Advanced techniques for radial velocity analysis with DACE notebooks

With theoretical complements and other resources (including open-source codes)

Nathan Hara

Université de Genève

Sagan Exoplanet Summer Virtual Workshop
20 - 24 July 2020

Content of the presentation

DACE notebooks: key steps

Main advantage over the graphical interface: the correlated noise handling.

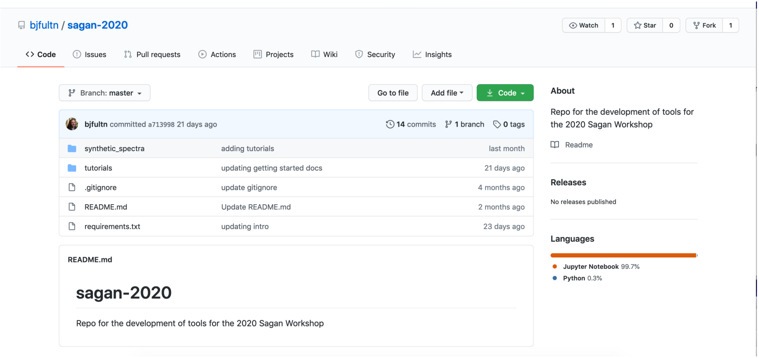
1. Base models: including offsets, trends, stellar activity
2. Gaussian noise representation: parameter choices and computational aspects
3. Periodograms: using correlated noise models

Other resources

Residuals analysis, ℓ_1-periodogram, PlanetPack, stacked periodograms, AGATHA, Kima, ExoFAST

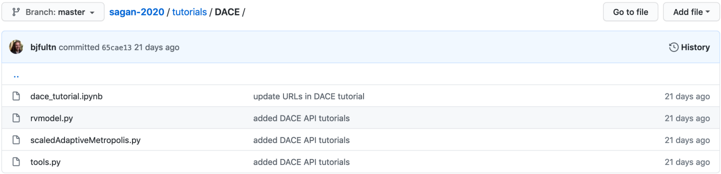
Downloading the notebook I

Go to `https://github.com/bjfultn/sagan-2020`
Green code icon to download, or for the repository.



Downloading the notebook II

The DACE tutorial is in tutorials/DACE



Open dace_tutorial.ipynb with Jupyter

DACE notebooks central class: rvModel

Central class: contains the input data, the base model, noise models, Keplerian planets etc.

```
rv_model = rvModel(
    rv_data['rjd']-epoch_rjd,
    rv_data['rv'],
    rv_data['rv_err']**2,
    inst_id=rv_data['inst_id'],
    var_jitter_inst=var_jitter,
  var_cos_qper=np.array([]), # No quasi periodic component
  var_sin_qper=np.array([]),
  lambda_qper=np.array([]),
  nu_qper=np.array([]),
  var_exp = np.array([1.0]),
  lambda_exp = np.array([1/1.0]),  # Noise time scale is 1 day
)
```

```
rv_model.addlin
```

Radial velocity models

Noise models

Radial velocity data $\boldsymbol{y}$ modelled with:

$$\underbrace{\boldsymbol{y}}_{\textit{data}} = \underbrace{\boldsymbol{f}}_{\textit{model}}(\underbrace{\boldsymbol{\theta}}_{\substack{\textit{model}\\ \textit{parameters}}}) + \underbrace{\boldsymbol{\epsilon}}_{\textit{noise}} \tag{1}$$

$$\boldsymbol{\epsilon} \sim \underbrace{d}_{\substack{\textit{noise}\\ \textit{distribution}}}(\underbrace{\boldsymbol{\eta}}_{\substack{\textit{noise}\\ \textit{parameters}}}) \tag{2}$$

In almost all analyses, d is assumed to be a Gaussian distribution of mean 0 and covariance $V(\boldsymbol{\eta})$

Priors and likelihood

We define

- The likelihood: $p(\boldsymbol{y}|\boldsymbol{\theta})$. It reads "probability density of $\boldsymbol{y}$ knowing $\boldsymbol{\theta}$"
- The prior distribution: $p(\boldsymbol{\theta})$
- The posterior distribution: $p(\boldsymbol{\theta}|\boldsymbol{y})$

How to update your information on $\boldsymbol{\theta}$ with prior information $p(\boldsymbol{\theta})$ once data $\boldsymbol{y}$ is available, assuming a form for $p(\boldsymbol{y}|\boldsymbol{\theta})$. The Bayes formula:

$$\underbrace{p(\boldsymbol{\theta}|\boldsymbol{y})}_{\textit{posterior}} = \frac{\overbrace{p(\boldsymbol{y}|\boldsymbol{\theta})}^{\textit{likelihood}}\,\overbrace{p(\boldsymbol{\theta})}^{\textit{prior}}}{\underbrace{p(\boldsymbol{y})}_{\textit{normalization}}} \tag{3}$$

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})\,\mathrm{d}\,\boldsymbol{\theta} \tag{4}$$

In the case of RV analysis

Likelihood

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{\epsilon} \tag{5}$$

$$\boldsymbol{\epsilon} \sim G(0, V(\boldsymbol{\eta})) \tag{6}$$

$$p(\boldsymbol{y}|\boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^N |V|}} \, \mathrm{e}^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{f}(\boldsymbol{\theta}))^T V^{-1}(\boldsymbol{y}-\boldsymbol{f}(\boldsymbol{\theta}))} \tag{7}$$

Exponent T denotes matrix transposition.

Priors

Usually it is assumed that $p(K, e, P, M_0, \omega) = p(K)p(e)p(P)p(M_0)p(\omega)$

- Prior on K and P : log-uniform
- Prior on e: Beta distribution (Kipping, 2014) or uniform
- Prior on M_0 and ω : uniform on $[0, 2\pi]$

Noise models

Noise models

What about $V(\boldsymbol{\eta})$ in

$$p(\boldsymbol{y}|\boldsymbol{\theta},\boldsymbol{\eta}) = \frac{1}{\sqrt{(2\pi)^N |V(\boldsymbol{\eta})|}} \mathrm{e}^{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{f}(\boldsymbol{\theta}))^T V^{-1}(\boldsymbol{\eta})(\boldsymbol{y}-\boldsymbol{f}(\boldsymbol{\theta}))}? \tag{8}$$

The noise is assumed to be **stationary**: the correlation matrix $V(\boldsymbol{\eta})$ is defined by

$$V(\boldsymbol{\eta})_{ij} = c(|t_i - t_j|, \boldsymbol{\eta}) \tag{9}$$

where i and j represent the row and column indices.

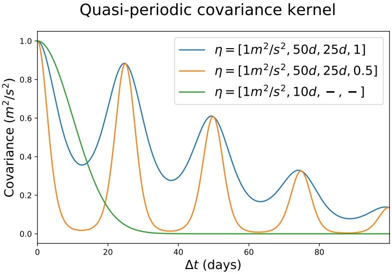
Computing a matrix inverse and determinants is computationally costly .
In the general case the cost is in $O(N^3)$, where N is the number of observations.
For estimating evidences and uncertainties and computing error bars on the orbital parameters, you need to compute the covariance matrix inverse and determinant many times.

Covariance kernel

One of the most classic form is

$$c(\Delta t, \sigma_J, \boldsymbol{\eta}) = \delta_{ij}(\sigma_i^2 + \sigma_J^2) + \eta_1^2 \exp\left(-\frac{\Delta t^2}{2\eta_2^2} - \frac{2\sin^2\left(\frac{\pi \Delta t}{\eta_3}\right)}{\eta_4^2}\right).$$

as defined in Haywood et al. (2014). σ_i is the nominal uncertainty on measurement i, σ_J is a free additional error term, $\delta_{ij} = 1$ if $i = j$, 0 otherwise.



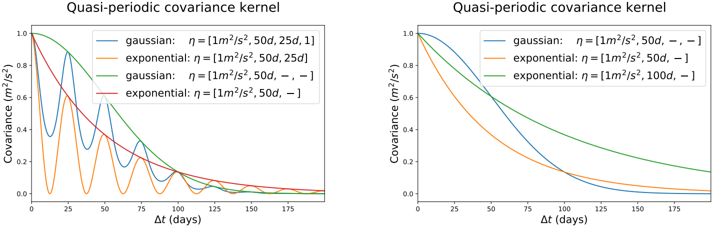
The CELERITE model

An alternative form is given in Foreman-Mackey et al. (2017), called the CELERITE model.

$$V_{i,j} = \sigma_i^2 \delta_{i,j} + \sum_{s<n_c} \left(a_s \cos(\nu_s \Delta t_{i,j}) + b_s \sin(\nu_s \Delta t_{i,j})\right) e^{-\lambda_s \Delta t_{i,j}} \,. \quad (10)$$

Simple quasi-periodic kernel

$$V_{i,j} = \frac{\sigma_{QP}^2}{2} \left(1 + \cos \frac{2\pi}{P_\star} \Delta t\right) \quad (11)$$



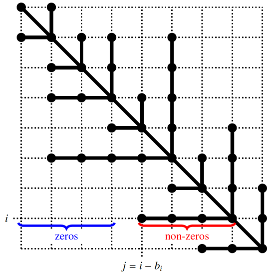
Covariance matrices of that form have a mathematical, they are "semi-separable" $\rightarrow$ **inversion is much faster: in $O(N)$**

Code available at `https://github.com/dfm/celerite/issues/48`

Noise models III: The S-LEAF model

Delisle et al. (2020) generalises the CELERITE model to covariance matrices that can be written as a sum of a semi-separable matrix and a so-called leaf matrix.

A leaf matrix is zero everywhere except close to the diagonal, like this:



inversion is still faster: in $O(bN)$, where b is the average number of non-zero diagonal terms. Allows to model for instance **calibration noise**. Code available at `https://gitlab.unige.ch/Jean-Baptiste.Delisle/spleaf`

Noise models IV: In practice

Covariance matrix defined by:

$$V_{i,j} = (\sigma_i^2 + \sigma_J(i)^2)\delta_{i,j} + \sigma_C^2 c_{i,j} + \sum_{k=1}^{n_\mathrm{e}} \sigma_{e,k}^2 \mathrm{e}^{-\lambda_k |t_i - t_j|} \tag{12}$$

$$+ \sum_{s=1}^{n_\mathrm{qp}} (a_s \cos(\nu_s |t_i - t_j|) + b_s \sin(\nu_s |t_i - t_j|))\, \mathrm{e}^{-\lambda_s |t_i - t_j|}. \tag{13}$$

```
rv_model = rvModel(
    rv_data['rjd']-epoch_rjd,
    rv_data['rv'],
    rv_data['rv_err']**2,
    inst_id=rv_data['inst_id'],
    calib_file=rv_data['cal_thfile'],
    var_calib_meas = rv_data['cal_therror']**2,
    var_jitter_inst=var_jitter,
  var_cos_qper=np.array([]), # No quasi periodic component
  var_sin_qper=np.array([]),
  lambda_qper=np.array([]),
  nu_qper=np.array([]),
  var_exp = np.array([1.0]),
  lambda_exp = np.array([1/1.0]),  # Noise time scale is 1 day
)
```

Warning: `nu_qper` is in radian per time unit: $= 2\pi/$[units of the time stamps = first argument of `rv_model`]

Linear base models

Base model I

$$\underbrace{\boldsymbol{y}}_{data} = \underbrace{\boldsymbol{f}}_{model}(\underbrace{\boldsymbol{\theta}}_{\substack{model\\parameters}}) + \underbrace{\boldsymbol{\epsilon}}_{noise} \tag{14}$$

$$\boldsymbol{\epsilon} \sim \underbrace{d}_{\substack{noise\\distribution}}(\underbrace{\boldsymbol{\eta}}_{\substack{noise\\parameters}}) \tag{15}$$

Model

In the case of RV (with negligible planet-planet interactions), for k planets

$$f(\boldsymbol{\theta}) = \sum_{i=1}^{k} \mathrm{Kep}(t, e_i, K_i, P_i, \omega_i, M_0^i) + g(\boldsymbol{\beta}) \tag{16}$$

where "Kep" is a Keplerian function (See Jason Wright's talk). We here focus on how to define $g(\boldsymbol{\beta})$ in the case where it is linear. It can be written of the form, where ϕ_H is a matrix: $g(\boldsymbol{\beta}) = \phi_H \boldsymbol{\beta}$

Syntax: `rv_model.addlin(matrix_column, name of thee variable )`

Base model II: offsets and trend

Offsets

You want to adjust the value of the "zero velocity point" of different instruments. Suppose you have data from three instruments $j = 1, 2, 3$, so that in total you have N observations at times t_i, $i = 1..N$. Then define ϕ_H as a $N \times 3$ matrix where $M_{ij} = 1$ if measurement i is taken by instrument j and 0 otherwise

```
for kinst in range(ninst):
    rv_model.addlin(1.0*(rv_data['inst_id']==kinst), 'offset_inst_{}'.format(kinst))
```

Trends

Once you have defined offsets, if you want to add a polynomial trend of degree d to your data, add columns to ϕ_H of the form $\phi_{H_{i,\cdot}} t_i^j$ for $j = 1..d$.

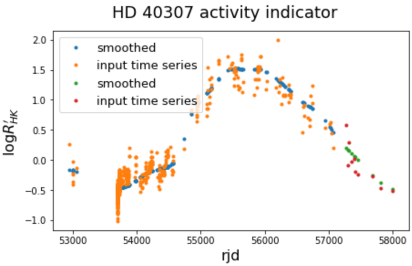
```
for kpow in range(dpow):
    rv_model.addlin(rv_model.t**(kpow+1), 'drift_pow{}'.format(kpow+1))
```

Base models III: Activity models

Suppose you have a variability in the S-index or $\log R'_{HK}$ (Noyes, 1984), that you want to use to model the activity.

You can smooth it: simple smoothing with a moving kernel (as in DACE) of a Gaussian or smoothing with a Gaussian process (and other operations such as differentiating Haywood et al. (2014) for instance). Very similar in practice (see a discussion in Rasmussen and Williams (2005) chapter 6).

You can use the raw, smoothed, or even more processed as a linear predictor (a column of matrix M).




```
for kind, indic in enumerate(indicators):
    tmp = rv_data[indic].copy()
    tmp_in = tmp - np.mean(tmp)
    if indic_kernel_smoother[kind] is not None:
        tmp_smooth = tools.smooth_series(
            rv_data['rjd']/(365.25*indic_filter_timescale_yr[kind]),
            tmp,
            indic_kernel_smoother[kind])
        if indic_filter_type[kind] == 'high':
            tmp -= tmp_smooth
        else:
            tmp = tmp_smooth
    tmpmx = tmp.max()
    tmpmn = tmp.min()
    tmp = 2.0*(tmp-tmpmn)/(tmpmx-tmpmn) - 1.0
    tmp = tmp - np.mean(tmp)
    tmp_in = tmp_in/np.std(tmp_in) * np.std(tmp)
    indic_name = indic
    if indic_filter_type[kind] is not None:
        indic_name += "_" + indic_filter_type[kind]
    rv_model.addlin(tmp, indic_name)
```

Periodogram definition

Periodograms I

we use the framework of Delilse, Hara, Ségransan 2020, generalizing Baluev 2008, MNRAS
The base model is

$$H(x) : y = \phi_H x + \epsilon \tag{17}$$

where ϵ is a Gaussian noise of covariance matrix V and ϕ_H is a fixed $N \times p$ matrix. The alternative model is

$$\begin{aligned} K_\omega(x, A, B) : y &= \phi_H x + A\cos\omega t + B\sin\omega t + \epsilon & (18)\\ &= [\phi_H, \cos\omega t, \sin\omega t]\begin{bmatrix} x \\ A \\ B \end{bmatrix} + \epsilon & (19)\\ &= \phi_K(\omega)x' + \epsilon & (20) \end{aligned}$$

Where $\phi_K(\omega) = [\phi_H, \cos\omega t, \sin\omega t]$, $x' = \begin{bmatrix} x \\ A \\ B \end{bmatrix}$

Periodograms II

One can compare the likelihoods of H and K_ω maximized over their parameters (x and x'). The models H and K_ω are linear, and the noise is fixed and Gaussian. This means that maximising the likelihood is equivalent to minimising the χ^2 of the residuals

Denoting by $\boldsymbol{x}_i$ the least square solution of the minimisation $\mathrm{argmin}_x(\boldsymbol{y} - \Phi_i x)^T V^{-1}(\boldsymbol{y} - \Phi_i x)$ for $i = H$ or $i = K_\omega$

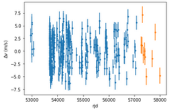
$$\chi_i^2 = (\boldsymbol{y} - \Phi_i x_i)^T V^{-1}(\boldsymbol{y} - \Phi_i x_i) \tag{21}$$

We use the definition of the periodogram
$P(\omega) = \frac{\chi_H^2 - \chi_{K_\omega}^2}{\chi_H^2}$
where χ_M^2 is the χ^2 of the residuals after fitting model M.

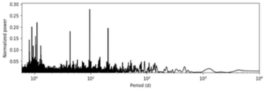
Periodogram III

```
#Define frequency  grid
Pmin = 0.6
Pmax = 10000
nfreq = 100000
nu0 = 2 * np.pi / Pmax
dnu = (2 * np.pi / Pmin - nu0) / (nfreq - 1)
for _ in range(max_pla):

    #Compute priodogram
    nu, power = rv_model.periodogram(nu0, dnu, nfreq)
    P = 2 * np.pi / nu

    # Compute FAP
    kmax = np.argmax(power)
    faplvl = rv_model.fap(power[kmax], nu.max())

    # Add new planet
    rv_model.smartaddpla(P[kmax])

    # Global fit of the model
    rv_model.fit(method=fit_method, options=fit_options)
    rv_model.show_params()
```

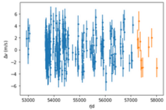
Periodogram III



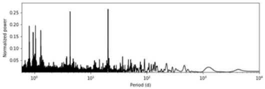

```
Periodogram of residuals:
```




```
Highest periodogram peak:
P=9.62203292065871 d, power=0.2771653464105409, FAP=6.76845073991474e-28

Fit all parameters:
Parameter                          Value     Error
lin.offset_inst_0              31334.350  ±  0.104
lin.offset_inst_1              31348.735  ±  0.523
lin.rhk_low                        1.855  ±  0.122
pla.0.n                        0.6529803  ±  0.0000479
pla.0.la0                        -2.5941  ±  0.0634
pla.0.K                            2.303  ±  0.126
pla.0.k                          -0.0895  ±  0.0541
pla.0.h                          -0.0210  ±  0.0514

loglikelihood = -947.0154752108538

Plot residuals:
```




```
Periodogram of residuals:
```




```
Highest periodogram peak:
P=20.40224211457698 d, power=0.2642084945298696, FAP=3.6632394742080033e-26
```

MCMC

MCMC

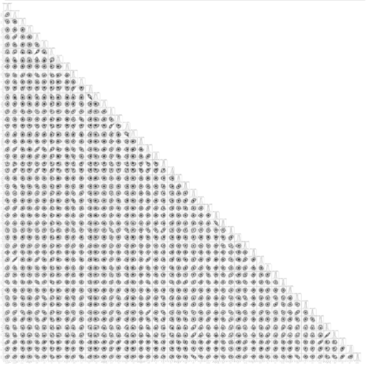
Launch a Monte-Carlo markov chain to estimate the uncertainties.The MCMC does the computation automatically on the model

Launch MCMC

Entrée [12]:
```
samples, diagnos = rv_model.sample(nsamples=nsamples, logprior=lambda fitparams, x:0)
```

```
Step 100000, acceptance rate (since last printing): 0.1970
```

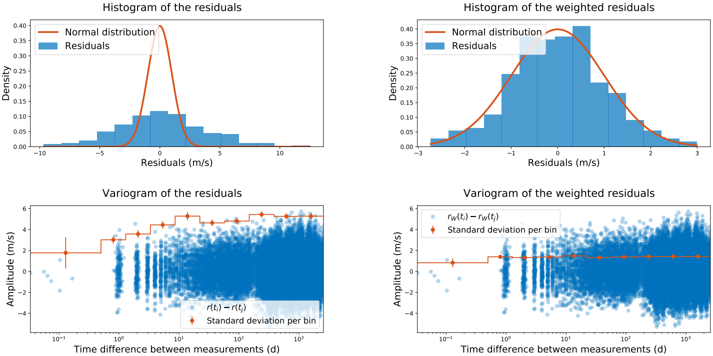
Algorithm: adaptive metropolis from Delisle et al. (2018), based on Haario et al. (2001).



Other tools

Looking at the residuals

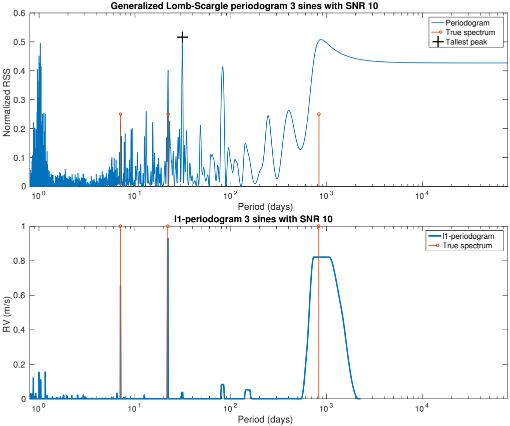
If the model you select is reasonable, the residuals should be well behaved.



See Hara et al. (2019, 2020) and Baluev (2013) for details.

ℓ_1 periodogram

Search for several sinusoidal components at the same time. Based on sparse recovery techniques (fast).

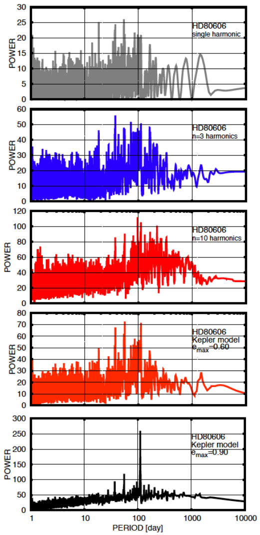


See Hara et al. (2017, 2020) for details, code + tutorial available `https://github.com/nathanchara/l1periodogram`, similar philosophy as DACE codes.

PlanetPack

From Baluev (2018). Keplerian periodograms, general non linear periodograms, joint fit of photometric and RV data.

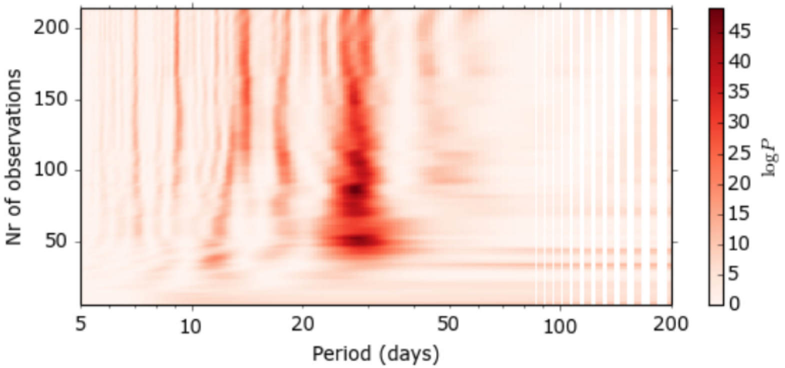
Code (in C++) available at `https://sourceforge.net/projects/planetpack/`



From Baluev (2015)

Stacked periodograms

Compute the periodograms adding one point at a time. The power of planetary signals increases steadily



See Mortier and Collier Cameron (2017). Idea already proposed by Schuster (1898)!

AGATHA

Periodogram marginalized on nuisance parameters (as opposed to maximised).

3 MARGINALIZED LIKELIHOOD PERIODOGRAM

Although the BFP penalizes model complexity by applying BIC-estimated BF, it assumes a Gaussian-like posterior for each parameter and treats each parameter equally as a free parameter of the model. But such assumptions are not always valid, especially when the posterior is multimodal. According to the Bayesian theorem, the posterior distribution of parameters $\boldsymbol{\theta}$ for a given model M is

$$P(\boldsymbol{\theta}|D, M) = \frac{P(D|\boldsymbol{\theta}, M)P(\boldsymbol{\theta}|M)}{P(D|M)}, \quad (11)$$

where $P(D|M) = \int P(D|\boldsymbol{\theta}, M)P(\boldsymbol{\theta}|M)\mathrm{d}\boldsymbol{\theta}$ is the so-called 'evidence' or integrated likelihood, $P(D|\boldsymbol{\theta}, M)$ is the likelihood function, and $P(\boldsymbol{\theta}|M)$ is the prior probability density. The evidence ratio of two models is the BF.

Assuming uniform prior distributions for all parameters, the posterior of frequency f is

$$P(f|D, M) \equiv \int P(\boldsymbol{\theta}', \boldsymbol{\theta}_{\rm fix}, f|D, M)\mathrm{d}\boldsymbol{\theta}' \propto \int \mathcal{L}(\boldsymbol{\theta}', \boldsymbol{\theta}_{\rm fix}, f)\mathrm{d}\boldsymbol{\theta}', \quad (12)$$

See Feng et al. (2017)

Kima

Sampling the posterior distribution of orbital elements with a nested sampling algorithm (Skilling, 2006),
Kima (Faria et al., 2018) allows to compute evidences, defined as:

$$p(\boldsymbol{y}|k \text{ planets}) = \int p(\boldsymbol{y}|\boldsymbol{\theta}, k \text{ planets}) p(\boldsymbol{\theta}|k \text{ planets}) \, \mathrm{d}\, \boldsymbol{\theta} \qquad (22)$$

Code at `https://github.com/j-faria/kima/wiki`

ExoFAST

Modelling RV data, and/or photometric light curves, with any combination of wavelengths
Code available at `https://github.com/jdeast/EXOFASTv2`

EXOFASTv2: A public, generalized, publication-quality exoplanet modeling code

JASON D. EASTMAN,[1] JOSEPH E. RODRIGUEZ,[1] ERIC AGOL,[2,3] KEIVAN G. STASSUN,[4,5] THOMAS G. BEATTY,[6] ANDREW VANDERBURG,[7,8] B. SCOTT GAUDI,[9] KAREN A. COLLINS,[1] AND RODRIGO LUGER[10]

[1] *Center for Astrophysics | Harvard & Smithsonian, 60 Garden St, Cambridge, MA 02138, USA*
[2] *Virtual Planetary Laboratory, University of Washington, Seattle, WA, USA*
[3] *Department of Astronomy, University of Washington, Seattle, WA, USA*
[4] *Department of Physics and Astronomy, Vanderbilt University, 6301 Stevenson Center Ln., Nashville, TN 37235, USA*
[5] *Department of Physics, Fisk University, 1000 17th Avenue North, Nashville, TN 37208, USA*
[6] *Department of Astronomy and Steward Observatory, University of Arizona, Tucson, AZ 85721, USA*
[7] *Department of Astronomy, The University of Texas at Austin, 2515 Speedway, Stop C1400, Austin, TX 78712*
[8] *NASA Sagan Fellow*
[9] *Department of Astronomy, The Ohio State University, 140 West 18th Avenue, Columbus, OH 43210, USA*
[10] *Center for Computational Astrophysics, Flatiron Institute, New York, NY*

ABSTRACT

We present the next generation public exoplanet fitting software, EXOFASTv2. It is capable of fitting an arbitrary number of planets, radial velocity data sets, astrometric data sets, and/or transits observed with any combination of wavelengths. We model the star simultaneously in the fit and provide several state-of-the-art ways to constrain its properties, including taking advantage of the now-ubiquitous all-sky catalog photometry and Gaia parallaxes. EXOFASTv2 can model the star by itself, too. Multi-planet systems are modeled self-consistently with the same underlying stellar mass that defines their semi-major axes through Kepler's law and the planetary period. Transit timing, duration, and depth variations can be modeled with a simple command line option.

Bibliography I

Baluev, R. (2018). Planetpack3: A radial-velocity and transit analysis tool for exoplanets. *Astronomy and Computing*, 25:221 – 229.

Baluev, R. V. (2013). The impact of red noise in radial velocity planet searches: only three planets orbiting GJ 581? *MNRAS*, 429:2052–2068.

Baluev, R. V. (2015). Keplerian periodogram for Doppler exoplanet detection: optimized computation and analytic significance thresholds. *MNRAS*, 446:1478–1492.

Delisle, J. B., Hara, N., and Ségransan, D. (2020). Efficient modeling of correlated noise II. A flexible noise model with fast and scalable methods. *arXiv e-prints*, page arXiv:2004.10678.

Bibliography II

Delisle, J.-B., Ségransan, D., Dumusque, X., Diaz, R. F., Bouchy, F., Lovis, C., Pepe, F., Udry, S., Alonso, R., Benz, W., Coffinet, A., Collier Cameron, A., Deleuil, M., Figueira, P., Gillon, M., Lo Curto, G., Mayor, M., Mordasini, C., Motalebi, F., Moutou, C., Pollacco, D., Pompei, E., Queloz, D., Santos, N. C., and Wyttenbach, A. (2018). The HARPS search for southern extra-solar planets. XLIII. A compact system of four super-Earth planets orbiting HD 215152. *A&A*, 614:A133.

Faria, J. P., Santos, N. C., Figueira, P., and Brewer, B. J. (2018). kima: Exoplanet detection in radial velocities. *Journal of Open Source Software*, 3(26):487.

Feng, F., Tuomi, M., and Jones, H. R. A. (2017). Agatha: disentangling periodic signals from correlated noise in a periodogram framework. *MNRAS*, 470:4794–4814.

Foreman-Mackey, D., Agol, E., Ambikasaran, S., and Angus, R. (2017). Fast and Scalable Gaussian Process Modeling with Applications to Astronomical Time Series. *AJ*, 154:220.

Bibliography III

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive metropolis algorithm. *Bernoulli*, 7(2):223–242.

Hara, N. C., Bouchy, F., Stalport, M., Boisse, I., Rodrigues, J., Delisle, J. B., Santerne, A., Henry, G. W., Arnold, L., Astudillo-Defru, N., Borgniet, S., Bonfils, X., Bourrier, V., Brugger, B., Courcol, B., Dalal, S., Deleuil, M., Delfosse, X., Demangeon, O., Díaz, R. F., Dumusque, X., Forveille, T., Hébrard, G., Hobson, M. J., Kiefer, F., Lopez, T., Mignon, L., Mousis, O., Moutou, C., Pepe, F., Rey, J., Santos, N. C., Ségransan, D., Udry, S., and Wilson, P. A. (2020). The SOPHIE search for northern extrasolar planets. XVI. HD 158259: A compact planetary system in a near-3:2 mean motion resonance chain. *A&A*, 636:L6.

Hara, N. C., Boué, G., Laskar, J., and Correia, A. C. M. (2017). Radial velocity data analysis with compressed sensing techniques. *MNRAS*, 464:1220–1246.

Bibliography IV

Hara, N. C., Boué, G., Laskar, J., Delisle, J. B., and Unger, N. (2019). Bias and robustness of eccentricity estimates from radial velocity data. *MNRAS*, 489(1):738–762.

Haywood, R. D., Collier Cameron, A., Queloz, D., Barros, S. C. C., Deleuil, M., Fares, R., Gillon, M., Lanza, A. F., Lovis, C., Moutou, C., Pepe, F., Pollacco, D., Santerne, A., Ségransan, D., and Unruh, Y. C. (2014). Planets and stellar activity: hide and seek in the CoRoT-7 system. *MNRAS*, 443:2517–2531.

Kipping, D. M. (2014). Bayesian priors for the eccentricity of transiting planets. *MNRAS*, 444:2263–2269.

Mortier, A. and Collier Cameron, A. (2017). Stacked Bayesian general Lomb-Scargle periodogram: Identifying stellar activity signals. *A&A*, 601:A110.

Noyes, R. W. (1984). The Study of Stellar Global Oscillations by CAII H and K Variations. In Mangeney, A. and Praderie, F., editors, *Space Research in Stellar Activity and Variability*, page 113.

Bibliography V

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Schuster, A. (1898). On the investigation of hidden periodicities with application to a supposed 26 day period of meteorological phenomena. *Terrestrial Magnetism*, 3(1):13–41.

Skilling, J. (2006). Nested sampling for general bayesian computation. *Bayesian Anal.*, 1(4):833–859.