data-driven discovery in the astronomical time domain

dan foreman-mackey
cca@flatiron dfm.io github.com/dfm @exoplaneteer

noise models, algorithm development, and open source scientific software

dan foreman-mackey
cca@flatiron dfm.io github.com/dfm @exoplaneteer

my goals for today

Discuss the role of **open source software** in astrophysics motivated by the study of **time domain astronomical surveys**.

Demonstrate the impact of **interdisciplinary collaboration**, **good documentation**, and **open development** on this research.

time domain astronomy

Measure **[something]** as a function of time.

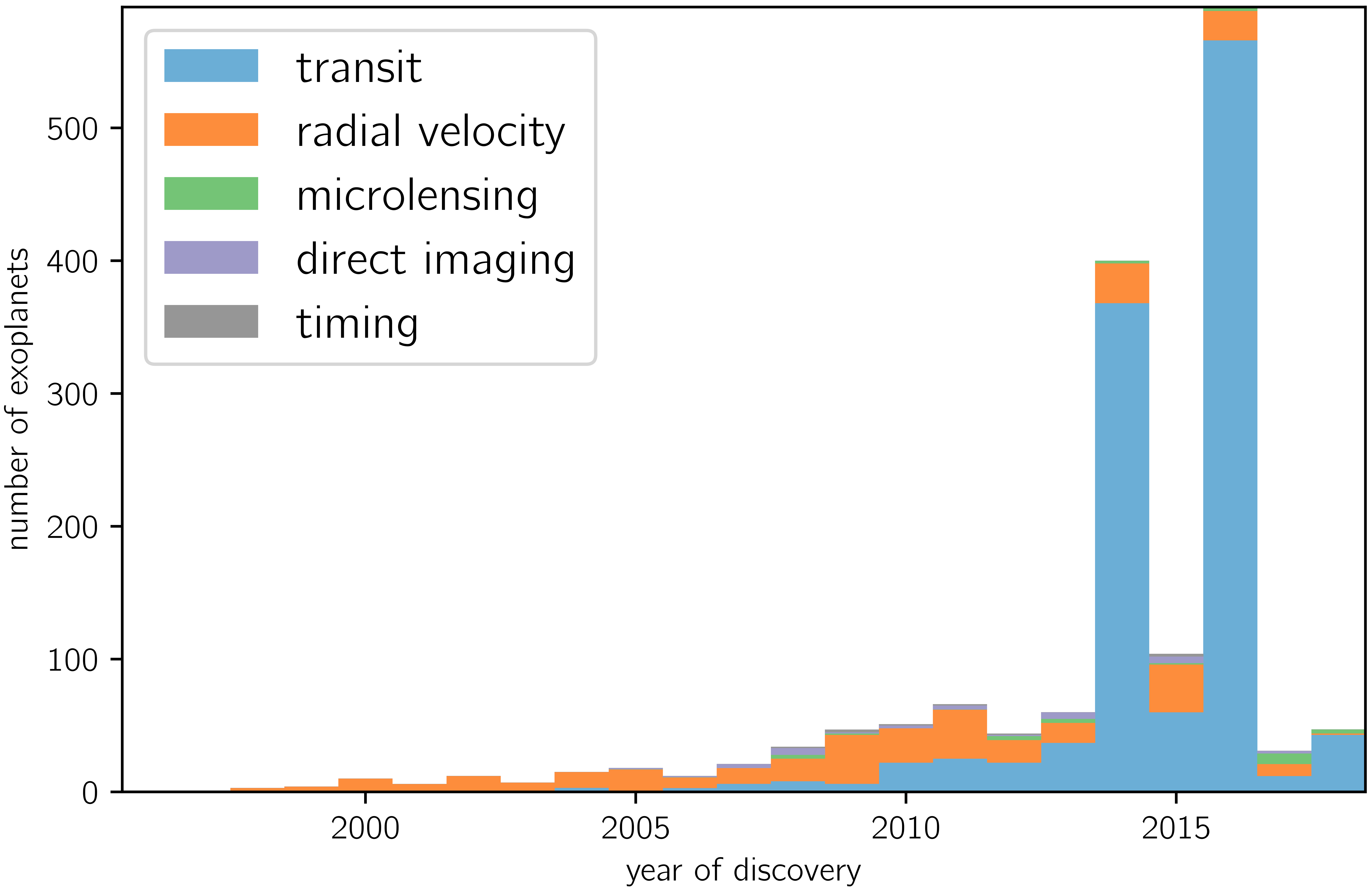
time domain astronomy

Measure **[something]** as a function of time.

where **[something]** = position
velocity
brightness
color
...

exoplanets

One of the recent time domain success stories.

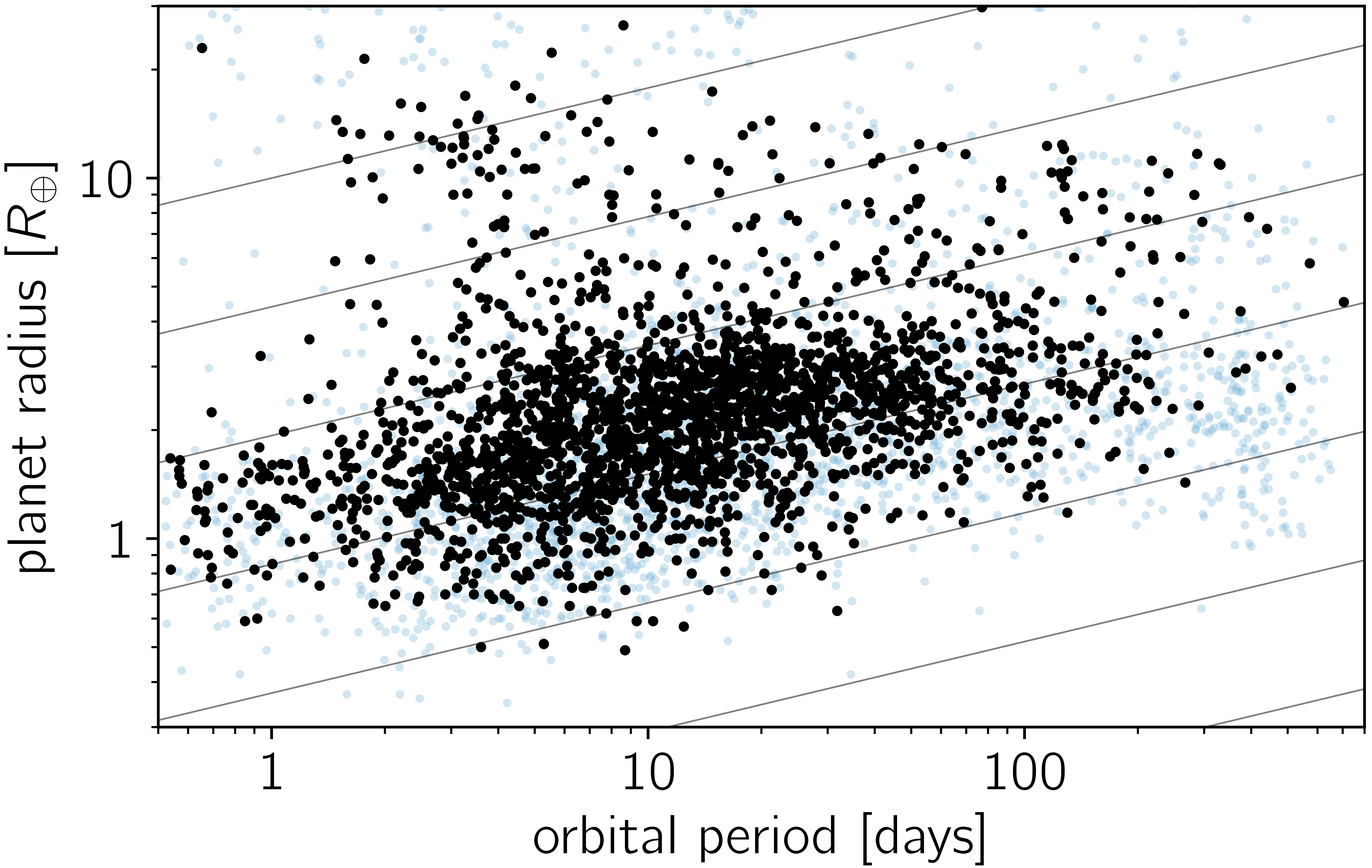


source: The Open Exoplanet Catalogue

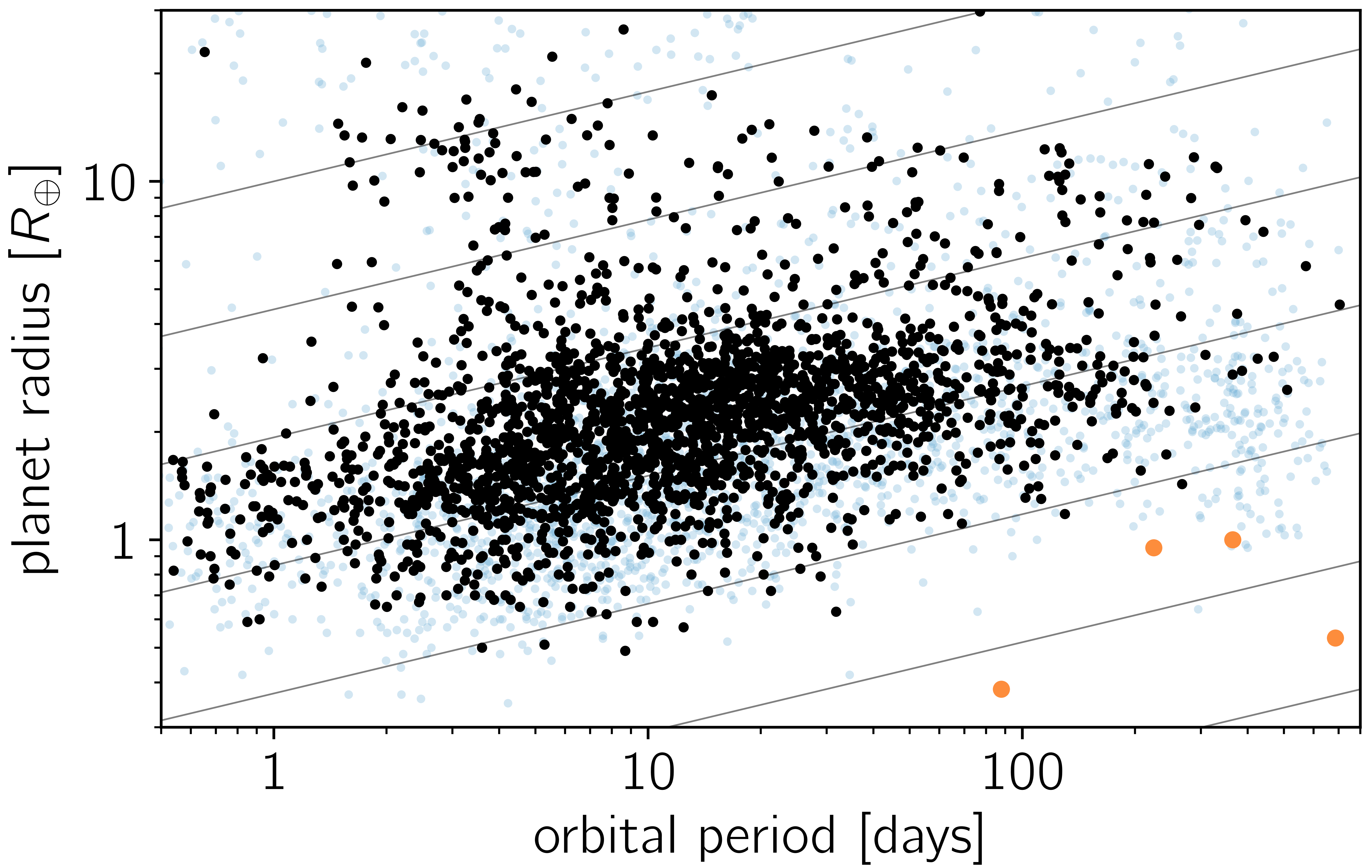
so what?

kepler

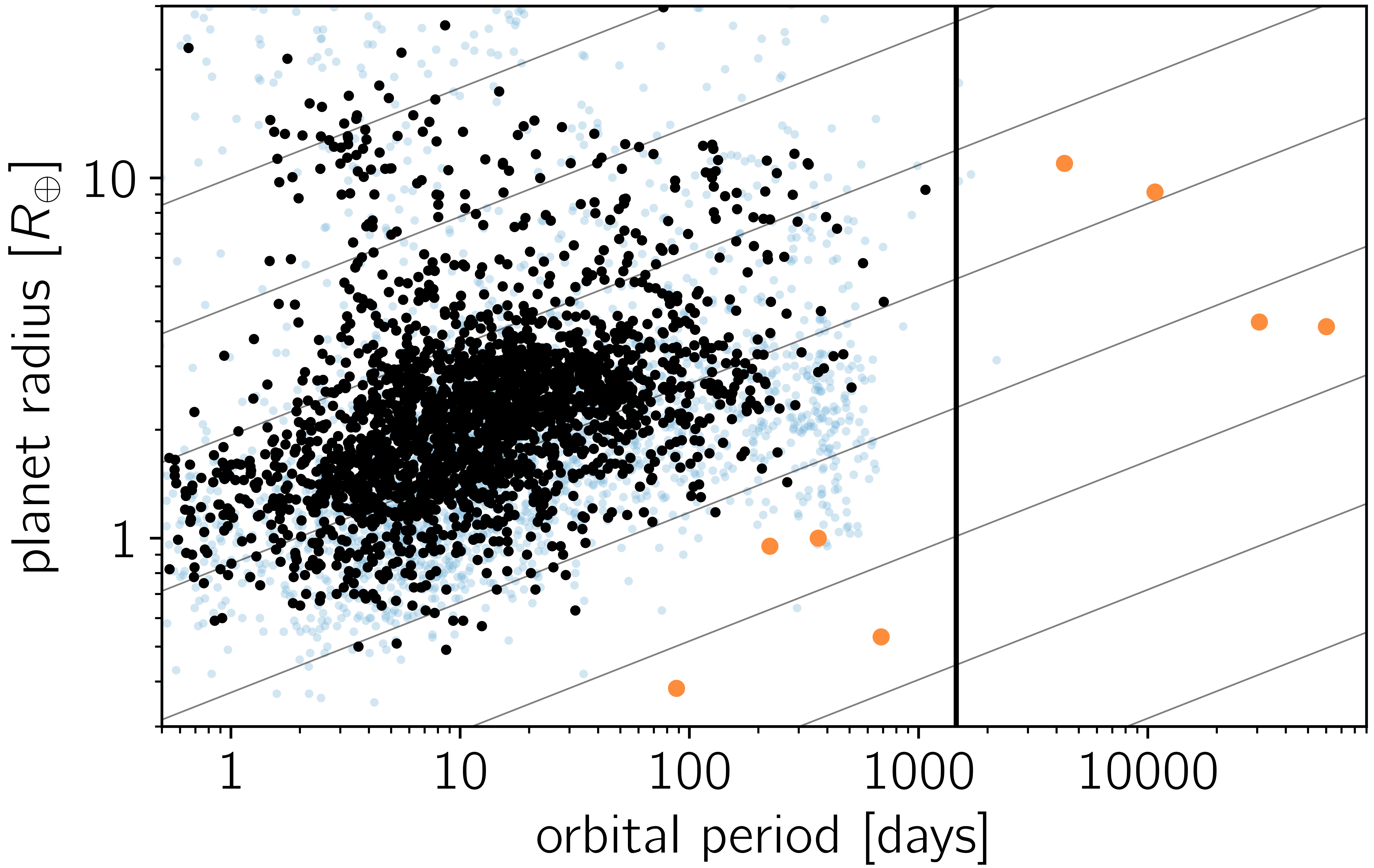
was designed as a **statistical** mission.



data: NASA Exoplanet Archive



data: NASA Exoplanet Archive



data: NASA Exoplanet Archive

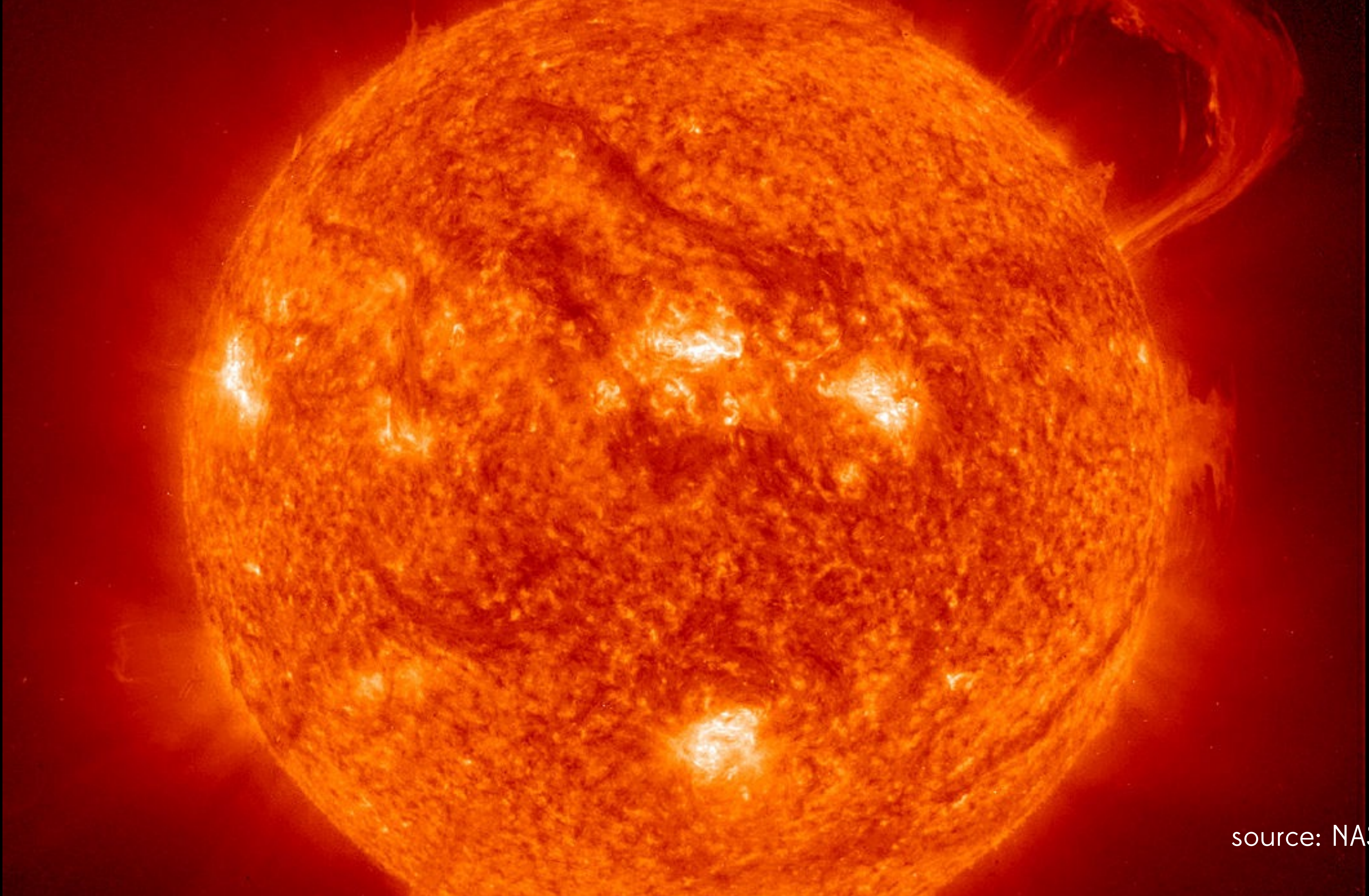
state-of-the-art discovery
requires
pushing the technical boundaries

when you push the boundaries

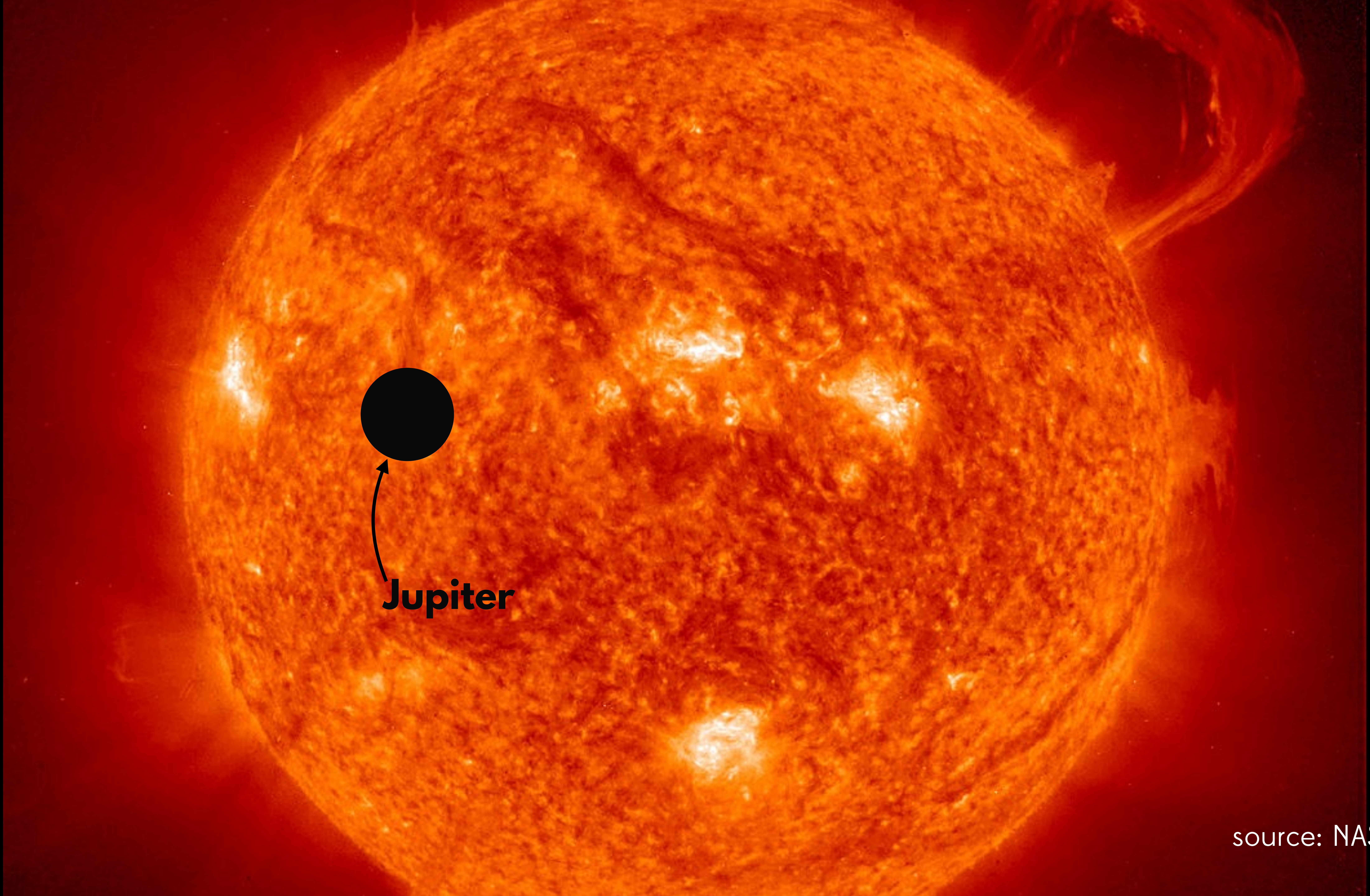
you won't have

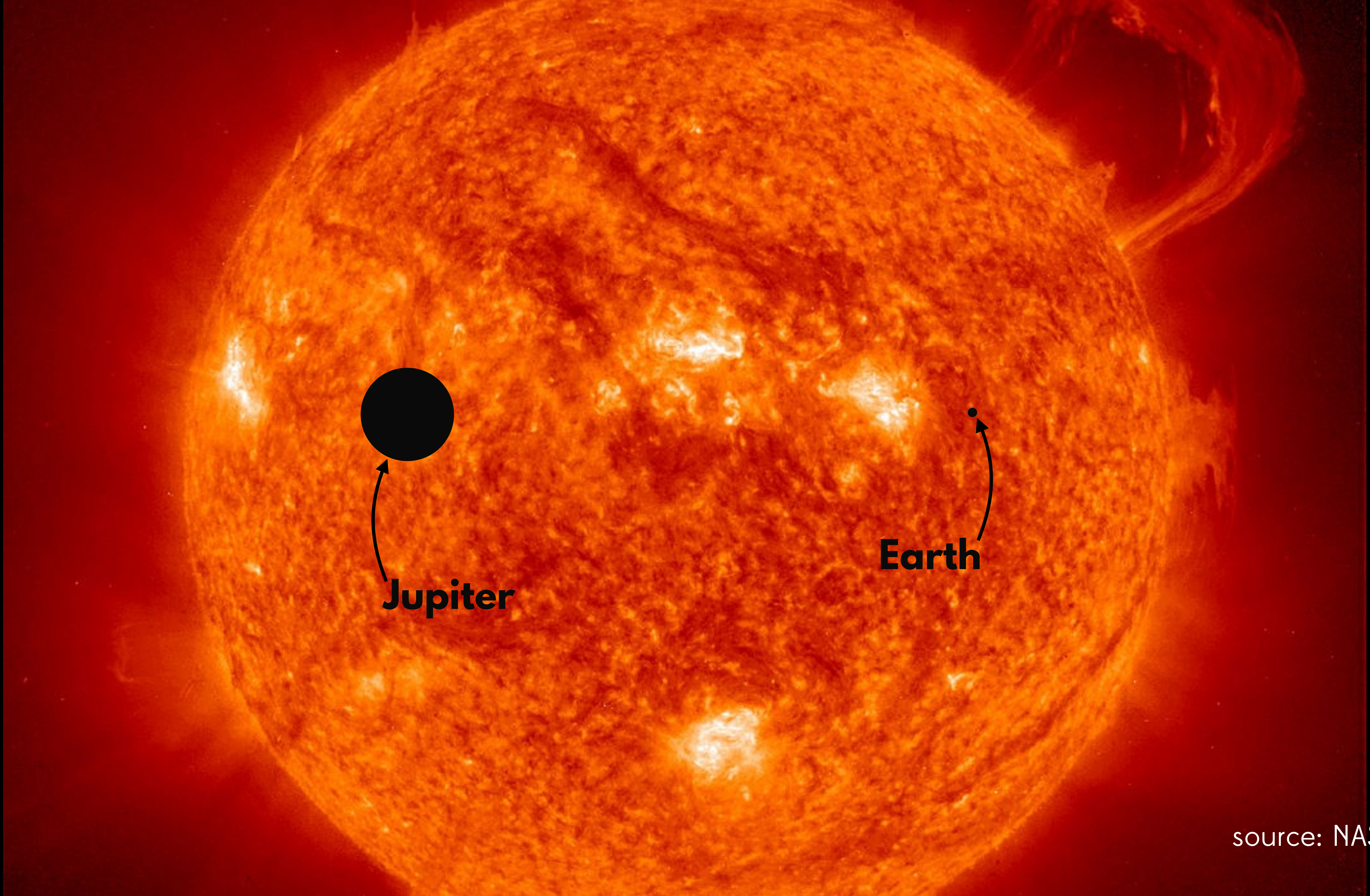
a good training set

the transit method

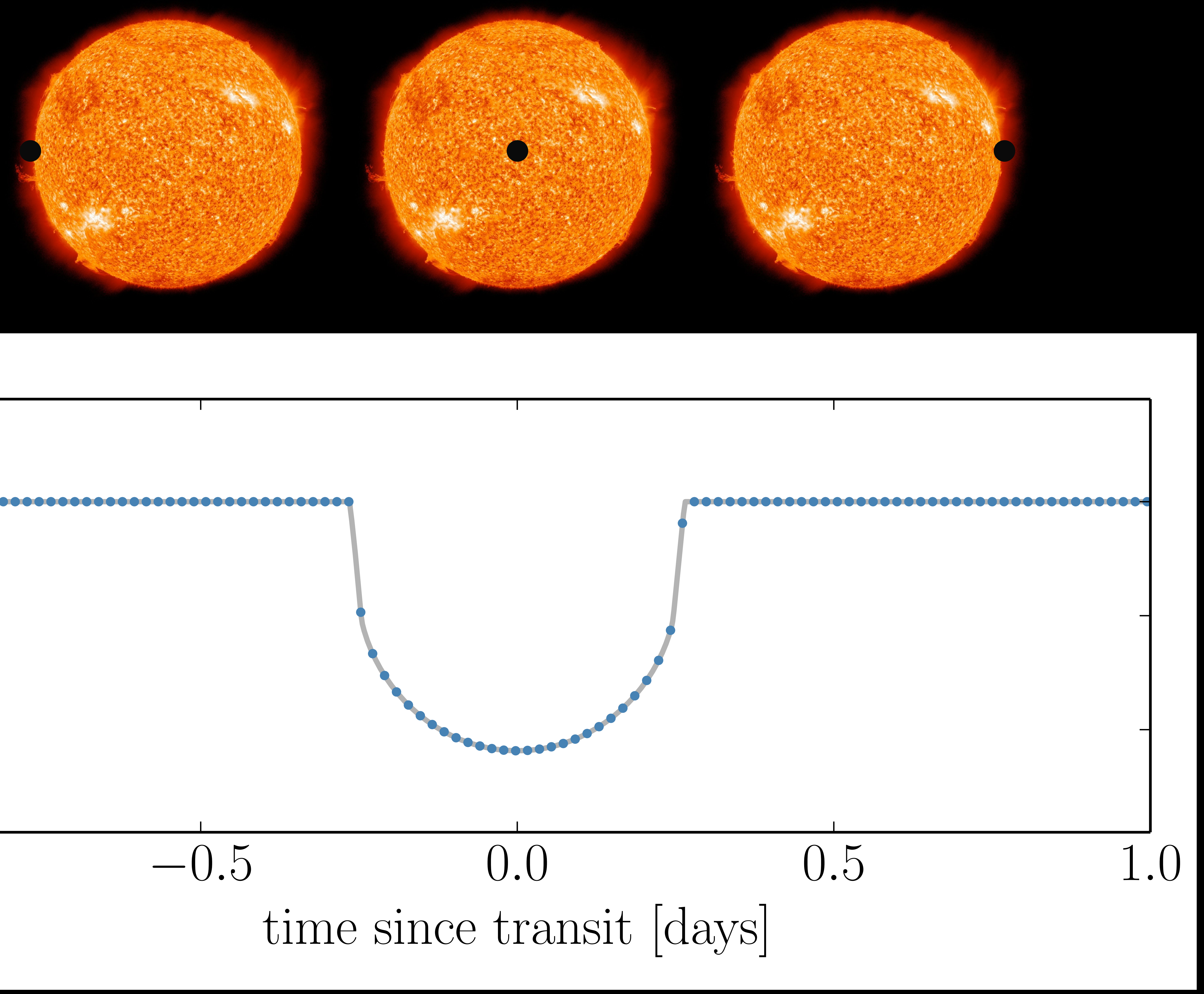


source: NASA/ESA


Jupiter
source: NASA/ESA

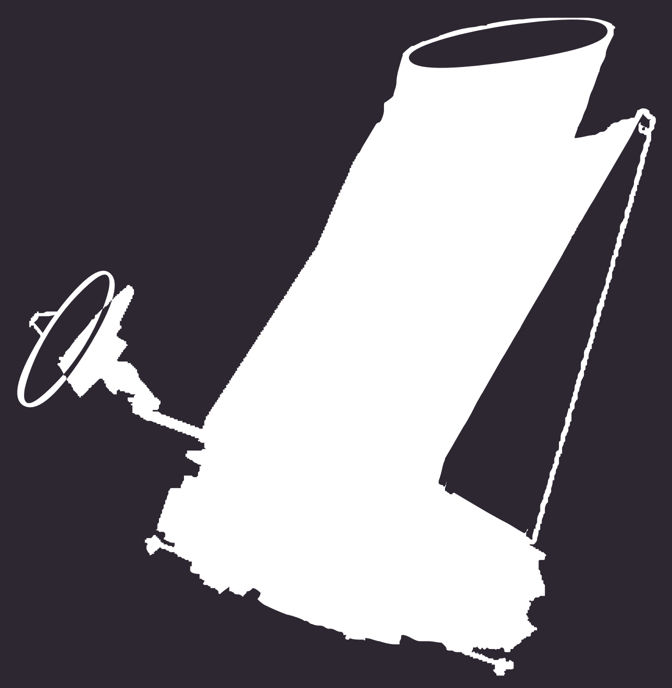


source: NASA/ESA



relative brightness [ppm]
0
−50
−100
−1.0
−0.5
0.0
0.5
1.0
time since transit [days]

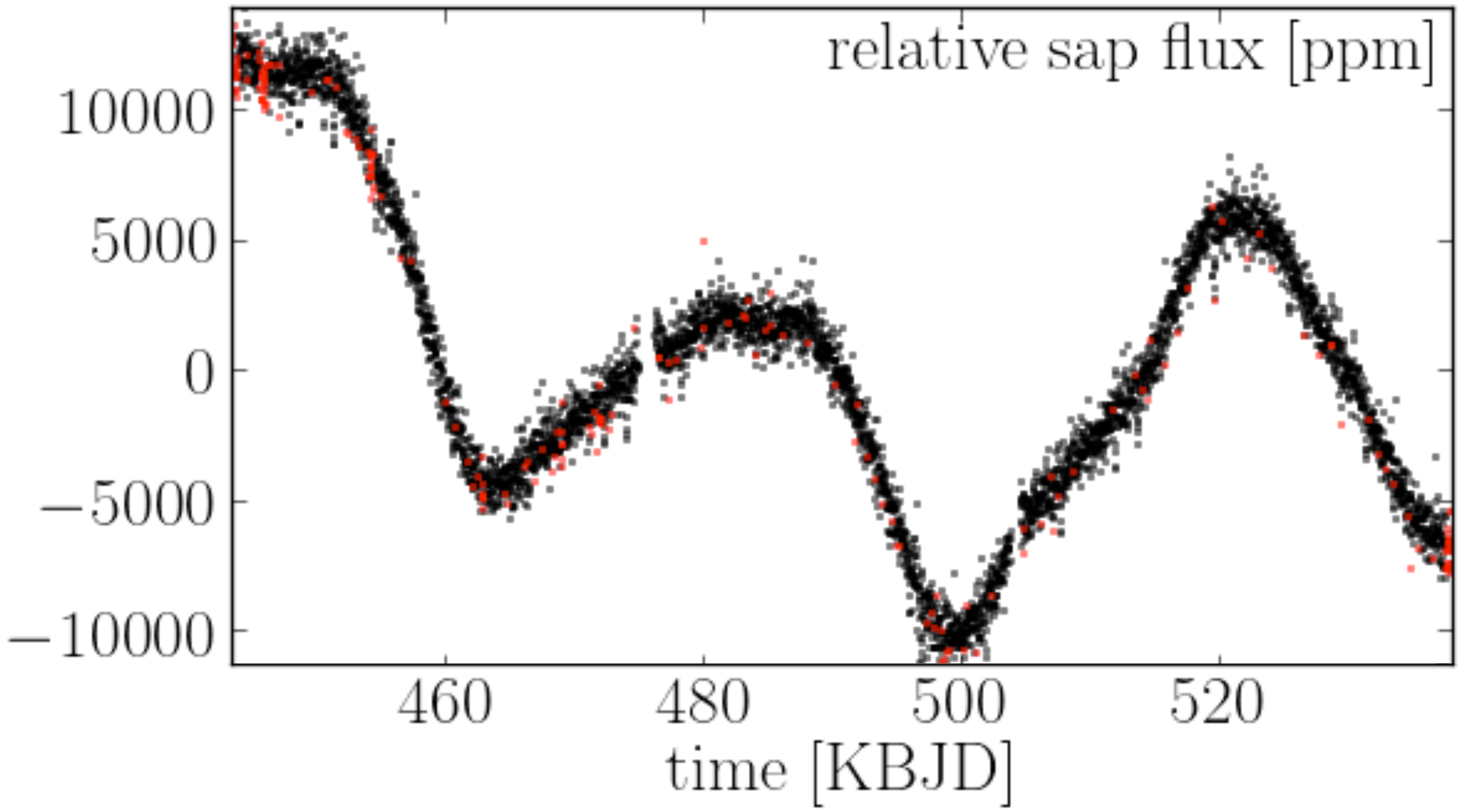
we need to **precisely** measure the brightness
of **many** stars
at **high cadence**
for a **long time**



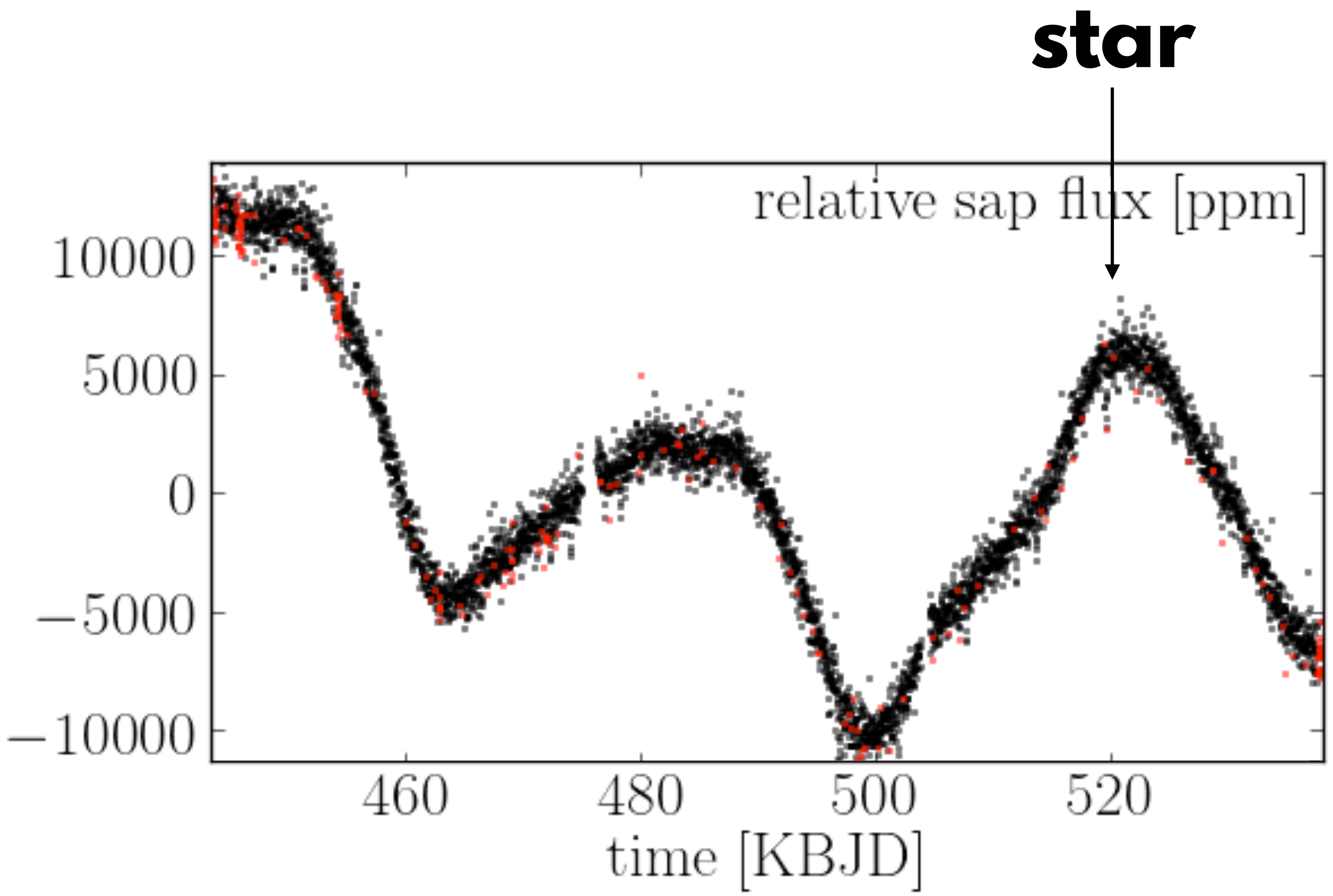
kepler

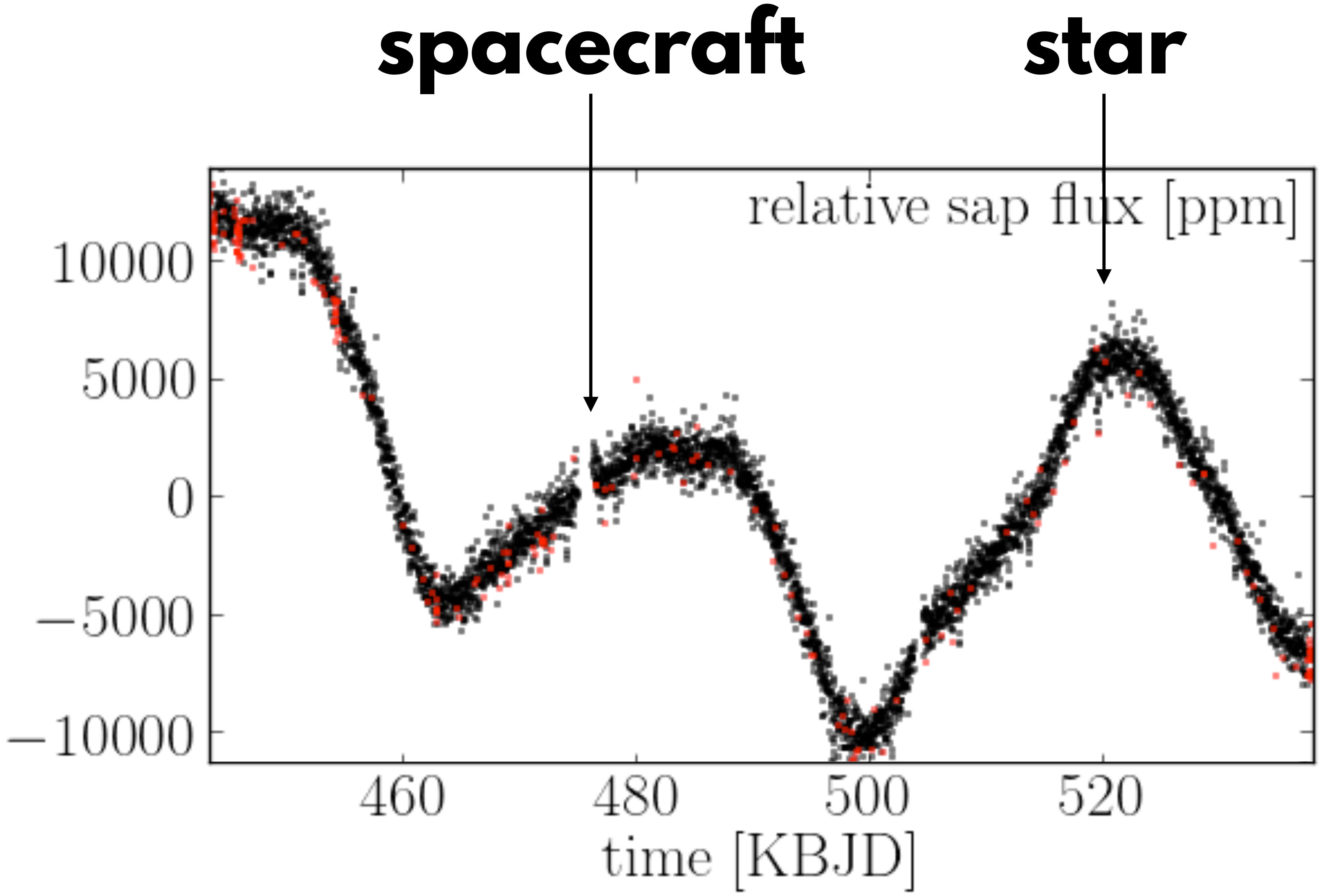
190k targets
30 min cadence
4 year baseline
5,000 planet candidates

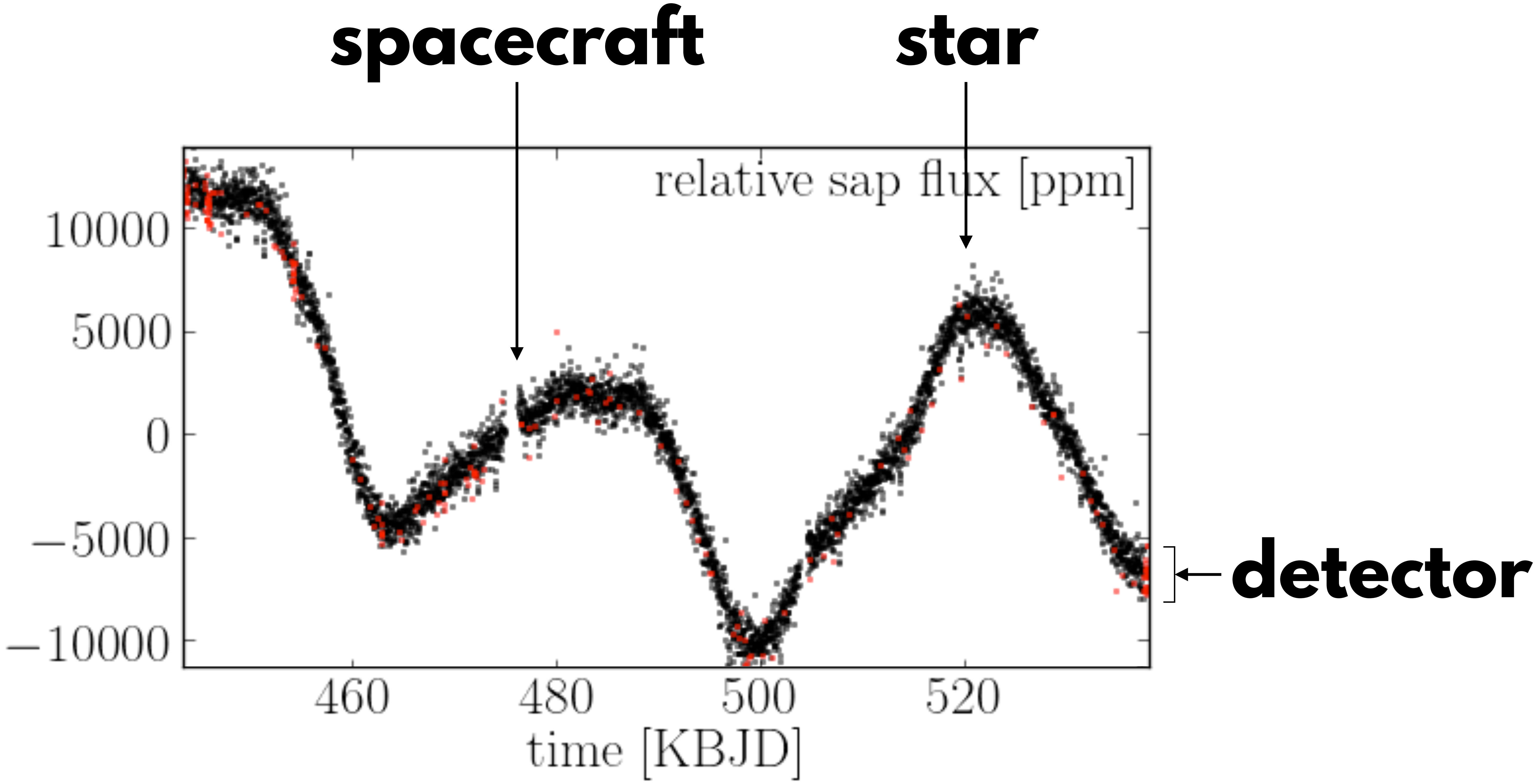
*note: all numbers are approximate



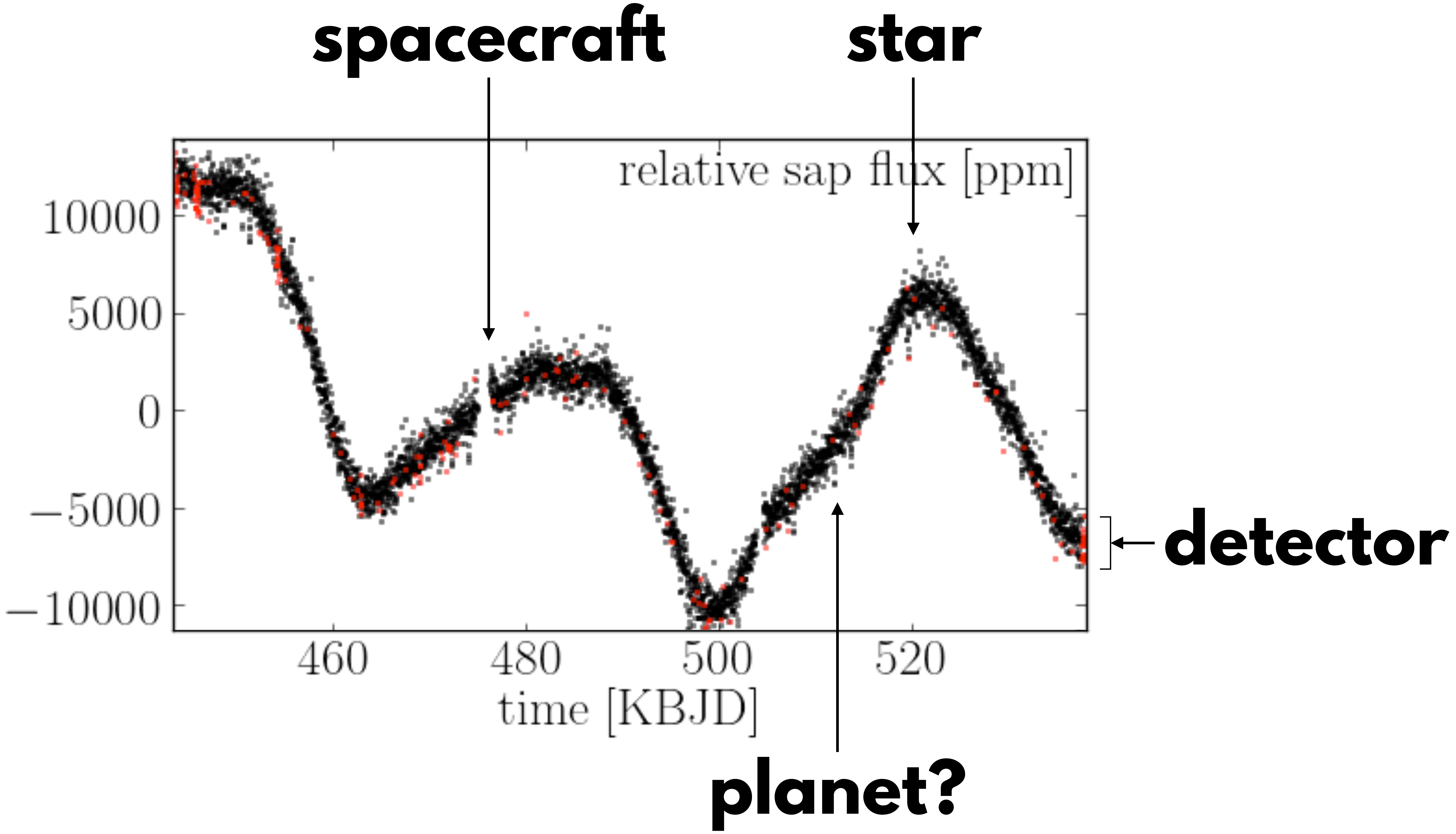
relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]


star
relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]

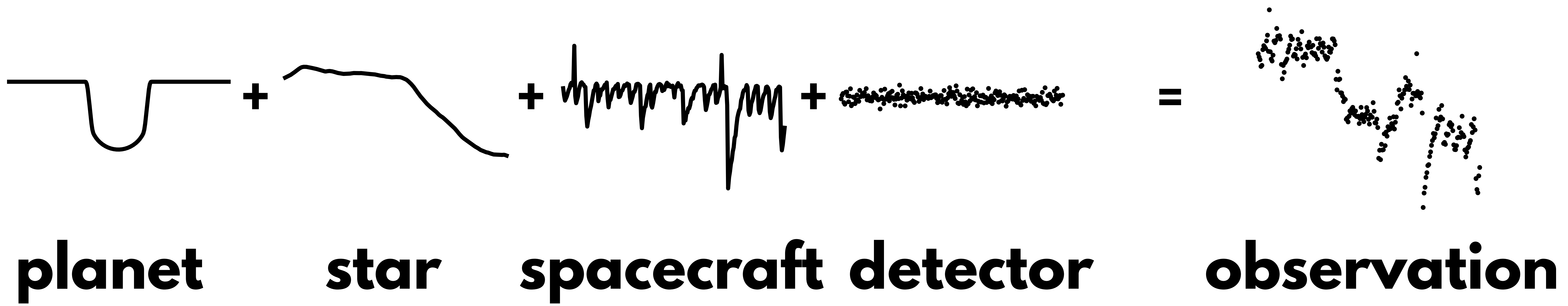

spacecraft
star
relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]



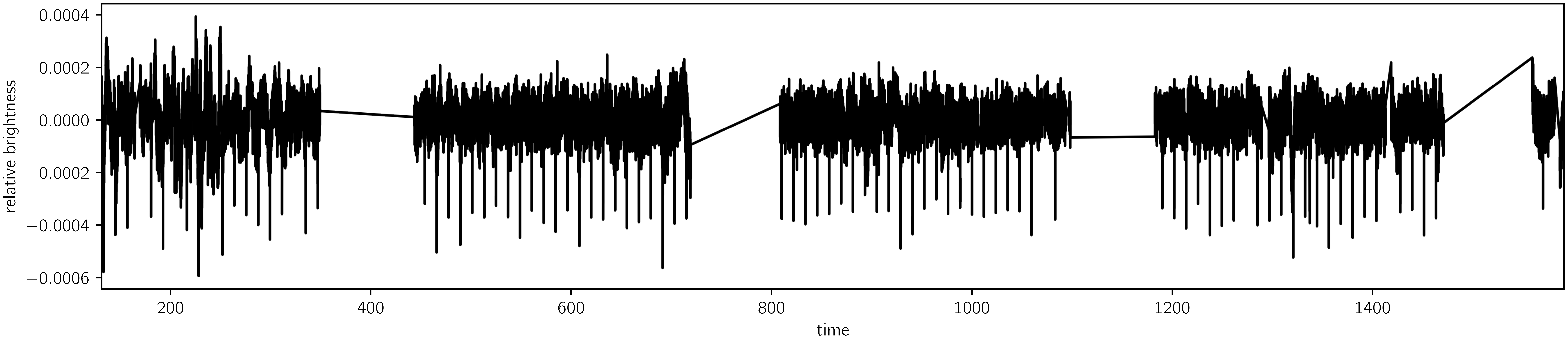
spacecraft
star
detector
relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]



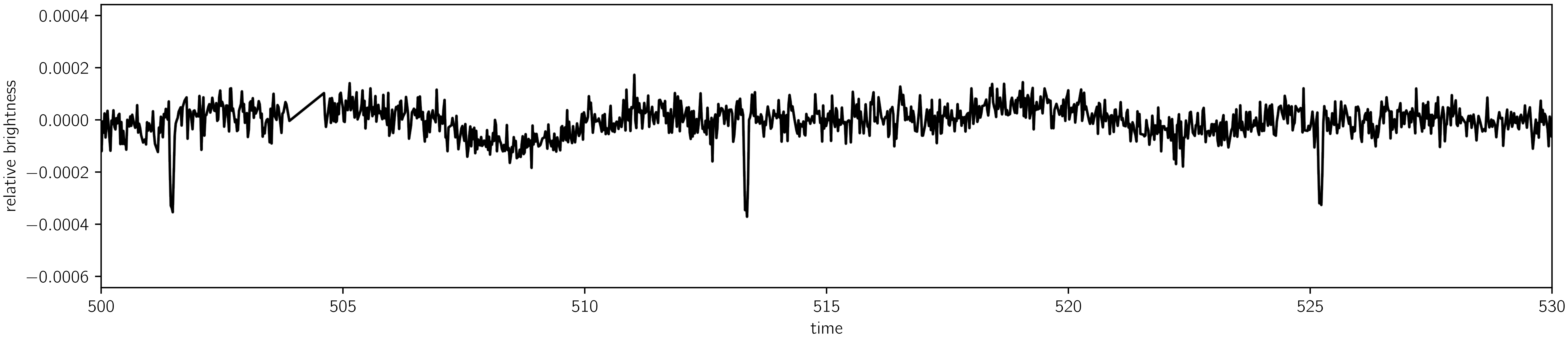
spacecraft
star
relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]
detector
planet?

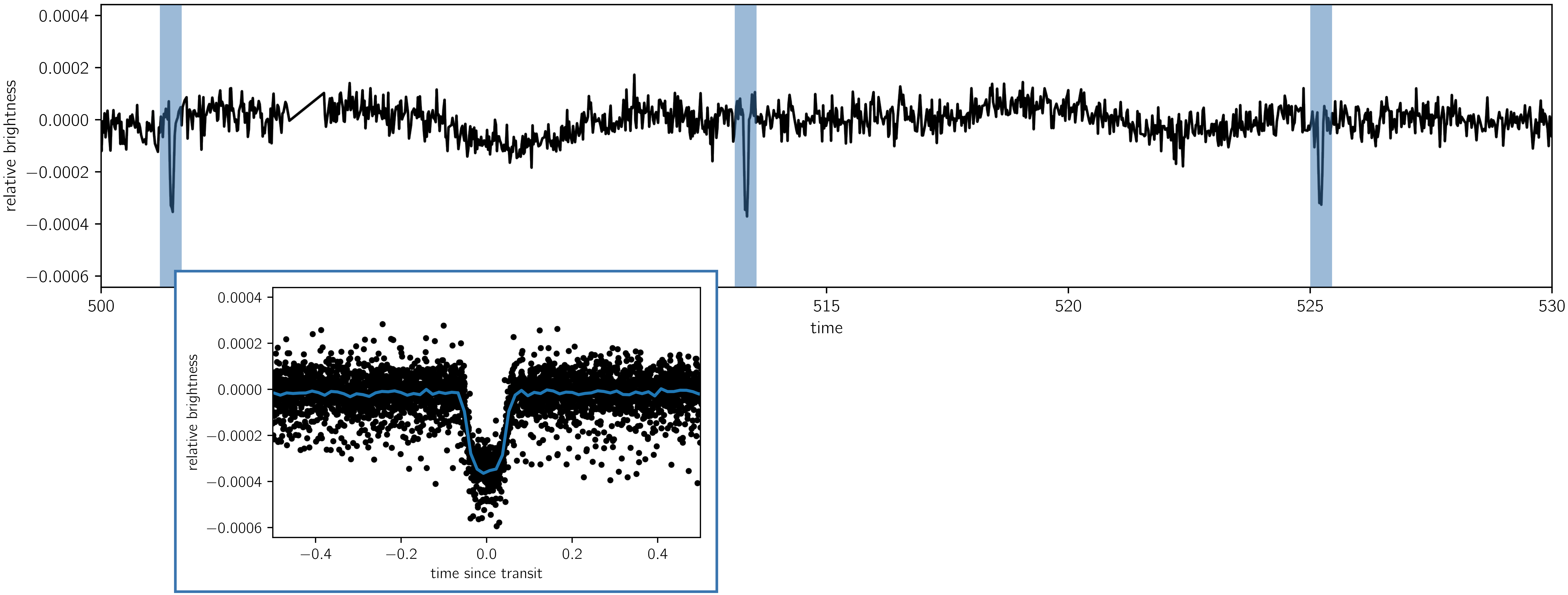

+
+
+
=
planet
star
spacecraft
detector
observation

auxiliary science

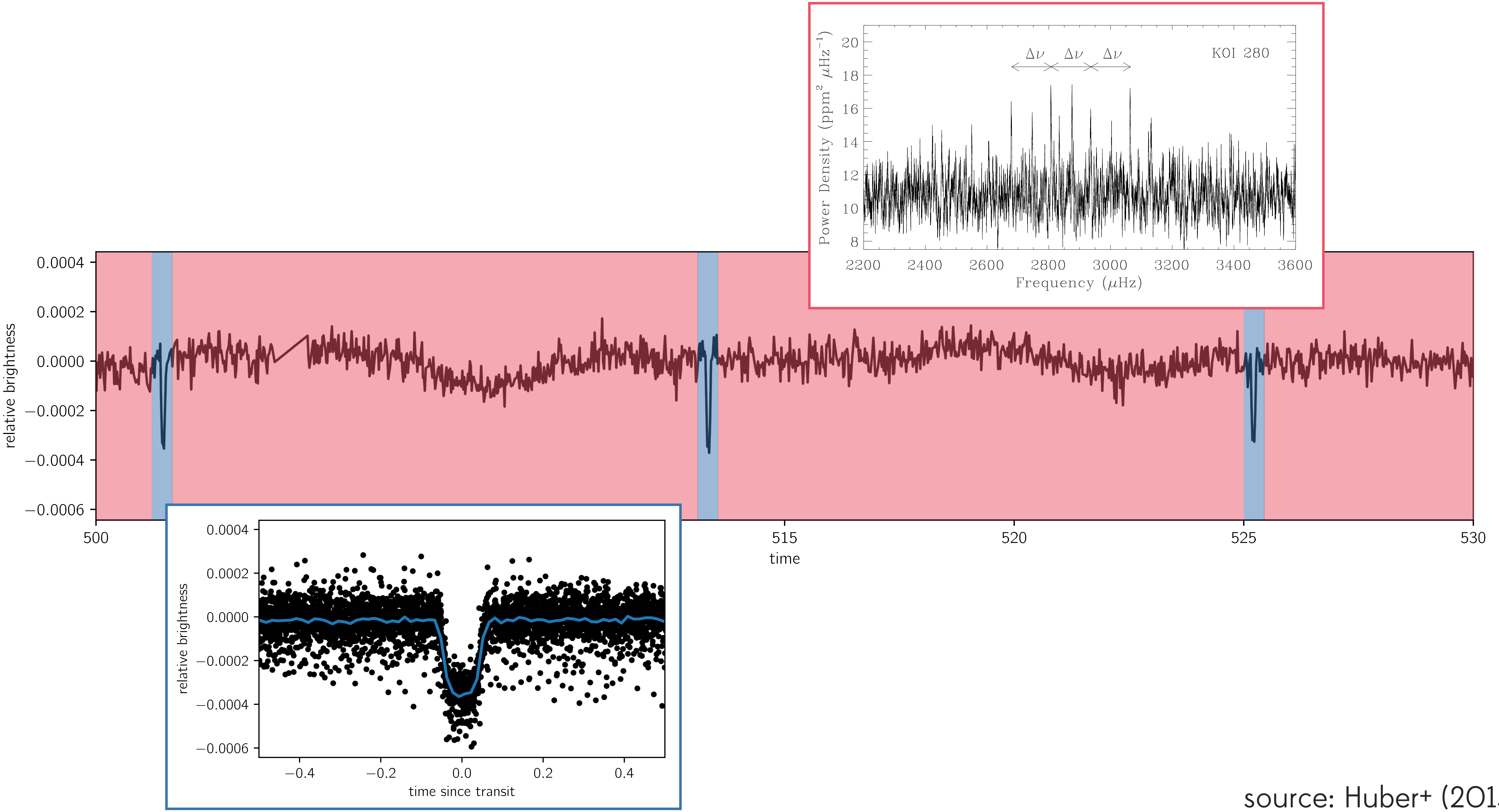


0.0004
0.0002
0.0000
−0.0002
−0.0004
−0.0006
relative brightness
200
400
600
800
1000
1200
1400
time


0.0004
0.0002
0.0000
−0.0002
−0.0004
−0.0006
relative brightness
500
505
510
515
520
525
530
time

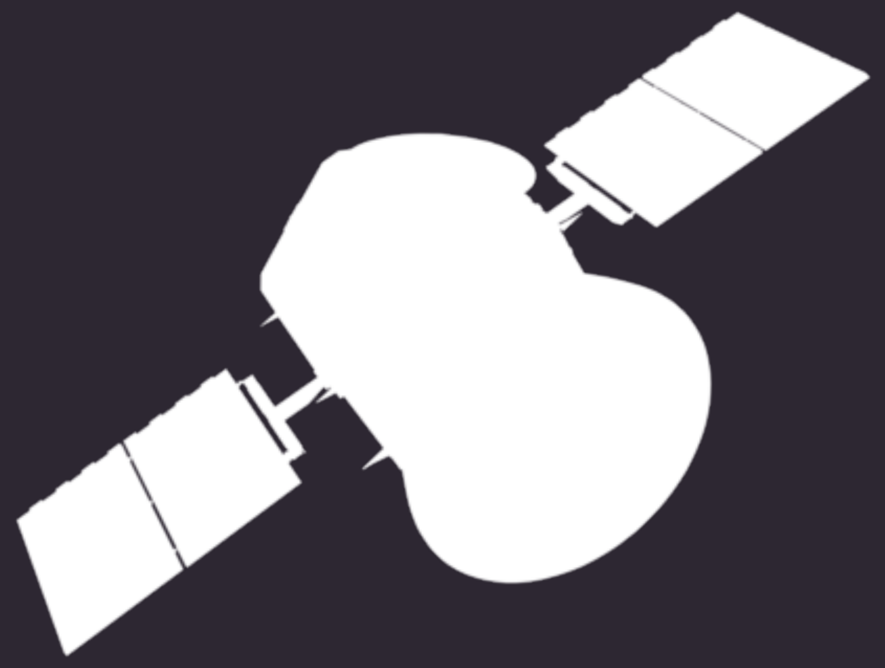


0.0004
0.0002
0.0000
−0.0002
−0.0004
−0.0006
relative brightness
500
515
520
525
530
time
0.0004
0.0002
0.0000
−0.0002
−0.0004
−0.0006
relative brightness
−0.4
−0.2
0.0
0.2
0.4
time since transit



source: Huber+ (2013)

it's not stopping anytime soon



tess

3.2M / 500k targets
30 min / 2 min cadence
30 day baseline
15,000 planet candidates

*note: all numbers are **predictions**

what do we do with all these data?

$$p(\text{physics} \mid \text{data})$$

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

1. A rigorous inference method
2. Efficient computation of the likelihood

1. Markov chain Monte Carlo (**MCMC**)
2. Gaussian processes (**GPs**)

Markov chain Monte Carlo

$$\int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \boldsymbol{D})\, \mathrm{d}\boldsymbol{\theta}$$

$$\int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \boldsymbol{D})\, \mathrm{d}\boldsymbol{\theta}$$

physics data

my advisor's advisor's fortran code

$$\int f(\boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid \boldsymbol{D})\, \mathrm{d}\boldsymbol{\theta}$$

physics

data

MCMC

$$\theta_n \sim p(\theta \mid D)$$

physics — data

MCMC

samples from

$$\theta_n \sim p(\theta \mid D)$$

physics

data

samples from

WOAH!

$$\theta_n \sim p(\theta \mid D)$$

physics

data

in astronomy, we use **MCMC**

MCMC is brittle

Either: **algorithm requires tuning** or **model must be simplified**.

Publications of the Astronomical Society of the Pacific, **125**:306–312, 2013 March

emcee: The MCMC Hammer

Daniel Foreman-Mackey,[1] David W. Hogg,[1,2] Dustin Lang,[3,4] and Jonathan Goodman[5]

Received 2013 January 09; accepted 2013 January 30; published 2013 February 25

MCMC

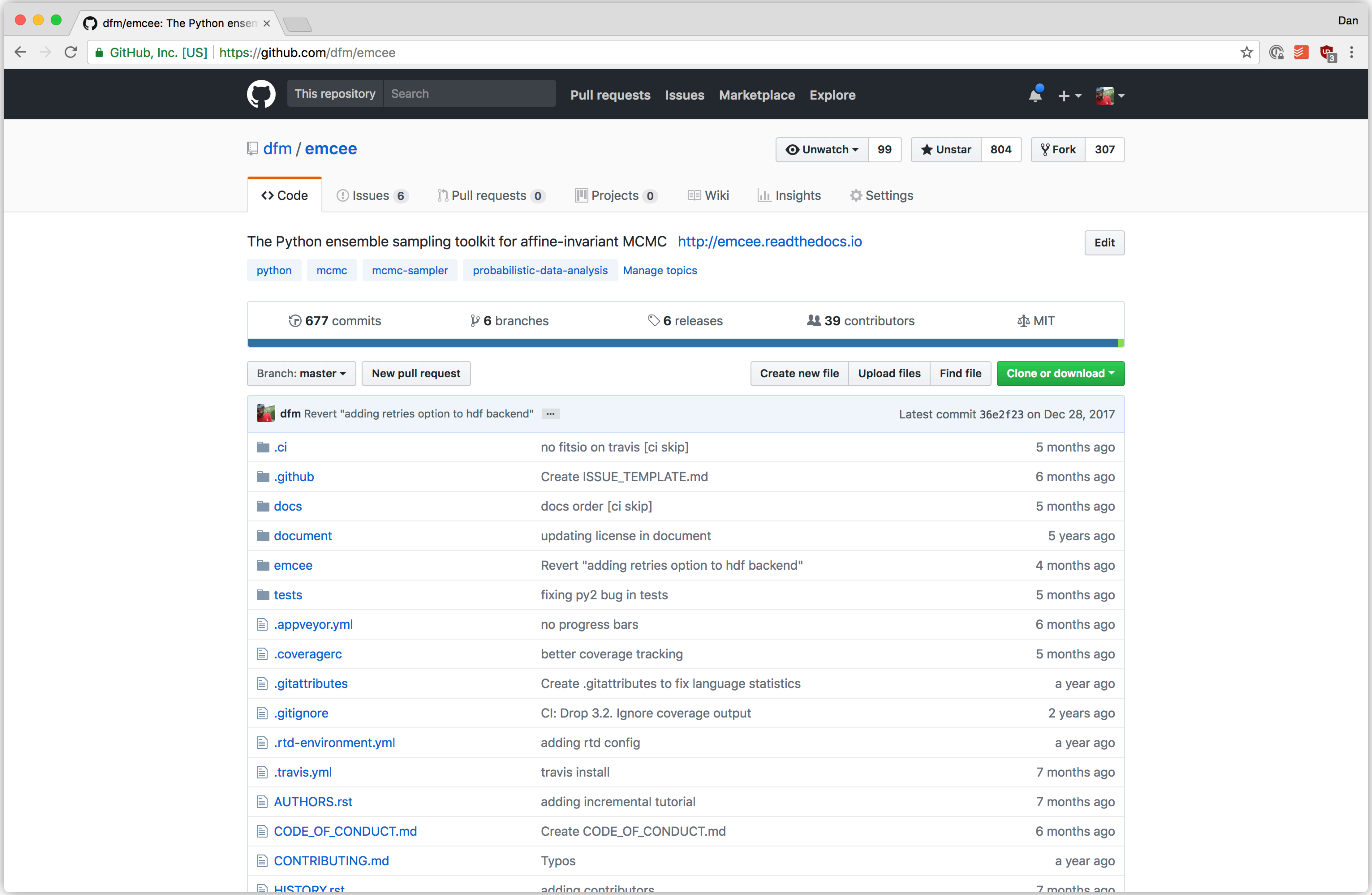
Algorithm 3 The parallel stretch move update step

for $i \in \{0, 1\}$ **do**
  for $k = 1, \ldots, K/2$ **do**
    // This loop can now be done in parallel for all k
    Draw a walker X_j at random from the complementary ensemble $S^{(\sim i)}(t)$
    $X_k \leftarrow S_k^{(i)}$
    $z \leftarrow Z \sim g(z)$, Equation (10)
    $Y \leftarrow X_j + z\,[X_k(t) - X_j]$
    $q \leftarrow z^{n-1}\, p(Y)/p(X_k(t))$
    $r \leftarrow R \sim [0, 1]$
    if $r \leq q$, Equation (9) **then**
      $X_k(t + \frac{1}{2}) \leftarrow Y$
    else
      $X_k(t + \frac{1}{2}) \leftarrow X_k(t)$
    end if
  end for
  $t \leftarrow t + \frac{1}{2}$
end for

from: DFM, Hogg, Lang, Goodman (2013)

this project has been quite popular.

why?



dfm/emcee: The Python ensen
Dan
GitHub, Inc. [US] https://github.com/dfm/emcee
This repository Search
Pull requests Issues Marketplace Explore
dfm / emcee
Unwatch 99
Unstar 804
Fork 307
Code
Issues 6
Pull requests 0
Projects 0
Wiki
Insights
Settings
The Python ensemble sampling toolkit for affine-invariant MCMC http://emcee.readthedocs.io
Edit
python
mcmc
mcmc-sampler
probabilistic-data-analysis
Manage topics
677 commits
6 branches
6 releases
39 contributors
MIT
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
dfm Revert "adding retries option to hdf backend"
Latest commit 36e2f23 on Dec 28, 2017
.ci no fitsio on travis [ci skip] 5 months ago
.github Create ISSUE_TEMPLATE.md 6 months ago
docs docs order [ci skip] 5 months ago
document updating license in document 5 years ago
emcee Revert "adding retries option to hdf backend" 4 months ago
tests fixing py2 bug in tests 5 months ago
.appveyor.yml no progress bars 6 months ago
.coveragerc better coverage tracking 5 months ago
.gitattributes Create .gitattributes to fix language statistics a year ago
.gitignore CI: Drop 3.2. Ignore coverage output 2 years ago
.rtd-environment.yml adding rtd config a year ago
.travis.yml travis install 7 months ago
AUTHORS.rst adding incremental tutorial 7 months ago
CODE_OF_CONDUCT.md Create CODE_OF_CONDUCT.md 6 months ago
CONTRIBUTING.md Typos a year ago

MCMC

emcee — emcee 2.2.1 docume ×

emcee.readthedocs.io/en/stable/

Fork me on GitHub

emcee
The MCMC Hammer

emcee is an extensible, pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo (MCMC) Ensemble sampler. It's designed for Bayesian parameter estimation and it's really sweet!

Feedback

Feedback is greatly appreciated. If you have any questions, comments, issues or anything else really, shoot me an email.

Quick search

Go

emcee

Seriously Kick-Ass MCMC

emcee is an MIT licensed pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo (MCMC) Ensemble sampler and these pages will show you how to use it.

This documentation won't teach you too much about MCMC but there are a lot of resources available for that (try this one). We also published a paper explaining the emcee algorithm and implementation in detail.

emcee has been used in quite a few projects in the astrophysical literature and it is being actively developed on GitHub.

Basic Usage

If you wanted to draw samples from a 10 dimensional Gaussian, you would do something like:

```
import numpy as np
import emcee

def lnprob(x, ivar):
    return -0.5 * np.sum(ivar * x ** 2)

ndim, nwalkers = 10, 100
ivar = 1. / np.random.rand(ndim)
p0 = [np.random.rand(ndim) for i in range(nwalkers)]

sampler = emcee.EnsembleSampler(nwalkers, ndim, lnprob, args=[ivar])
sampler.run_mcmc(p0, 1000)
```

A more complete example is available in the quickstart documentation.

User Guide

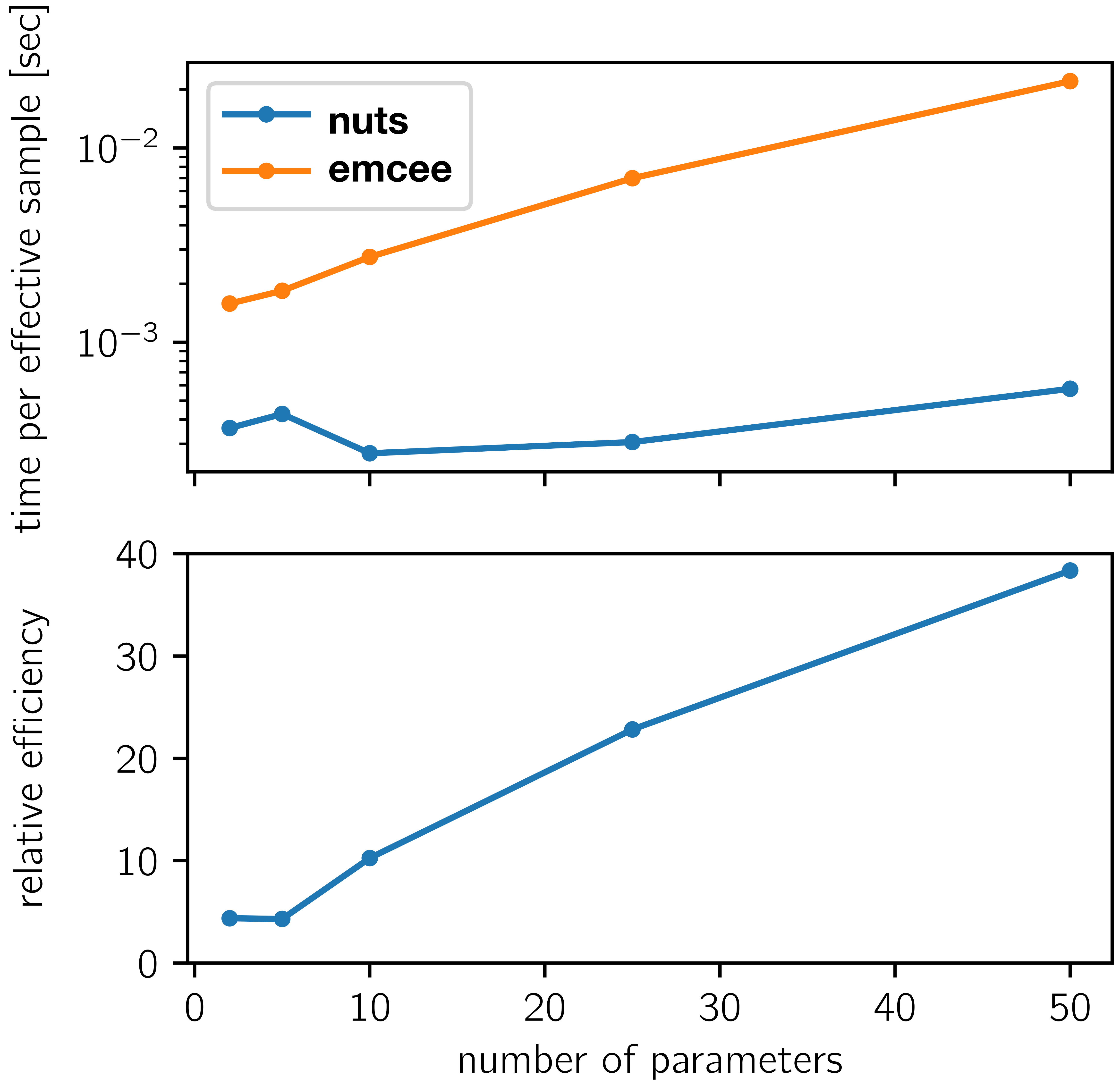
- Installation

v: stable

MCMC

limitations

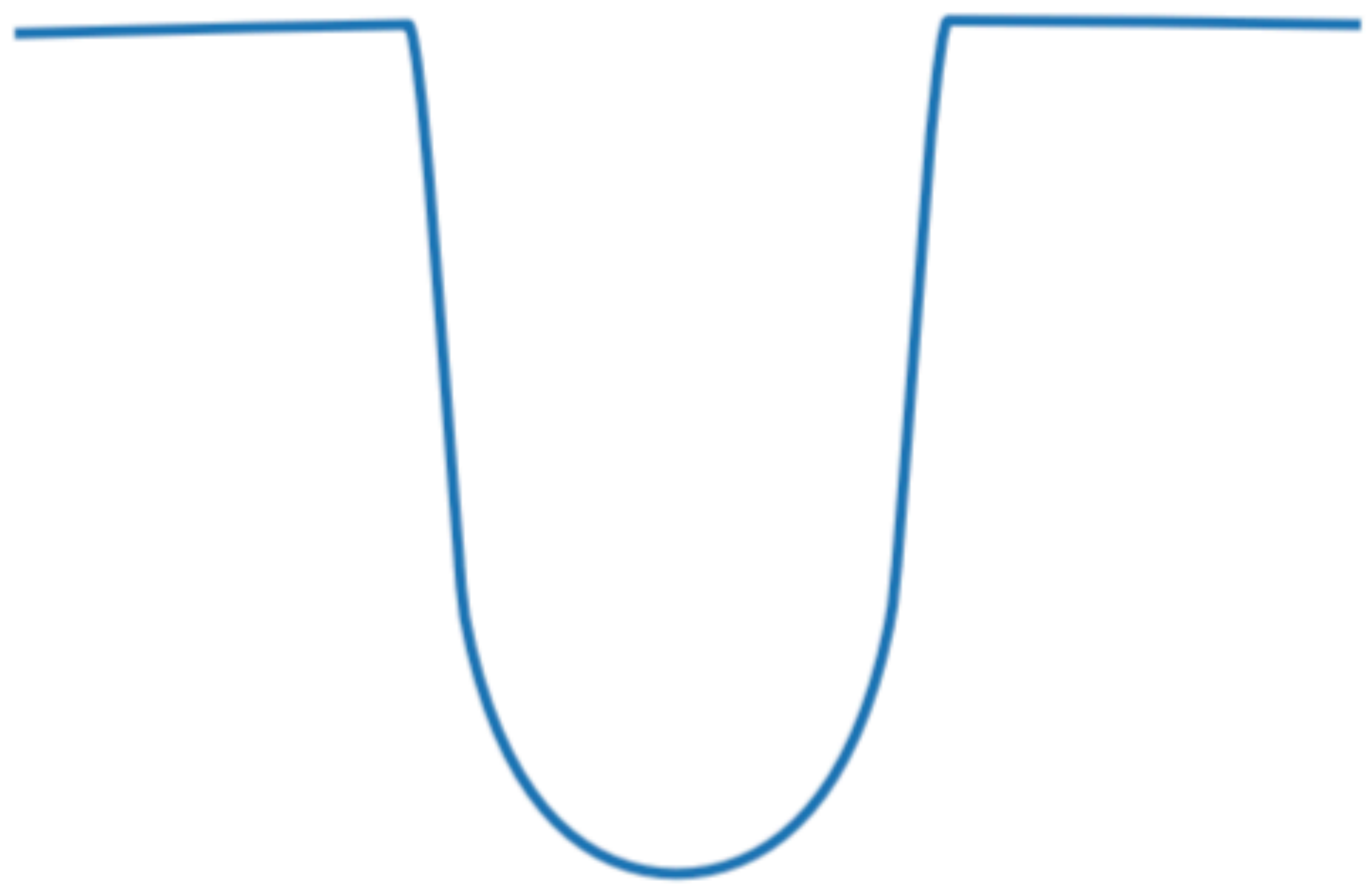
emcee does not scale well with the number of dimensions.


nuts
emcee
time per effective sample [sec]
10^{-2}
10^{-3}
relative efficiency
40
30
20
10
0
0
10
20
30
40
50
number of parameters

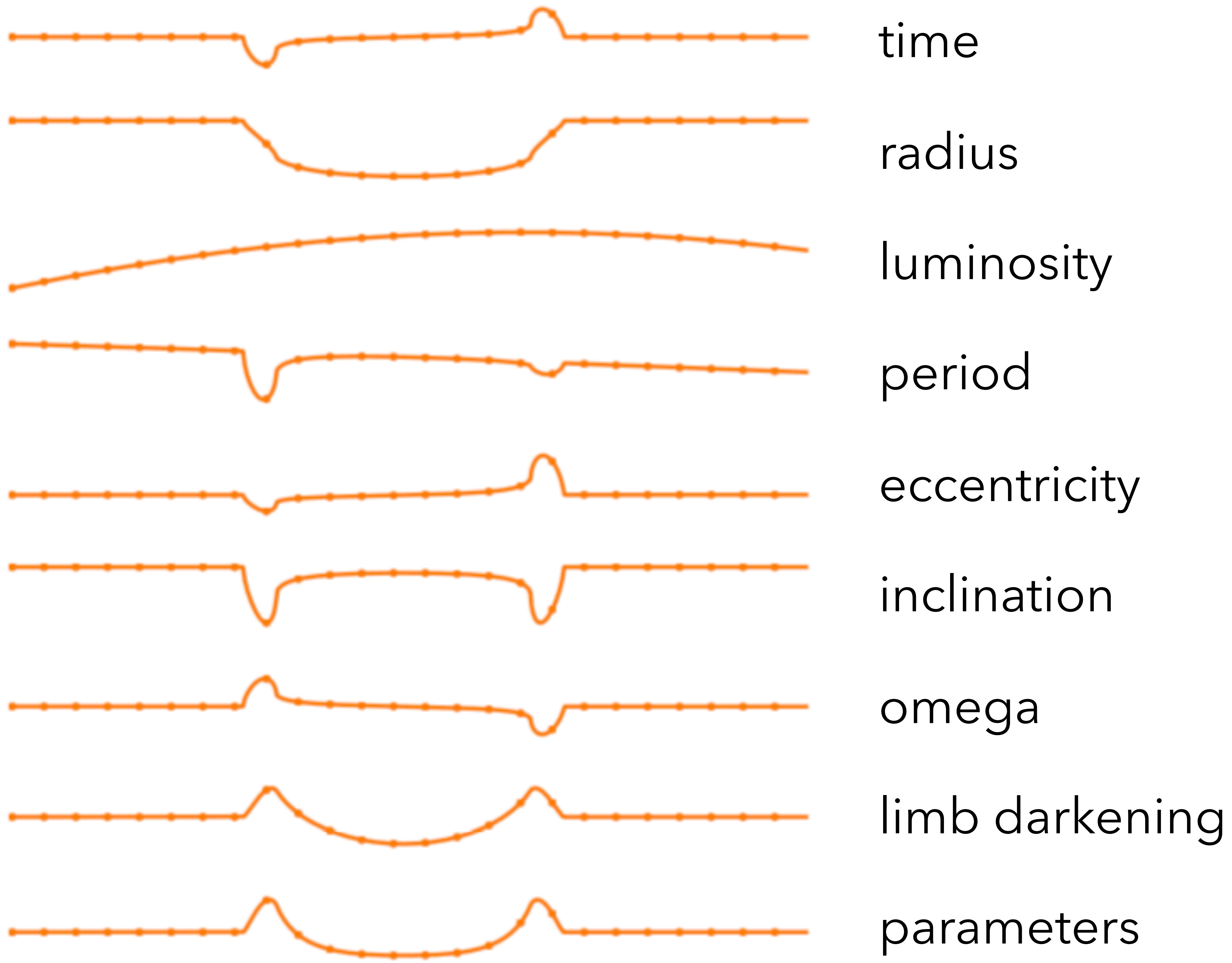
what to do?

1. **Compute derivatives** by hand or using **autodiff**.
2. Use an **emulator** or **surrogate model**.

transit light curve



derivative w.r.t.



credit: Rodrigo Luger

Table Of Contents

Related Topics

Documentation index
- Next: Installation

Quick search

Go

exoplanet

GitHub dfm/exoplanet | license MIT | build passing | docs passing | DOI 10.5281/zenodo.2536576
powered by starry | powered by celerite | powered by PyMC3 | powered by AstroPy

exoplanet is a toolkit for probabilistic modeling of transit and/or radial velocity observations of exoplanets and other astronomical time series using PyMC3. *PyMC3* is a flexible and high-performance model building language and inference engine that scales well to problems with a large number of parameters. *exoplanet* extends *PyMC3*'s language to support many of the custom functions and distributions required when fitting exoplanet datasets. These features include:

- A fast and robust solver for Kepler's equation.
- Scalable Gaussian Processes using celerite.
- Fast and accurate limb darkened light curves using starry.
- Common reparameterizations for limb darkening parameters, and planet radius and impact parameter.
- And many others!

All of these functions and distributions include methods for efficiently calculating their *gradients* so that they can be used with gradient-based inference methods like Hamiltonian Monte Carlo, No U-Turns Sampling, and variational inference. These methods tend to be more robust than the methods more commonly used in astronomy (like ensemble samplers and nested sampling) especially when the model has more than a few parameters. For many exoplanet applications, *exoplanet* (the code) can improve the typical performance by orders of magnitude.

exoplanet is being actively developed in a public repository on GitHub so if you have any trouble, open an issue there.

User guide

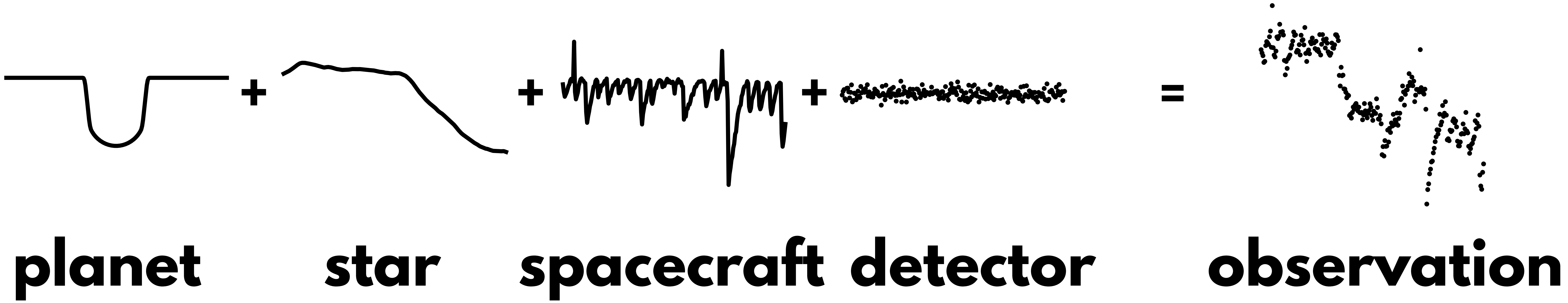
- Installation
  - Dependencies
  - Using pip
  - From Source

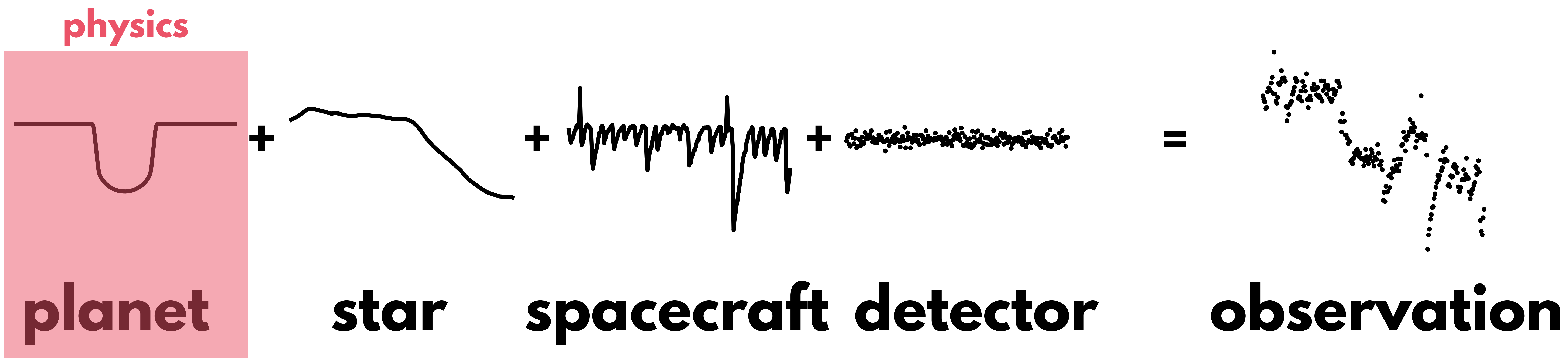
v: latest

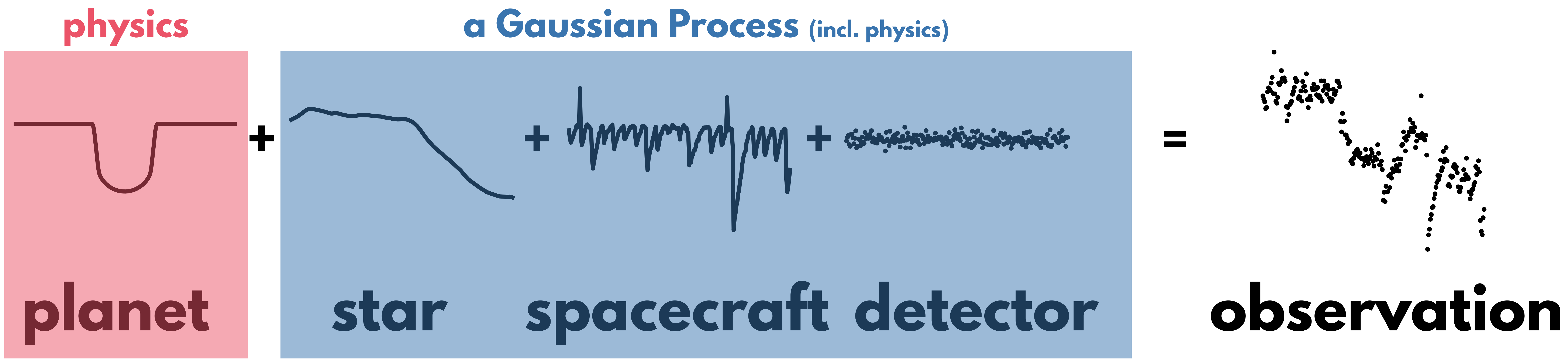
MCMC

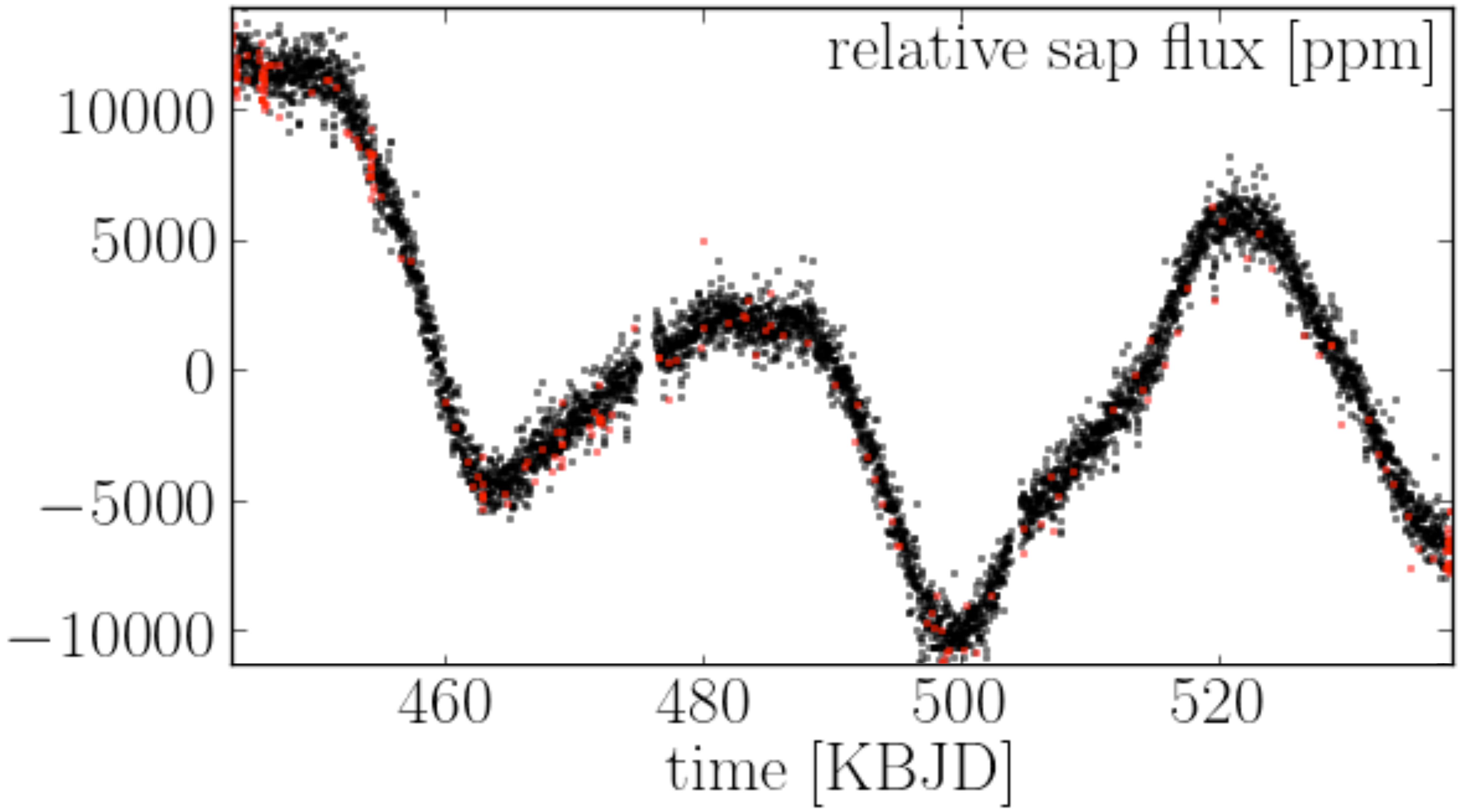
1. Markov chain Monte Carlo (**MCMC**)
2. Gaussian processes (**GPs**)

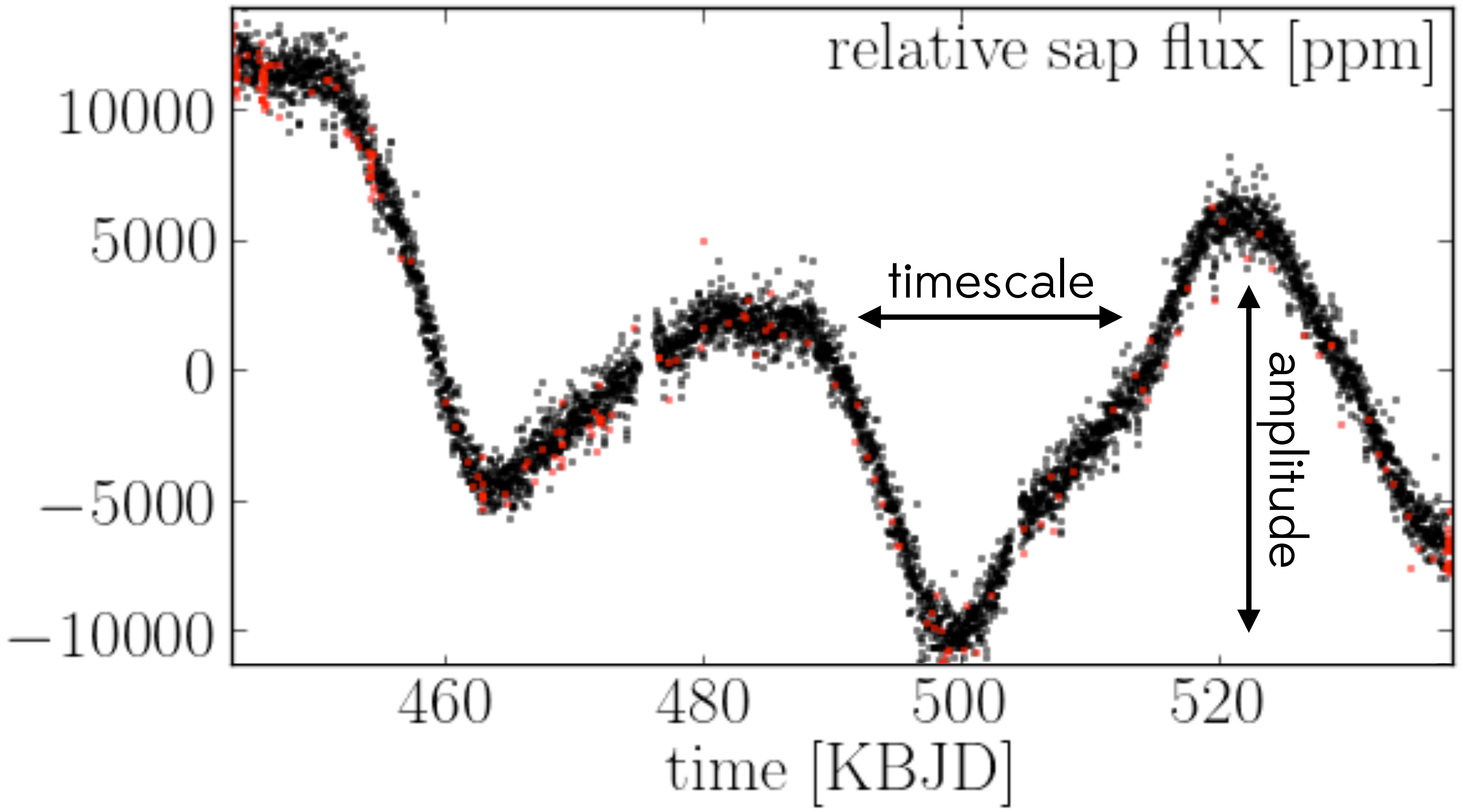
Gaussian processes

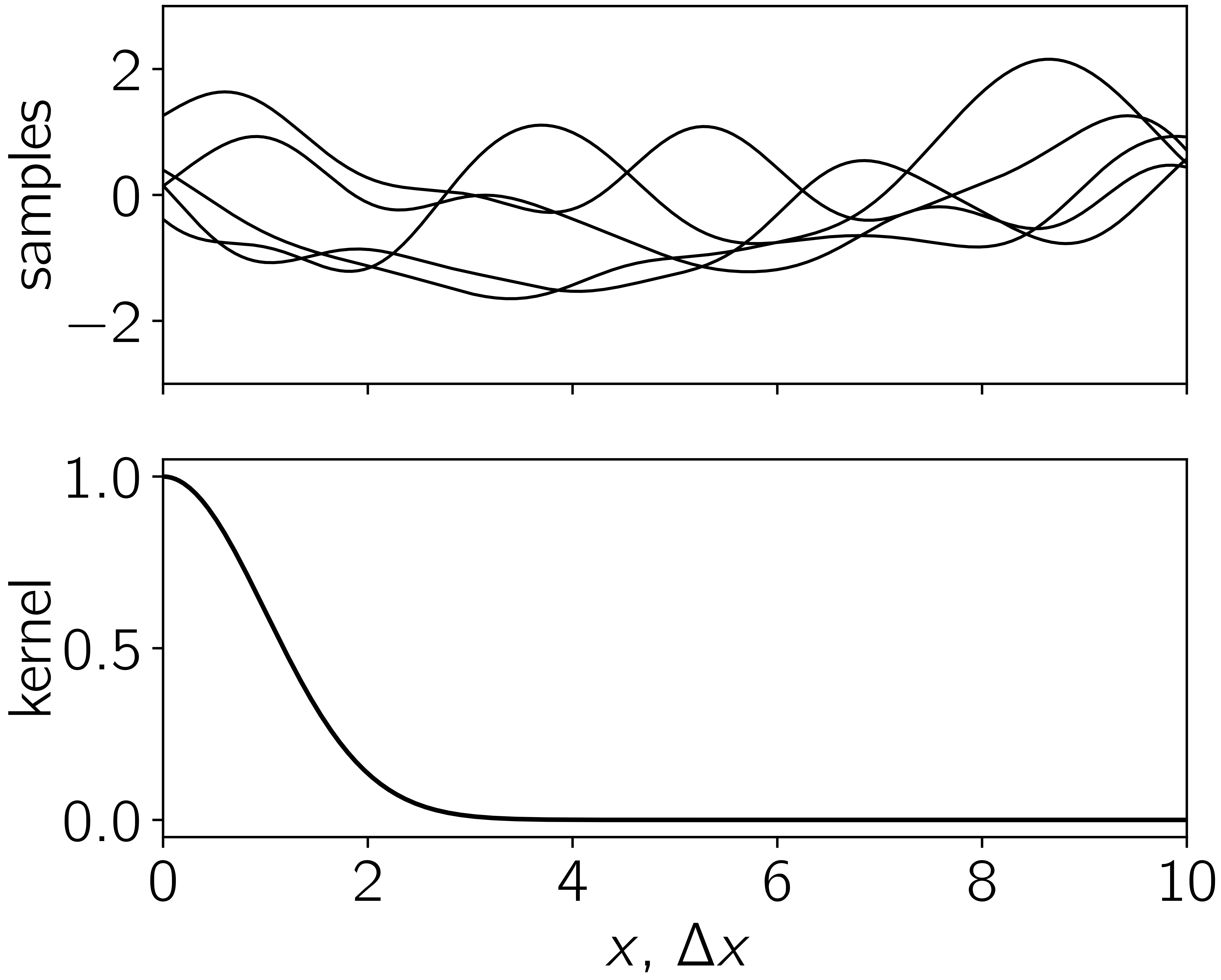

+
+
+
=
planet
star
spacecraft
detector
observation


physics
planet
+
star
+
spacecraft
+
detector
=
observation

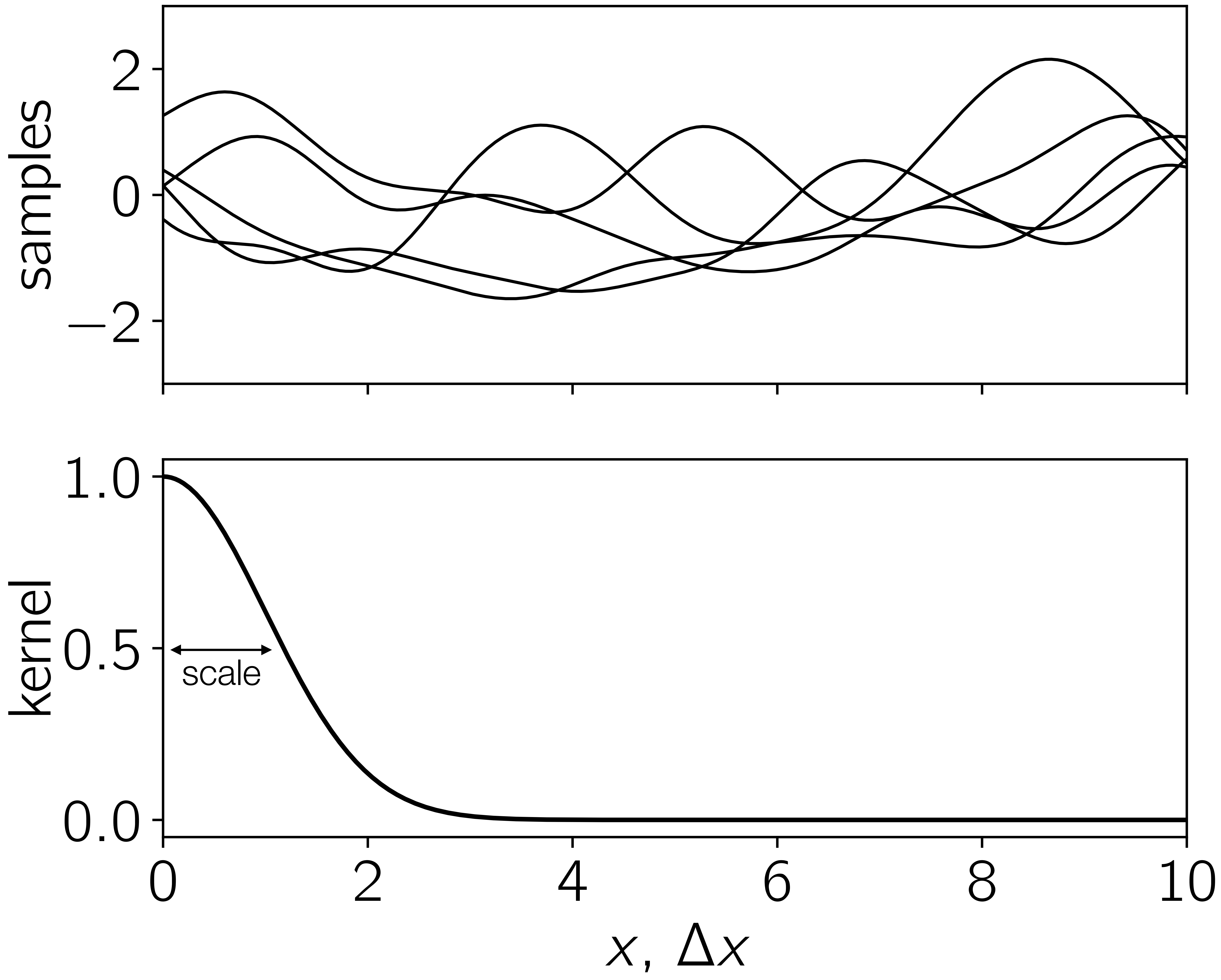

physics
a Gaussian Process (incl. physics)
planet
star
spacecraft
detector
observation


relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]

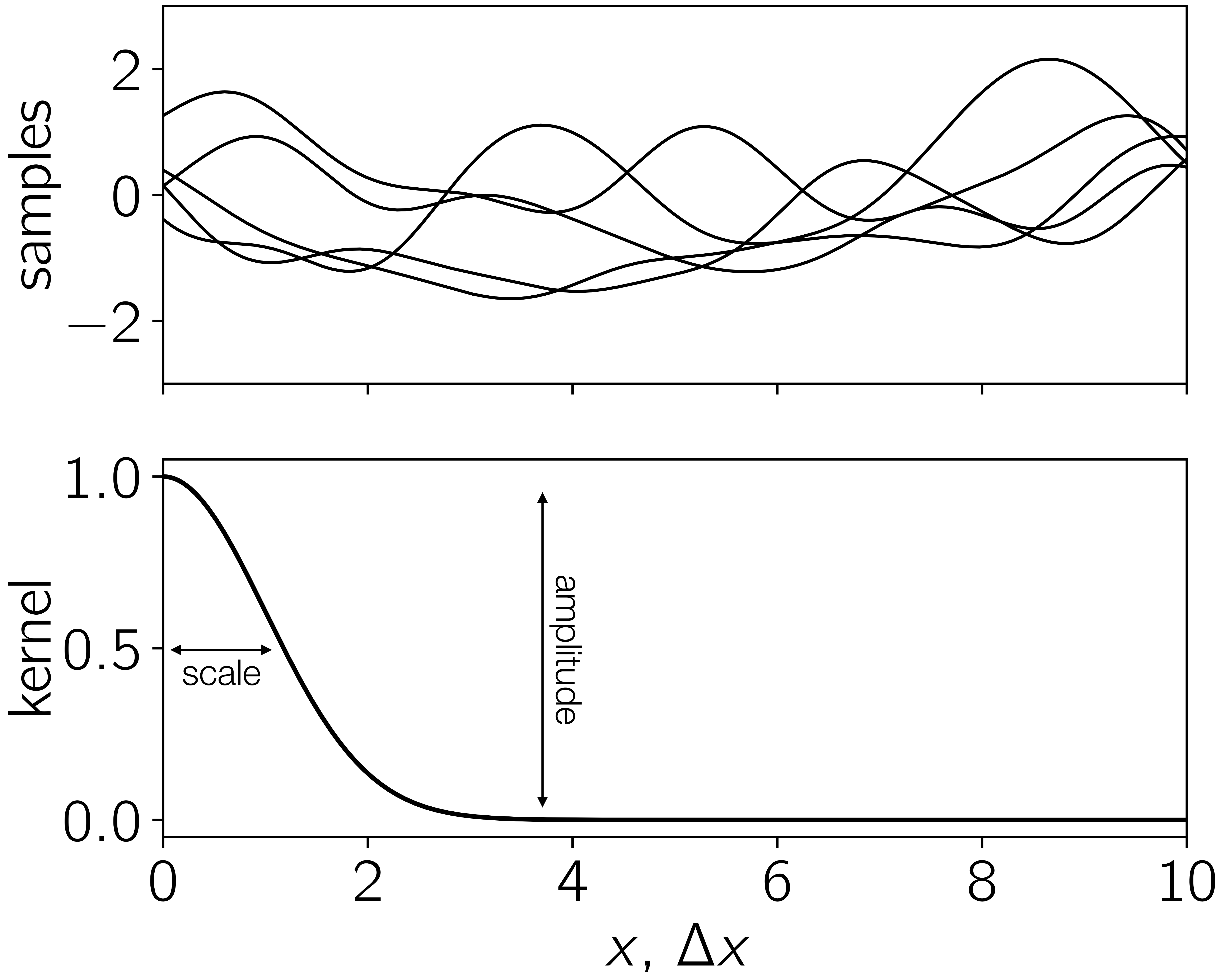

relative sap flux [ppm]
10000
5000
0
−5000
−10000
460
480
500
520
time [KBJD]
timescale
amplitude



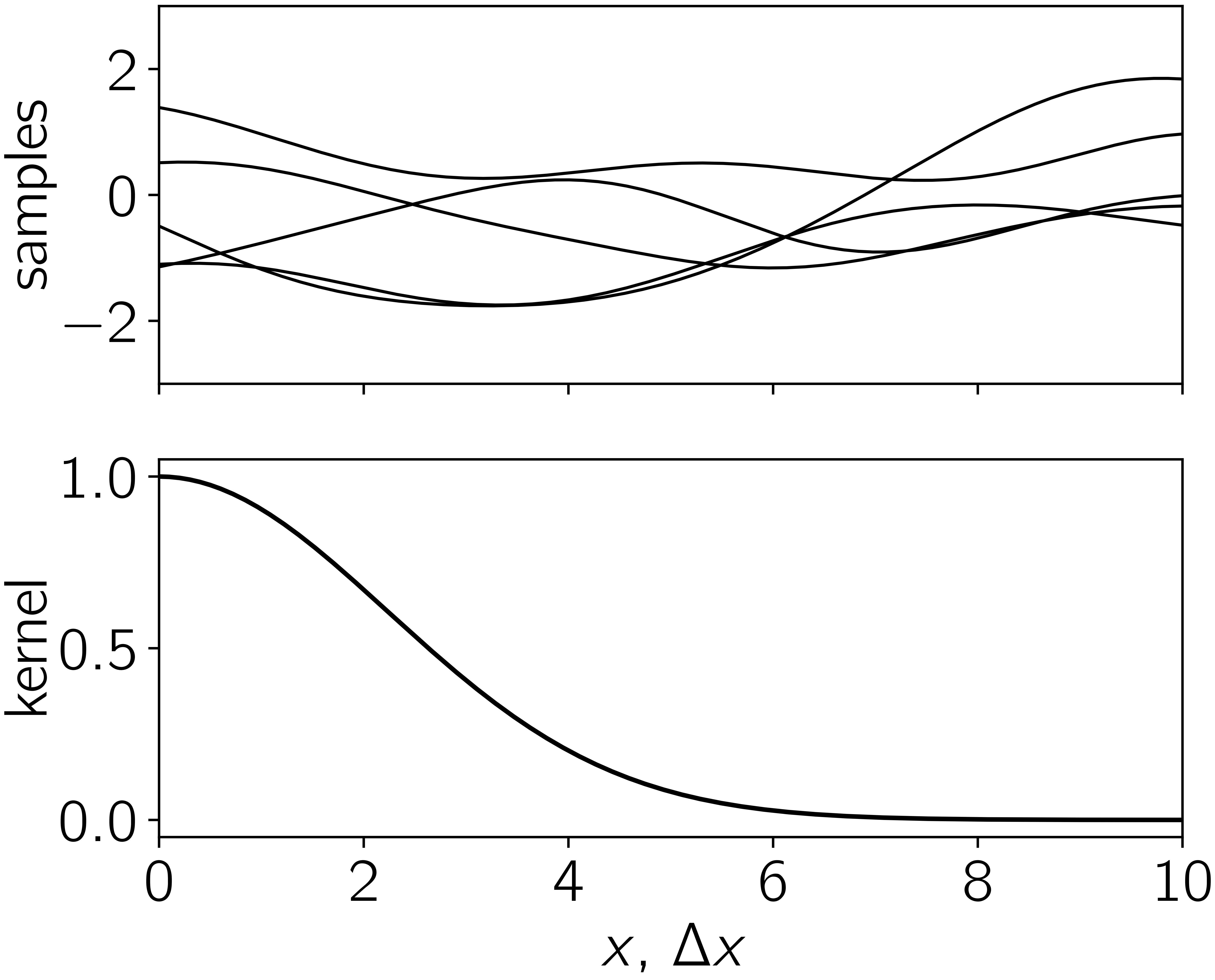
samples
2
0
−2
kernel
1.0
0.5
0.0
0
2
4
6
8
10
x, Δx



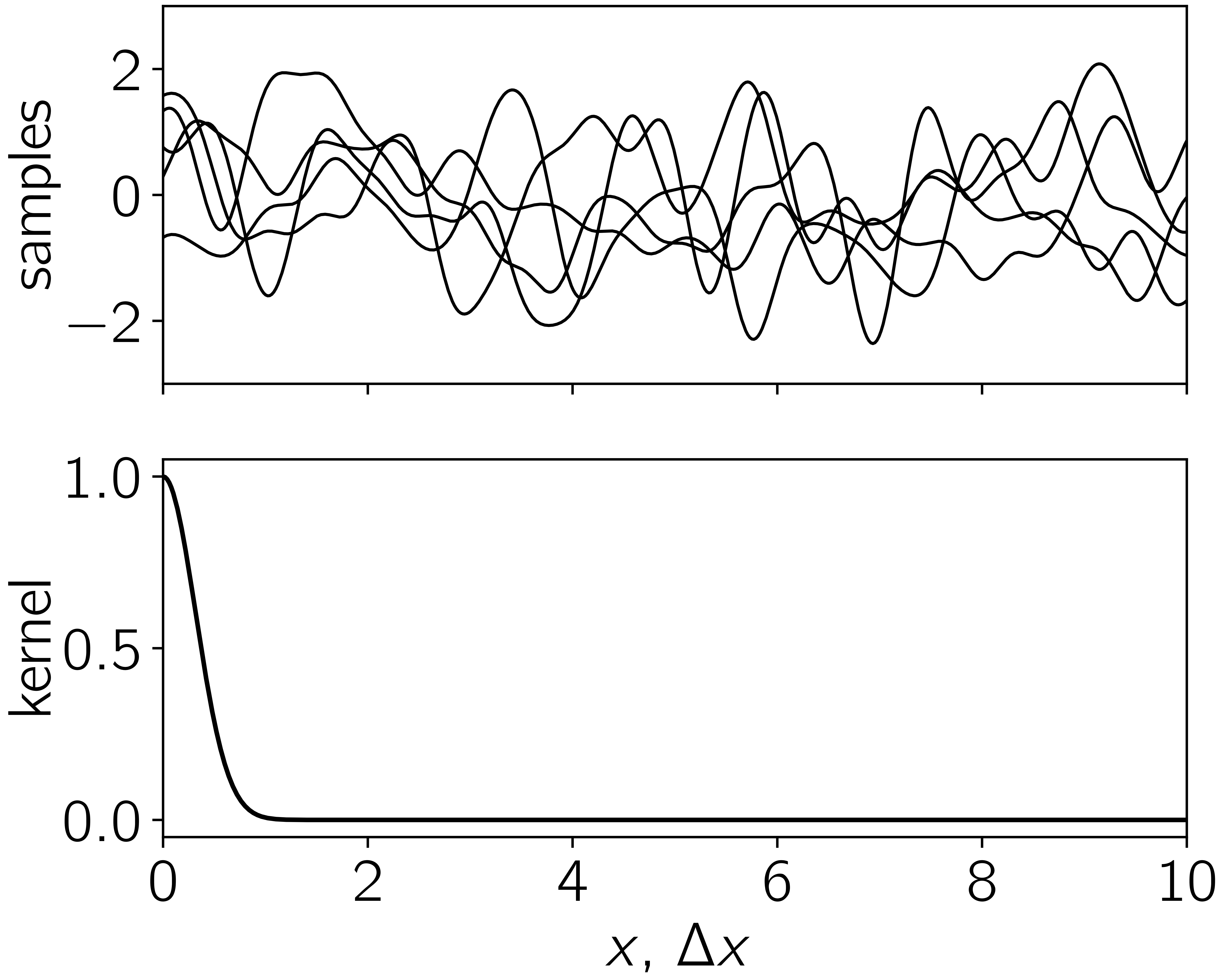
samples
2
0
−2
kernel
1.0
0.5
0.0
scale
0
2
4
6
8
10
x, Δx



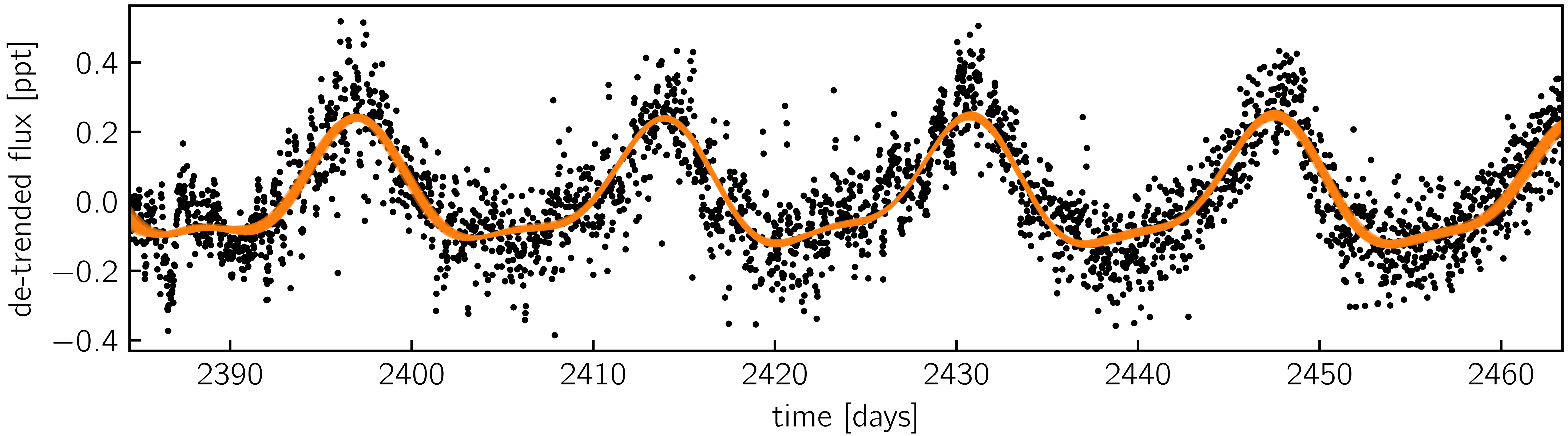
samples
2
0
−2
kernel
1.0
0.5
0.0
scale
amplitude
0
2
4
6
8
10
x, Δx



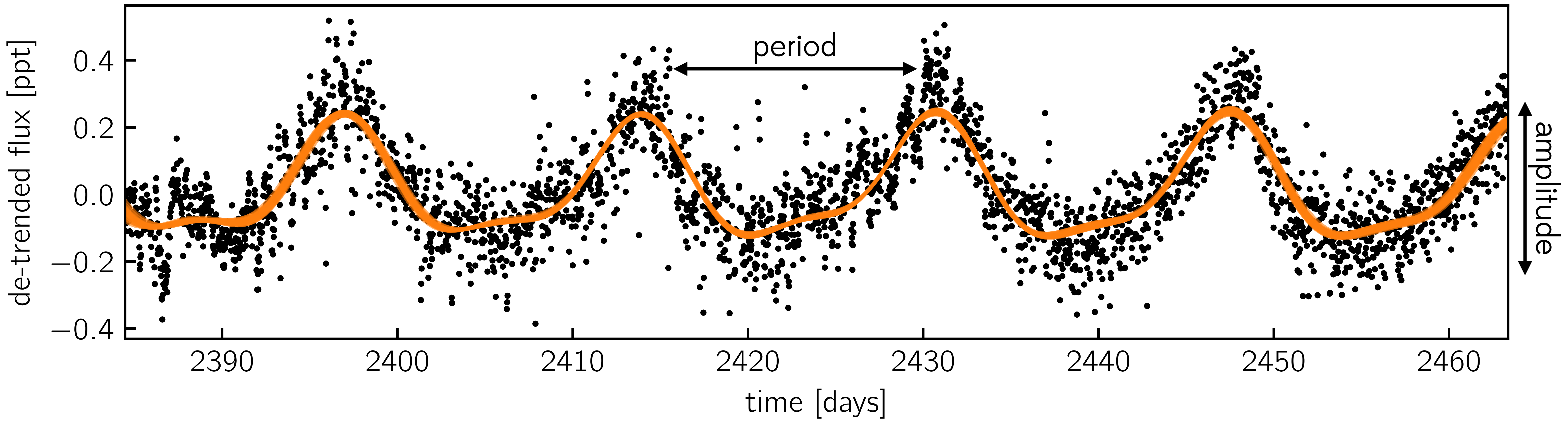
samples
2
0
−2
kernel
1.0
0.5
0.0
0
2
4
6
8
10
x, Δx



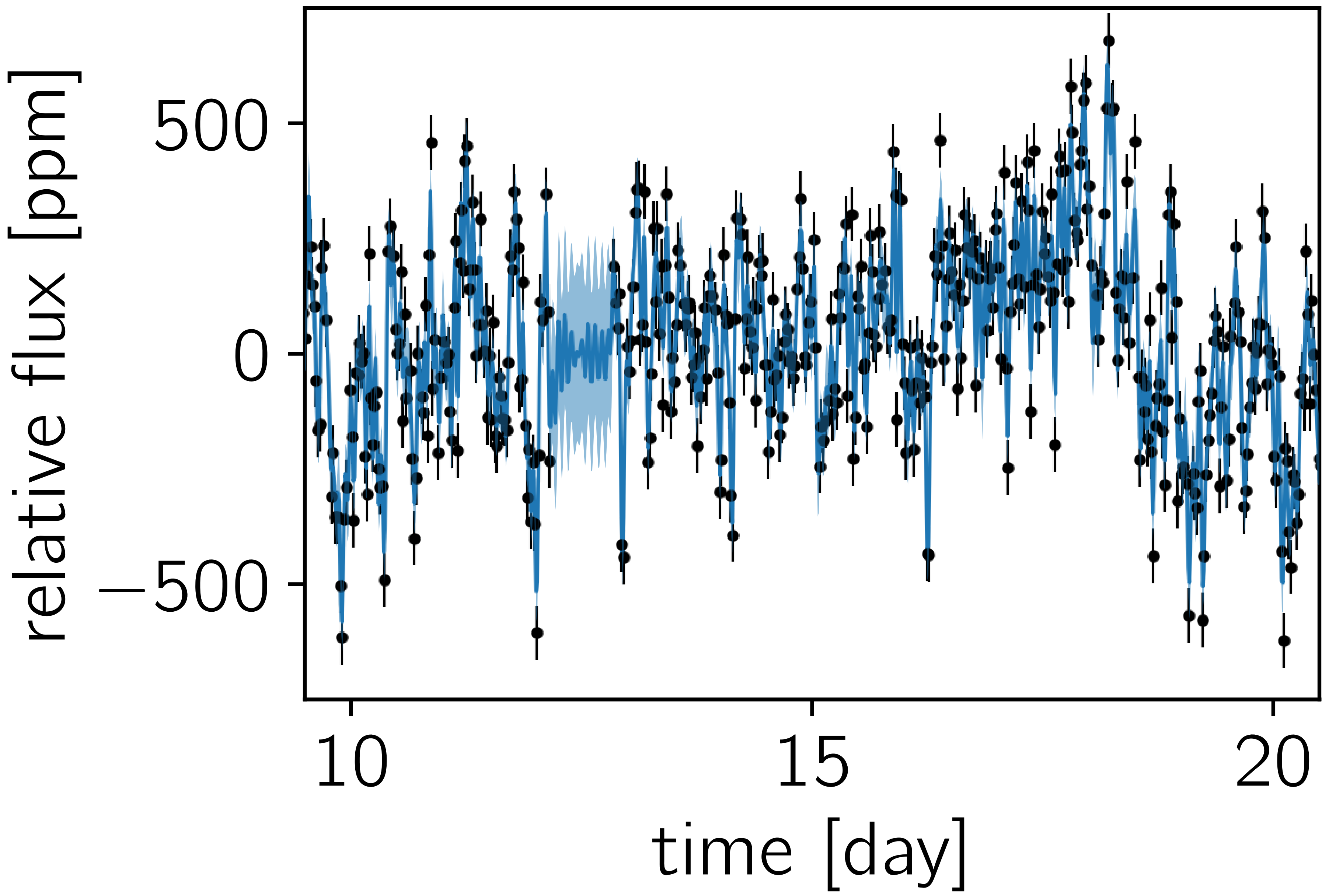
samples
2
0
−2
kernel
1.0
0.5
0.0
0
2
4
6
8
10
x, Δx

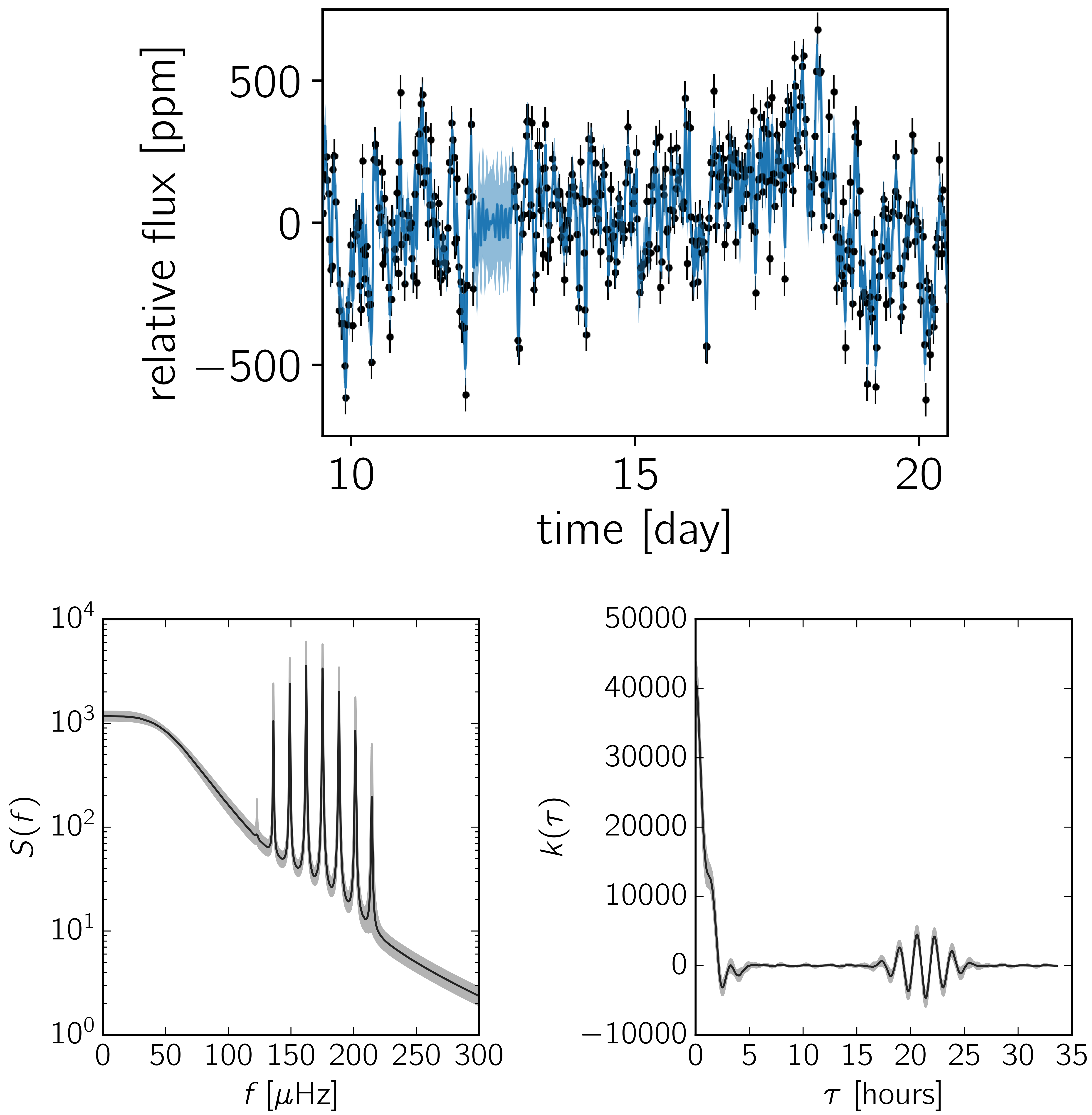

de-trended flux [ppt]
0.4
0.2
0.0
−0.2
−0.4
2390
2400
2410
2420
2430
2440
2450
2460
time [days]

GPs



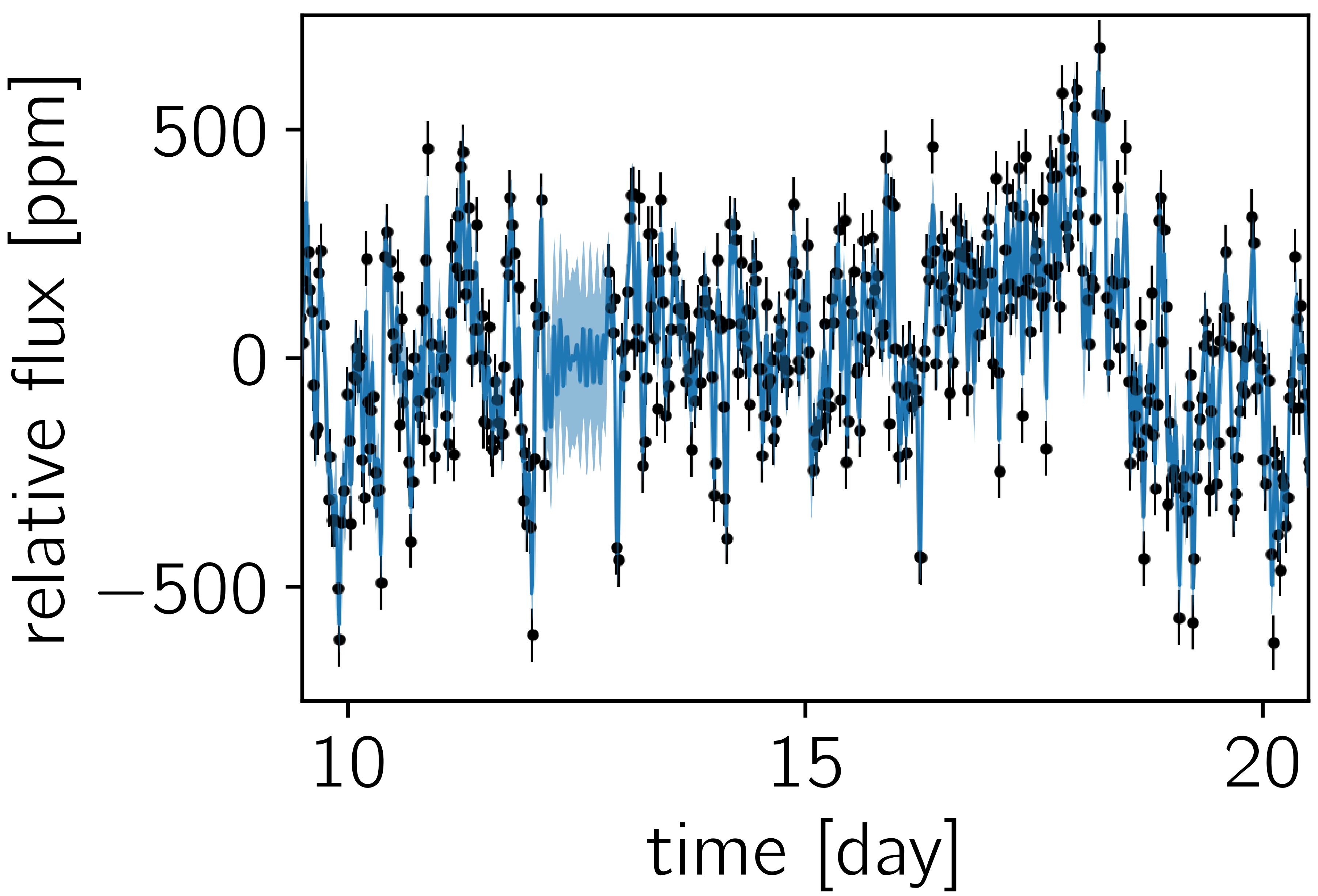
period
amplitude
de-trended flux [ppt]
0.4
0.2
0.0
−0.2
−0.4
2390
2400
2410
2420
2430
2440
2450
2460
time [days]

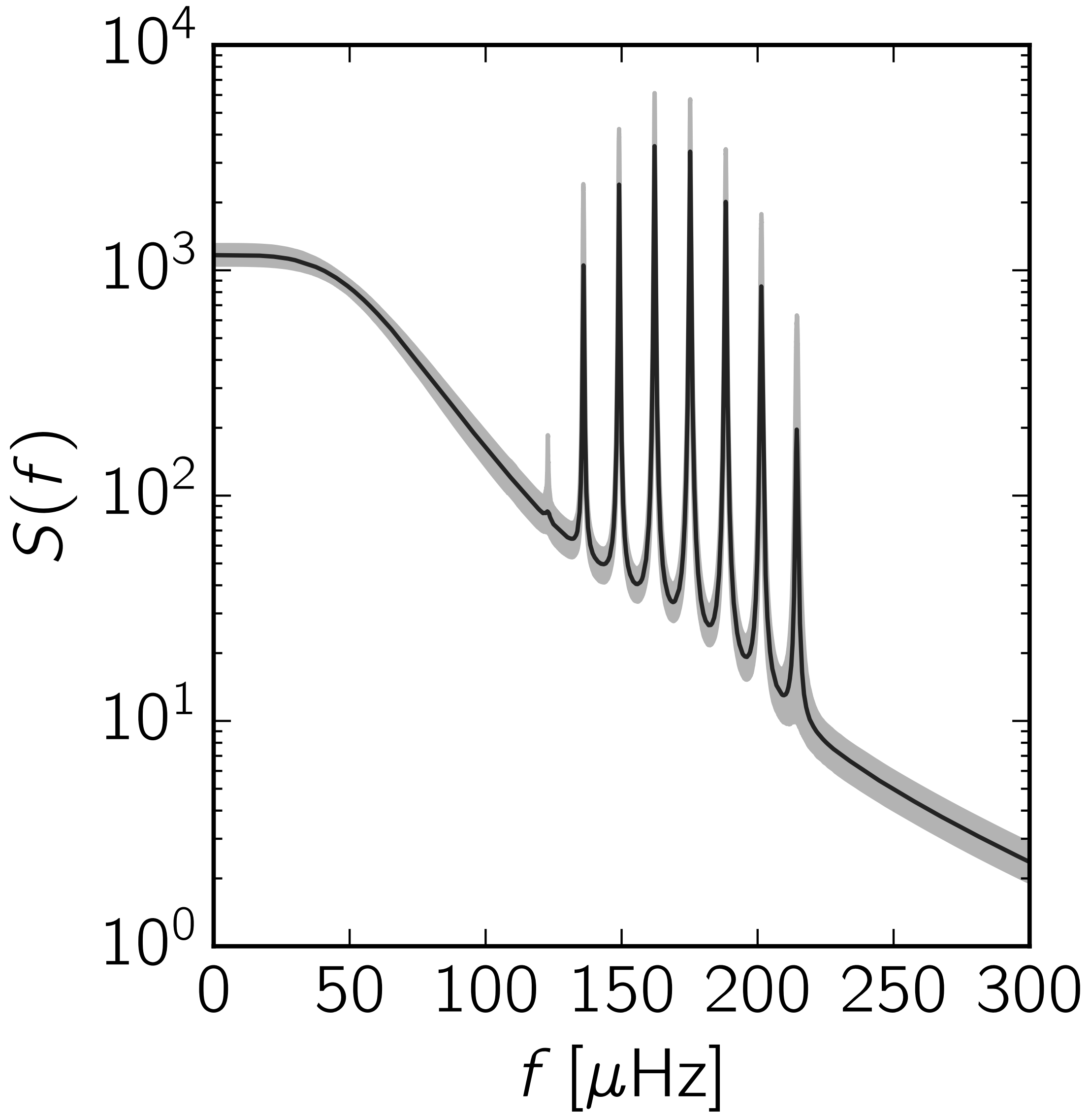

500
0
−500
relative flux [ppm]
10
15
20
time [day]

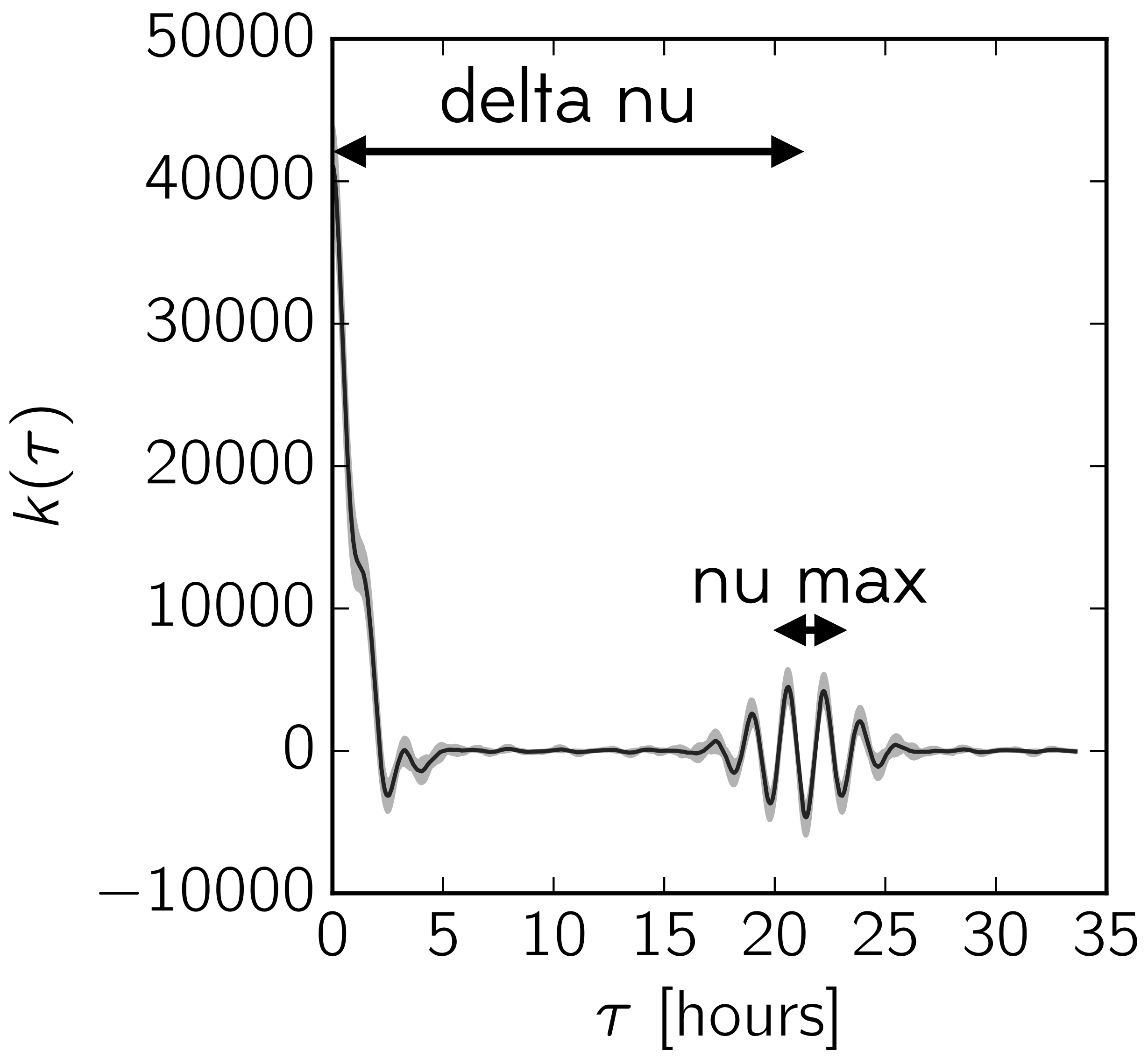


relative flux [ppm]
500
0
−500
10
15
20
time [day]
10^4
10^3
10^2
10^1
10^0
S(f)
0
50
100
150
200
250
300
f [μHz]
50000
40000
30000
20000
10000
0
−10000
k(τ)
0
5
10
15
20
25
30
35
τ [hours]

GPs


500
0
−500
relative flux [ppm]
10
15
20
time [day]


10^4
10^3
10^2
10^1
10^0
$S(f)$
0 50 100 150 200 250 300
f [μHz]


50000
40000
30000
20000
10000
0
−10000
$k(\tau)$
delta nu
nu max
0 5 10 15 20 25 30 35
τ [hours]

GPs

but...

There's a **problem**!

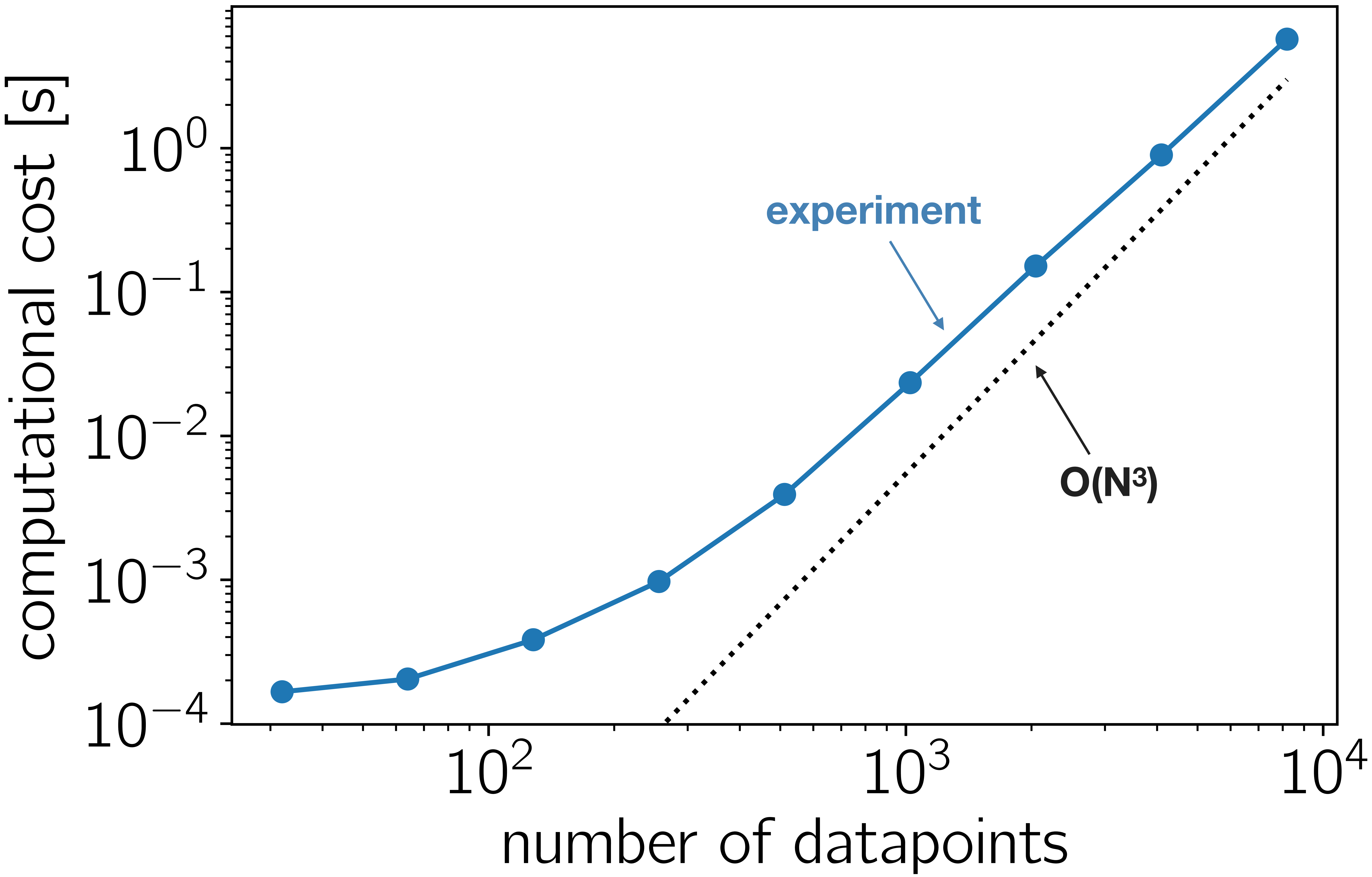
$$\log \mathcal{L}_{\mathrm{GP}} = -\frac{1}{2}\boldsymbol{r}^{\mathrm{T}} K^{-1} \boldsymbol{r} - \frac{1}{2} \log \det K - \frac{N}{2} \log 2\pi$$

uh oh!

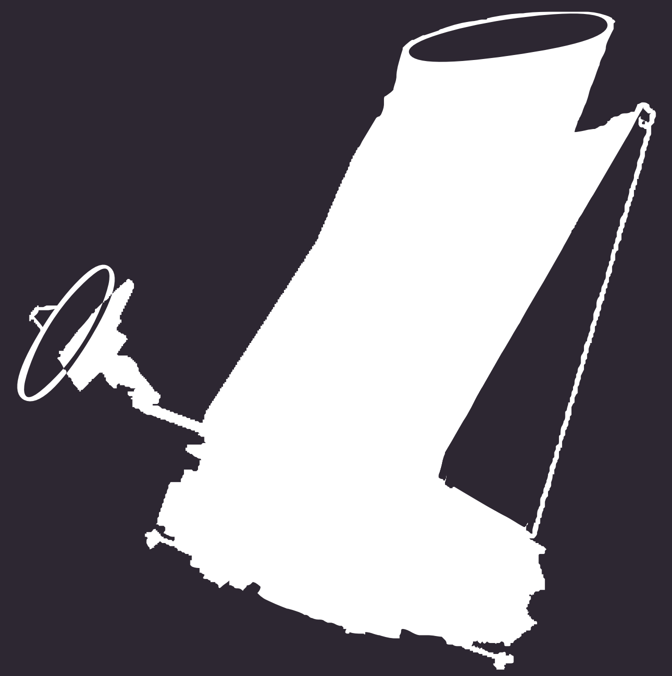
$$\log \mathcal{L}_{\mathrm{GP}} = -\frac{1}{2}\boldsymbol{r}^{\mathrm{T}} K^{-1} \boldsymbol{r} - \frac{1}{2}\log\det K - \frac{N}{2}\log 2\pi$$

uh oh! uh oh!!!

$$\log \mathcal{L}_{\mathrm{GP}} = -\frac{1}{2}\boldsymbol{r}^{\mathrm{T}} K^{-1}\boldsymbol{r} - \frac{1}{2}\log \det K - \frac{N}{2}\log 2\pi$$

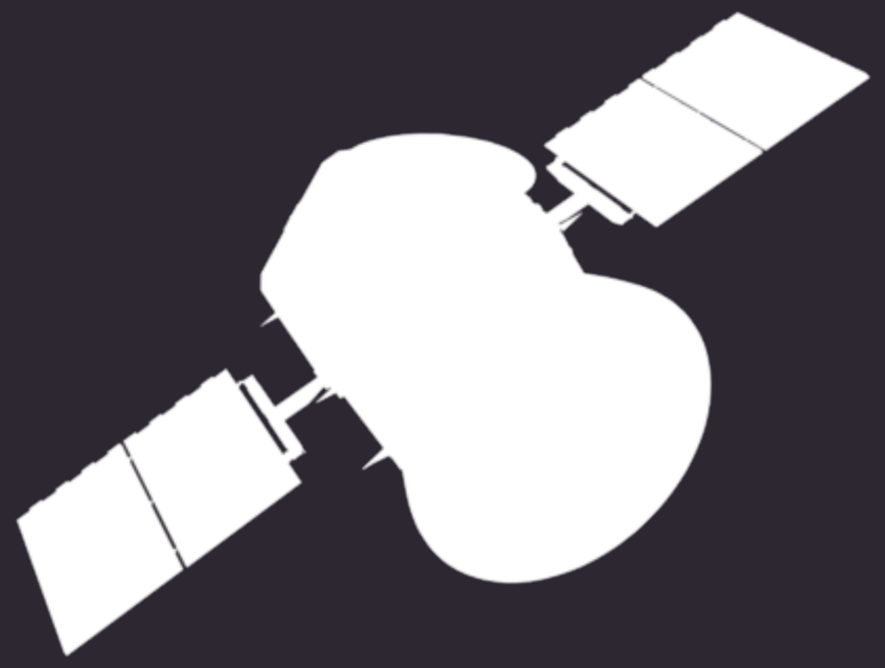


computational cost [s]
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
experiment
O(N³)
10^{2}
10^{3}
10^{4}
number of datapoints



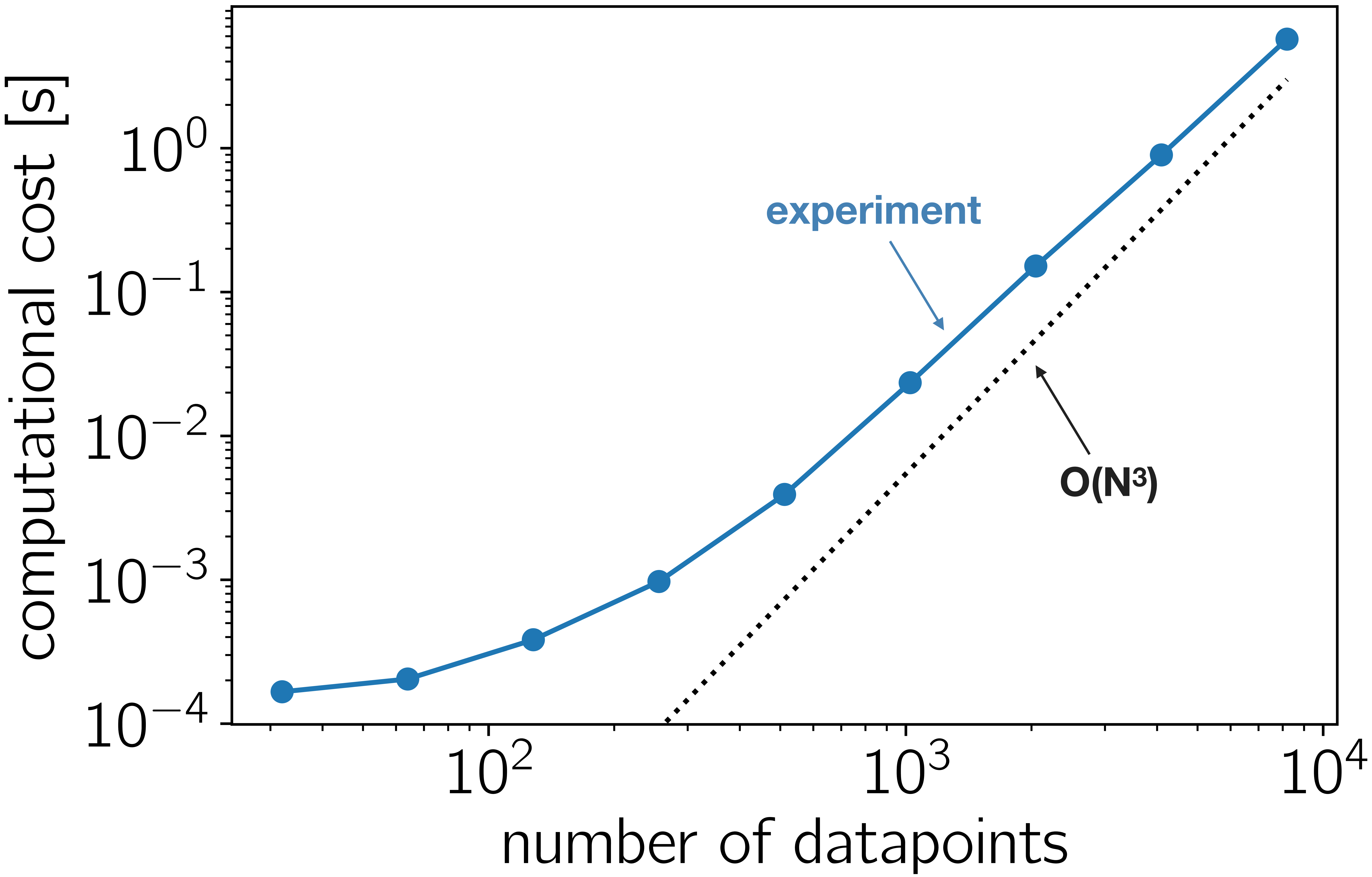
kepler

190k targets
70k obs./target



tess

500k targets
20k obs./target



computational cost [s]
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
experiment
O(N³)
10^2
10^3
10^4
number of datapoints

GPs

my first try: hodlr

Hierarchical **O**ff-**D**iagonal **L**ow **R**ank approximations.

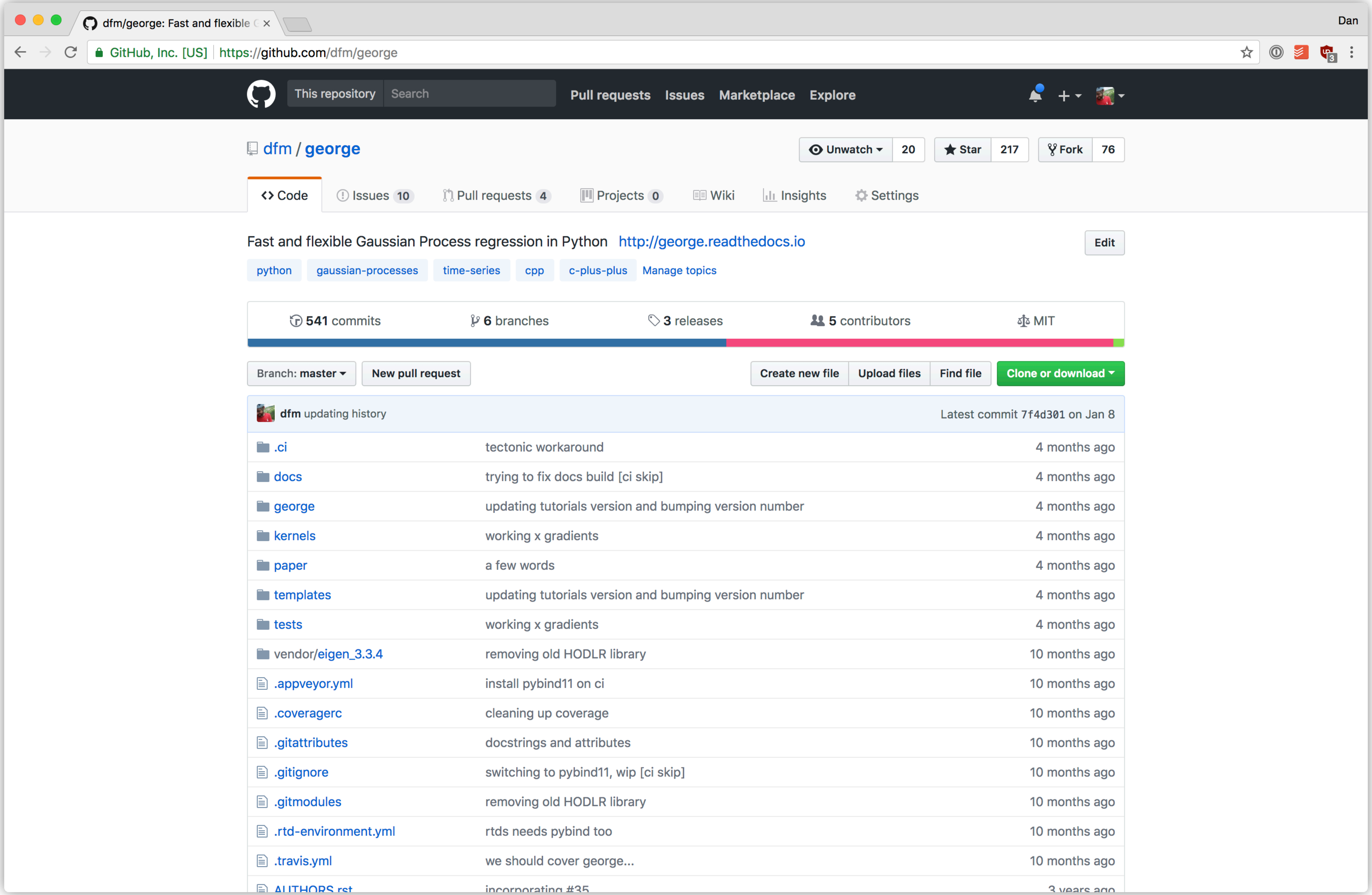
In collaboration: Sivaram Ambikasaran (NYU→IIT)

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 38, NO. 2, 2015

Fast Direct Methods for Gaussian Processes

Sivaram Ambikasaran, Daniel Foreman-Mackey, Leslie Greengard, *Member, IEEE,*
David W. Hogg, and Michael O'Neil, *Member, IEEE*

reference: arXiv:1403.6015


dfm/george: Fast and flexible
Dan
GitHub, Inc. [US] https://github.com/dfm/george
This repository Search
Pull requests Issues Marketplace Explore
dfm / george
Unwatch 20
Star 217
Fork 76
Code
Issues 10
Pull requests 4
Projects 0
Wiki
Insights
Settings
Fast and flexible Gaussian Process regression in Python http://george.readthedocs.io
Edit
python gaussian-processes time-series cpp c-plus-plus Manage topics
541 commits
6 branches
3 releases
5 contributors
MIT
Branch: master
New pull request
Create new file Upload files Find file Clone or download
dfm updating history
Latest commit 7f4d301 on Jan 8
.ci tectonic workaround 4 months ago
docs trying to fix docs build [ci skip] 4 months ago
george updating tutorials version and bumping version number 4 months ago
kernels working x gradients 4 months ago
paper a few words 4 months ago
templates updating tutorials version and bumping version number 4 months ago
tests working x gradients 4 months ago
vendor/eigen_3.3.4 removing old HODLR library 10 months ago
.appveyor.yml install pybind11 on ci 10 months ago
.coveragerc cleaning up coverage 10 months ago
.gitattributes docstrings and attributes 10 months ago
.gitignore switching to pybind11, wip [ci skip] 10 months ago
.gitmodules removing old HODLR library 10 months ago
.rtd-environment.yml rtds needs pybind too 10 months ago
.travis.yml we should cover george... 10 months ago
AUTHORS.rst incorporating #35 3 years ago

GPs

but...

That's still not **fast enough**!

my second try: célérité

Semi-separable matrices.

In collaboration: Sivaram Ambikasaran (IIT)

The Astronomical Journal, 154:220 (21pp), 2017 December

Fast and Scalable Gaussian Process Modeling with Applications to Astronomical Time Series

Daniel Foreman-Mackey[1,2,6], Eric Agol[1,7], Sivaram Ambikasaran[3], and Ruth Angus[4,5]
[1] Astronomy Department, University of Washington, Seattle, WA, USA
[2] Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, 6th Floor, New York, NY 10010, USA
[3] Department of Computational and Data Sciences, Indian Institute of Science, Bangalore, India
[4] Department of Astronomy, Columbia University, 550 W 120th Street, New York, NY 10027, USA
Received 2017 March 27; revised 2017 October 9; accepted 2017 October 10; published 2017 November 9

reference: arXiv:1703.09710

$\tau_{ij} = |t_i - t_j|$

$$k(\tau_{ij}) = {\sigma_i}^2 \delta_{ij} + \sum_{j=1}^{J} \left[a_j\, e^{-c_j \tau}\ \cos(d_j \tau) + b_j\, e^{-c_j \tau}\ \sin(d_j \tau)\right]$$

reference: DFM+ (2017), arXiv:1703.09710

$$\tau_{ij} = |t_i - t_j|$$

stochastically-driven damped simple harmonic oscillator

$$k(\tau_{ij}) = {\sigma_i}^2 \delta_{ij} + \sum_{j=1}^{J} \left[a_j\, e^{-c_j \tau}\, \cos(d_j \tau) + b_j\, e^{-c_j \tau}\, \sin(d_j \tau)\right]$$

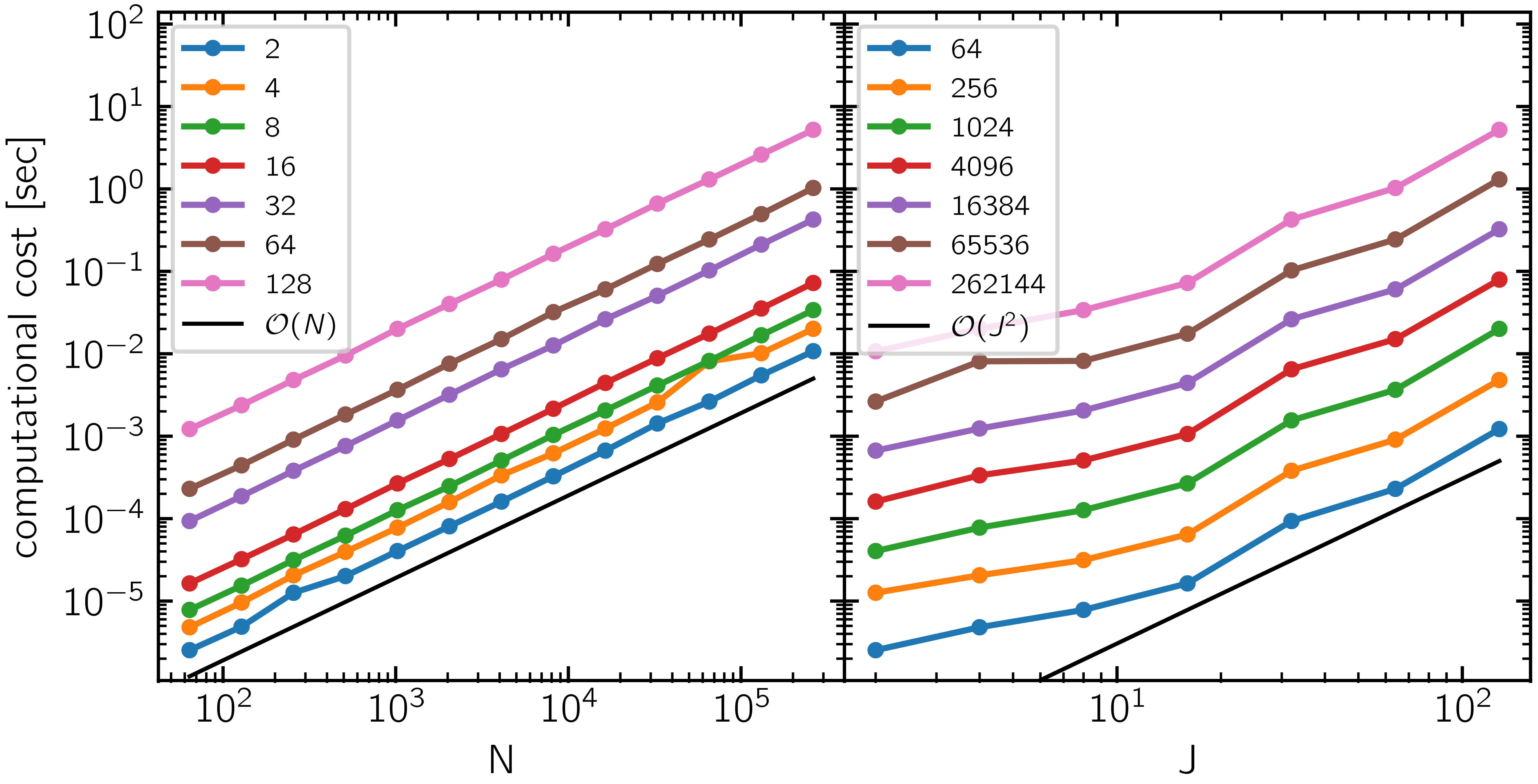
reference: DFM+ (2017), arXiv:1703.09710

```
function celerite_factor(U, P, d, W)
    # Initially d = a and W = V
    S ← zeros(J, J)
    w_1 ← w_1 / d_1
    for n = 2, ..., N:
        S ← diag(p_{n−1}) [S + d_{n−1} w_{n−1}^T w_{n−1}] diag(p_{n−1})
        d_n ← d_n − u_n S u_n^T
        w_n ← [w_n − u_n S] / d_n
    return d, W, S
```

$$
\begin{aligned}
&\textbf{function}\ \texttt{celerite\_factor}(U,\ P,\ \boldsymbol{d},\ W)\\
&\quad \#\ \text{Initially}\ \boldsymbol{d} = \boldsymbol{a}\ \text{and}\ W = V\\
&\quad S \leftarrow \texttt{zeros}(J,\ J)\\
&\quad \boldsymbol{w}_1 \leftarrow \boldsymbol{w}_1 / d_1\\
&\quad \text{for}\ n = 2, \ldots, N:\\
&\qquad S \leftarrow \texttt{diag}(\boldsymbol{p}_{n-1})\,[S + d_{n-1}\,{\boldsymbol{w}_{n-1}}^{\mathrm{T}}\,\boldsymbol{w}_{n-1}]\,\texttt{diag}(\boldsymbol{p}_{n-1})\\
&\qquad d_n \leftarrow d_n - \boldsymbol{u}_n\,S\,{\boldsymbol{u}_n}^{\mathrm{T}}\\
&\qquad \boldsymbol{w}_n \leftarrow [\boldsymbol{w}_n - \boldsymbol{u}_n\,S]\,/d_n\\
&\quad \textbf{return}\ \boldsymbol{d},\ W,\ S
\end{aligned}
$$

reference: DFM (2018), arXiv:1801.10156

$$\mathcal{O}(NJ^2)$$



reference: DFM (2018), arXiv:1801.10156

GitHub - dfm/celerite: Scalable

GitHub, Inc. [US] | https://github.com/dfm/celerite

dfm / celerite

Watch 12 | Star 112 | Fork 28

Code | Issues 8 | Pull requests 2 | Projects 0 | Insights

Scalable 1D Gaussian Processes in C++, Python, and Julia http://celerite.rtfd.io

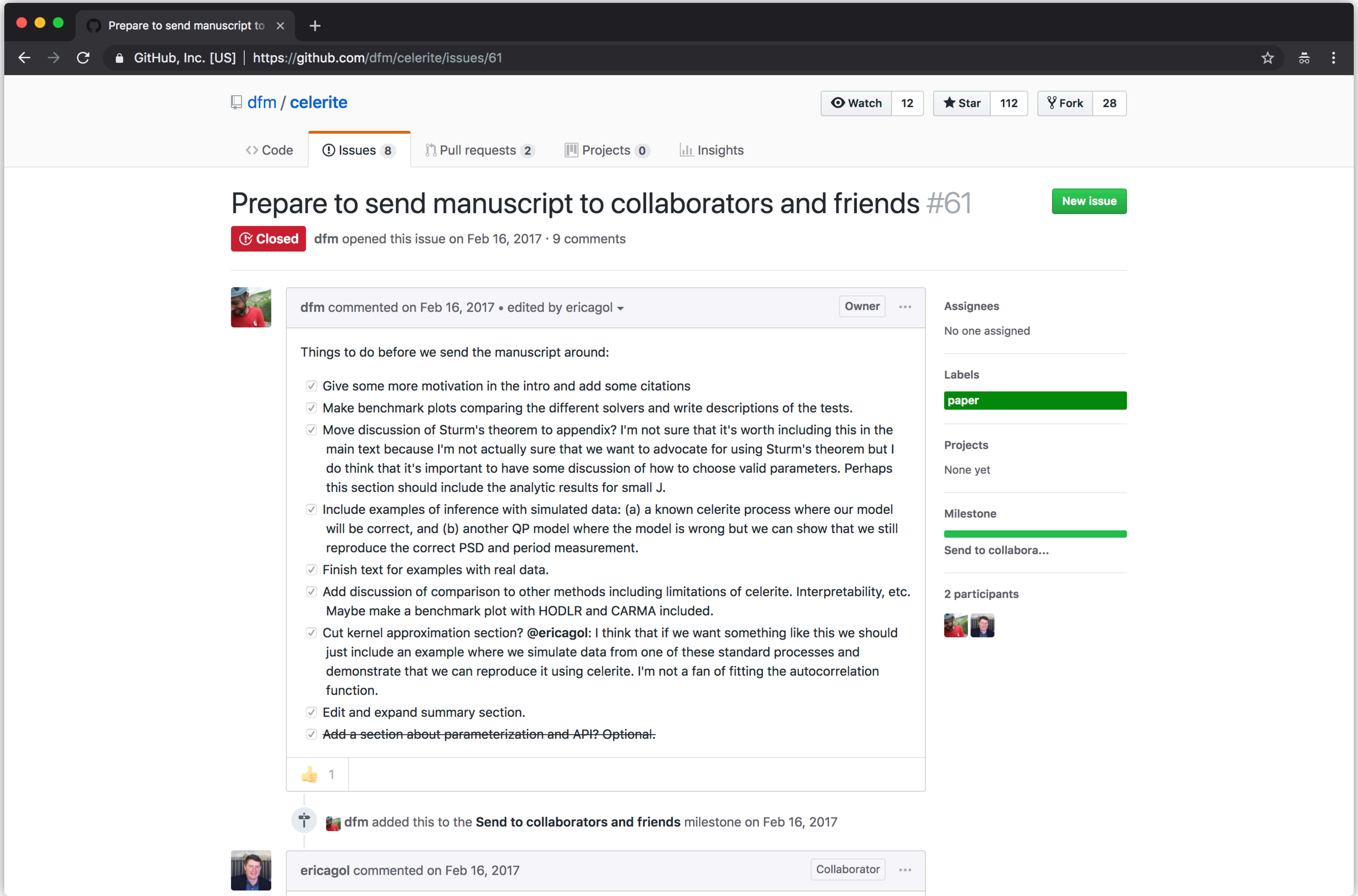
python | gaussian-processes | astronomy | time-series | stars | exoplanets | julia | cpp | c-plus-plus

1,008 commits | 4 branches | 7 releases | 9 contributors | MIT

Branch: master | New pull request | Find file | Clone or download

dfm updating bibtex reference — Latest commit 846c239 on Nov 28, 2017

.ci	develop on ci	2 years ago
celerite	updating bibtex reference	a year ago
cpp	speeding up dot products and improving dot interface	a year ago
docs	updating zenodo DOI	2 years ago
examples/benchmark	updating benchmark plots	2 years ago
paper	zenodo links	a year ago
tests	noqa annoyingness	2 years ago
.appveyor.yml	Update .appveyor.yml	a year ago
.coveragerc	adding coverage and coveralls	2 years ago
.gitattributes	Update .gitattributes	2 years ago
.gitignore	bumping version number to 0.3.0	a year ago
.rtd-environment.yml	Update .rtd-environment.yml	a year ago
.travis.yml	develop on ci	2 years ago
AUTHORS.rst	working on docs [ci skip]	2 years ago
CITATION	bumping version number	2 years ago
HISTORY.rst	bumping version number to 0.3.0	a year ago
LICENSE	Adding 2017 to license [ci skip]	2 years ago



Prepare to send manuscript to ✕

GitHub, Inc. [US] | https://github.com/dfm/celerite/issues/61

dfm / celerite

Watch 12 | Star 112 | Fork 28

Code | Issues 8 | Pull requests 2 | Projects 0 | Insights

Prepare to send manuscript to collaborators and friends #61

New issue

Closed **dfm** opened this issue on Feb 16, 2017 · 9 comments

dfm commented on Feb 16, 2017 • edited by ericagol — Owner

Things to do before we send the manuscript around:

- [x] Give some more motivation in the intro and add some citations
- [x] Make benchmark plots comparing the different solvers and write descriptions of the tests.
- [x] Move discussion of Sturm's theorem to appendix? I'm not sure that it's worth including this in the main text because I'm not actually sure that we want to advocate for using Sturm's theorem but I do think that it's important to have some discussion of how to choose valid parameters. Perhaps this section should include the analytic results for small J.
- [x] Include examples of inference with simulated data: (a) a known celerite process where our model will be correct, and (b) another QP model where the model is wrong but we can show that we still reproduce the correct PSD and period measurement.
- [x] Finish text for examples with real data.
- [x] Add discussion of comparison to other methods including limitations of celerite. Interpretability, etc. Maybe make a benchmark plot with HODLR and CARMA included.
- [x] Cut kernel approximation section? **@ericagol**: I think that if we want something like this we should just include an example where we simulate data from one of these standard processes and demonstrate that we can reproduce it using celerite. I'm not a fan of fitting the autocorrelation function.
- [x] Edit and expand summary section.
- [x] ~~Add a section about parameterization and API? Optional.~~

👍 1

dfm added this to the **Send to collaborators and friends** milestone on Feb 16, 2017

ericagol commented on Feb 16, 2017 — Collaborator

Assignees
No one assigned

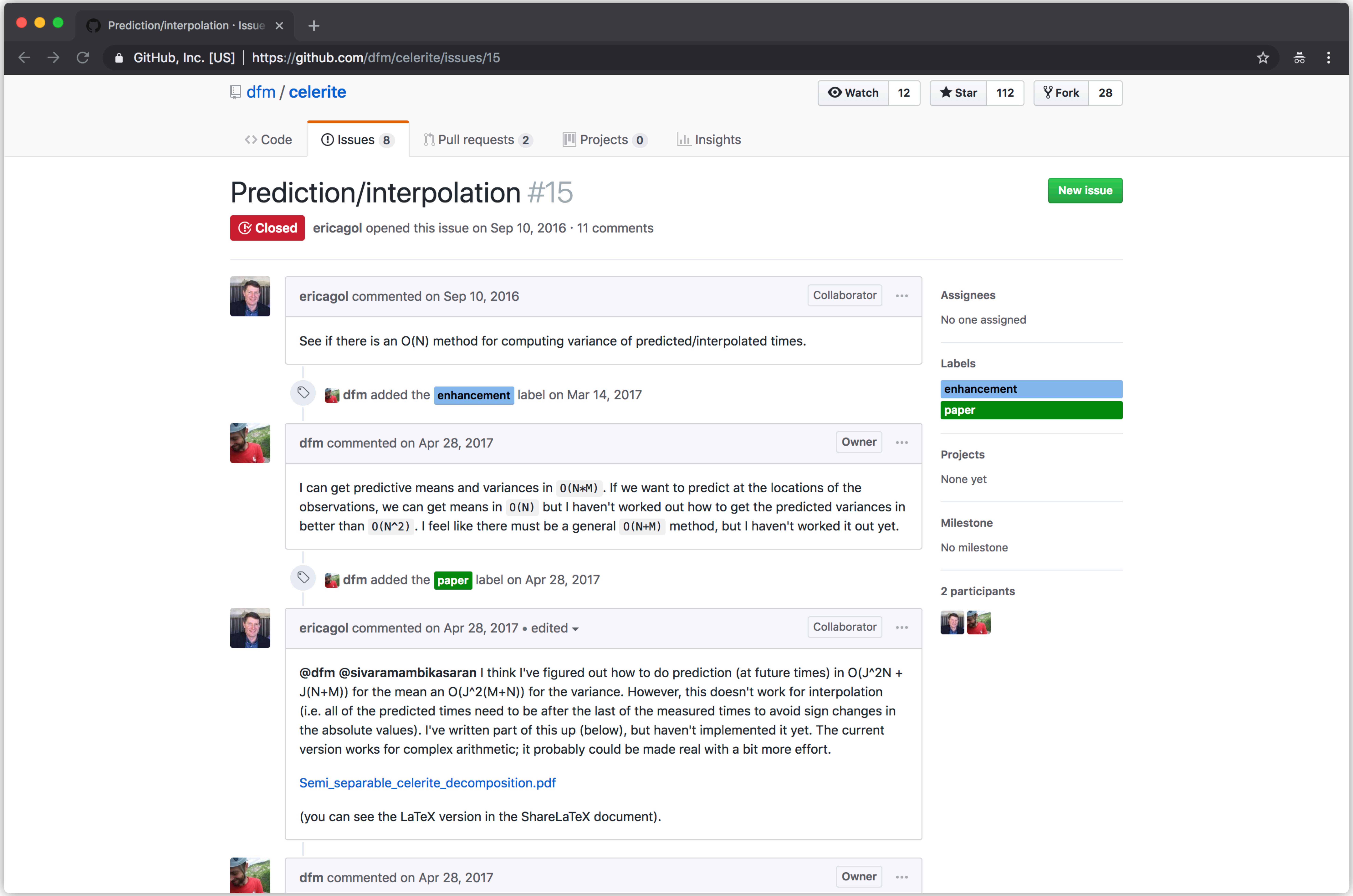
Labels
paper

Projects
None yet

Milestone
Send to collabora...

2 participants

GPs



Prediction/interpolation · Issue

GitHub, Inc. [US] | https://github.com/dfm/celerite/issues/15

dfm / celerite

Watch 12 | Star 112 | Fork 28

Code | Issues 8 | Pull requests 2 | Projects 0 | Insights

Prediction/interpolation #15

New issue

Closed ericagol opened this issue on Sep 10, 2016 · 11 comments

ericagol commented on Sep 10, 2016 — Collaborator

See if there is an O(N) method for computing variance of predicted/interpolated times.

dfm added the enhancement label on Mar 14, 2017

dfm commented on Apr 28, 2017 — Owner

I can get predictive means and variances in `O(N*M)`. If we want to predict at the locations of the observations, we can get means in `O(N)` but I haven't worked out how to get the predicted variances in better than `O(N^2)`. I feel like there must be a general `O(N+M)` method, but I haven't worked it out yet.

dfm added the paper label on Apr 28, 2017

ericagol commented on Apr 28, 2017 • edited — Collaborator

@dfm @sivaramambikasaran I think I've figured out how to do prediction (at future times) in O(J^2N + J(N+M)) for the mean an O(J^2(M+N)) for the variance. However, this doesn't work for interpolation (i.e. all of the predicted times need to be after the last of the measured times to avoid sign changes in the absolute values). I've written part of this up (below), but haven't implemented it yet. The current version works for complex arithmetic; it probably could be made real with a bit more effort.

Semi_separable_celerite_decomposition.pdf

(you can see the LaTeX version in the ShareLaTeX document).

dfm commented on Apr 28, 2017 — Owner

Assignees: No one assigned

Labels: enhancement, paper

Projects: None yet

Milestone: No milestone

2 participants

GPs

limitations

célérité is restricted to **semi-separable matrices** and (in my implementation) this means **1D inputs** with a **stationary kernel**.

summary

célérité provides a **scalable** method for evaluating GP likelihoods. This makes GP inference feasible for astronomical surveys.

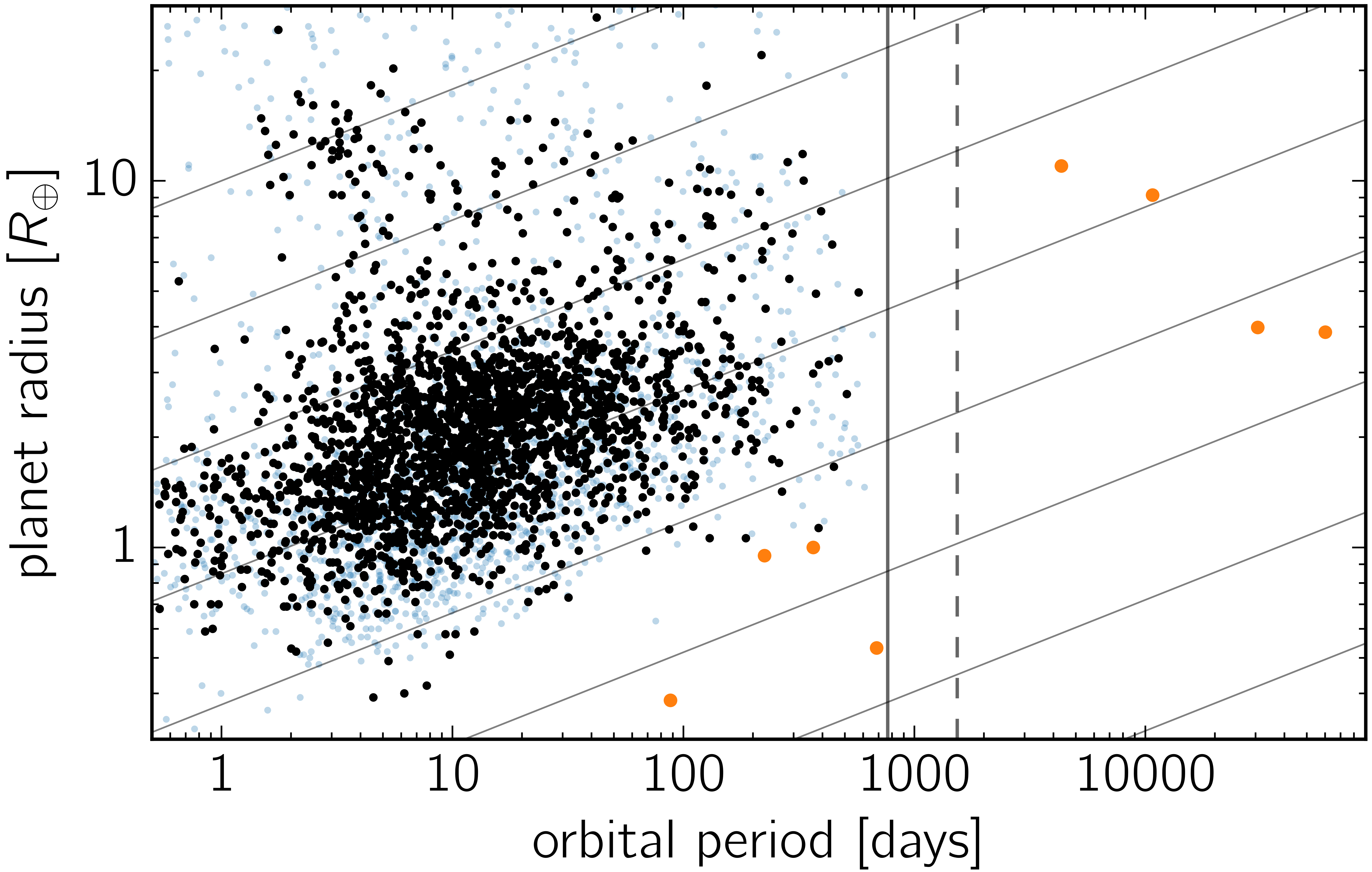
The class of kernel functions have a **useful physical interpretation**.

future directions

1. Combine with **deep kernel learning** to make the kernel more flexible.
2. Extend to **multiple dimensions** for **structured** data (e.g. parallel time series).

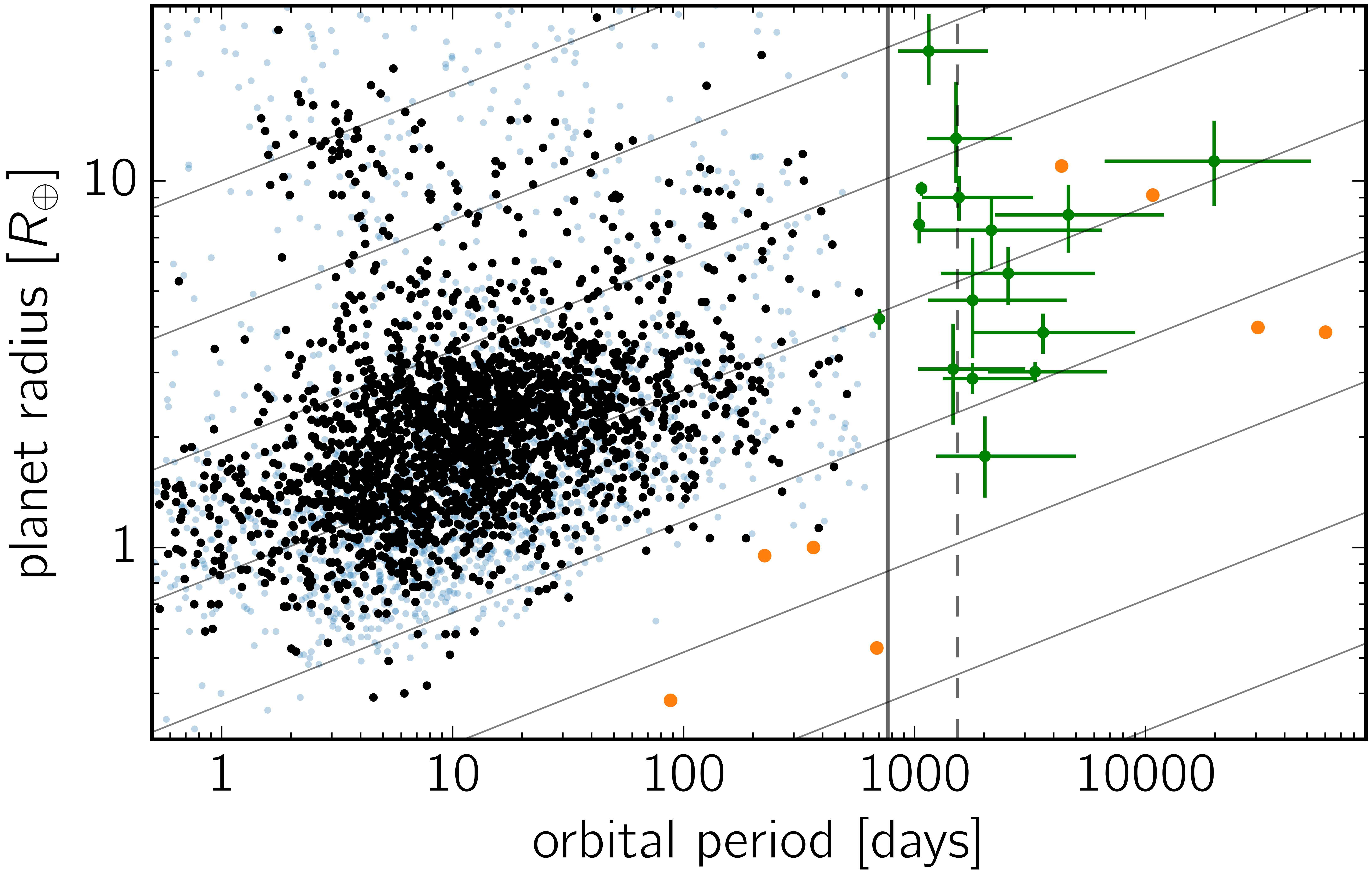
long period transiting planets

it all comes together...



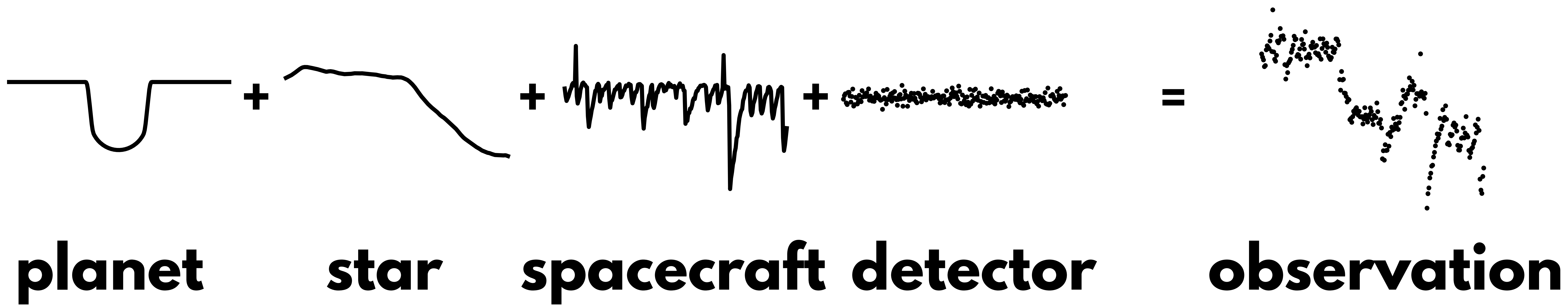
source: NASA Exoplanet Archive

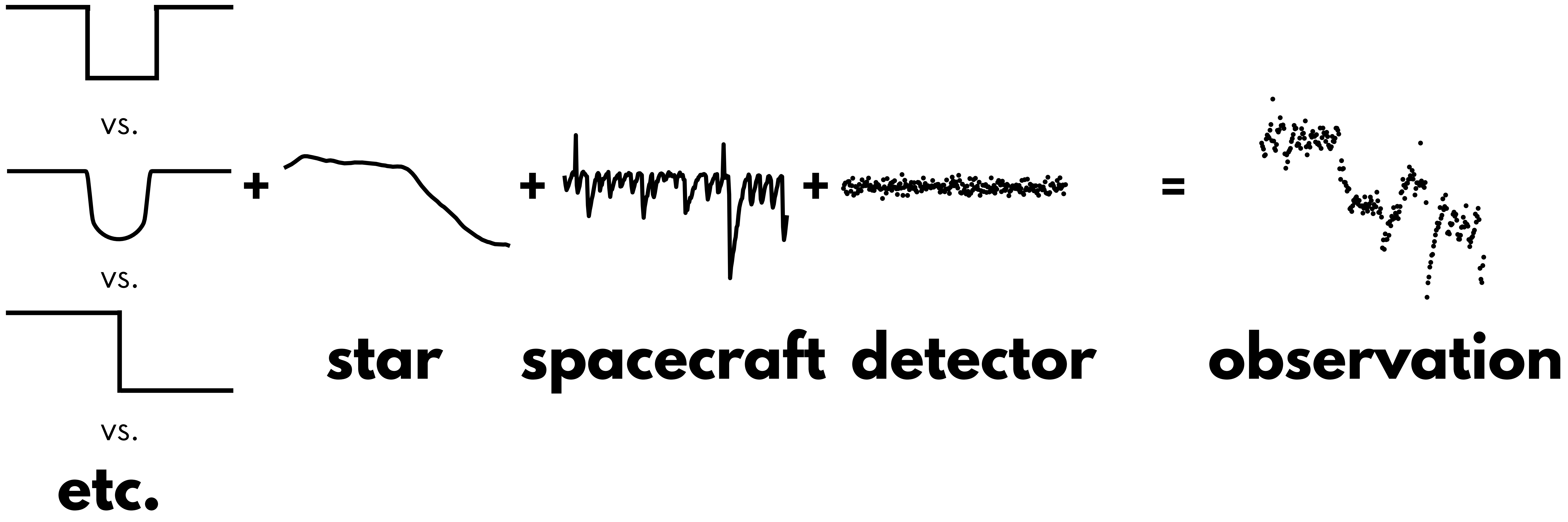
DFM+ (2016); arXiv:1607.08237

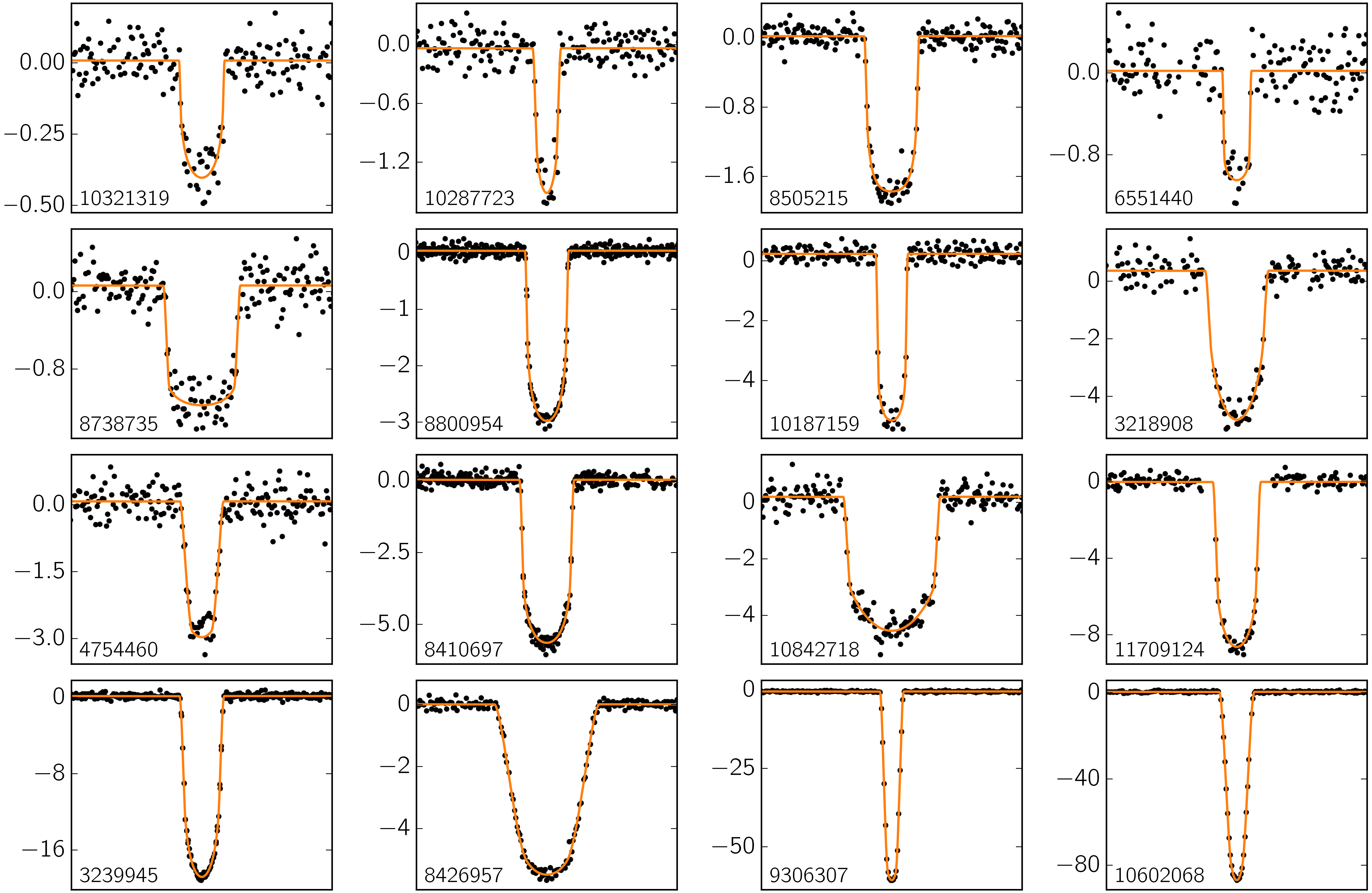


source: NASA Exoplanet Archive

these are (mostly) single transits

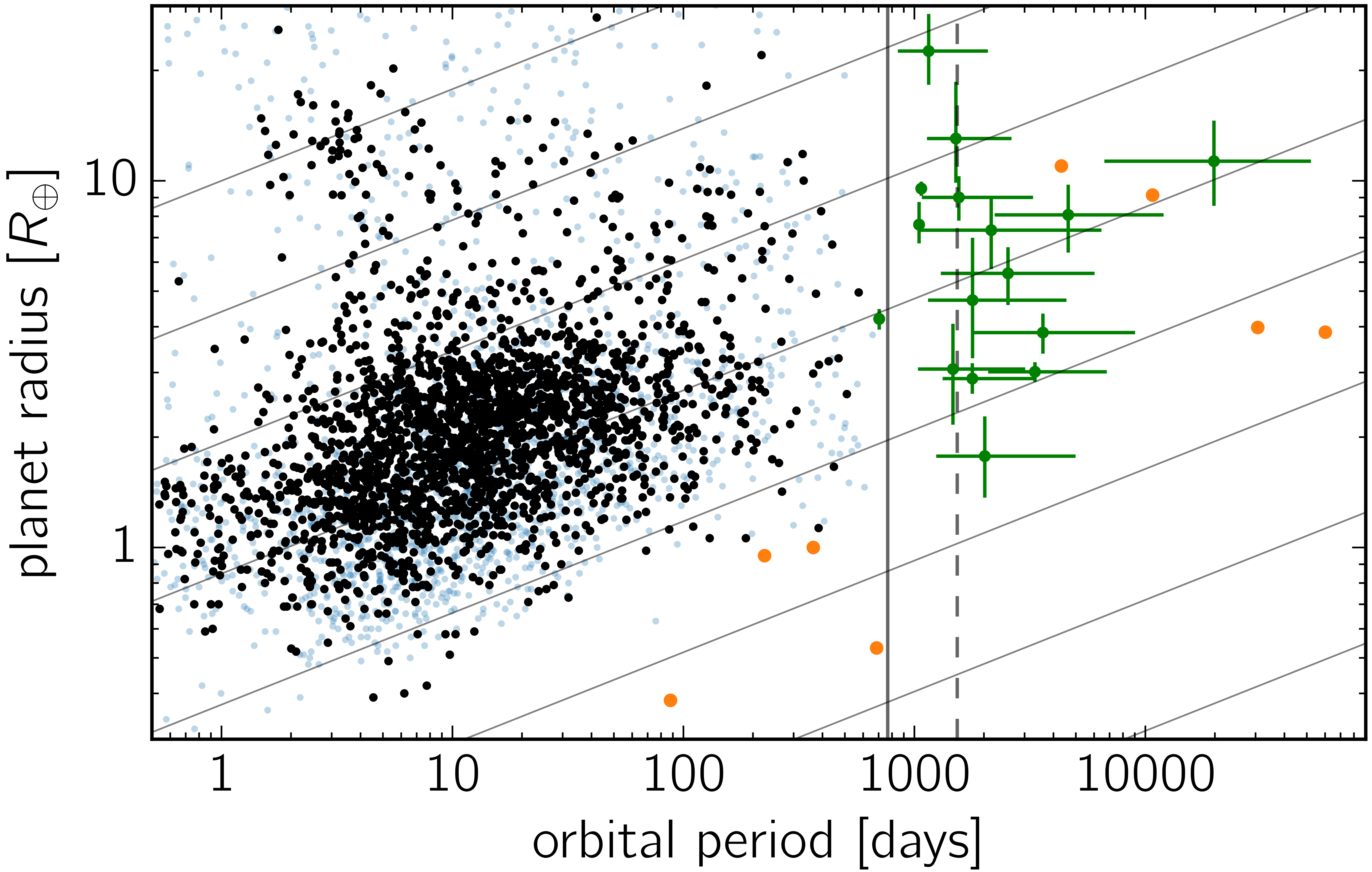

+
+
+
=
planet
star
spacecraft
detector
observation


vs.
vs.
vs.
etc.
+
star
+
spacecraft
+
detector
=
observation

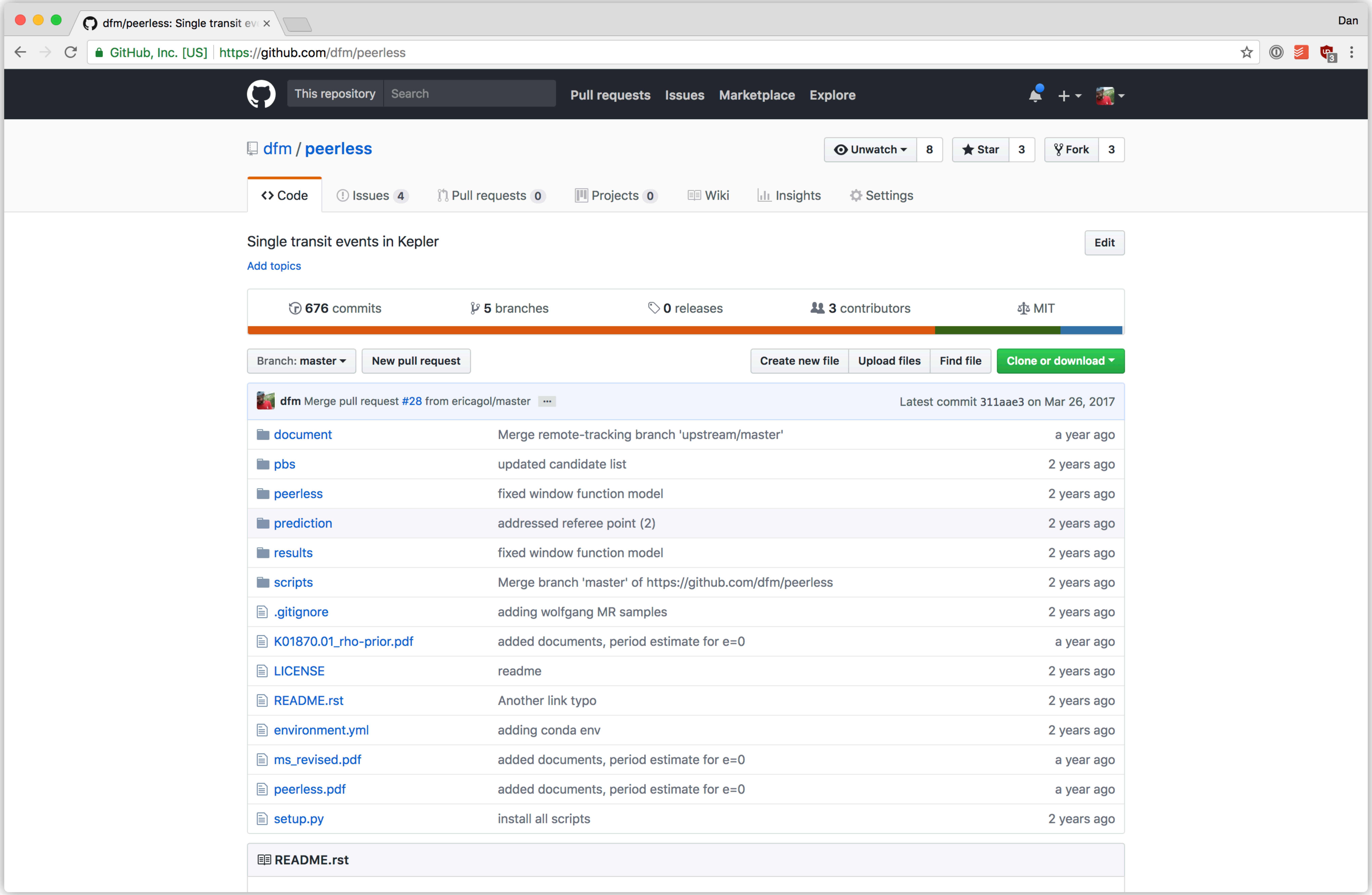


DFM+ (2016); arXiv:1607.08237

DFM+ (2016); arXiv:1607.08237



source: NASA Exoplanet Archive


dfm/peerless: Single transit ev
Dan
GitHub, Inc. [US]
https://github.com/dfm/peerless
This repository
Search
Pull requests
Issues
Marketplace
Explore
dfm / peerless
Unwatch 8
Star 3
Fork 3
Code
Issues 4
Pull requests 0
Projects 0
Wiki
Insights
Settings
Single transit events in Kepler
Edit
Add topics
676 commits
5 branches
0 releases
3 contributors
MIT
Branch: master
New pull request
Create new file
Upload files
Find file
Clone or download
dfm Merge pull request #28 from ericagol/master
Latest commit 311aae3 on Mar 26, 2017
document
Merge remote-tracking branch 'upstream/master'
a year ago
pbs
updated candidate list
2 years ago
peerless
fixed window function model
2 years ago
prediction
addressed referee point (2)
2 years ago
results
fixed window function model
2 years ago
scripts
Merge branch 'master' of https://github.com/dfm/peerless
2 years ago
.gitignore
adding wolfgang MR samples
2 years ago
K01870.01_rho-prior.pdf
added documents, period estimate for e=0
a year ago
LICENSE
readme
2 years ago
README.rst
Another link typo
2 years ago
environment.yml
adding conda env
2 years ago
ms_revised.pdf
added documents, period estimate for e=0
a year ago
peerless.pdf
added documents, period estimate for e=0
a year ago
setup.py
install all scripts
2 years ago
README.rst

BEB predictions and catalog s ×

Dan

GitHub, Inc. [US] | https://github.com/dfm/peerless/pull/3

This repository | Search | **Pull requests** | **Issues** | **Marketplace** | **Explore**

dfm / **peerless**

Unwatch 8 | Star 3 | Fork 3

Code | Issues 4 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

BEB predictions and catalog simulation scripts #3

Edit

Merged **timothydmorton** merged 13 commits into `dfm:master` from `timothydmorton:master` on Mar 3, 2016

Conversation 2 | Commits 13 | Files changed 6 | +1,640 −321

timothydmorton commented on Feb 14, 2016 — Collaborator

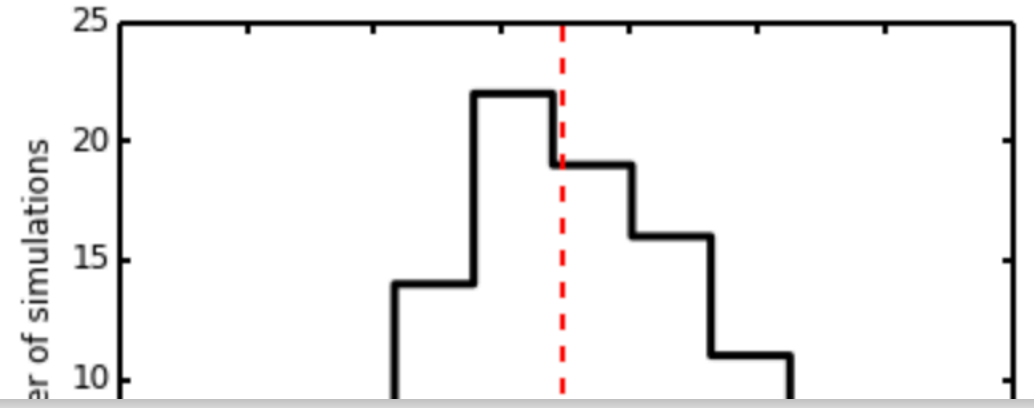
Finished the BEB predictions too. Seems to predict about ~1 in the right depth range. I'm now running 100x catalog simulations of both EB and BEB, to get a better sense of things.

timothydmorton added some commits on Feb 13, 2016

`BG_BinaryPopulation implemented`	647e7d1
`BG demo checked`	8a56d74
`added EB simulation script`	835b861
`BEB simulation script`	c4ecb87
`bug fixed`	2aed875
`added population analysis notebook`	2df6bc6

timothydmorton commented on Feb 15, 2016 — Collaborator

Money figures:



Reviewers
No reviews

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

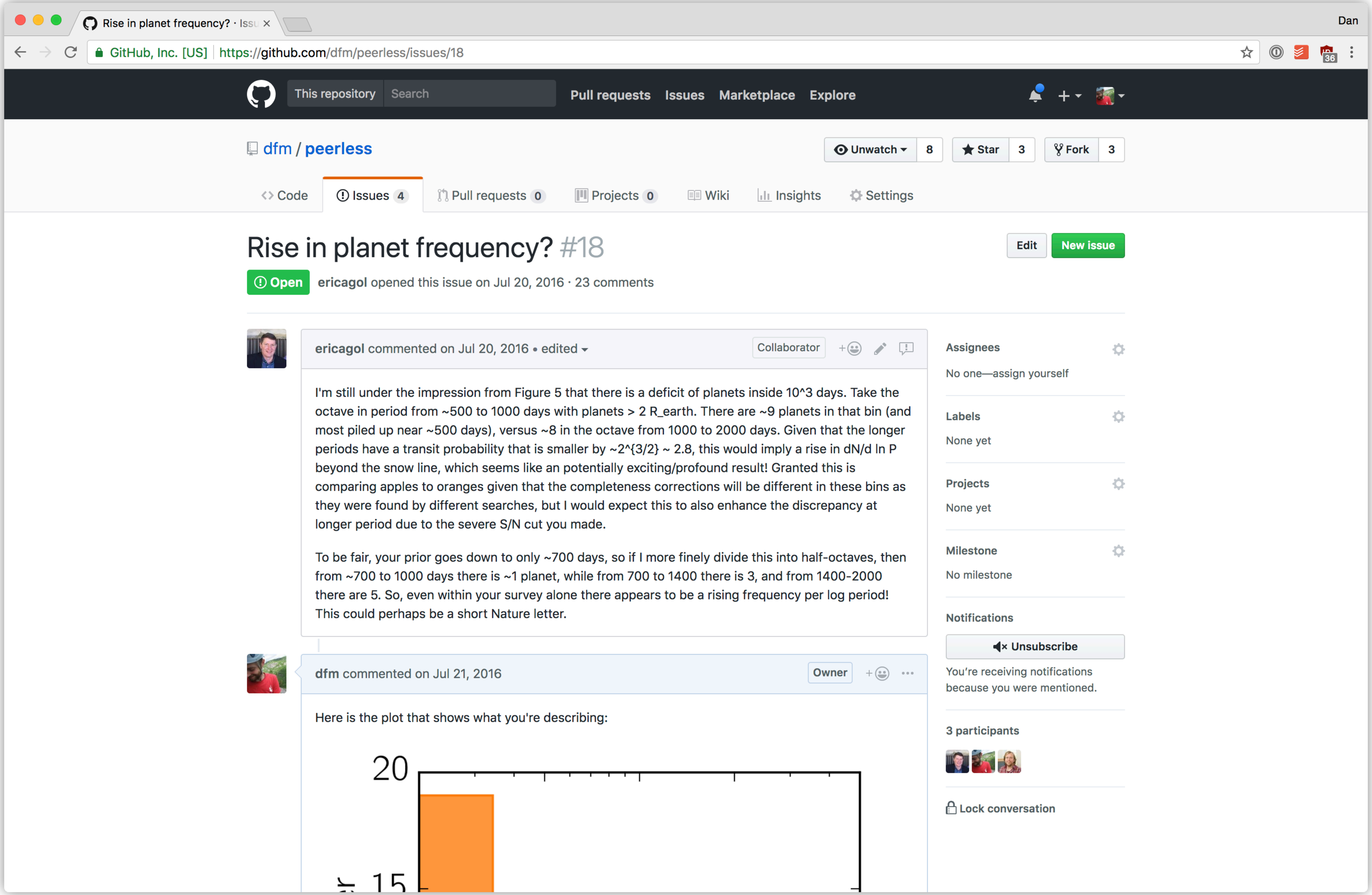
Milestone
No milestone

Notifications

Unsubscribe

You're receiving notifications because you commented.

2 participants



Rise in planet frequency? · Issu

GitHub, Inc. [US] https://github.com/dfm/peerless/issues/18

This repository Search Pull requests Issues Marketplace Explore

dfm / peerless

Unwatch 8 Star 3 Fork 3

Code Issues 4 Pull requests 0 Projects 0 Wiki Insights Settings

Rise in planet frequency? #18

Edit New issue

Open **ericagol** opened this issue on Jul 20, 2016 · 23 comments

ericagol commented on Jul 20, 2016 • edited — Collaborator

I'm still under the impression from Figure 5 that there is a deficit of planets inside 10^3 days. Take the octave in period from ~500 to 1000 days with planets > 2 R_earth. There are ~9 planets in that bin (and most piled up near ~500 days), versus ~8 in the octave from 1000 to 2000 days. Given that the longer periods have a transit probability that is smaller by ~2^{3/2} ~ 2.8, this would imply a rise in dN/d ln P beyond the snow line, which seems like an potentially exciting/profound result! Granted this is comparing apples to oranges given that the completeness corrections will be different in these bins as they were found by different searches, but I would expect this to also enhance the discrepancy at longer period due to the severe S/N cut you made.

To be fair, your prior goes down to only ~700 days, so if I more finely divide this into half-octaves, then from ~700 to 1000 days there is ~1 planet, while from 700 to 1400 there is 3, and from 1400-2000 there are 5. So, even within your survey alone there appears to be a rising frequency per log period! This could perhaps be a short Nature letter.

dfm commented on Jul 21, 2016 — Owner

Here is the plot that shows what you're describing:

20

15

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe
You're receiving notifications because you were mentioned.

3 participants

Lock conversation

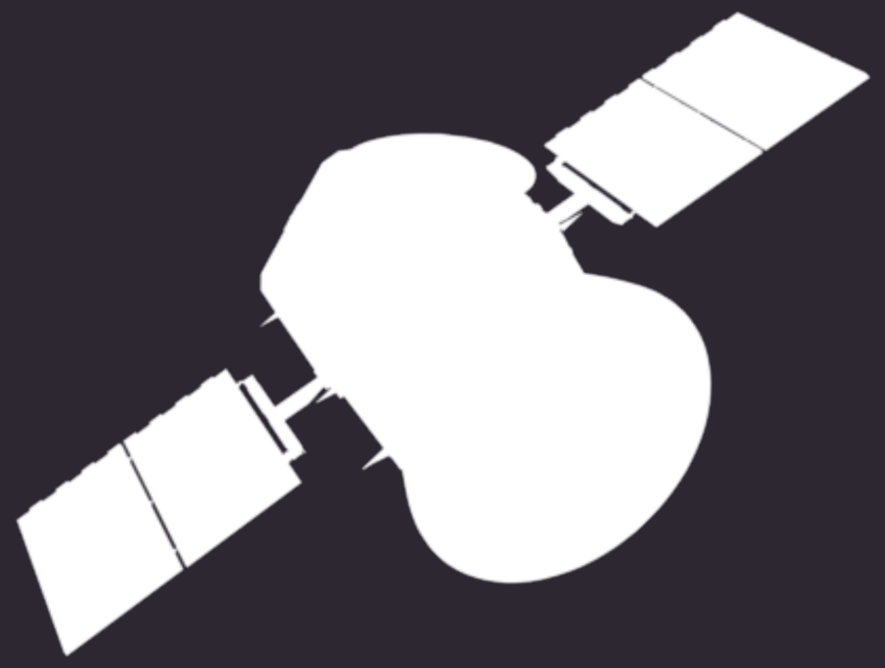
REVISITING THE LONG-PERIOD TRANSITING PLANETS FROM *KEPLER*

Miranda K. Herman[1], Wei Zhu (祝伟)[2], and Yanqin Wu (武延庆)[1]

[1]Astronomy & Astrophysics, University of Toronto, 50 St. George St., Toronto, ON M5S 3H4, Canada; herman@astro.utoronto.ca and
[2]Canadian Institute for Theoretical Astrophysics, University of Toronto, 60 St. George St., Toronto, ON M5S 3H8, Canada

Draft version March 7, 2019

reference: arXiv:1901.01974



tess

3.2M / 500k targets
30 min / 2 min cadence
30 day baseline
15,000 planet candidates

*note: all numbers are **predictions**

take homes & themes

Modern astrophysics **requires** the development (and implementation) of new algorithms.

take homes & themes

Modern astrophysics **requires** the development (and implementation) of new algorithms.

This really benefits from **interdisciplinary collaboration**, **good documentation**, and **open development**.

take homes & themes

Every project described here is **open source software** with an **associated journal article**.

take homes & themes

Every project described here is **open source software** with an **associated journal article**.

Is this a hack?

references

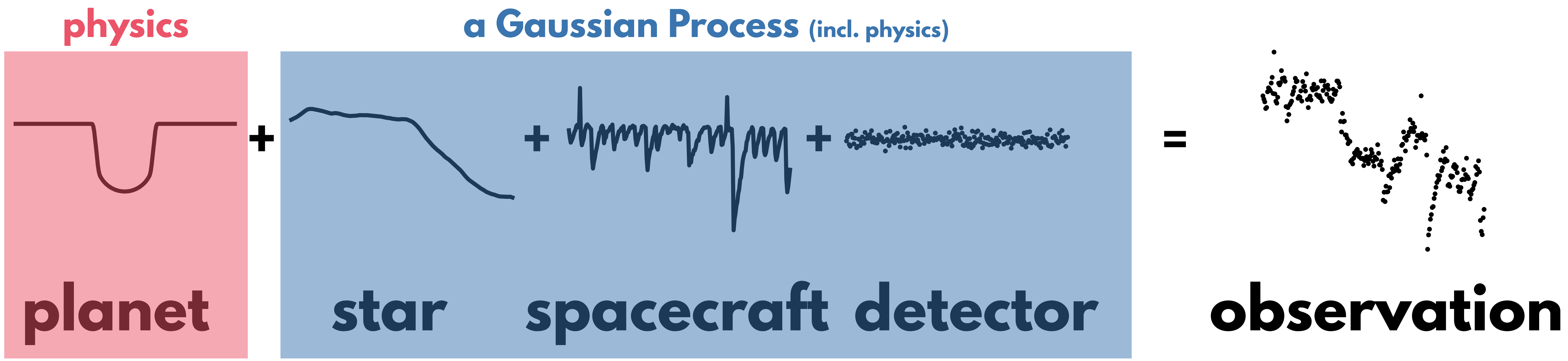
dfm/emcee	gradient-free MCMC in Python
dfm/exoplanet	gradient-based inference for time series
dfm/george	simple Gaussian processes in Python
dfm/celerite	fast & scalable Gaussian processes
dfm/peerless	long period transiting exoplanets

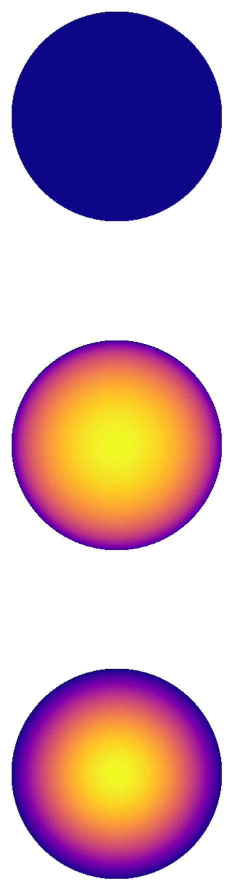
dan foreman-mackey
cca@flatiron dfm.io github.com/dfm @exoplaneteer

fast/accurate light curve models

work led by **Rodrigo Luger** (Flatiron)

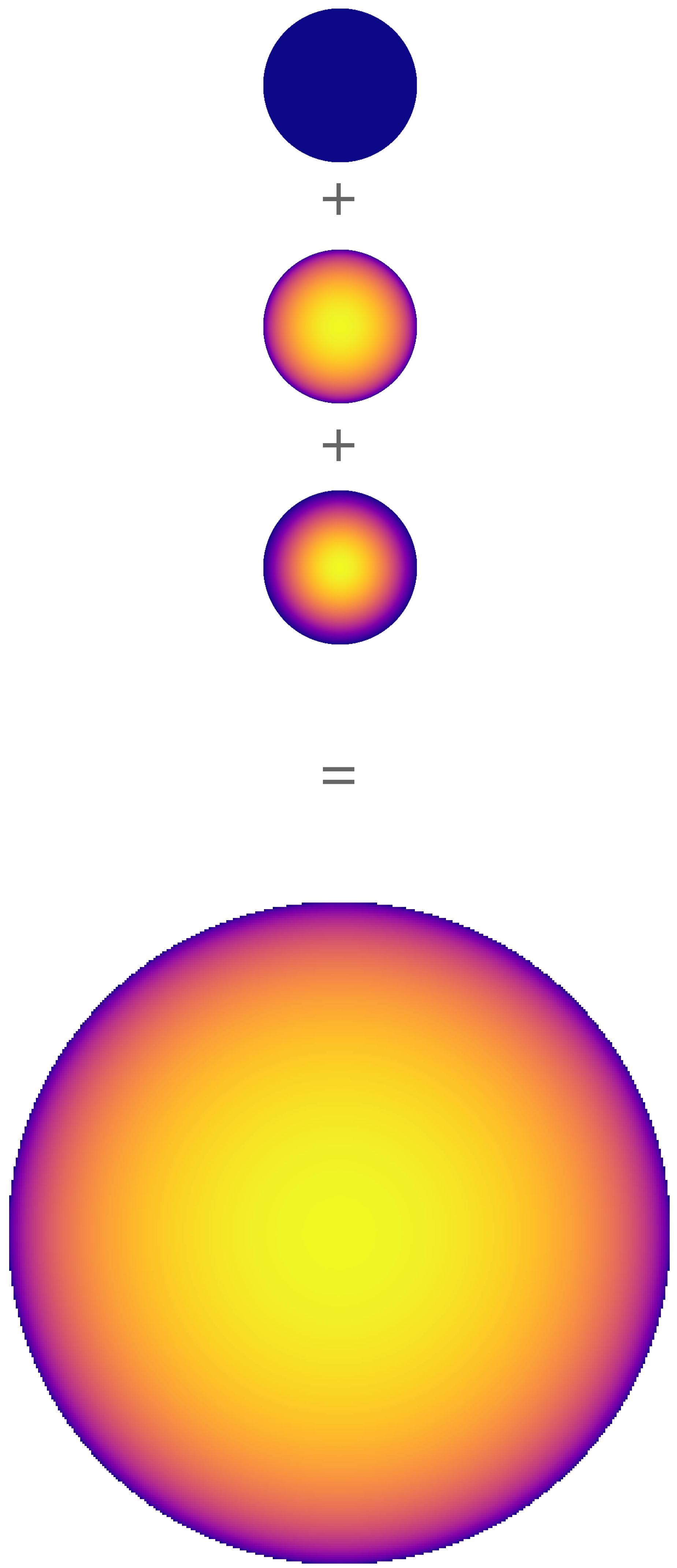
STARRY


physics
a Gaussian Process (incl. physics)
+
+
+
=
planet
star
spacecraft
detector
observation



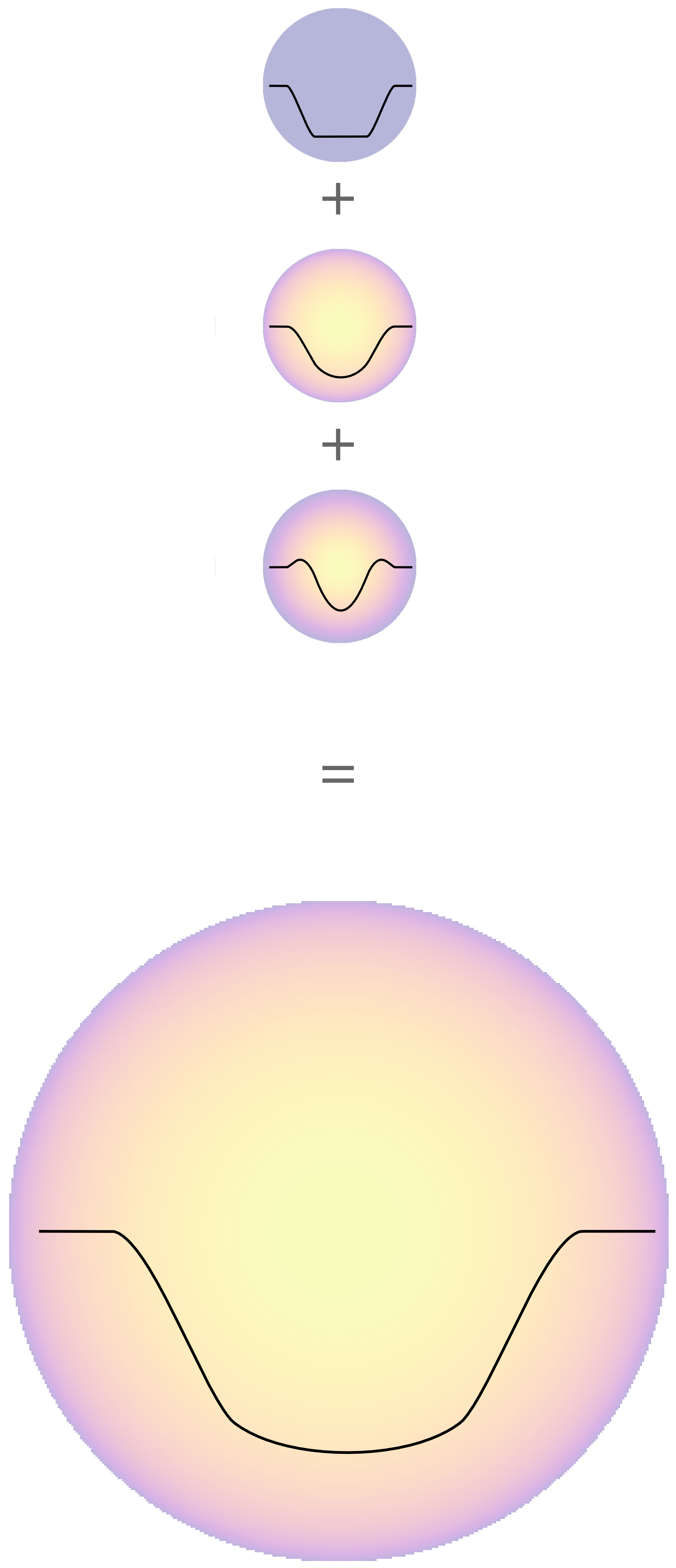
credit: Rodrigo Luger

STARRY



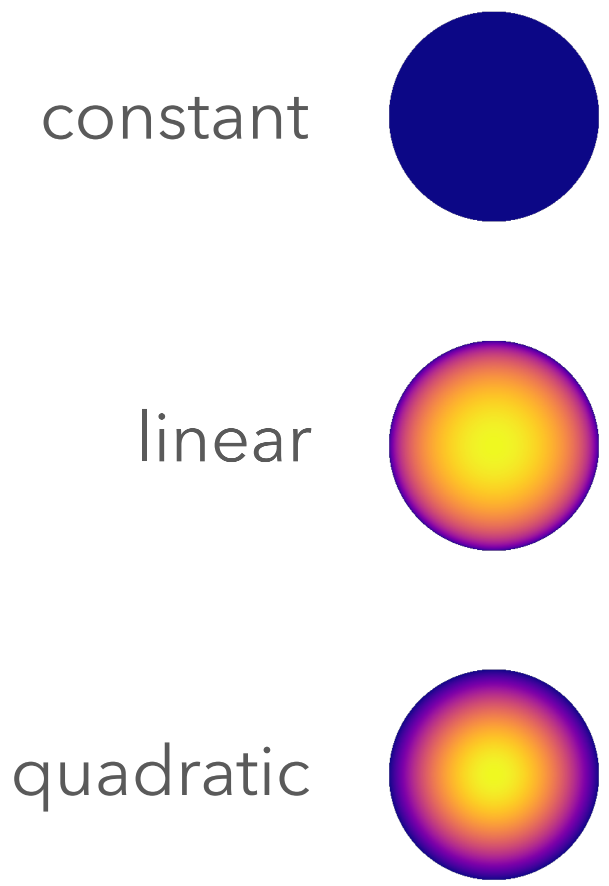
credit: *Rodrigo Luger*

STARRY

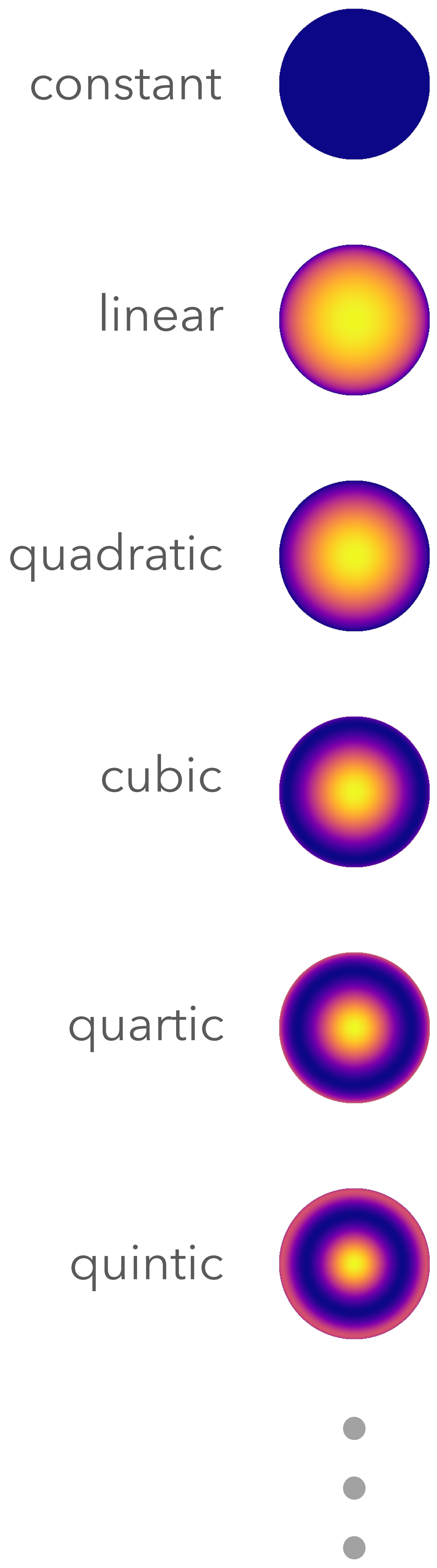


credit: Rodrigo Luger

STARRY

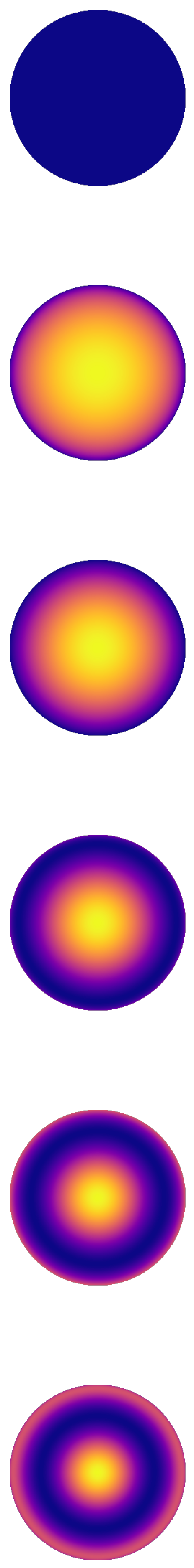


credit: *Rodrigo Luger*



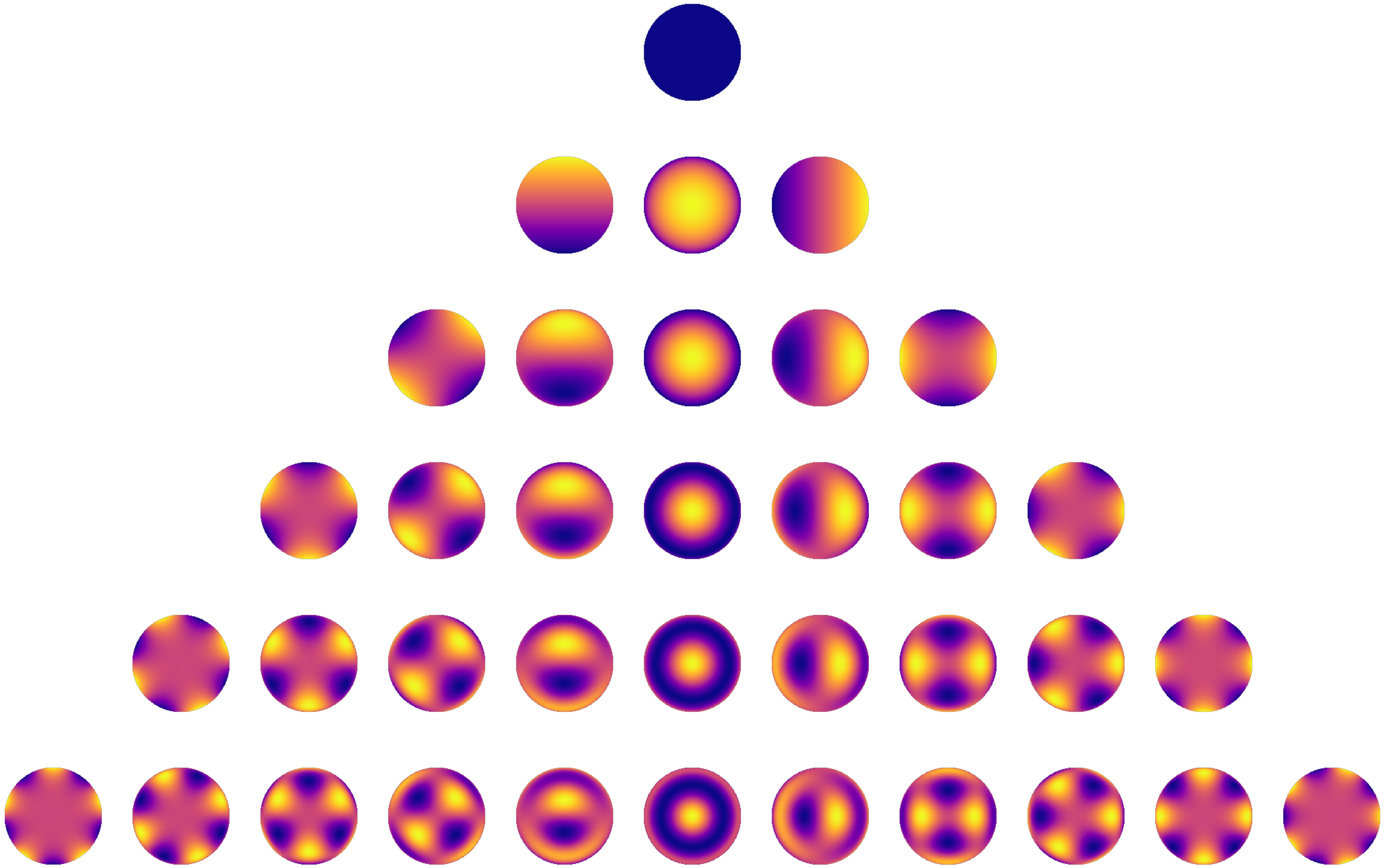
credit: Rodrigo Luger

STARRY



credit: Rodrigo Luger

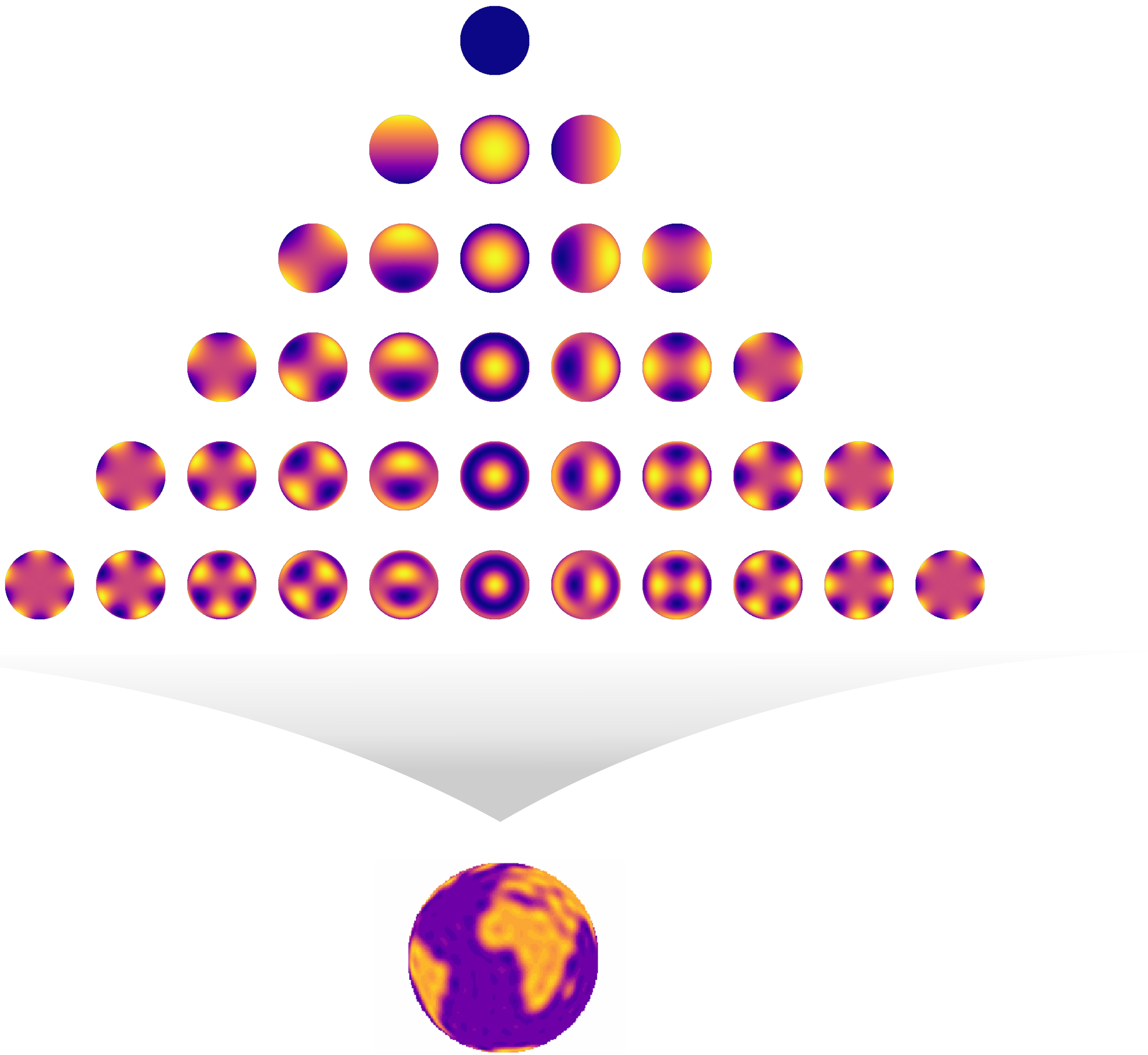
STARRY



The real spherical harmonics

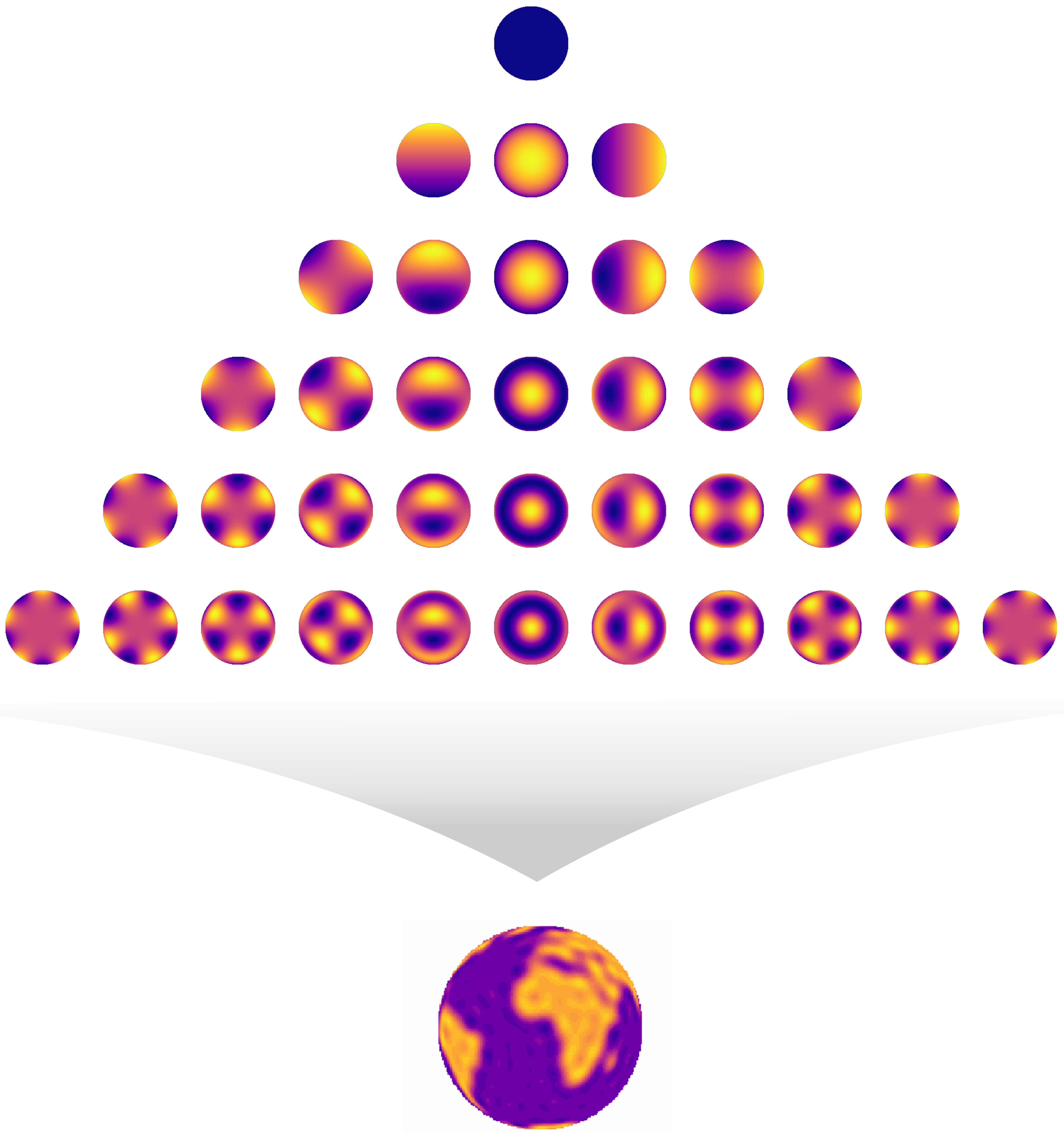
credit: Rodrigo Luger

STARRY



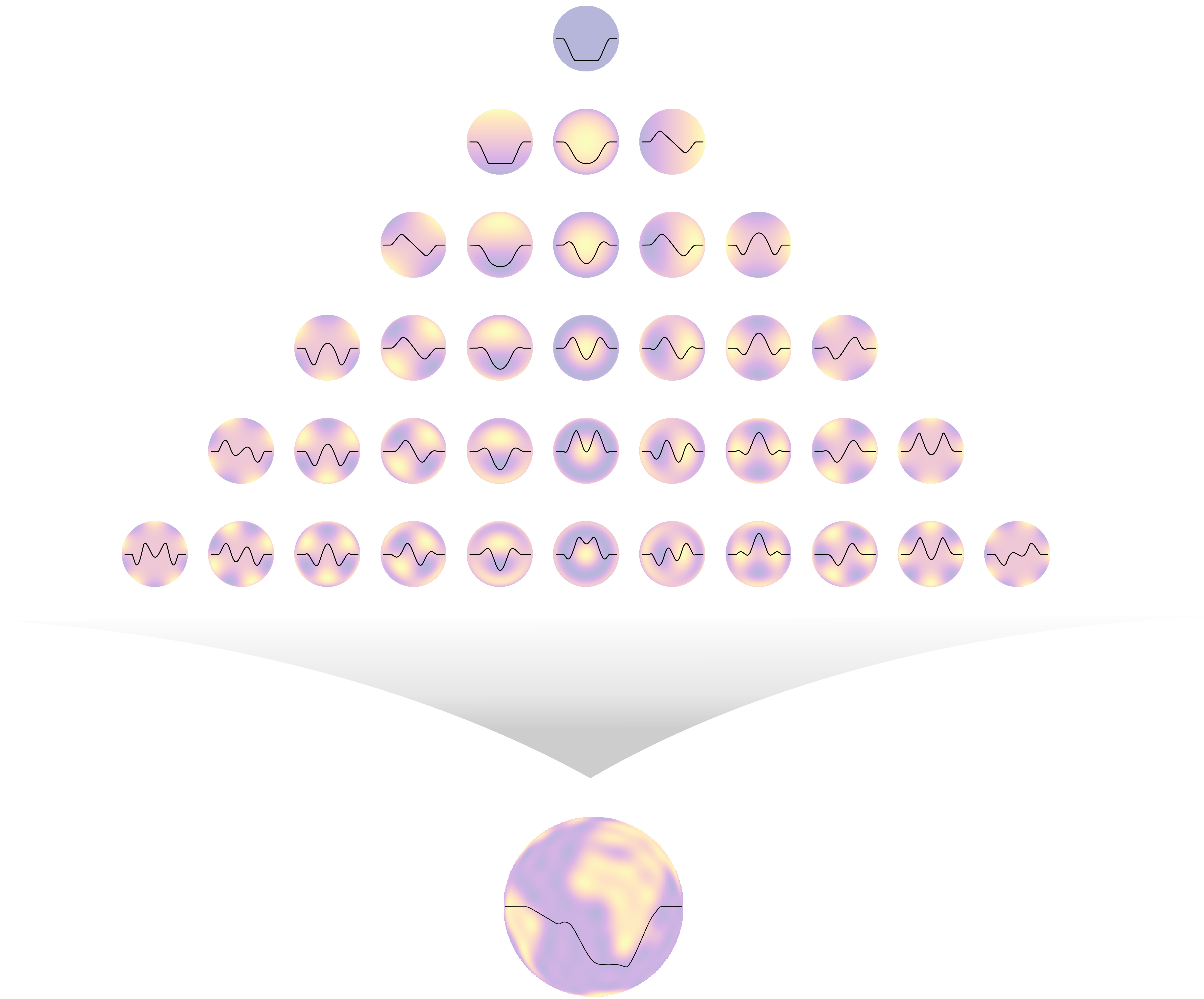
credit: Rodrigo Luger

STARRY



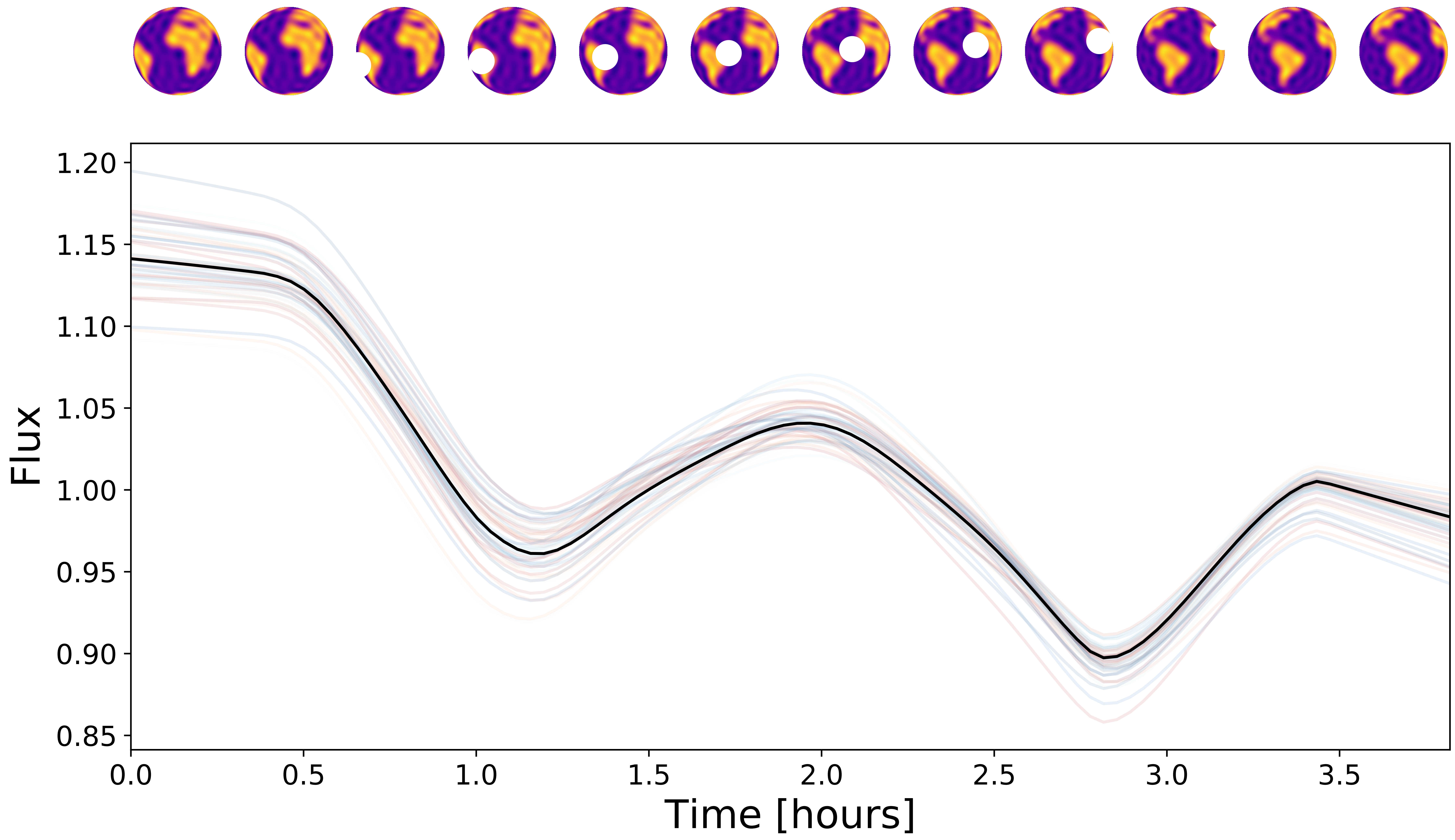
credit: Rodrigo Luger

STARRY



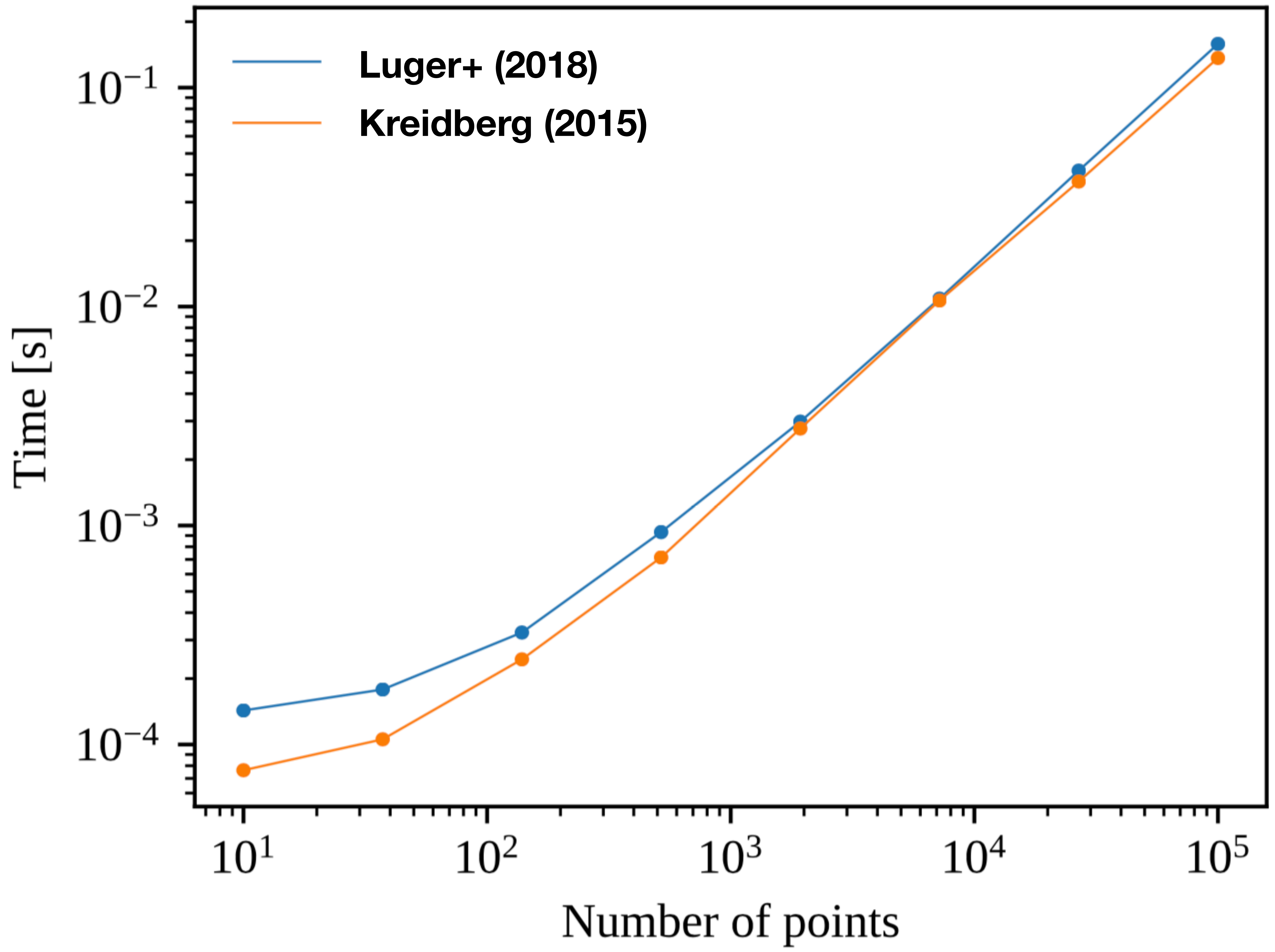
credit: Rodrigo Luger

STARRY



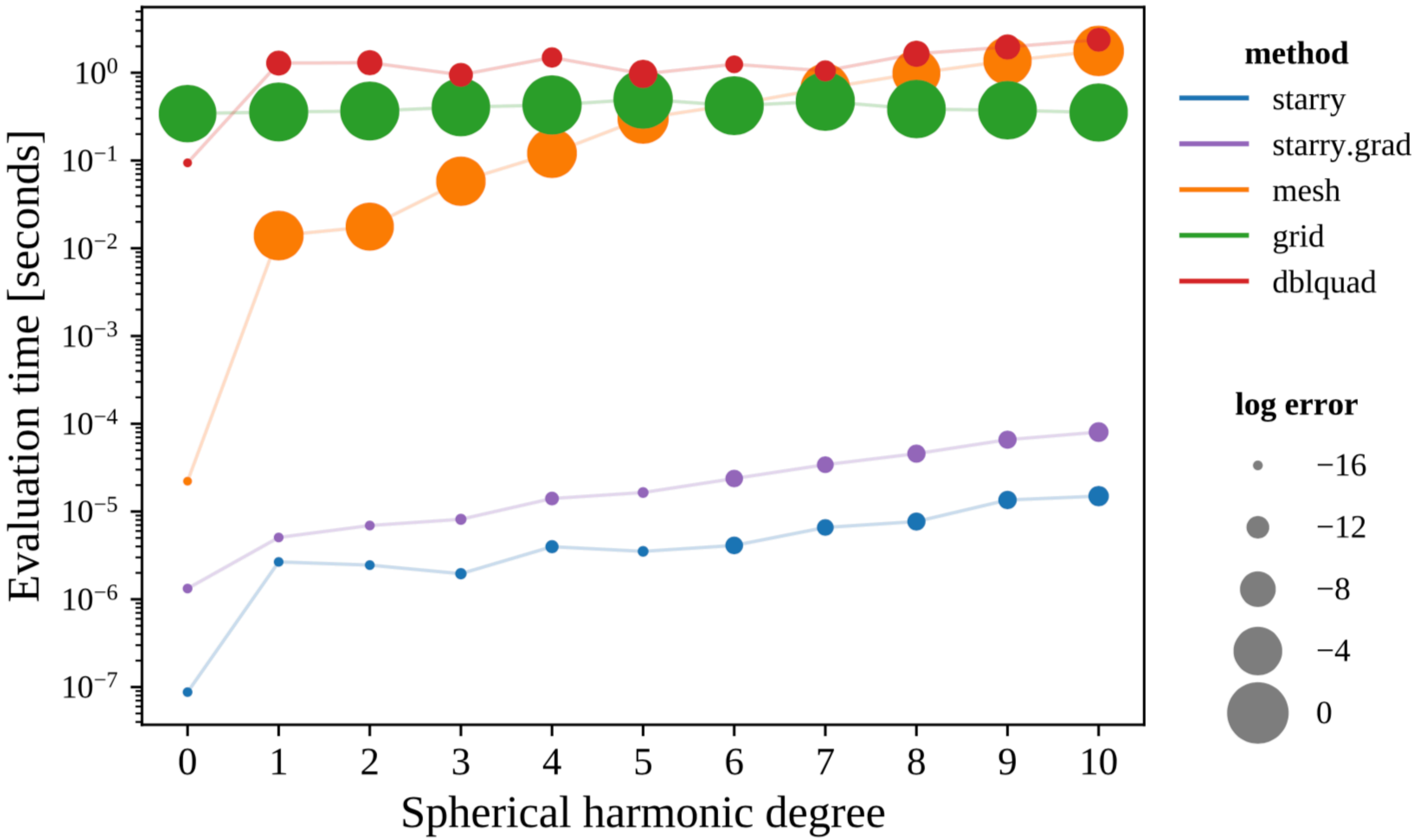
credit: Rodrigo Luger

STARRY



credit: Rodrigo Luger

STARRY



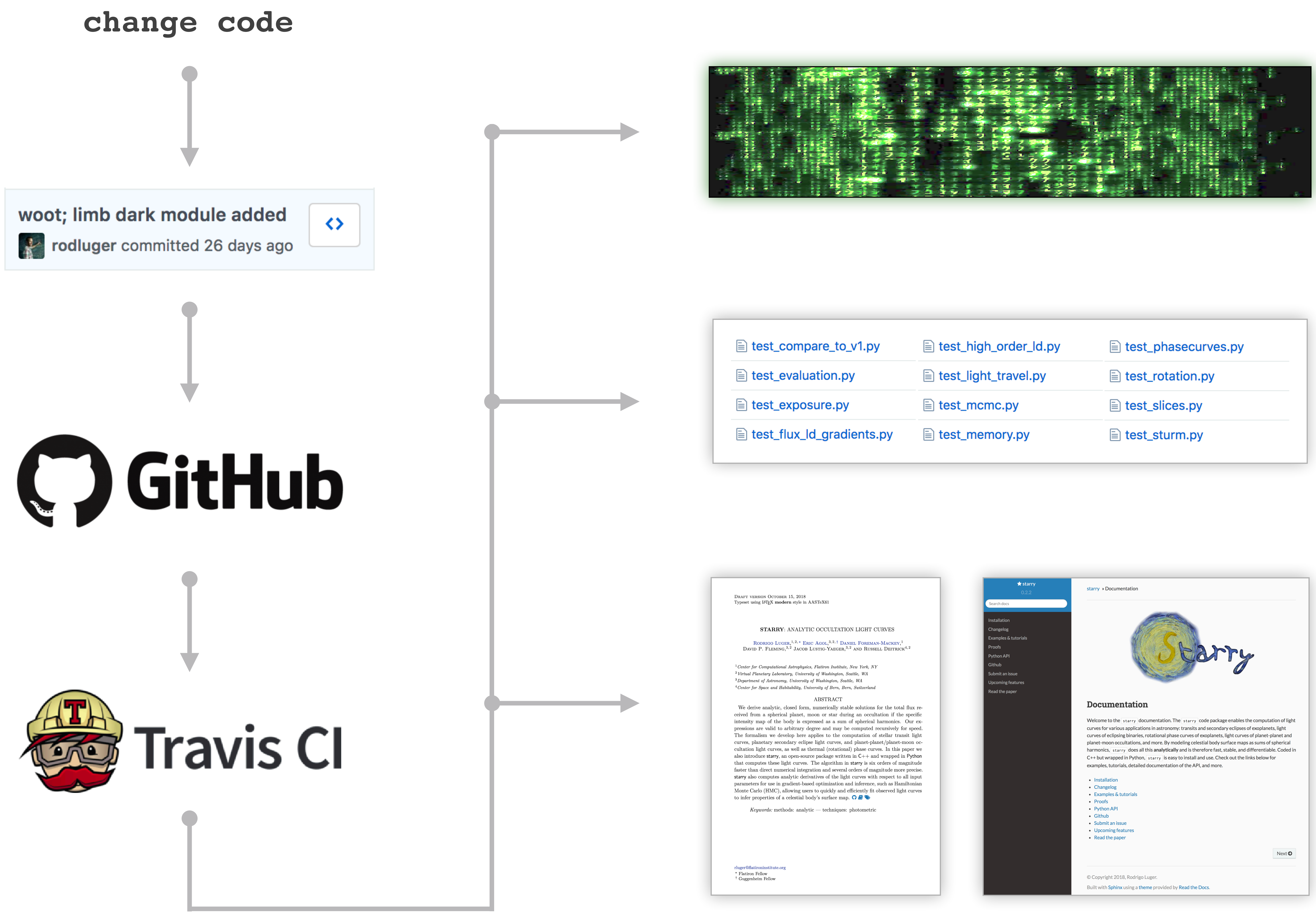
credit: Rodrigo Luger

STARRY

github.com/rodluger/starry

rodluger.github.io/starry

arxiv.org/abs/1810.06559



credit: Rodrigo Luger

STARRY

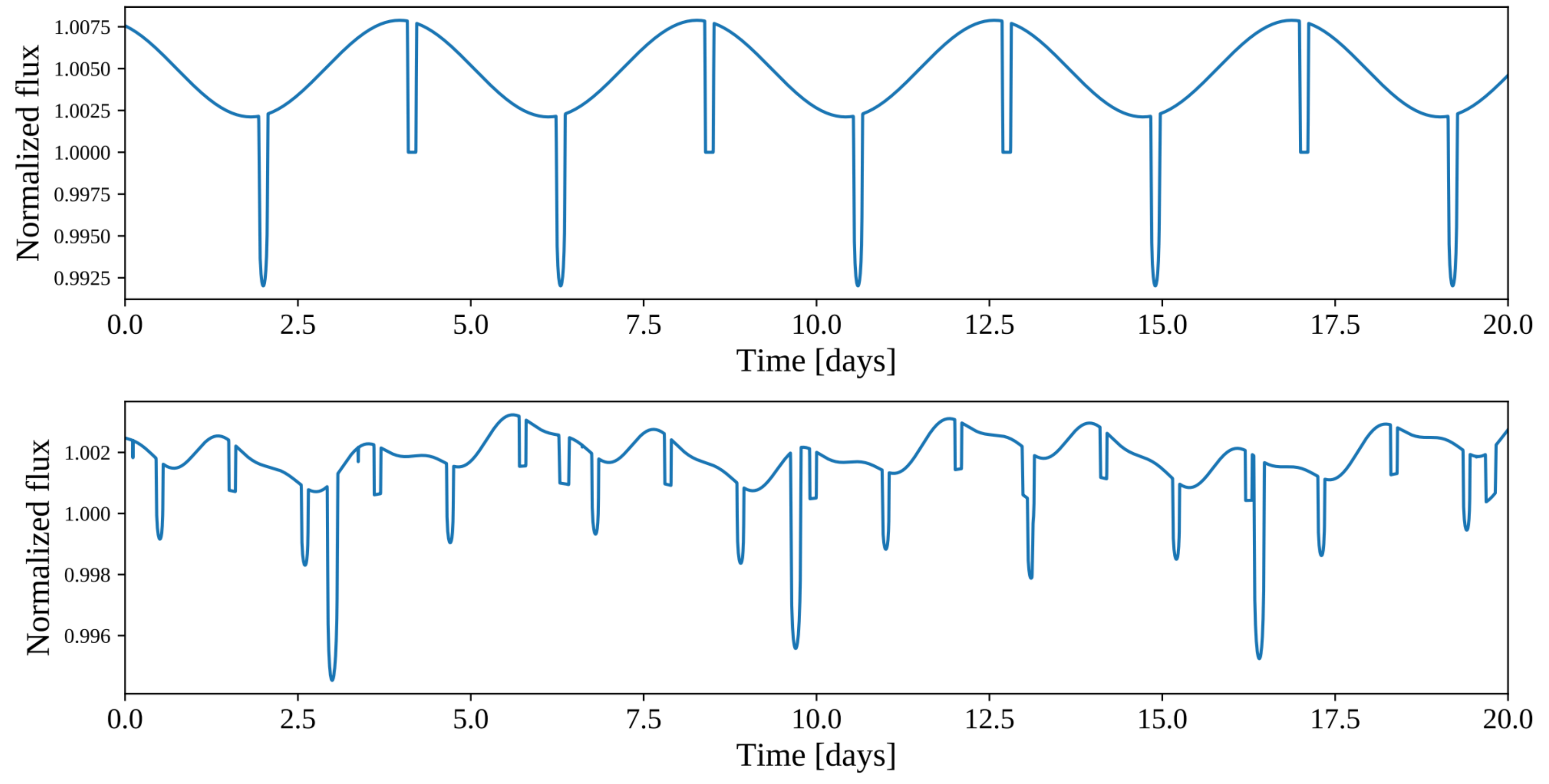
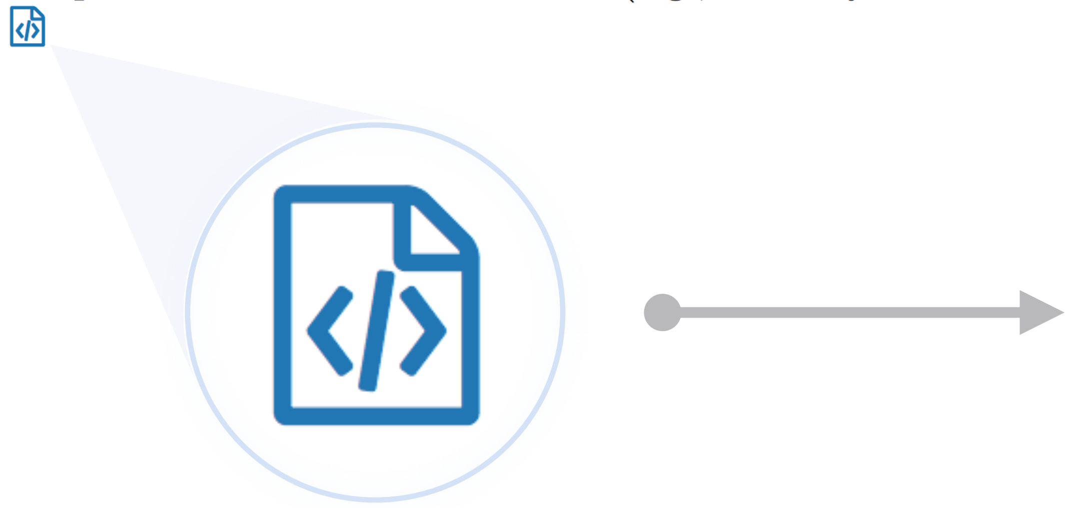
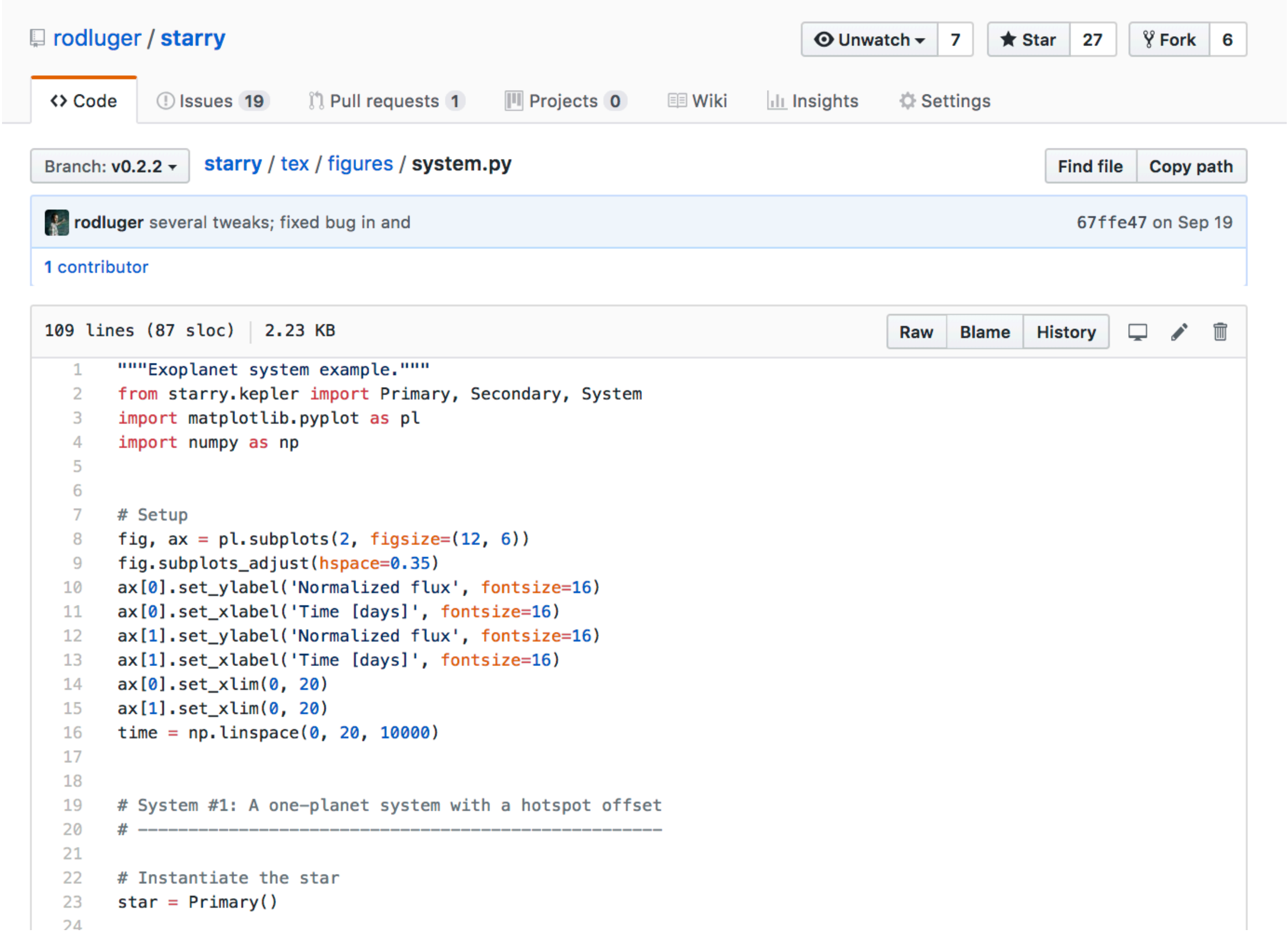


Figure 9. Sample analytic exoplanet system light curves computed with starry. *Top:* a hot Jupiter transiting a Sun-like star. The planet's map is a simple dipole, with the hotspot offset 15° from stellar noon; the offset in the secondary eclipse from the peak of the phase curve is apparent. *Bottom:* a two-planet system with more complex surface maps. In addition to transits and secondary eclipses, a few planet-planet occultations are visible (e.g., the very short events at $t = 0.1$ and $t = 3.4$ days).





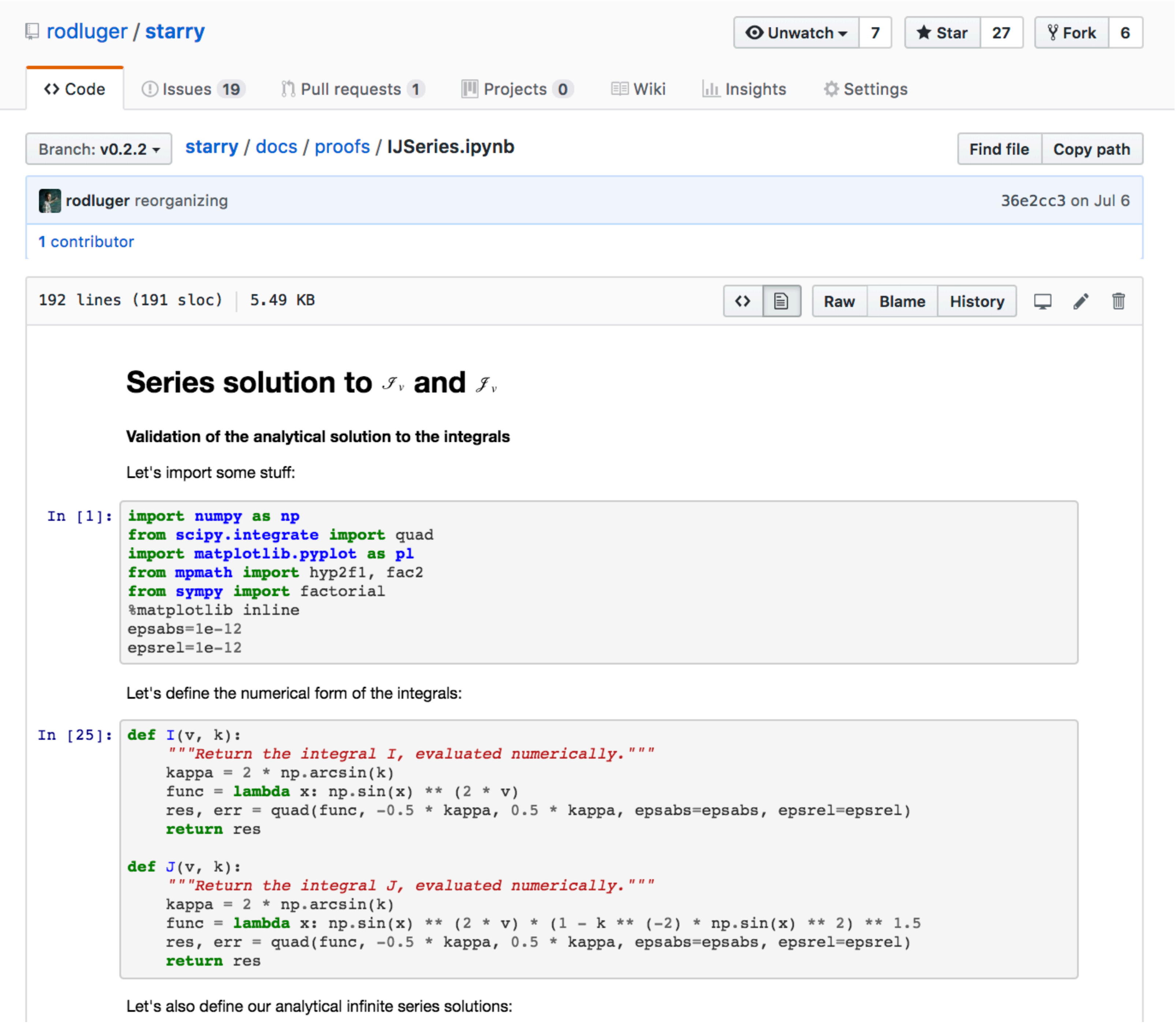
credit: Rodrigo Luger

STARRY

where ${}_2F_1(a,b;c;x)$ is the generalized Hypergeometric function. These functions can alternatively be expressed as series in k^2 by expanding $(1-k^2w)^{-1/2}$ as a series in k^2w, and then integrating each term over w, giving

$$\mathcal{I}_v = 2k^{1+2v}\sum_{j=0}^{\infty}\frac{(2j-1)!!}{2^j j!(2j+2v+1)}(k^2)^j ,$$

$$\mathcal{J}_v = \frac{3\pi}{4}k^{1+2v}\sum_{j=0}^{\infty}\frac{(2j-1)!!(2j+2v-1)!!}{2^{2j+v}j!(j+v+2)!}(k^2)^j . \quad \text{(D42)}$$



credit: Rodrigo Luger

STARRY