MNRAS 000, 1–6 (2020) Preprint 21 July 2020 Compiled using MNRAS LATEX style file v3.0

Modelling stars with Gaussian Process Regression – I: Augmenting Stellar Model Grid

Tanda Li,[1]★ Guy R. Davies,[1]† Alex Lyttle,[1] Lindsey Carboneau,[1] and A. N. Others[1]
[1] *School of Physics and Astronomy, University of Birmingham, Birmingham, B15 2TT, United Kingdom*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
Grid-based modelling is widely used for estimating stellar parameters. However, theoretical model grid is sparse. This paper demonstrates an application of Gaussian Process Regression (GPR), which transfer a sparse model grid to a continuing function. We trained a GPR model with a model grid to learn the function from 5 independent fundamental inputs (Mass, fractional age, initial metallicity, initial helium fraction, and the mixing-length parameter) to 6 model outputs (age, seismic large separation, effective temperature, surface gravity, radius, and surface metallicity). The GPR function was then validated and the typical standard deviations of GPR predictions are 1% for the age, 0.5μHz for the seismic large separation, 12K for the effective temperature, 0.004dex for the surface gravity, 0.008$R_\odot$ for the radius, and 0.008dex for the surface metallicity. This indicates that GPR can be used to augmenting stellar grids and furthered model stars. We lastly applied the GPR function for modelling the Sun-as-a-star. The GPR function gives nicer statistical estimates than grid-based modelling does.

Key words: keyword1 – keyword2 – keyword3

1 INTRODUCTION

This will be the intro.

2 THEORETICAL MODEL GRID

We built up a model grid as the base data of GP model. In this work, we aimed cover stars with approximate solar mass on main-sequence and subgiant phases. Thus, the mass range is set up as 0.8 – 1.2$M_\odot$, and we computed each stellar evolutionary track from the Hayashi line and to the base of red-giant branch where $/logg = 3.5$ dex. The gird includes four independent model inputs: stellar mass (M), initial helium fraction ($Y_{\rm init}$), initial metallicity ([Fe/H]), and the mixing-length parameter ($\alpha_{\rm MLT}$). Ranges and grid steps of the four model inputs are summarized in Table 1.

2.1 Stellar models and input physics

We used Modules for Experiments in Stellar Astrophysics (MESA, version 12115) to establish a grid of stellar models. MESA is an open-source stellar evolution package which is undergoing active development. Descriptions of input physics and numerical methods can be found in **???**. We adopted the solar chemical mixture $[(Z/X)_\odot = 0.0181]$ provided by **?**. The initial chemical composition was calculated by:

$$\log(Z_{\rm init}/X_{\rm init}) = \log(Z/X)_\odot + [{\rm Fe/H}]. \quad (1)$$

We used the MESA $\rho - T$ tables based on the 2005 update of OPAL EOS tables (**?**) and OPAL opacity supplemented by low-temperature opacity (**?**). The MESA 'simple' photosphere were used as the set of boundary conditions for modelling the atmosphere. The mixing-length theory of convection was implemented, where $\alpha_{\rm MLT} = \ell_{\rm MLT}/H_p$ is the mixing-length parameter. We also applied the MESA predictive mixing scheme (**??**) in the model computation. The MESA inlist used for the computation is available on https://github.com/litanda/mesa_inlist.

The evolution time step was mainly controlled by the set-up tolerances on changes in surface effective temperature and luminosity. We saved one structural model at every time step at main sequence and every two steps after central hydrogen exhaustion. For each evolutionary track, we obtained ~100 at the main-sequence stage and 500 – 700 at evolved stages.

2.2 Oscillation models and seismic $\Delta\nu$

Theoretical stellar oscillations were calculated with the GYRE code (version 5.1), which was developed by **?**. And we computed radial modes (for $\ell = 0$) by solving the adiabatic stellar pulsation equations with the structural models generated by MESA. We computed a seismic large separation($\Delta\nu$) for each model with theoretical radial modes to avoid the systematic offset of the scaling relation. We

★ E-mail: t.li.2@bham.ac.uk
† E-mail: G.R.Davies@bham.ac.uk

Table 1. Stellar model computations for training and validating sets.

Training model set (Grid-based)			
Input Parameter	Range	Increment	$N_{\rm track}$
M [$M_\odot$]	0.80 – 1.20	0.01	
[Fe/H] [dex]	-0.5 – 0.2/0.2 – 0.5	0.1/0.05	15,375
$Y_{\rm init}$	0.24 – 0.32	0.02	
$\alpha_{\rm MLT}$	1.7 – 2.5	0.2	
Validating model set (Randomly computed in above parameter ranges)			
GPR model	Varying parameters	Fixed parameters	$N_{\rm track}$
2D	M, $t_{\rm frac}$	[Fe/H] = 0.0, $Y_{\rm init}$ = 0.28, $\alpha_{\rm MLT}$ = 2.1	15
3D	M, $t_{\rm frac}$, [Fe/H]	$Y_{\rm init}$ = 0.28, $\alpha_{\rm MLT}$ = 2.1	200
4D	M, $t_{\rm frac}$, [Fe/H], $Y_{\rm init}$	$\alpha_{\rm MLT}$ = 2.1	1,000
5D	M, $t_{\rm frac}$, [Fe/H], $Y_{\rm init}$, $\alpha_{\rm MLT}$	-	4,000

derived $\Delta\nu$ with the approach given by **?**, which is a weighted least-squares fit to the radial frequencies as a function of n.

3 GPR MODEL

An introduction of Gaussian Process Regression (GPR) model and the package GPy. What does a GPR model learn from the data and how it predict stellar parameters?

3.1 Discussion about kernels

The analysing is mainly based on the MLP kernel because data is multiple-demission, and we also use other kernels (Exponential) to combine with the MLP kernel to improve the fitting.

3.2 Selection of GPR Model Inputs

We aim to derive observables from fundamental input parameters of the grid. As mentioned in Section 2, our model grid has five independent fundamental inputs, says, mass, initial metallicity, initial helium fraction, mixing-length parameter, and age. The GPR model inputs are ideally in fixed dynamic ranges (form a cube-like space), however, the age ranges significantly vary on different tracks. We hence define an age index to replace the age as an input, and putted the age as an output. The age index is calculated on each evolutionary track following

$$t' = 10^{(t/t_{\rm max})}, \tag{2}$$

where t is the age, $t_{\rm max}$ is the maximum age of a track. Note that $t_{\rm max}$ must be defined in the same way for all evolutionary tracks. In this work, we defined $t_{\rm max}$ as the age when a track evolve to $\log g$ = 3.8. The purpose to use an exponential formula is flattening sharp changes of observables in the last 10% lifetime. This make GPR models converge easily. A comparison between GPR models using $t/t_{\rm max}$ and $10^{(t/t_{\rm max})}$ is illustrated in Figure 1. As it can be seen that, the effective temperature evolutaion presents a hump around 0.7 of the lifetime (which is the hook) and quickly drops in the last 10% lifetime (the subgiant phase). The GPR model in the top panel does not well learn these two stages, and the residuals (blue dots in the bottom graph) go up to ±1K. By using $10^{(t/t_{\rm max})}$ as the input, those sharp changes are flattened and hence the GPR model predictions are significantly improved (the maximum residual is down to $\sim 10^4$ K). We lastly summary our selections of GPR model inputs and outputs as below.

- GPR model inputs:
  Mass (M = 0.8 – 1.2$M_\odot$)
  Age index (t' = 0.0 – 10)
  Initial metallicity ([Fe/H]$_{\rm init}$ = -0.5 – 0.5)
  Initial helium fraction ($Y_{\rm init}$ = 0.24 – 0.32)
  Mixing-length parameter ($\alpha_{\rm MLT}$ = 1.7 – 2.3)
- GPR model outputs:
  Effective temperature ($T_{\rm eff}$)
  Surface gravity ($\log g$)
  Radius (R)
  Surface metallicity ([Fe/H])
  Two global seismic parameters ($\Delta\nu$ and $\nu_{\rm max}$)

Thus, our GPR model can be described as

$$\text{Observables} = f(M, t', (\text{Fe/H})_{\rm init}, Y_{\rm init}, \alpha_{\rm MLT}). \tag{3}$$

4 MODEL AUGMENTATION WITH GPR MODELS

In this section, we demonstrated the application of GPR on stellar model augmentation.

4.1 Training GPR Models

To demonstrate our work flow, we present the training process of a GPR model for the effective temperature with 2-demission inputs (M, $t_{\rm frac}$). The grid used here includes stellar models with fixed $[Fe/H]_{\rm init}$ (0.0 dex), $Y_{\rm init}$ (0.28), and $\alpha_{\rm MLT}$ (2.1). The total number of models is 24,485.

4.1.1 Step1: Selecting Training, Testing, and Validating Data

For properly training and validating a GPR model, three types of data are required. Firstly, training data are used to train a GPR model. Because the computational and memory complexity exponentially increase with the number of training data, a practical limit is about 10,000. Secondly, we required a testing data set, which were also selected from the model grid but not used in any training processes. Testing data are used to examine wether a GPR model gives proper description for the model grid. Once a GPR model provides good

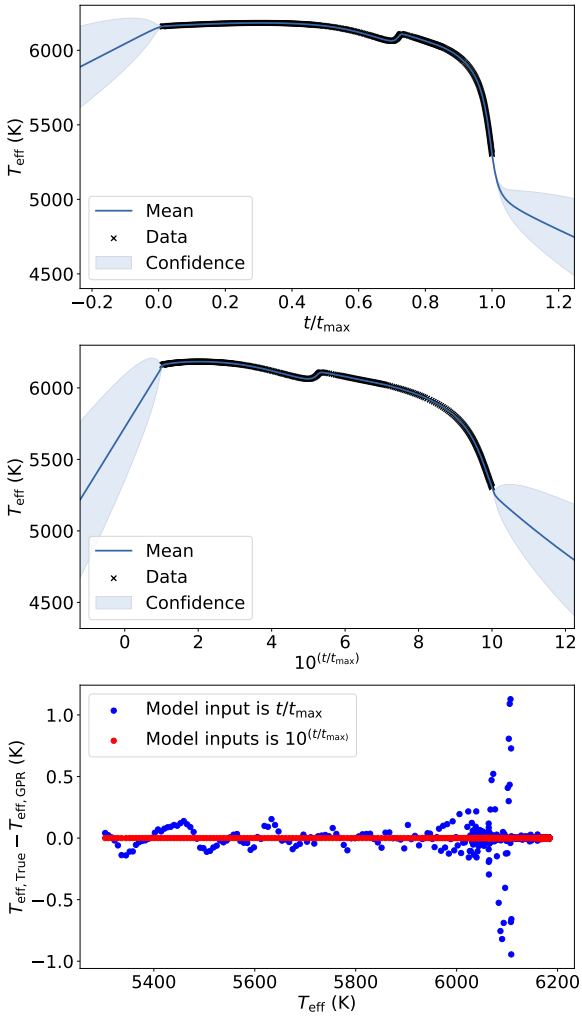

Figure 1. The comparison between GPR model predictions with two different input age indices ($t/t_{\rm max}$ and $10^{(t/t_{\rm max})}$) for a 1.1$M_\odot$ track. Top: the GPR model of the effective temperature as a function of $t/t_{\rm max}$. The adopted kernel is MLP. Middle: same as the top, but the GPR model input is $10^{(t/t_{\rm max})}$). Bottom: residuals between true values and GPR model predictions.

agreement with the testing data, the training succeeds. The last is validating data, which were randomly distributed in the input parameters space. Validating data are used to validate the GPR model performance and estimate the systematical uncertainty.

We selected these three model data following one principle: the data uniformly distribute on all evolutionary stages. Due to the step-control strategy, our MESA models do not uniformly distribute. Models are dense at the main-sequence and the red-giant phases but quite sparse on subgiant stage. Random sampling is hence not appropriate. We tested a few methods and lastly used the

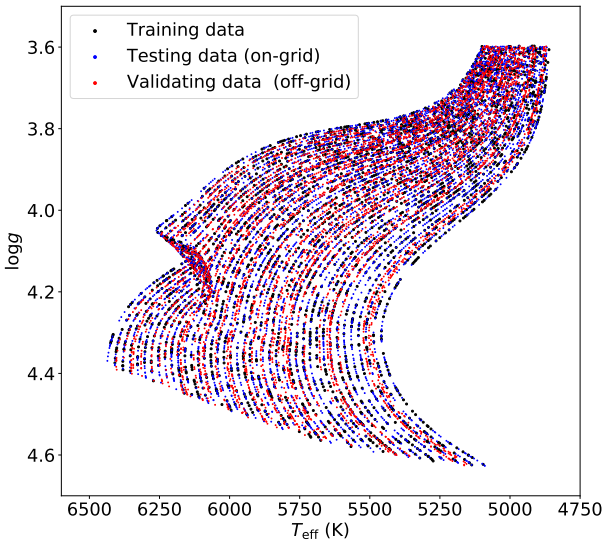

Figure 2. Selected training and testing data for the GPR model with 2-demission inputs on the $T_{\rm eff}$ - $\log g$ (Top) and M - $t_{\rm frac}$ (bottom) diagrams. Black and red dots represent training and testing data.

gradient on the $T_{\rm eff}$ - $\log g$ diagram as the weights for sampling. For each evolutionary track, we calculated the gradient as $\delta d = (\delta T_{\rm eff}^2 + \delta \log g^2)^{1/2}$, which gives relatively uniform data distribution. Selected data for training the 2-demission inputs GPR model are demonstrated in Figure 2 . We selected 3,000 on-grid models as training data, reserved 5,000 on-grid models as testing data, and sampled 5,000 off-grid models as validating data.

4.1.2 *Step 2: Training the Primary GPR Model with the MLP Kernel*

We trained a GPR model with the MLP kernel. The GPR model function is demonstrated in Figure 8.

We used testing data to examine wether the model gives proper description for the grid. The $T_{\rm eff}$ residuals of 95% models are below 1K. This indicate that the MLP kernel is appropriate for the data. However, the GPR model does not well reproduce the features for $M > 1.1$ and $t' = 5 - 6$. This corresponds to the 'hook' and the turn-off point on the $T_{\rm eff} - \log g$ diagram shown at the bottom.

4.1.3 *Step3: Training GPR Models for Residuals*

We then worked on an additional GPR model to fit residuals.

As it can be seen in Figure **??**., the residuals are mostly close to zero in the parameter space and only arise in some particular areas. They hence can be treated as 'spikes'. We selected a number of kernels which are suitable for spike data (MLP, EXP, RQ, Mat32, and RBF), used each them to fit the residuals, and examined which one gives best predictions. The validation (on-grid) errors of each GPR model are illustrated in Figure **??**. As it shown that the combined GPR model with the MLP kernel for data and the RBF kernel for residuals gives the best predictions.

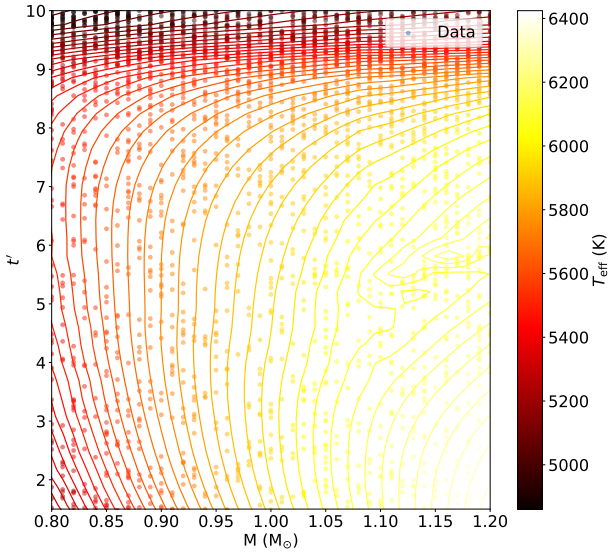

Figure 3. Top: The GPR model (using the MLP kernel) for the effective temperature ($T_{\rm eff}$) as a function of mass (M) and age index (t'). Grey counters describes the original grid and coloured counters are GPR predictions. Note that grey counters are interpolated based on the model grid, hence they are not very smooth. Bottom: Residuals of the GPR model on the $T_{\rm eff} - \log g$ diagram.

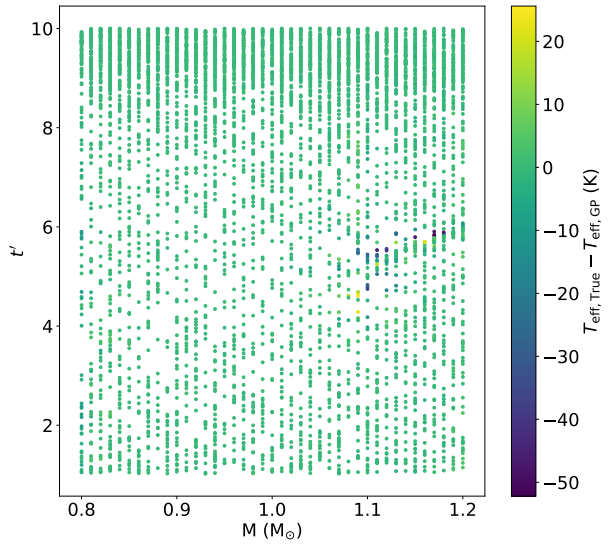

Figure 4. Validation (on-grid) for the GPR model.

4.1.4 *Validating the GPR Model and Estimating Systematical Uncertainty*

We lastly validated the GPR model obtained in the previous step (kernel is MLP/RBF) with off-grid stellar models. The distribution of offsets between true values and the GPR predictions are shown in Figure **??**.

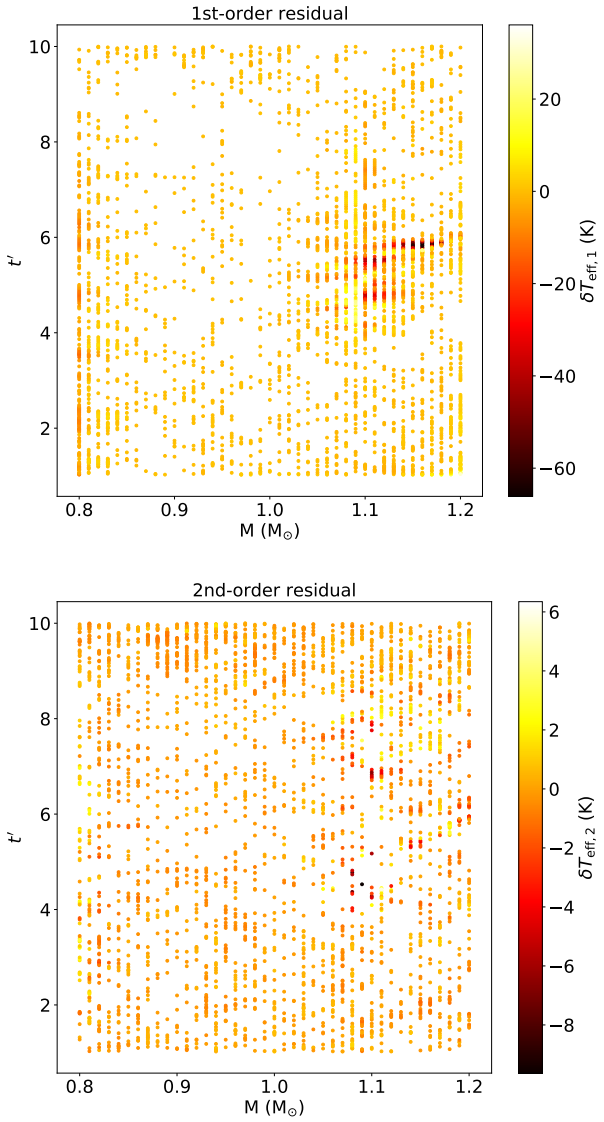

Figure 5. Training data for the first and second order residuals.

4.1.5 *Summary of the Work Flow*

4.2 The GPR model with Five-demission Inputs

- Accuracy goes down. Because the multiple-demission space size increase while the training dataset has a limitation of 10,000 points.
- We hence need to divide the grid into small chunks and train each chunk separately. (illustrate how chunks are divided)
- show training and validation results in a fancy way.

5 DISCUSSION

Do some disccusion.

Table 2. Training and validating for the 5D GPR models

GPR model inputs	Training information		Validations of GPR outputs (50/95/99.8)					
	$N_{\rm grid}$ (10^3)	$N_{\rm training}$ (10^3)	δt (Myr)	$\delta\Delta\nu$ (μHz)	$\delta T_{\rm eff}$ (K)	$\delta \log g$ (10^{-3}dex)	δR ($10^{-3} R_\odot$)	δ[Fe/H] (10^{-3}dex)
M, $t_{\rm frac}$, [Fe/H]$_{\rm init}$, $Y_{\rm init}$, $\alpha_{\rm MLT}$	8,212	10	6/47/102	0.06/0.4/1	3/15/62	0.6/3/12	1/6/26	

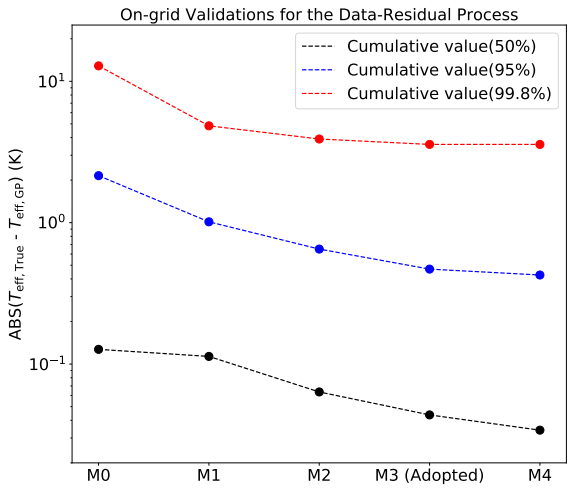

Figure 6. On-grid validations.

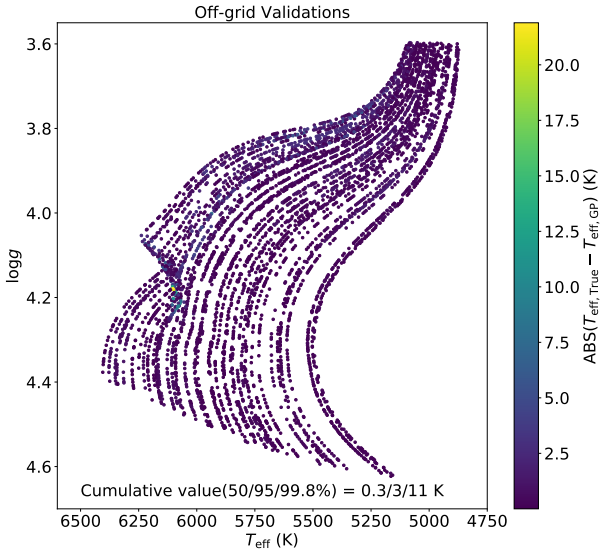

Figure 7. Off-grid validations.

6 CONCLUSIONS

Restate the main results of the paper.

ACKNOWLEDGEMENTS

The Acknowledgements section is not numbered. Here you can thank helpful colleagues, acknowledge funding agencies, telescopes and facilities used etc. Try to keep it short.

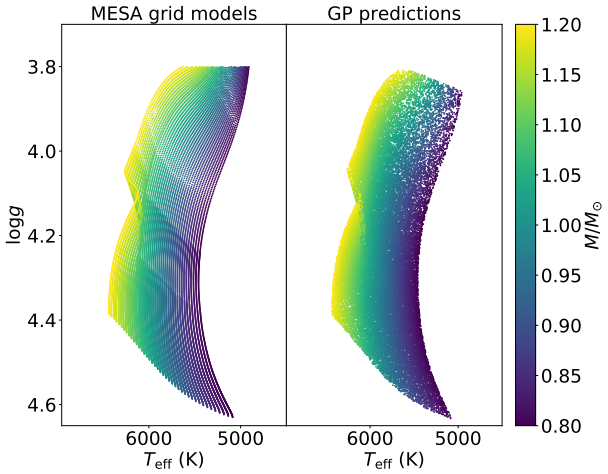

Figure 8. MESA grid models (sparse) and GP predictions (non-sparse) on the $T_{\rm eff}$ - log g diagram.

APPENDIX A: VALIDATION AND PREDICTION OF GPR MODELS

A1 GPR model with 3-D inputs

- Training set
- validation
- GP predictions

A2 GPR model with 4-D inputs

- Training set
- validation
- GP predictions

A3 GPR model with 5-D inputs

- Training set
- validation
- GP predictions

This paper has been typeset from a TEX/LATEX file prepared by the author.

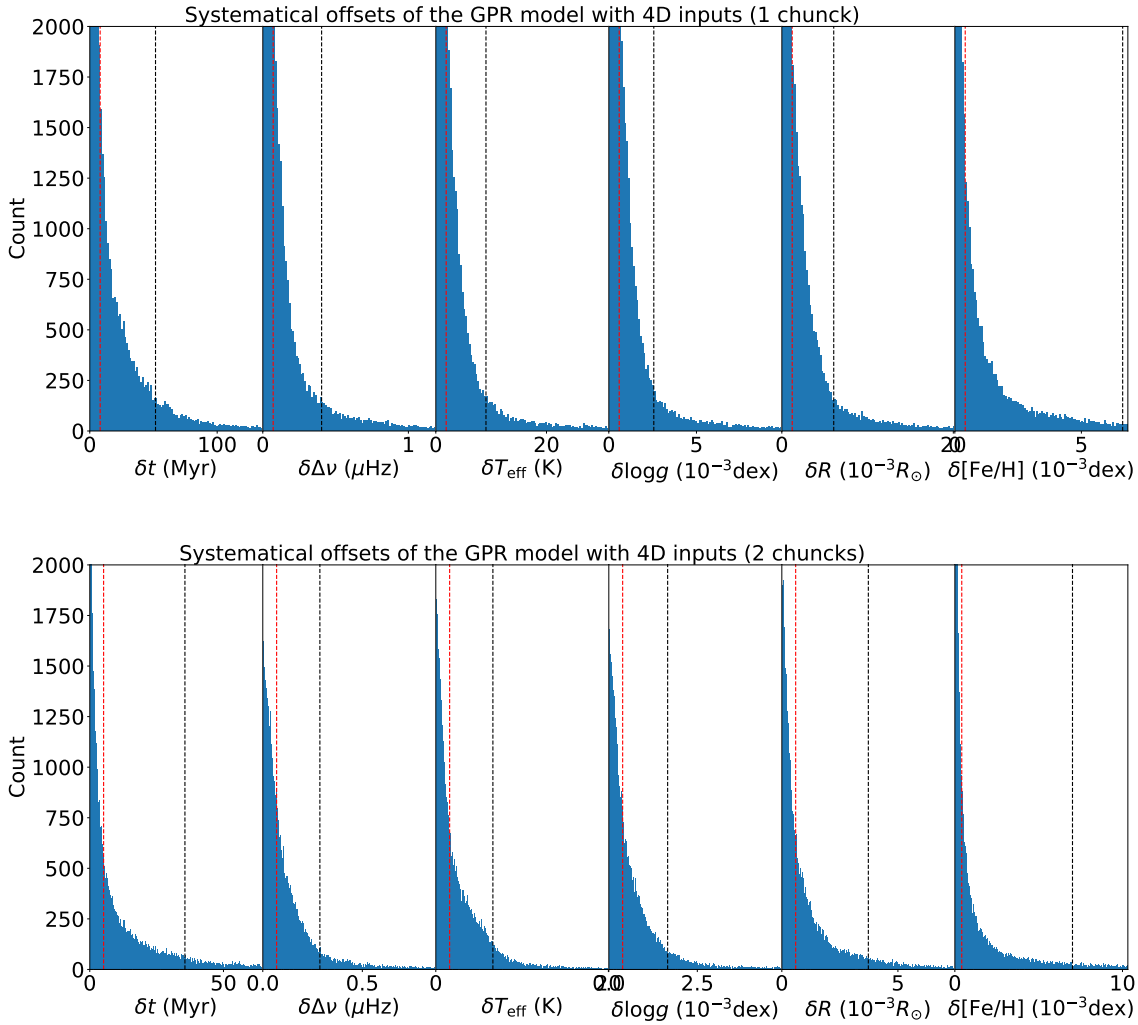

Figure A1. Validations for 4D inputs GPR models before and after chunking.

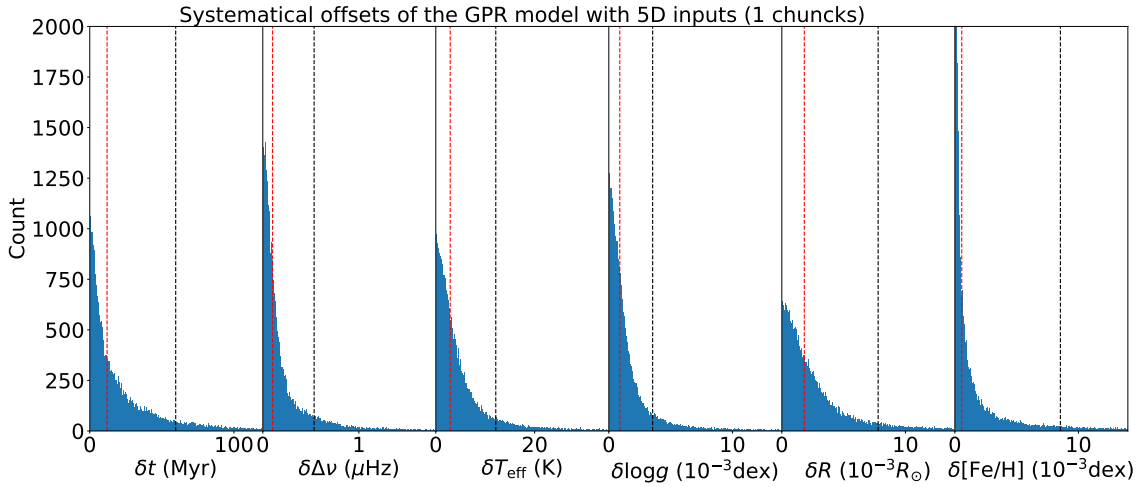

Figure A2. Validations for 5D inputs GPR models before and after chunking.