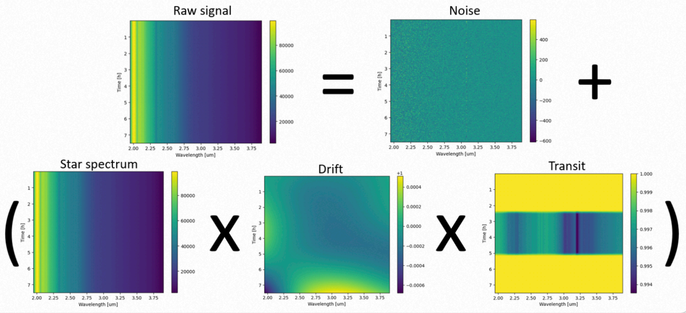
My solution to this challenge is built on the related concepts of **Bayesian Inference** and **Gaussian Processes**.

**Bayesian Inference** (BI) is a powerful statistical approach, based around defining a *prior* (a statistical belief about reality) and *observations* (some form of new information). Using Bayes' law, we then combined these to find the *posterior* (an updated belief about reality). In our case, this means:

- **Prior**: a description of the physics that affect the final measured signal, describing for example detector noise, drift, and the transit behavior (including the transit depth itself) as formal distributions.
- **Observations**: the provided measurements.
- **Posterior**: a breakdown of the observations into the various elements defined in the prior (see figure below). From this we can simply read out the desired transit depth. Importantly, the posterior is not just a single point, it's a distribution. By taking samples from this distribution we can find the required confidence intervals (and even full covariance matrices, although that is not asked of us here).

To me, the most attractive element of BI is how it lets us split our physical thinking from our solver. All our domain knowledge goes into defining the prior, where we can consider one physical element at a time; doing the actual Bayesian Inference to find the posterior is then 'just math'. Not necessarily easy math - but it's entirely separate from our domain knowledge.

The figure below shows how one example transit is split in the posterior:

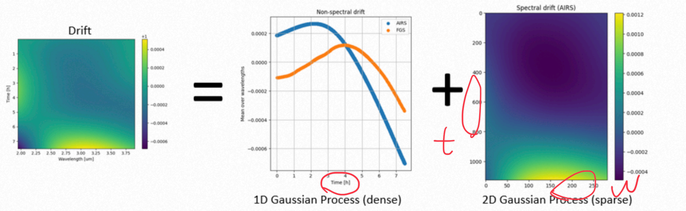


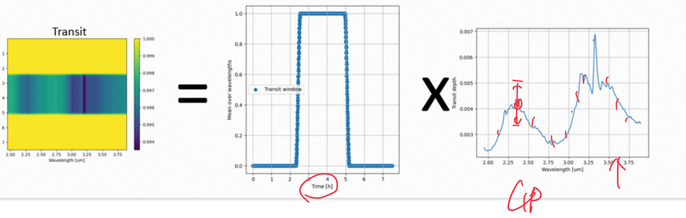

Several of our prior elements are smooth functions; for example, transit depth is a function of wavelength, drift is a function of wavelength and time, etc. This means that we need to describe a distribution of functions in the prior. **Gaussian Processes** (GPs) are a natural way to do this. GPs are non-parametric methods, meaning they do not rely on a fixed set of basis functions. Instead, they allow any function, but assign different probabilities to different functions. For example, a low-frequent drift is more likely than a high-frequent one. An excellent starting point to learn GPs is https://arxiv.org/abs/2009.10862

In my post here, I'll mainly focus on providing the details of the model, which in the BI framework just means describing the prior. At the end I'll cover some odds and ends (preprocessing and how we actually do the BI math).

## Prior definition

In this section I'll provide some visual breakdowns, an overview of the various elements, and some notes. You will see references to separate sensors (AIRS and FGS), but I don't really discuss my use of these sensors; The AIRS is the main sensor (the spectroscope). This section assumes familiarity with GPs, but hopefully you can get the gist of it in any case. If you want to know all the details, you'll have to go into the code (ariel_gp.py).




| Prior element | Description | Tuning and hyperparameters | Degrees of freedom |
|---|---|---|---|
| Noise | Uncorrelated Gaussian per time and wavelength | Standard deviations found in preprocessing | 114340 |
| Star spectrum | Uncorrelated value per wavelength | Not regularized (infinite sigma) | 283 |
| Drift: 1D | 2x GP over time, 1 for AIRS and 1 for FGS | Tuned on training set | 816 |
| Drift: 2D | GP over time and wavelength | Tuned on training set | 113928 → 800 with KISS-GP |
| Transit window | Fixed function, ingress/egress time and width are fit | Fixed function is found on training set | 3 |
| Transit depth: mean | Single value | Not regularized | 1 |
| Transit depth: variation FGS | Single Gaussian value | Standard deviation found on training set | 1 |
| Transit depth: variation AIRS | GP over wavelength | Tuned on training set | 282 |
| Transit depth: PCA | Fixed basis functions obtained from PCA analysis | PCA shapes found from an initial rough fit on test data | 1 |

- All GPs use multiple squared-exponential kernels (i.e. they are themselves multiple GPs combined, each with their own fixed length scale). The hyperparameters (sigma values per length scale) are tuned on the training data using various techniques (not currently included in the submission code).
- We tune one hyperparameter during inference on the test set, per planet: the magnitude of the non-mean part of the transit depth (i.e. the variation and PCA components). This is essentially a scaling applied to all underlying hyperparameters. It is found using maximum likelihood estimation, with a minimum value applied (MLE tends to estimate zero too often).
- All GPs are solved as dense GPs, except the spectral drift (trying this would lead to a 100k by 100k dense matrix); there we use KISS-GP to sparsify the GP. This works well because the shape is very low-frequent to begin with.
- There are common shapes between the transit depths per planet, corresponding to specific elements in the atmosphere. I use principal component analysis (PCA) without centering to find these shapes. We can't do this on the training labels, because the test set follows a very different distribution. So the approach is:
  - Do a rough fit on all 800 planets using the full model except the PCA shapes.
  - Do PCA on the 800 found transit depths (1 or 2 components seems best on the test set; I use 1 for the final submission).
  - Redo the fit on all 800 planets, this time including the PCA shapes we just found. This leads to the final reported transit depths.
- The ingress and egress profiles are a fixed function, found on the training data. We do fit three parameters: the width (i.e. is the ingress abrupt or broad), the ingress time, and the egress time.
- The star spectrum is an uncorrelated Gaussian per wavelength, constant over time. This is not optimal; I spent a lot of time trying to make use of the fact that different planets for one star have the same star spectrum. This worked quite well on training (+0.005), but was disastrous on test (my final score with it was 0.110).
- After all the proper Bayesian modeling, some additional fudge factors are applied. These are optimized on the training set, and an additional offset is applied to the test set (found by hill climbing). These are:
  - A fixed scaling factor applied to all confidence intervals. For the final submission, this value is around +10%. Its impact on the final score is limited.
  - A scaling factor applied to the mean of the transit depth. For the final submission, the mean is multiplied by around 1.0064. This is critical (score would be under 0.7 without), and I have no idea why it's needed...