SF2A 2019
P. Di Matteo, O. Creevey, A. Crida, G. Kordopatis, J. Malzac, J.-B. Marquette, M. N'Diaye, O. Venot (eds)

DETERMINING SURFACE ROTATION PERIODS OF SOLAR-LIKE STARS OBSERVED BY THE KEPLER MISSION USING MACHINE LEARNING TECHNIQUES

S.N. Breton[1,2], L. Bugnet[1,2], A.R.G. Santos[3], A. Le Saux[1,2], S. Mathur[4,5], P.L. Pallé[4,5] and R.A. García[1,2]

Abstract. For a solar-like star, the surface rotation evolves with time, allowing in principle to estimate the age of a star from its surface rotation period. Here we are interested in measuring surface rotation periods of solar-like stars observed by the NASA mission *Kepler*. Different methods have been developed to track rotation signals in *Kepler* photometric light curves: time-frequency analysis based on wavelet techniques, autocorrelation and composite spectrum. We use the learning abilities of random forest classifiers to take decisions during two crucial steps of the analysis. First, given some input parameters, we discriminate the considered *Kepler* targets between rotating MS stars, non-rotating MS stars, red giants, binaries and pulsators. We then use a second classifier only on the MS rotating targets to decide the best data-analysis treatment.

Keywords: asteroseismology, rotation, solar-like stars, kepler, machine learning, random forest

1 Introduction

Rotation plays an important role in stellar evolution. For cool main-sequence (MS) dwarfs (G, K and M spectral type), age may be determined thanks to gyrochronology (Skumanich 1972): surface rotation evolves roughly as the square root of its age. Even if recent studies suggested that at a given stage of its evolution, the braking of a solar-like star is reduced (van Saders et al. 2016), this relation seems to be verified while the stars remain on the main sequence.

In this work, we use *Kepler* (Borucki et al. 2010) photometric light curves obtained with the KADACS pipeline (*Kepler* Asteroseismic Data Analysis and Calibration Software, García et al. 2011; García et al. 2014). The KADACS pipeline has been specifically designed to correct for *Kepler* light curves from instrumental effects and properly stitch the quarters in an optimized way for asteroseismology studies. We consider different high-pass filters (20, 55, and 80 days) for the processing of the light curves to be sure that rotation period is not filtered out. Rotation period is then extracted thanks to a combination of different methods (Global Wavelet Power Spectrum GWPS, AutoCorrelation Function ACF and Composite Spectrum CS) as described in Mathur et al. (2010), García et al. (2014), Ceillier et al. (2016) and Ceillier et al. (2017). However, the computed rotation period may differ from one KADACS filter to another and from one method to another. Dozens of thousands of stars have to be considered with, until now, no other solution than to use a pre-defined hierarchical decision tree and make a final visual inspection of the conflicting cases. A machine learning algorithm seems the ideal tool to make the decision over the stars of our data set. Before performing this analysis and to avoid any bias, it is nevertheless necessary to distinguish main sequence rotators from other types of targets.

2 Rotators classification

[1] IRFU, CEA, Université Paris-Saclay, 91191 Gif-sur-Yvette, France
[2] AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, 91191 Gif-sur-Yvette, France
[3] Space Science Institute, 4750 Walnut Street Suite 205, Boulder, CO 80301, USA
[4] Instituto de Astrofísica de Canarias, 38200 La Laguna, Tenerife, Spain
[5] Universidad de La Laguna, Dpto. de Astrofísica, 38205 La Laguna, Tenerife, Spain

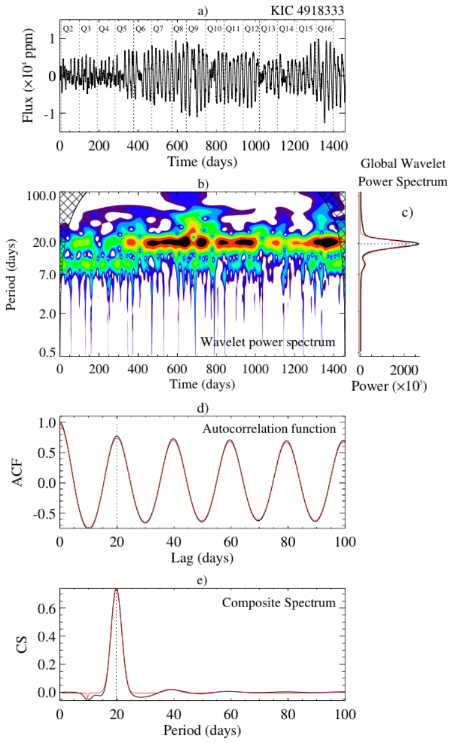

Fig. 1. *From top to bottom* - example of *Kepler* photometric lightcurve analyzed with A2Z pipeline (Mathur et al. 2010), wavelet power spectrum, autocorrelation function and composite spectrum. Extracted from Santos et al. (submitted to ApJ).

Our set of stars consist of 14,441 M and K dwarfs based on the *Kepler* star properties catalog from Mathur et al. (2017), whose rotation periods have been studied by Santos et al. (submitted to ApJ). However, the sample can be polluted by red giants (RG), classical pulsators (CP) or eclipsing binaries. Santos et al. identified these pollutors by visually checking the light curves. Here, we propose to use artificial intelligence methods to automatically detect such pollutions. We train a first random forest algorithm (with the Python package *scikit-learn*, Pedregosa et al. 2011) in order to identify those different targets. The principle of a random forest algorithm is briefly reminded in annex A1. The input parameters are:

- periods computed by each method, $P_{\rm GWPS}$, $P_{\rm ACF}$, $P_{\rm CS}$, and related control values $H_{\rm ACF}$, $G_{\rm ACF}$ and $H_{\rm CS}$ (see Figure 1 and Ceillier et al. 2017 for further explanation);
- photometric activity proxy $S_{\rm ph}$ (García et al. 2010, 2014; Mathur et al. 2014);
- FliPer values (see Bugnet et al. 2018, 2019);
- effective temperature $T_{\rm eff}$ and surface gravity $\log g$ from Mathur et al. (2017).

Those parameters have been chosen because their values are directly related to stellar types and rotation properties. $T_{\rm eff}$ and $\log g$ are good parameters to distinguish between red giants and MS stars. The FliPer metric allows us to help disentangling the proposed classes of stars attending to their power in the PSD. Rotation periods computed by the three methods combined with all the other related parameters ($H_{\rm ACF}$, $G_{\rm ACF}$ and $H_{\rm CS}$) allow

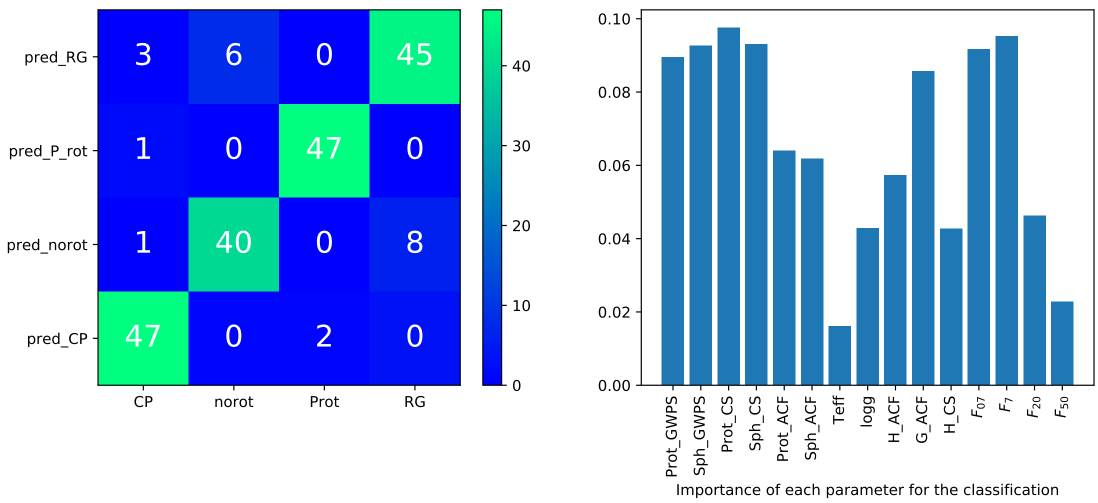

Fig. 2. *Left panel* - classification result for a test set of 200 stars. The algorithm has been trained with 600 stars that were visually classified before-hand. CP stands for classical pulsator, norot for MS non-rotating star, Prot for MS rotating star, RG for red giant. The real class of an element corresponds to its column label, the class assignated by the classifier corresponds to the line label. Accuracy of the classification is 0.895. *Right panel* : relative importance of each parameter used for the classification.

the classifier to decide whether the measured signal corresponds to a rotation period or not. Figure 2 shows the result of the classification of the test set. Stars are globally well classified, except for some non-rotating main-sequence stars and red giants with close $T_{\rm eff}$ and $\log g$. On a $T_{\rm eff}$-$\log g$ diagram, those stars would lay in the subgiants region. Thus, one of the next improvements of the classifier will be to add the possibility for the algorithm to give a label *subgiant*.

3 Period determination

A second random forest classifier is trained to determine the best filter to consider (20, 55, 80 days) to retrieve the most probable rotation period $P_{\rm rot}$. We assume that this $P_{\rm rot}$ will be given by the wavelet method of the correct filter. Comparing the best filter choice and the classifier choice gives us an estimation of the classifier accuracy. Sometimes, even when the classifier choice is not the filter chosen by Santos et al. (submitted to ApJ), the period estimate is approximately the same. If the period differs by less than 10% from the true period labelled on the training set, we consider that the classifier is right and compute what we call the true accuracy (e.g. the true accuracy score is given between the ratio of stars with a retrieved period laying between $\pm 10\%$ error according to the right period over the total number of stars). Our 14,441 stars are distributed between a training set of 12,275 stars (85 %) and a test set of 2,166 stars (15 %). The distribution is randomly chosen. To check whether it could be responsible for a bias in the training, we compute ten trainings of the algorithm with different distributions each time. The average classifier accuracy and true accuracy over those ten runs are respectively 0.936 and 0.979 (see Figure 3 for an example of the training).

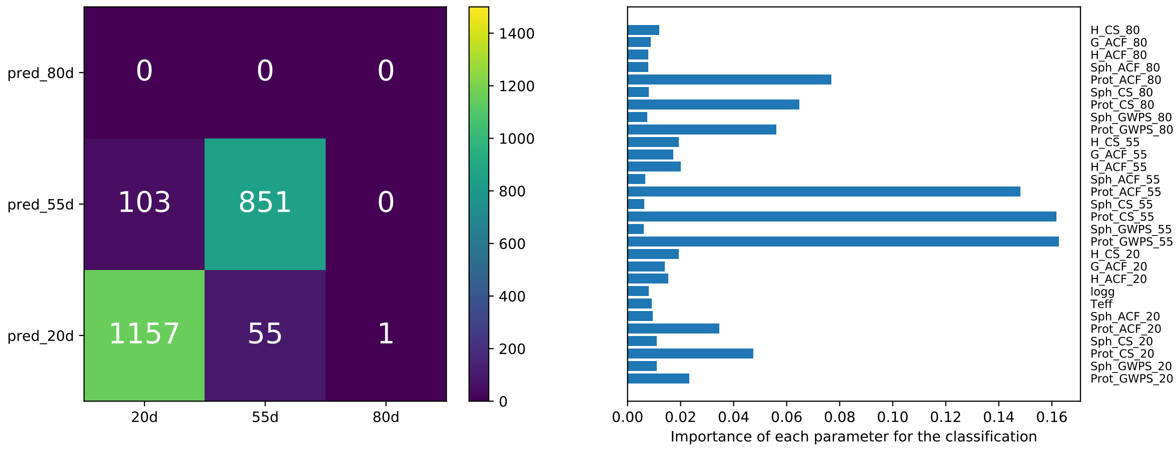

Fig. 3. *Left panel*: filter-choice result for a test set of 2166 rotating MS stars. The algorithm was trained with 12,275 stars. The total number of stars is 14,441. Classifier accuracy is 0.927. The true accuracy on the period is 0.974. *Right panel*: relative importance of each parameter used for the classification. $P_{\rm GWPS}$, $P_{\rm ACF}$, $P_{\rm CS}$, $H_{\rm ACF}$, $G_{\rm ACF}$, $H_{\rm CS}$ and $\rm S_{ph}$ values are considered for each filter (20, 55, 80) and consequently subscripted in the legend of the plot.

4 Conclusions

Random forest classifiers prove themselves to be an excellent tool to study stellar rotation properties and allow us to deal with large datasets. On the two distinct steps of the analysis, we get promising accuracy values of 0.895 for the rotators classification and 0.979 for the retrieval of rotation period. Especially, the classifier seems particularly efficient to retrieve the filter that leads to the rotation period. However, we still need to improve the accuracy of the results in the future, especially by using larger data sets. One of the goal of future work will be to apply the analysis to datasets from other missions like K2 or TESS.

This paper includes data collected by the *Kepler* mission and obtained from the MAST data archive at the Space Telescope Science Institute (STScI). Funding for the *Kepler* mission is provided by the NASA Science Mission Directorate. STScI is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 526555. This work has been partially fund by the GOLF and PLATO grants at the CEA. A.R.G.S acknowledges the support from National Aeronautics and Space Administration (NASA) under the grant NNX17AF27G. S.M. acknowledges the support from the Ramon y Cajal fellowship number RYC-2015-17697. S.N.B. thanks all the SSEBE team at the IAC for the all the scientific discussions and support during the internship period at the IAC.

References

Borucki, W. J., Koch, D., Basri, G., et al. 2010, Science, 327, 977
Bugnet, L., García, R. A., Davies, G. R., et al. 2018, A&A, 620, A38
Bugnet, L., García, R. A., Mathur, S., et al. 2019, A&A, 624, A79
Ceillier, T., Tayar, J., Mathur, S., et al. 2017, A&A, 605, A111
Ceillier, T., van Saders, J., García, R. A., et al. 2016, MNRAS, 456, 119
García, R. A., Ballot, J., Mathur, S., Salabert, D., & Regulo, C. 2010, arXiv e-prints, arXiv:1012.0494
García, R. A., Ceillier, T., Salabert, D., et al. 2014, A&A, 572, A34
García, R. A., Hekker, S., Stello, D., et al. 2011, MNRAS, 414, L6
García, R. A., Mathur, S., Pires, S., et al. 2014, A&A, 568, A10
Mathur, S., García, R. A., Ballot, J., et al. 2014, A&A, 562, A124
Mathur, S., García, R. A., Régulo, C., et al. 2010, A&A, 511, A46
Mathur, S., Huber, D., Batalha, N. M., et al. 2017, ApJS, 229, 30
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Santos, A. R. G., Garca, R. A., Mathur, S., et al. submitted to ApJ, ApJ
Skumanich, A. 1972, ApJ, 171, 565
van Saders, J. L., Ceillier, T., Metcalfe, T. S., et al. 2016, Nature, 529, 181

A1 Random forest classifiers

A random forest algorithm is a useful machine learning tool for classification. Thanks to the training set, the algorithm is able to grow a *forest* of decision trees that will then be used to assign a label to new data. A simple example of decision tree is showed in Figure 4

Decision trees are built with the following principle. The training data set is split from the root of the tree according to the value of one of the parameters of the data. Each resulting node gets a Gini score G :

$$\mathrm{G} = \sum_{k=1}^{\mathrm{N_{classes}}} p_k \times (1 - p_k); \tag{4.1}$$

that quantifies its purity (e.g. the proportion of each class for the data assigned to the node). A Gini score of 0 means that a node is totally pure (i.e. that all the elements assigned to the node have the same class). When the score of a node is low enough, the splitting stops: the node becomes a leaf that assignates to new data the label of the dominant class.

In order to choose a split close to the optimal possibility without consuming too much computation time, a number of possible splits is randomly generated and the best one is chosen over this sample.

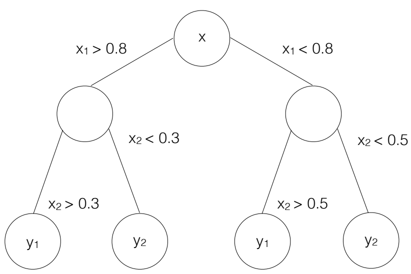

Fig. 4. Example of decision tree designed to classify two-parameters data $\boldsymbol{x} = \{x1, x2\}$ within two classes $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$.