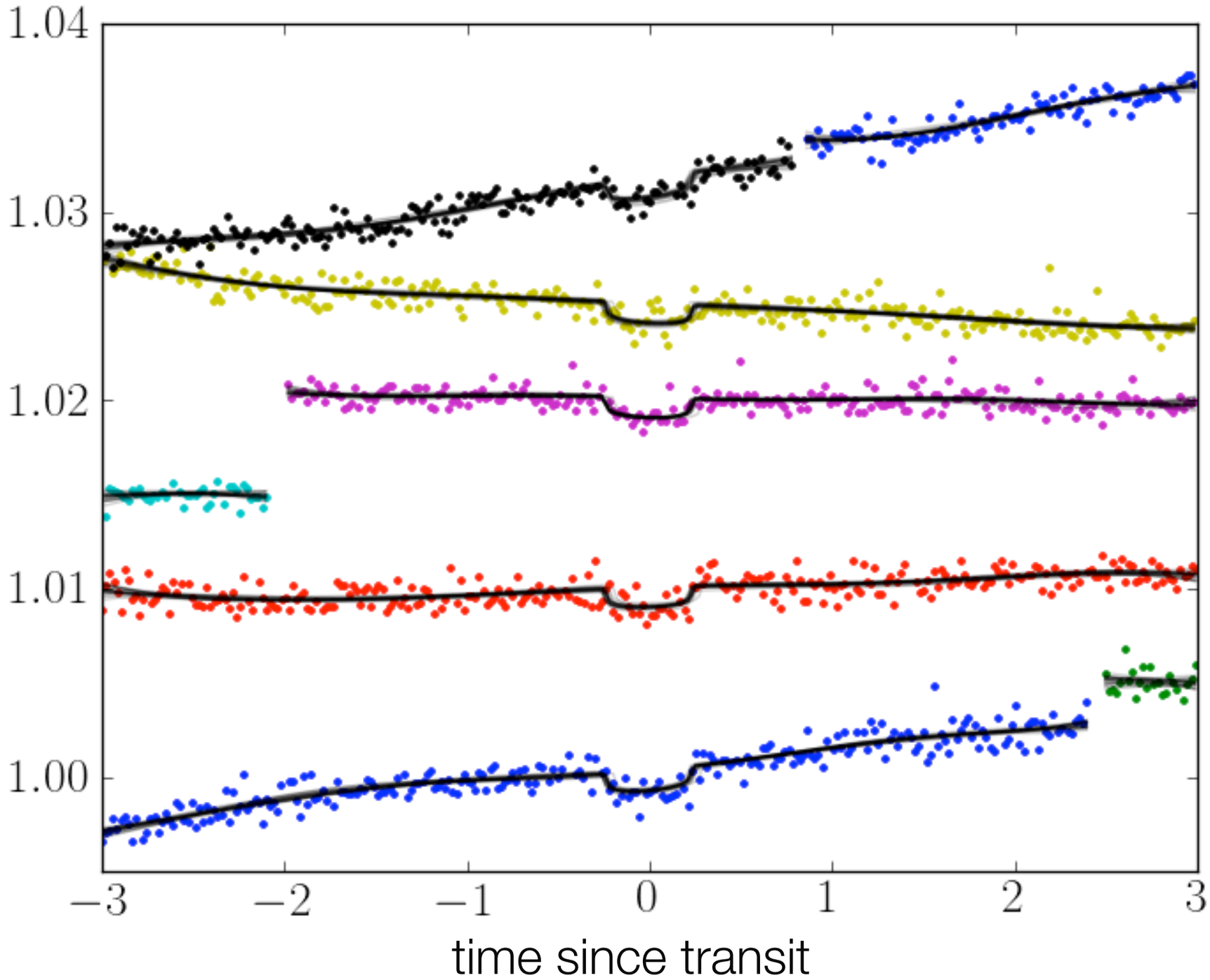
KIC 2301306 + injection

after median-filter detrending



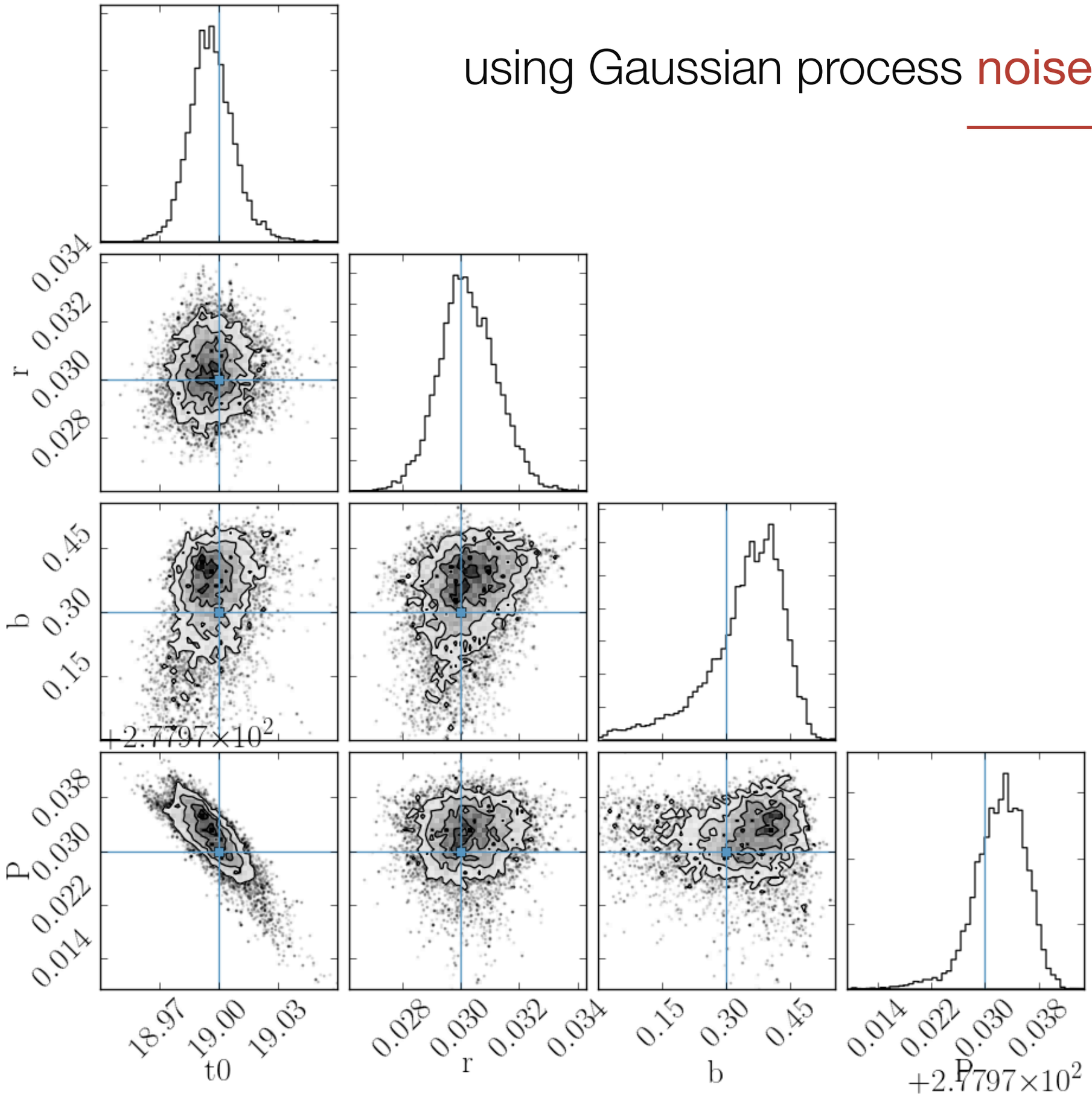


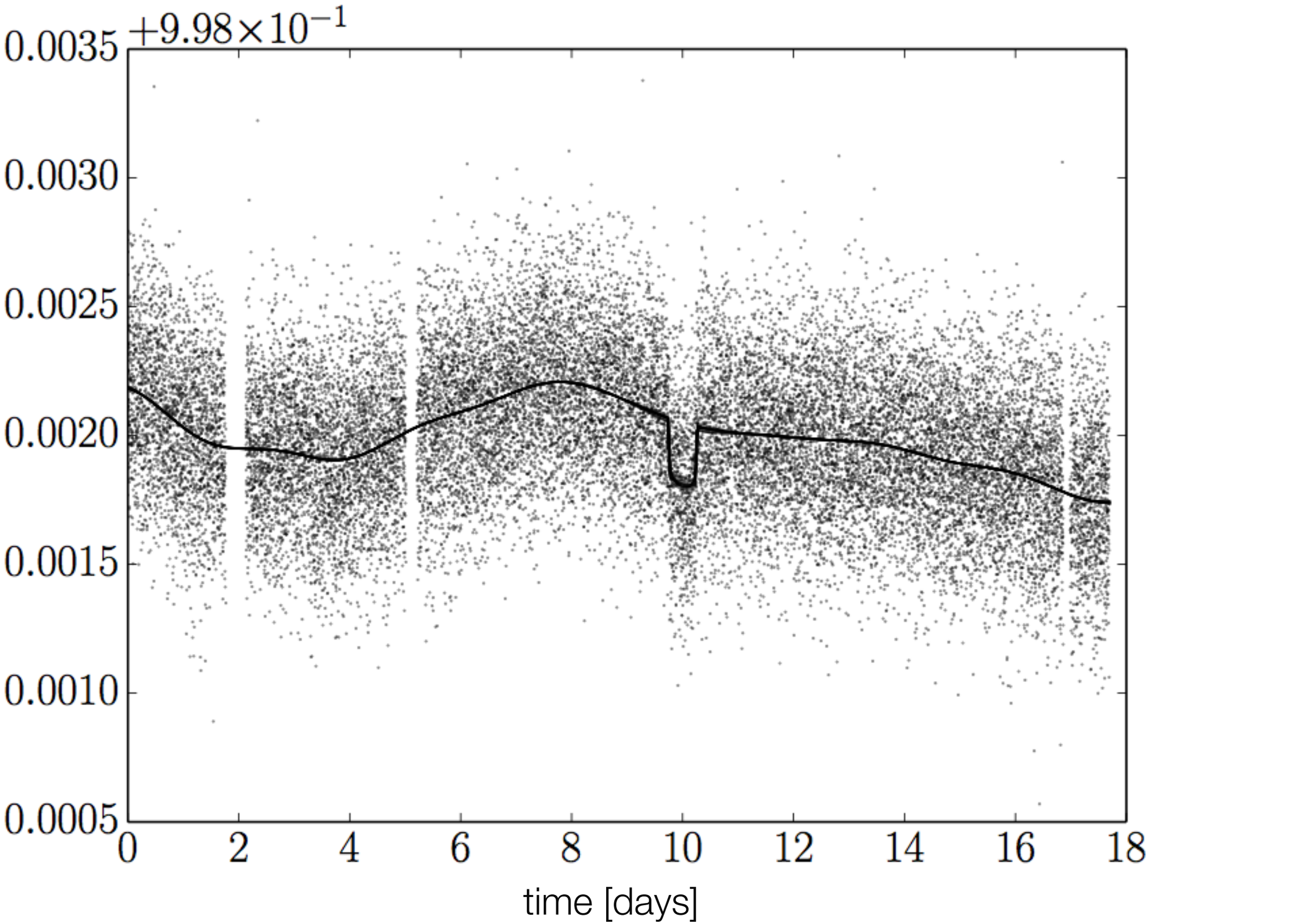
KIC 2301306 + injection



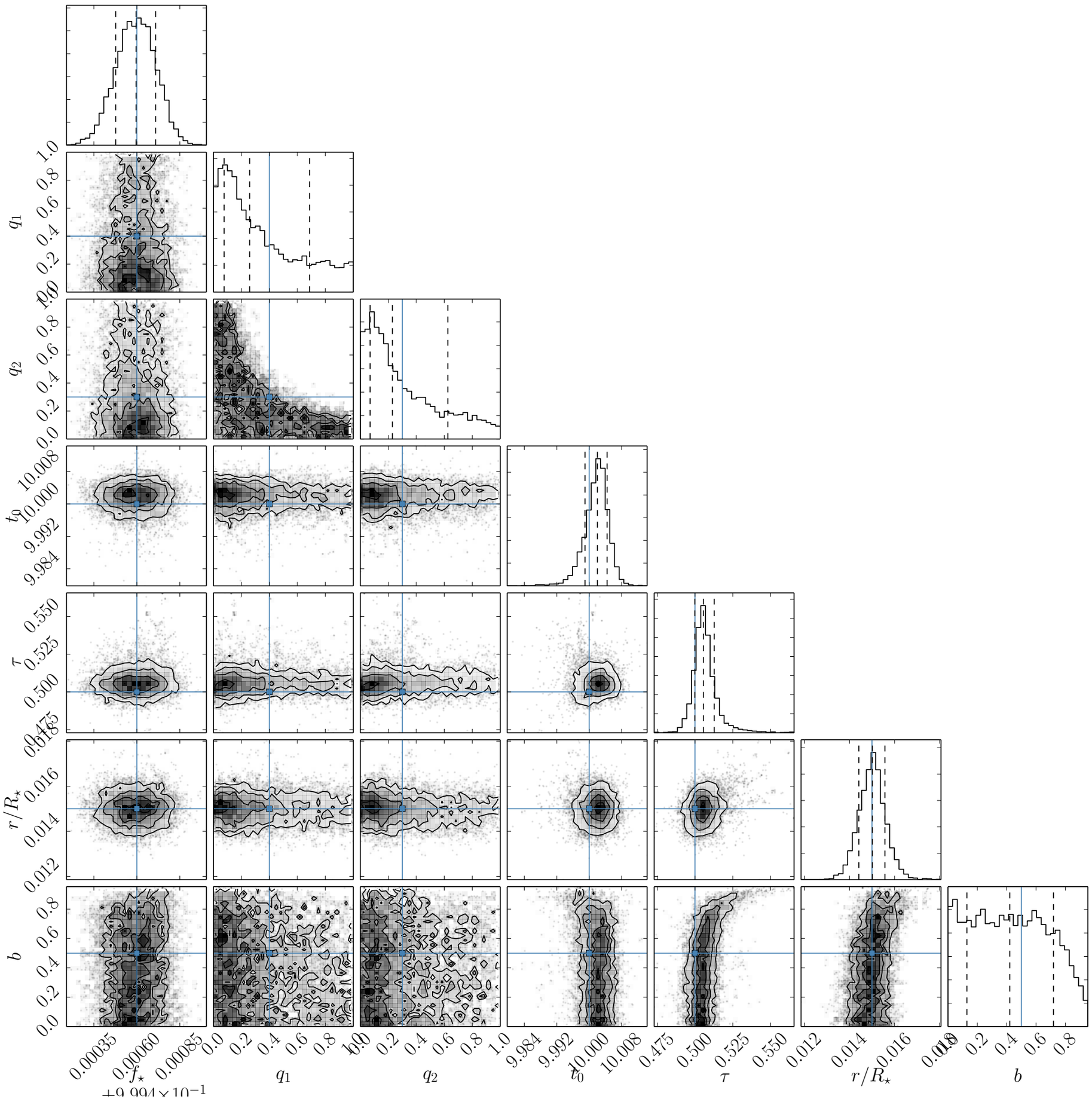
KIC 2301306 + injection

using Gaussian process noise model



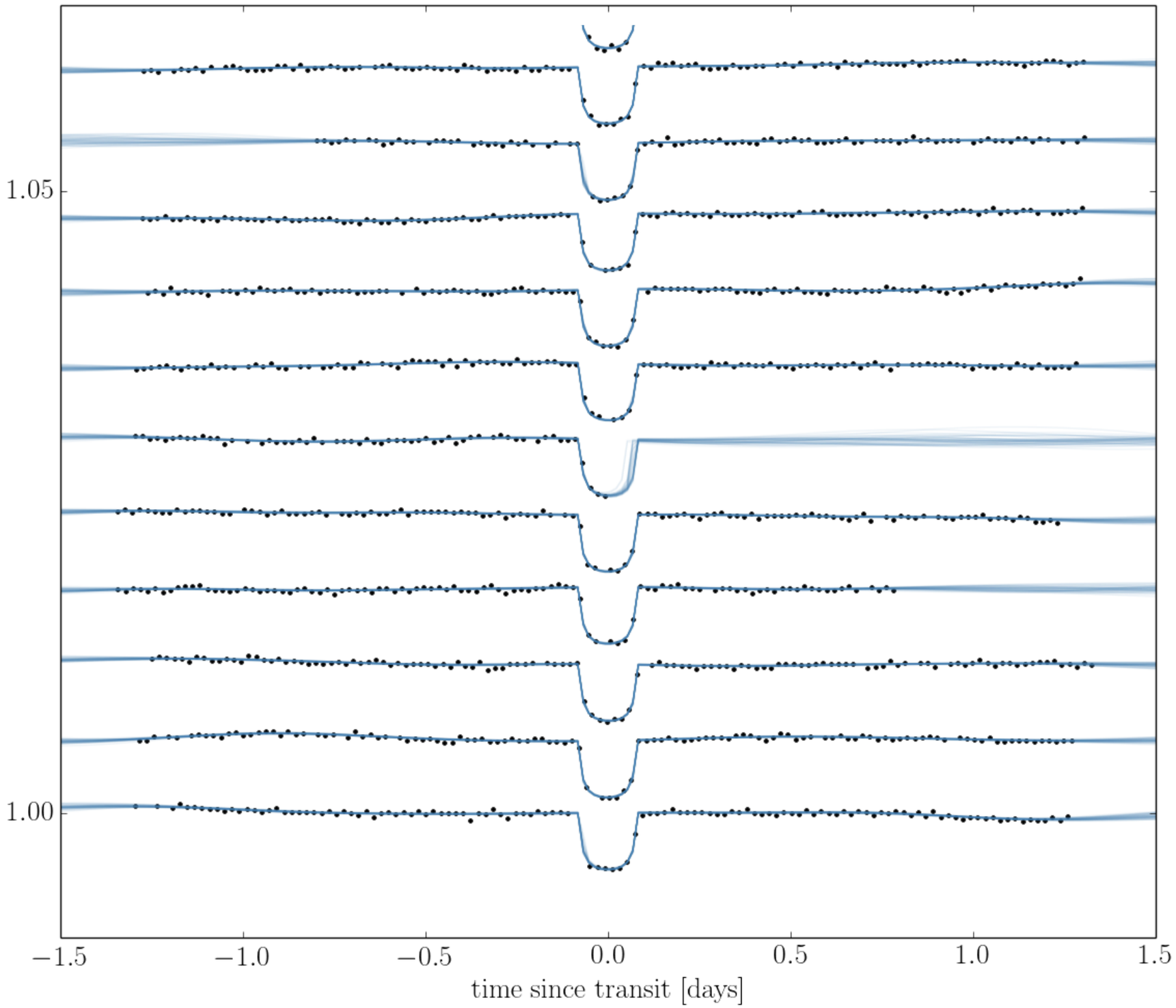


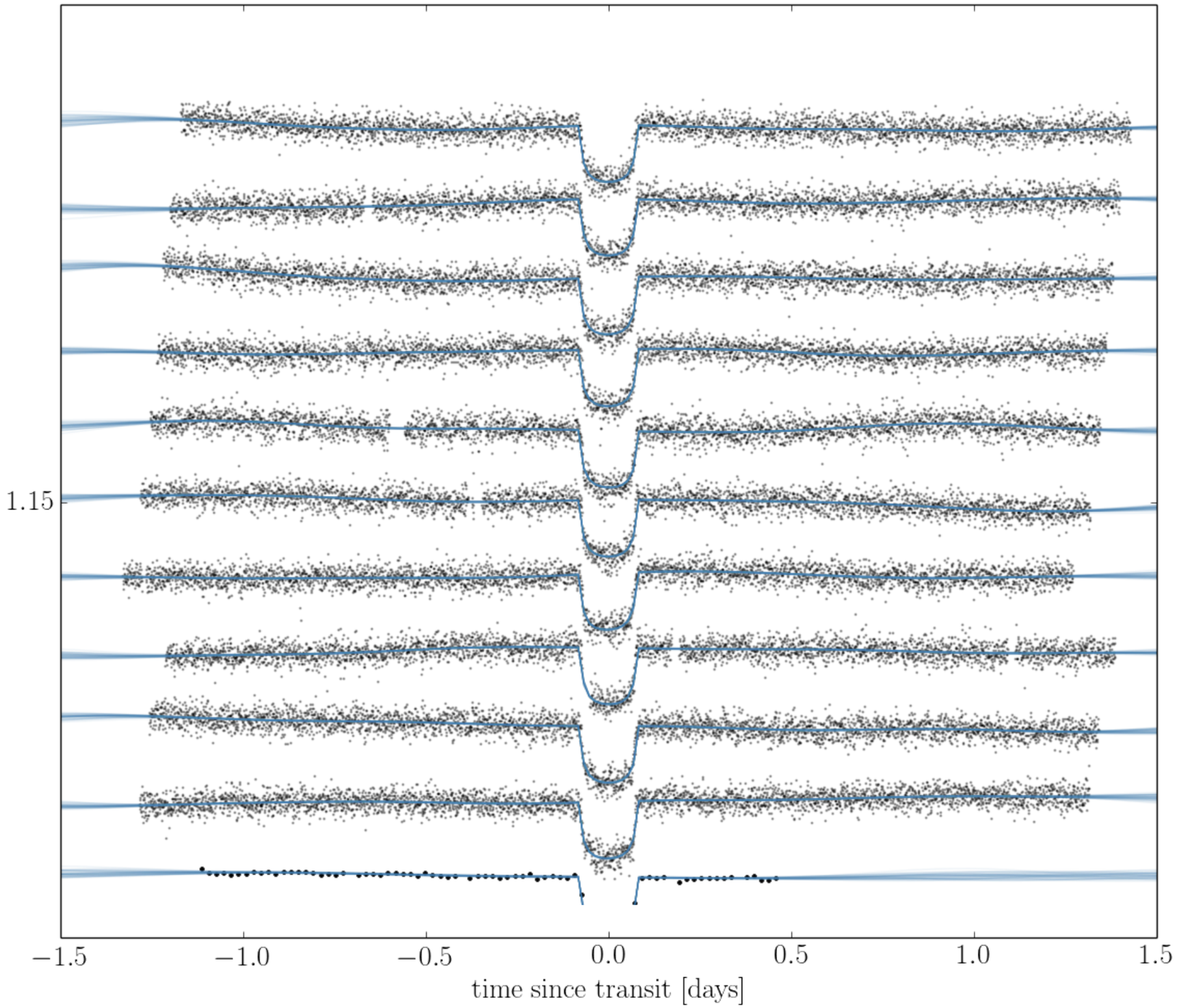
Ambikasaran, DFM, *et al.* (arXiv:1403.6015)


$f_\star$
$+9.994\times10^{-1}$
q_1
q_2
t_0
τ
$r/R_\star$
b

KOI 1474.01

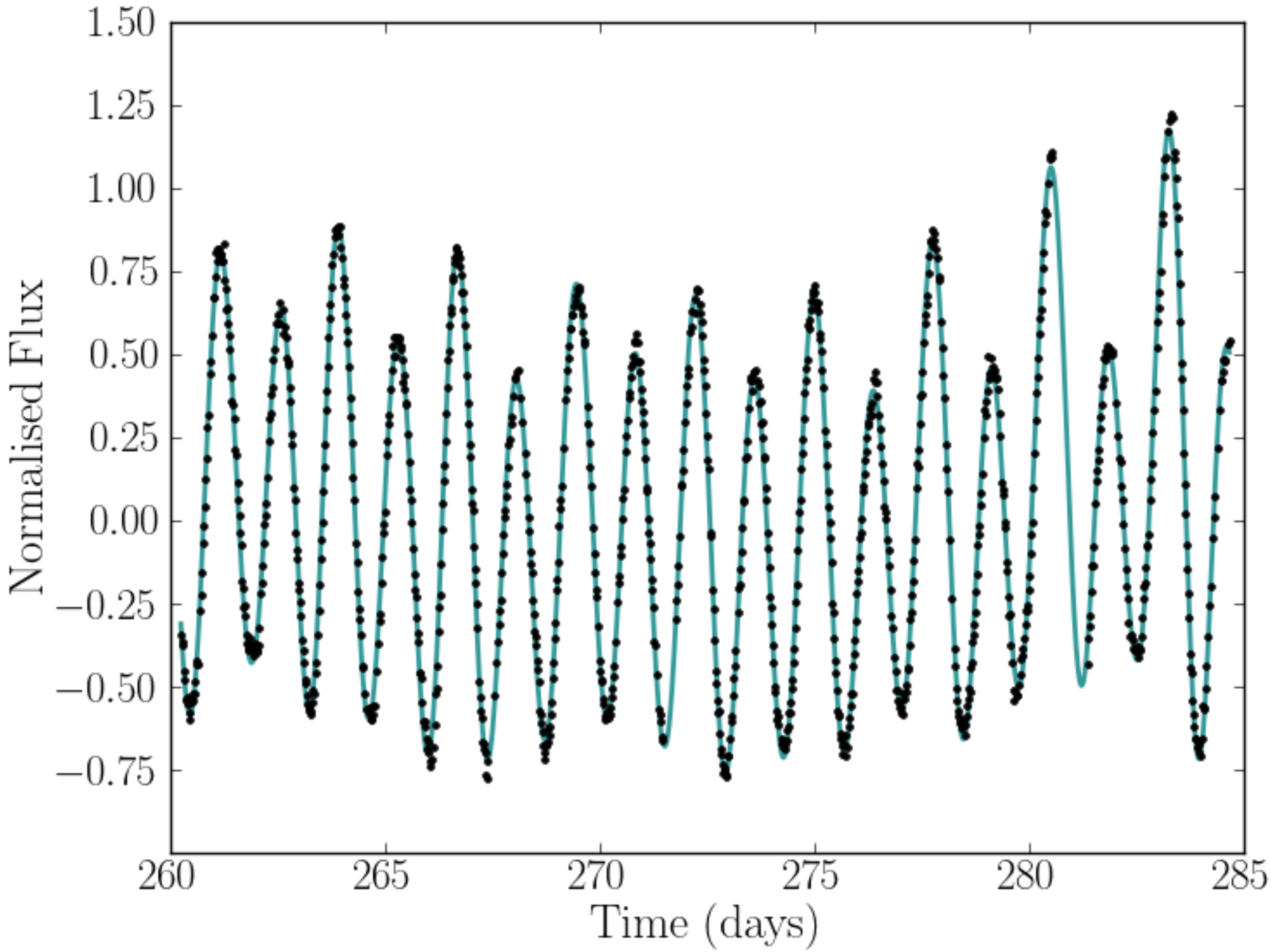
with Bekki Dawson, *et al.*


1.05
1.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
time since transit [days]

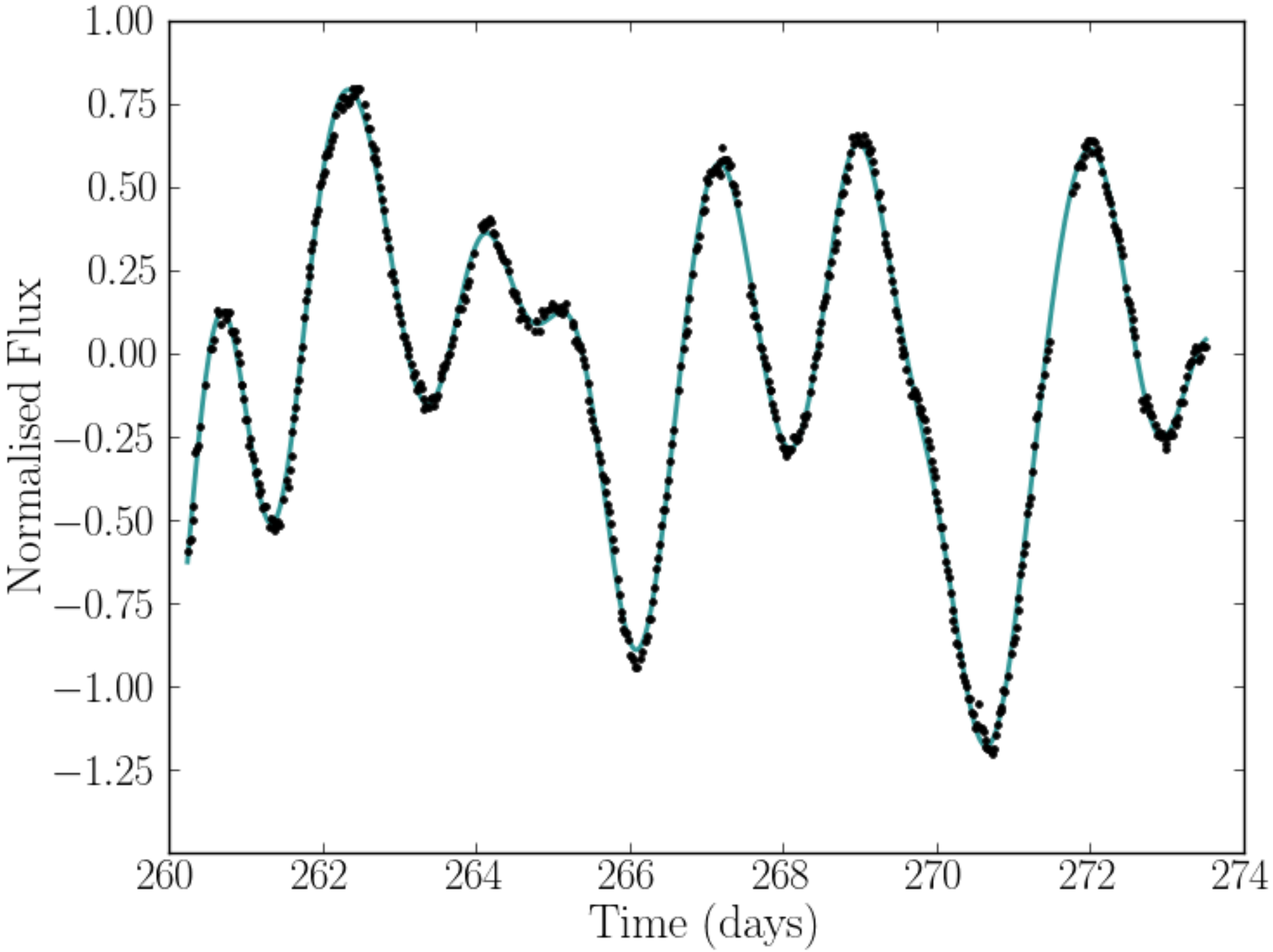

1.15
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
time since transit [days]

Stellar Rotation

with Ruth Angus



figures from Ruth Angus (Oxford)



figures from Ruth Angus (Oxford)

4

Conclusions & Summary.

correlated noise matters.

a Gaussian process provides a drop-in replacement likelihood function

if you can compute it

Resources

gaussianprocess.org/gpml

github.com/dfm/[gp
george

danfm@nyu.edu