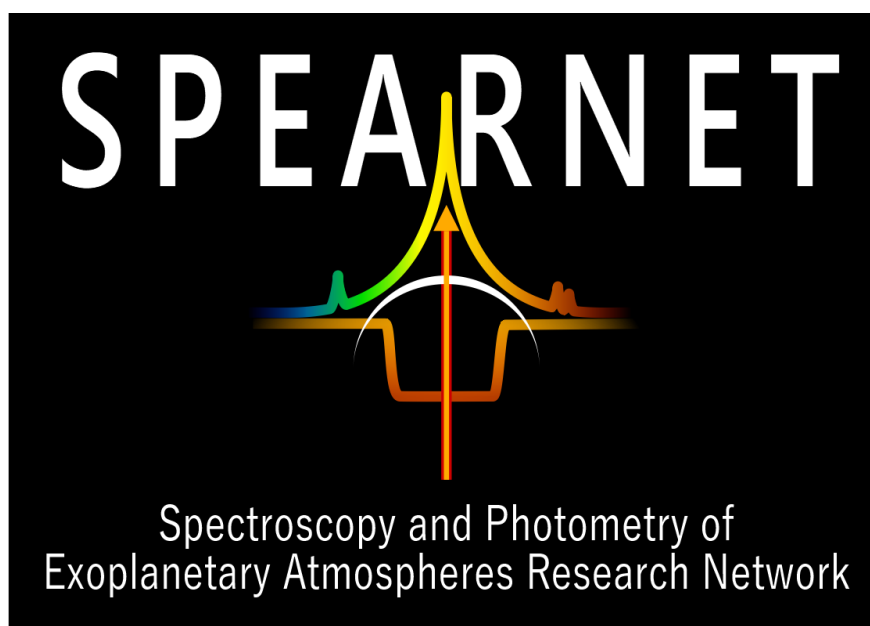


# PREFACE: An Automated Pipeline for the Selection of Transmission Spectroscopy Candidates

Development Overview and Operational Notes



Third Revision, August 2018

By  
Jake Staberg Morgan  
PhD Supervisor: Dr. Eamonn Kerins

Additional Project Members: Dr. Supachai Awiphan, Dr. Iain McDonald, Josh Hayes  
and the SPEARNet Team

# Contents

<b>Abstract</b>	<b>3</b>
<b>1 Motivation &amp; Developmental History</b>	<b>4</b>
1.1 Introduction . . . . .	4
1.2 Initial Pipeline Flow Diagram . . . . .	5
1.3 Initial Phase One Components . . . . .	6
<b>2 Subsequent Development</b>	<b>10</b>
2.1 TEPCat . . . . .	10
2.2 Telescope Data Collection & Storage . . . . .	11
2.3 The Phase One Metric . . . . .	14
2.3.1 Derivation . . . . .	14
2.3.2 Extending the Metric: Spectroscopic, Habitability & Long- Term Studies . . . . .	23
2.3.3 Planets with Known Masses . . . . .	24
2.4 Completing the Loop: Atmospheric Modelling . . . . .	24
<b>3 Using PREFACE</b>	<b>27</b>
3.1 Requirements . . . . .	27
3.2 Pipeline Architecture . . . . .	27
3.2.1 MasterShell . . . . .	27
3.2.2 Phase One . . . . .	28
3.2.3 Phase Two . . . . .	34
3.2.4 Final Outputs . . . . .	51
<b>Appendix A Initial Targets</b>	<b>53</b>
<b>Appendix B Defocused Photometry, Or How I Learned to Stop Worrying and Love 2D Gaussians</b>	<b>55</b>
<b>Appendix C How To Isolate a Fault in PREFACE Phase Two</b>	<b>57</b>
<b>BIBLIOGRAPHY</b>	<b>58</b>

## **Abstract**

This document details the development and use of the Pipeline for Ranking Exoplanets For Atmospheric CharactErization (PREFACE), built as part of Jake Staberg Morgan's PhD project as part of the SPEARNet team. The pipeline, constructed in Python, can be divided into two broad parts; Phase One applies a selection metric to choose the best targets for atmospheric study, and Phase Two calculates when these targets will be visible from a chosen location. The final output is a ranked list of transit events, which can then be used to direct observing efforts.

# Chapter 1

## Motivation & Developmental History

### 1.1 Introduction

From October 2016 to February 2017, development of a Python code to predict the transit of a given set of exoplanets from a known location on Earth was undertaken and mostly completed. By supplying co-ordinates of both observer and target, as well as the transit ephemeris, orbital period and transit duration, a set of events can be returned within a user-specified time window and ranked in order of their observability. Although this is not the first tool of its kind to be developed, it can effectively run for multiple targets and displays a clear graphical output, an example of which is shown in Figure 1.1.

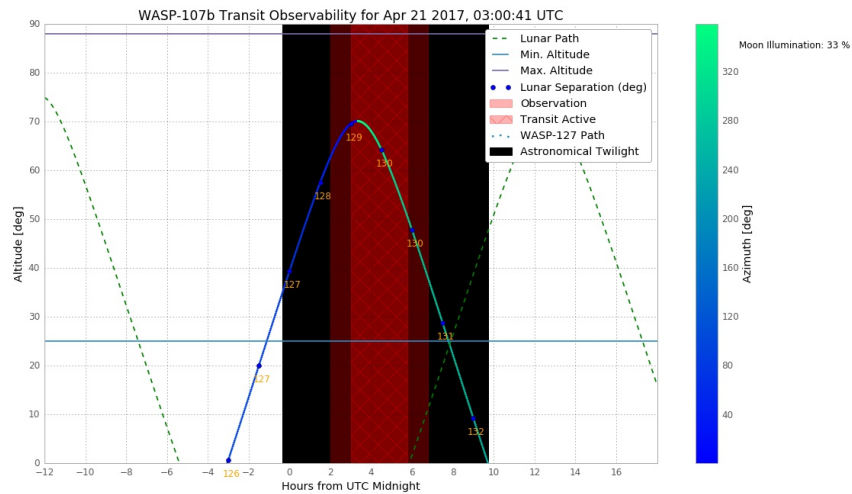


Figure 1.1: Example output of a highly-ranked event for the transiting exoplanet WASP-107b, observing from the CITO site in Chile.

This script would eventually become “Phase Two” of the completed PREFACE pipeline. Development of this phase was initially ended in Feb 2017; it worked effec-

tively and was used to support observing proposals by supplying transit dates. However, it was only capable of handling a small set of six, manually-selected exoplanets, all of which were already known to be good targets for transmission spectroscopy. These are detailed in Appendix A. However, with the upcoming launch of missions such as the *Transiting Exoplanet Survey Satellite (TESS)* [1] and the *PLANetary Transits and Oscillations of stars (PLATO)* [2], both of which are expected to discover thousands of new planets, the method of manually selecting promising targets to follow up will no longer be workable in a practical sense. In order to handle this volume of data and obtain representative target sets for statistical studies of exoplanets, a fully-automated method must be developed, capable of selecting the most promising/accessible planetary candidates from a catalogue and then determining the best observing dates from suitable locations. This candidate selection process is “Phase One”, which now feeds into the existing transit predictor already developed in a full, end-to-end set of scripts. The pipeline architecture and operations are detailed in Section 3.2.

## 1.2 Initial Pipeline Flow Diagram

The completed pipeline will have to accomplish the following:

- Take in a catalogue (of arbitrary size) of known planets, some of which will be viable for transmission spectroscopy observations.
- Employ a series of criteria (our selection metric) in order to rank the planets in terms of their spectroscopic viability and the potential for “new science” to be done.
- Cross-reference this ranked list with the catalogue of telescopes and filters available, to assign the most appropriate instruments to each target.
- With this done, pass the list of targets and telescopes to the completed Phase Two to determine the best observing dates.

The flowchart in Figure 1.2 illustrates this. The natural choice of catalogue was initially thought to be the one maintained at [exoplanet.eu](http://exoplanet.eu) [3], containing data on all known planets, but smaller catalogues/subsets could also be used, such as the planets discovered by the SuperWASP survey. However, early in the development of Phase One, the decision was taken to switch to the Transiting ExoPlanet Catalogue (TEPCat), maintained by Dr. John Southworth at Keele University. Section 2.1 explores the reasons for this. [4]

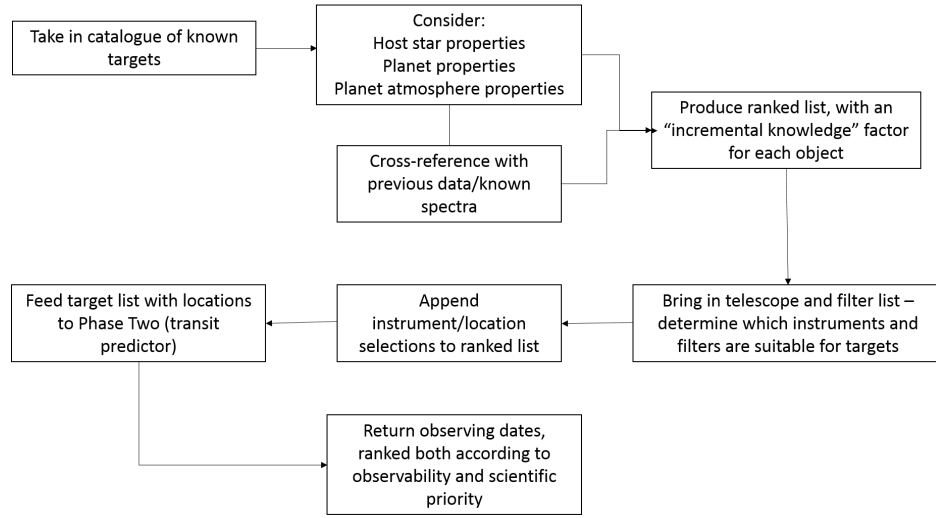


Figure 1.2: Initial flowchart for finished pipeline.

## 1.3 Initial Phase One Components

At the outset of developing Phase One, we attempted to bring together all of the various components and factors that would go into producing a final ranking. These are now described in turn:

### Host Star Properties

- **Location** - the target planet must be visible and transiting at night from a particular observing location; further to this, it must be at a high altitude during the course of an observing night, in order to minimise the air mass of the observations. Phase Two initially quantified this by means of an integration to return the area bounded by the path of the target during transit and some minimum altitude, typically  $30^\circ$  (refer to Figure 1.1.). The bigger this area is, the better the event is to observe. The subsequent evolution of this weighting is described in Section 3.2.3.
- **Magnitude** - relatively bright targets ( $V_{\text{mag}} > 12$ ) are preferred. The magnitude of the parent star will affect the final choice of instrument used; small telescopes will be unable to capture transits from fainter targets at a sufficiently high signal-to-noise ratio (SNR).
- **Stellar radius** - small stars will be preferred, as systems with a large star-planet radius ratio, expressed below, will produce a stronger (deeper) transit signal and so will have a higher SNR.

$$\text{Depth} = \left( \frac{R_p}{R_*} \right)^2 \quad (1.1)$$

This ratio can be calculated for most of the objects present in the exoplanet.eu catalogue, and so was anticipated to be useful as a rank-able parameter. Indeed, other teams have already begun to consider this parameter as a metric for useful observations, although only for small subsets of targets and not in the context of a pipeline [5] [6].

- Variability/activity - this includes such phenomena as star spots, which can mimic a transiting planet, as well as inherent stellar variability. Quiet, stable parent stars are preferred in order to avoid these possible false positives. In practice, known active stars could possibly have some kind of flag/marker attached to their data entry to caution observers.

#### Planet Properties

- Transiting planet - carrying out transmission spectroscopy relies on a planet transiting its parent star, such that star light passes through the atmospheric annulus.
- Planet radius - objects with large radii are preferred, in order to produce a large star-planet radius ratio as detailed previously.
- Orbital period - the catalogue of suitable targets will likely be biased towards large, close-in planets with short orbital periods. Short-period objects are useful, as they will transit more frequently, allowing for more opportunities to take data and a larger number of usable events (from an observational perspective).
- Transit duration - ideally, an observer will be able to capture a full transit, as well as an hour's baseline before and after the event. With this in mind, events that last for more than a few hours will be difficult to capture in their entirety in the space of one night. This could potentially be alleviated by observing from multiple sites around the world.

This is a required parameter for Phase Two to predict transit dates and times, but is not present in the exoplanets.eu catalogue. However, it can be calculated reasonably well using:

$$\Delta t = \frac{P \cdot \sqrt{R_*^2 - (a \cdot \cos(i))^2}}{\pi \cdot a}, \quad (1.2)$$

where  $\Delta t$  is the transit duration in days (also written as  $t_{14}$ ),  $P$  is the orbital period in days,  $R_*$  is the solar radius in  $R_\odot$ ,  $a$  is the semi-major axis in AU, and  $i$  is the inclination angle in degrees, which will be  $\approx 90^\circ$  for a transiting planet. This is calculable for around 600 objects in the exoplanet.eu catalogue (as of Feb. 2017); this comparatively small sample size was a key factor in transitioning to using TEPcat as the starting catalogue for PREFACE.

#### Planet Atmosphere Properties

- Scale height - this is a measure of atmospheric size; if one travels upwards through one scale height, the pressure and density decrease by a factor of 1/e. It is normally written as  $H$ , and can be derived from the ideal gas law as:

$$H = \frac{k_B \cdot T}{\mu_m \cdot g}, \quad (1.3)$$

where  $k_B$  is the Boltzmann constant,  $T$  is the atmospheric temperature,  $\mu_m$  is the mean molecular weight of the atmosphere constituents and  $g$  is the local gravity [7].

Planets with large atmospheric scale heights will produce a stronger signal (due to a large apparent growth from atmospheric absorption) when transiting, and so these should be priority targets for spectroscopic follow-up. However, the catalogues at [exoplanets.eu](http://exoplanets.eu) and [TEPCat](http://TEPCat) do not keep  $H$ -values as part of their archives, nor do they keep any information about molecular abundances, meaning  $\mu_m$ , and therefore  $H$ , cannot be immediately calculated.

- Known species/features - few planets have any known molecular species or spectral features, and the [exoplanets.eu](http://exoplanets.eu) catalogue reflects this - only a few dozen planets are listed as having any known species, and the catalogue contains no spectra [3].

If existing spectra are to be used - for comparison purposes as part of a follow-up on a target, for example, it is likely they will have to initially be sourced manually. The comparison, however, can be automated, through use of the *Exo-Transmit* program [8]. Using this, many forward model spectra can be generated for a variety of atmosphere pressures, temperatures and constituent species. If these were then stacked against each other and/or existing data, useful future observations could then be identified, these being data that would allow us to rule out/discard multiple hypothetical models, or to support an existing one. Such observations would warrant a high “incremental knowledge” factor. This possibility is discussed further in Section 2.4.

A good example of this would be the transitional “super-Neptune” WASP-107b; if it is truly a gas giant, its metallicity should be low. Model spectra could then be generated in advance and compared to observations once they are made, from which the atmospheric metallicity can be estimated and individual species identified. This was recently done successfully with the *Hubble Space Telescope* (HST) [9], which in turn helped constrain the parameters which go into producing these forward models.

#### Telescope/Location Properties

- Location - this can either be specified through astropy’s coordinates sub-module, or can be supplied manually if the desired location is not in the `EarthLocation` library. If the latter is used, latitude, longitude and altitude must be specified in



degrees, using the DD:MM:SS format, with North and East taken to be positive. The choice of observing location will affect the visibility of a given target; targets with a high declination will not be easily observable from southern skies, for instance.

One possibility for determining the best location would be to run Phase Two for a given target from several possible observing locations over the course of a year. By determining the number and quality of usable events observable from each site (and thus, the number of observing opportunities/quantity of data potentially collected), we can get a measure of how viable that target is from various different sites. However, the best observing sites will not necessarily have the most suitable instruments/filters for that target, so this is only one factor that must be considered. Additionally, while within the capabilities of the code, this process would be computationally expensive even for a single planet.

Different locations around the world will also have different weather patterns, which will affect observing throughout the year. However, as inclement weather would affect all possible targets equally, quantifying the effects of weather on particular sites should be considered a lower-priority component of the pipeline.

- Telescope/detector specifications - among the factors to be considered here are instrument aperture, the quantum efficiency of the instrument's CCD or similar, and telescope overheads. Where possible, exposure-time calculators should be sought out and used beforehand to determine the optimum exposure time for a particular set-up. Ideally, the output of such a calculator could be stored, perhaps called as a .csv, to reduce the pipeline's reliance on external programs/applications.
- Filters available - to perform transmission spectroscopy, a variety of filters must be used in order to probe how the transit depth changes with wavelength. This information could also be stored in a telescope .csv. (Such a .csv has now been developed, as discussed in Section 2.2.)

# Chapter 2

## Subsequent Development

This chapter discusses key developments and improvements to the pipeline as a whole, following from the initial plans and expectations laid out previously.

### 2.1 TEPCat

As mentioned in Section 1.3, PREFACE now uses the TEPCat catalogue for its initial input. As of mid-July 2018, it contains 1522 transiting planets usable by PREFACE, with extensive literature data compiled on each. The process of adding new entries/data into the catalogue is one of manual inspection, but the construction is automated through an IDL pipeline, and a detailed changelog is provided on the TEPCat website. [4]

The driving reason for the transition was the fact that exoplanets.eu does not keep the transit duration  $t_{14}$  as part of its catalogue, a necessary component to predict transit times and visibilities. In theory, this is calculable using Eq. 1.3, but the catalogue only held inclination values for 606 of the 3570 planets in the catalogue at the time of development, artificially and severely restricting the population of objects available to us. TEPCat does not have these restrictions; all targets in the catalogue are known transients, with transit durations from the literature already supplied for each target. This naturally helps to ensure an accurate prediction for any given event.

Other useful quantities and information given in TEPCat include:

- Semi-major axes in AU, at least for “well-studied planets” (600+ objects). For cases where this is not given, it can be calculated reasonably well using Kepler’s Third Law.
- Literature equilibrium temperatures ( $T_{\text{eq}}$ ) for a significant fraction of the catalogue, with this quantity being calculable if it is not initially present.
- Literature references - three of these are given; one for the discovery paper, the most recent one concerning a particular planet, and the paper from which

the current transit ephemeris is drawn. These can be used to gain some initial insight as to how well the planet has been studied; if all three of these references are identical, then only that one reference is available (the discovery paper), and so this object is likely to be a “high-ignorance” planet (see Section 2.4).

- Co-ordinates - RA in HH:MM:SS and Dec in DD:MM:SS. This eliminates an internet-dependency from the code, as the initial build of Phase Two called target co-ordinates from the SIMBAD astronomical database [10], which often resulted in a failed run if the connection was unstable.
- Near-complete lists of  $V$  and  $K$ -band magnitudes of the parent stars ( $V$ -band is complete,  $K$ -band is complete for all but two targets.)
- Transit depths in %.
- Transit times (ephemerides) in HJD/BJD. This is another critical weakness of the exoplanets.eu catalogue; at the time of development, only 568 transit times were available, restricting the subset of usable targets still further.

One parameter missing from this list that eventually became important was the impact parameter,  $b$ . Hence, alternative sources had to be used to recover this for our target set, as described in Section 3.2.2.

In order to qualify for inclusion in the TEPcat database, a target must meet the following criteria, quoted from the TEPcat website:

- A study of the system has been published in one of the refereed journal papers checked by John. These are A&A, AJ, ApJ, ApJS, MNRAS, Nature, PASP and the arXiv preprint server. [4]
- The full sky position and orbital ephemerides are available, allowing people to perform follow-up observations.
- The mass or radius of the putative planet must be within the planetary regime. Hence, objects which are not definitively within these regimes, such as Kepler-1625b, are excluded. [11]
- The physical properties of the planet must be adequately measured with a reasonable error analysis.
- It must be orbiting a “normal” star.

## 2.2 Telescope Data Collection & Storage

One part of the wider pipeline development has been the process of collating information about the capabilities of the various instruments we either have access to, or

may request time on in the future, and storing it such that the pipeline can read and interpret it. This eventually grew into one of PREFACE's "core files", Scope.csv. The information on this bank of telescopes is widely scattered across various sources and sometimes contradictory, so it has become necessary to bring it all into one place as the table has become populated and the project has grown in complexity.

Scope.csv uses the following columns:

- **Telescope** - An appropriate name for the instrument, used for pipeline reference.
- **Lat** and **Long** - the latitude and longitude of the telescope, given in decimal degrees. North and East are taken to be positive. (A converter is hosted at [12].) This eliminates another internet dependency; by storing these co-ordinates locally, it is no longer necessary for astropy to call them from its online library.
- **Alt** - The altitude of the telescope, in metres.
- **mzp<sub>-</sub>** - These are zero-point magnitudes in the filters stated; an object of this magnitude will produce one detected photo-electron (not ADU count!) per second. If the instrument does not have a certain filter (eg. PROMPT-8 has no Sloan filters), the zero-point will be displayed internally as NaN.
- **m.5 $\sigma$**  - For spectrograph setups only. This is the limiting broad-band magnitude calculated for a point source of zero colour (an A0V star) which would give an SNR of 5 in one hour with dark sky, clear conditions and a seeing FWHM of  $\Theta_{\text{see}}$  [13]. Target exposure times and a zero-point can then be recovered from this. This parameter is only used by spectrographic set-ups.
- **msky<sub>-</sub>** - the sky background in mag/arcsec<sup>2</sup>. In an aperture of area 1 arcsec<sup>2</sup>, the observer will detect photons equivalent to a star of this brightness. This parameter is band-dependent, and becomes particularly important for near and mid-IR observations, owing to the effects of water bands.
- **Res** - The spectral resolution R, calculated from:

$$R = \frac{\text{Lambda\_Cent}}{\Delta\lambda}, \quad (2.1)$$

where  $\Delta\lambda$  is the resolution element (effectively the bin size). R will be unity for broad-band filters. For spectrographs, this number will be quoted as being much greater, eg. the FORS2 spectrograph has  $R = 1000$ . In practice, these channels will be binned together to boost the SNR and alleviate seeing effects.

- **Lambda\_Cent** - The central wavelength (in ångströms) of the spectrographic grism.
- **Lambda\_Range** - The width (in ångströms) of the spectrographic grism. 200 is a NaN value.

- **Overhead** - This is the read-out time in seconds between exposures. This number does not include other overheads such as slewing to target or changing filters - see the documentation for the individual telescope for these, if applicable. For fast-readout cameras such as ULTRASPEC, this value is approximately zero.
- **Theta\_see** - The average seeing in arcseconds from the observing site. (Of course, this number may vary on the night!) This is taken using the FWHM (diameter) of a Gaussian PSF.
- **Theta\_DF** - This is the seeing diameter to be used for defocused operations. Defocusing is an important strategy to bring exposure times up while avoiding saturation.
- **(FoV\_Rad)** - the radius of the field of view in arcminutes. Simbad queries around a target using a circle of this radius, necessary to find comparison stars.
- **Detector\_Size** - The dimensions of the detector in pixels.
- **Pixel\_Scale** - this is the length of one side of a single (square) CCD pixel in arcseconds. If not explicitly stated, it can be recovered from:

$$\text{Pixel Scale} = \left( \frac{\text{Detector FoV (")}}{\# \text{ of pixels along 1 dimension}} \right) \quad (2.2)$$

- **Omega\_pix** - This is the area of 1 pixel in arcseconds square, given by:

$$\Omega_{pix} = \left( \frac{\text{Detector FoV (")}}{\# \text{ of pixels along 1 dimension}} \right)^2 = (\text{Pixel Scale})^2 \quad (2.3)$$

Note that this value will be constant even for non-square detectors like that in SPRAT, mounted on the Liverpool Telescope.

- **Gain** - This is a factor in converting from raw photo-electrons to counts registered (and eventually used by imaging software). The relation is:

$$\text{Counts (ADU)} = \frac{\text{Photo-electrons}}{\text{Gain}}, \quad (2.4)$$

assuming a CCD of 100% quantum efficiency (every photon results in a photo-electron).

- **Half\_Well** This is 50% of the depth of one CCD well, in electrons. Each well can only hold a fixed amount of charge (typically  $2^{16} = 65536 \text{ e}^-$ ), after which it will become saturated. However, the response curve of a CCD will often become non-linear in the run-up to saturation, so observers must keep to the linear regime wherever possible. This optimum CCD regime dictates the exposure time needed for a particular instrument and target, as detailed in the next section.

A .pdf exploring and citing all of the values in Scope.csv has also been made, and will continue to be updated as the file becomes populated with more instruments.

## 2.3 The Phase One Metric

Central to the working of PREFACE is the metric which carries out the ranking of individual targets. Its form is as follows:

$$\mathcal{D} \propto C_T(\lambda) 10^{-0.2m_*(\lambda)} t_{14}^{1/2} T_{\text{eq}} \frac{R_p^{1-n}}{0.8R_J} \delta. \quad (2.5)$$

### 2.3.1 Derivation

#### Metric Signal

Our initial discussions on which factors should go into the metric yielded four for initial inclusion:

- $t_{14}$  - the transit duration, given in days as part of the TEPcat catalogue, but converted to seconds to give consistent units in the calculation. A longer transit time will allow more exposures (and thus, more data) to be taken.
- The number of photons received per unit time  $N_{\text{ph}}$ , where:

$$N_{\text{ph}} = 10^{0.4(m_{\text{zp}} - m_*)}, \quad (2.6)$$

where  $m_*$  and  $m_{\text{zp}}$  are the apparent and zero-point magnitudes of the parent star in the desired waveband respectively. The choice of waveband will depend on what filters/grisms are available for the chosen instrument, what possible atmospheric features are accessible to each and which (if any) of these features have been previously studied.

- $T_{\text{eq}}$  - the equilibrium temperature of the planet. Hot planets are more likely to have bloated/tenuous atmospheric envelopes, which are easier to detect and study through transmission spectroscopy.
- The depth of the transit,  $\delta$ . Taking the star and planet as perfectly round disks and neglecting limb darkening, this is simply modelled as:

$$\delta \propto \left( \frac{R_p}{R_*} \right)^2, \quad (2.7)$$

but is also stated directly in the TEPcat database. The transit must be sufficiently deep to be detectable by the chosen instrument, with a larger depth providing a more favourable signal-to-noise ratio (hereafter abbreviated as SNR).

Hence, our initial ranking metric took the following form:

$$\text{Rank} = t_{14}^{\alpha} \cdot S^{\beta} \cdot T_{\text{eq}}^{\gamma} \cdot \delta^d, \quad (2.8)$$

where  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $d$  were weighting factors to be determined. These must be chosen carefully in order to avoid any one factor dominating the others without physical justification. In order to determine these, we consider a typical transit curve such as the one in Figure 2.1. It is important that transit light curves are well-sampled, particularly when conducting transmission photometry across different filters; any waveband-dependent changes in scale height will change the transit ingress and egress times of an individual event.

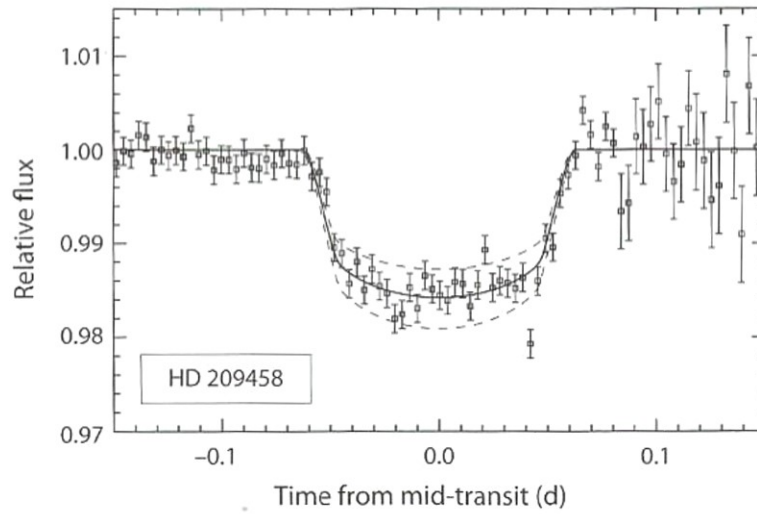


Figure 2.1: Example transit curve of the exoplanet HD 209458. This particular curve is well-sampled along the transit duration, with strong baselines either side of the transit. [14]

How effectively can we sample a transit curve as observers? The number of exposures (data points) captured over the duration of a transit is given by:

$$N_{\text{pts}} = \frac{t_{14}}{t_c}, \quad (2.9)$$

where  $t_c$  is the cadence time, given as:

$$t_c = t_{\text{exp}} + t_{\text{over}}, \quad (2.10)$$

where  $t_{\text{exp}}$  is the exposure time used to capture the target and  $t_{\text{over}}$  is the instrument overhead time between exposures (both in seconds). For fast-readout cameras such as ULTRASPEC on the Thai National Telescope (TNT) [15], the readout time between each exposure is negligible, and so  $t_{\text{over}} \approx 0$ . Conversely, for an instrument using a conventional CCD, such as the PROMPT-8 telescope, the readout time may be dominant compared to the exposure time. We can see that  $t_{\text{exp}}$  will also be instrument-dependent,

in that bright stars can be imaged with shorter exposures (and indeed, longer exposures may not be possible due to CCD saturation).

A standard method of calculating/choosing telescope exposure times had to be found, such that the rank outputs of different instruments could be meaningfully compared to each other. This was initially done for a fixed SNR, but then shifted to a more instrument-dependent method, where  $t_{\text{exp}}$  was chosen such that 50% of a CCD well (associated with a single pixel) would be filled during an exposure, assuming an evenly-illuminated detector chip. A detector with a large well would take longer to fill, but all other things being equal, its SNR would be commensurately higher, which would be reflected in its rank score for a particular target. Additionally, if the target flux is distributed over many pixels, particularly if defocusing is employed, each individual one will take longer to fill and thus the required exposure time will be longer. For broad-band filters, such as those used on our suite of small and medium telescopes, the optimum exposure time is:

$$t_{\text{exp}} = \left( \frac{\pi \cdot \left( \frac{\theta_{\text{see}}}{2} \right)^2}{\Omega_{\text{pix}}} \right) \cdot N_{\text{HWD}} \cdot 10^{-0.4 \cdot (m_{\text{zp}} - m_*)}, \quad (2.11)$$

where  $\theta_{\text{see}}$  and  $\Omega_{\text{pix}}$  are as defined in Section 2.2, and  $N_{\text{HWD}}$  is the half-well depth in electrons. Note that we take  $\theta_{\text{see}}/2$ , as it is a diameter. We can see that  $t_{\text{exp}}$  will be shorter for bright stars and/or focused observations, as we expect.

The first term of Eq. 2.11 describes how many pixels the target star occupies, assuming a Gaussian point spread function, or PSF. For a detector with small pixels, poor seeing conditions and/or defocusing, this term will be large, meaning an individual CCD well will take longer to fill. This effect can actually be advantageous for very bright targets; most systems have a minimum exposure time, and bright stars may saturate at or below this limit. At the request of the referee reporting on the paper describing the metric [24], defocusing can now be optionally toggled in the pipeline.

Extending Eq. 2.11 to spectrographs posed a challenge, as these will not produce a Gaussian PSF on a detector, but rather a whole spectrum along the detector's length. Recovering the zero-point magnitude for the spectrograph as a whole is not meaningful either, as the incoming flux is divided between some number of channels, affording the spectrograph its resolution. For such set-ups, the following formula is used:

$$t_{\text{exp}} = \frac{N_{\text{HWD}}}{25} \cdot 3600 \cdot 10^{-0.4 \cdot (m_{5\sigma} - m_*)} \quad (2.12)$$

The factor of 25 arises from the SNR of 5 corresponding to a 5- $\sigma$  mag-limit (giving 25photons/pixel/hour, for all channels), and the factor of 3600 is from the exposure time of 1 hour needed for the 5- $\sigma$  limit. The optimum exposure times for spectrographs are typically orders of magnitude greater than those of fast-readout cameras using broad filters (due to flux being distributed between many channels), but the increased spectral resolution and ability to immediately recover a complete transmission spectrum more than compensate for this. [9]



Overall, the characterisation of a transit curve will improve as  $\sqrt{N_{\text{pts}}}$ ; for a Poisson distribution, the standard deviation is simply the square root of the average number of events. Similarly, the SNR will improve as  $\sqrt{S}$ , assuming only photon noise (which follows a Poisson distribution). Hence,

$$\begin{aligned} N_{\text{Poisson}} &= 10^{0.2(m_{\text{zp}} - m_*)}, \\ \text{SNR} &= (\text{Signal})^{1/2} = (t_{\text{exp}} \cdot 10^{0.4(m_{\text{zp}} - m_*)})^{1/2}, \end{aligned} \quad (2.13)$$

As stated previously, if  $m_{\text{zp}}$  is not known from literature, it can be calculated using an out-of-transit frame from:

$$\text{Counts} = t_{\text{exp}} \cdot 10^{0.4(m_{\text{zp}} - m_*)}, \quad (2.14)$$

which can be rearranged to give:

$$2.5 \log_{10} \left( \frac{\text{Counts}}{t_{\text{exp}}} \right) + m_* = m_{\text{zp}}, \quad (2.15)$$

where the count number is the number of photons received from the target star. This is displayed as an ADU (Analogue-to-Digital) count in software, where:

$$\text{Counts (ADU)} = \frac{\text{Photo-electrons}}{G}, \quad (2.16)$$

where  $G$  is the gain of the instrument/CCD, with units of  $e^-/\text{ADU}$ .

Overall, this framework argues for both  $\alpha$  and  $\beta$  being  $1/2$ , and also argues for the inclusion of a factor of  $t_{\text{exp}}$  to be included in the numerator. To see where this comes from, we must consider Eqs. 2.6 and 2.13, which are both over some unit time. Over a single exposure of duration  $t_{\text{exp}}$ , the total number of photons collected,  $N_{\text{ph}}$ , will be:

$$N_{\text{ph}} = t_{\text{exp}} \cdot 10^{0.4(m_{\text{zp}} - m_*)}. \quad (2.17)$$

Taking  $N$  exposures over an entire transit, this total number becomes:

$$N_* = N_{\text{ph}} \cdot N_{\text{pts}} = \frac{t_{14} \cdot t_{\text{exp}} \cdot 10^{0.4(m_{\text{zp}} - m_*)}}{t_c}. \quad (2.18)$$

By including  $t_{\text{exp}}$  and  $t_{\text{over}}$  in the denominator (as  $t_c$ ), our ranking now has telescope-dependency; instruments able to take useful short exposures with minimum overhead (and thus, high operational efficiency) are now preferred. In the limit of  $t_{\text{over}} = 0$ , the factors of  $t_{\text{exp}}$  cancel, yielding an efficiency of 1 (100%) and a ranking metric independent of  $t_{\text{exp}}$ . Hence, the observation cadence and SNR (quantities returned separately by the pipeline) will still be important to consider, in order to determine optimal observing strategies. In this limiting case, the metric itself is blind to the observing strategy used; it only cares about efficiency and the number of photons captured.

The actual strength of our transit signal will be not the number of photons received, but the number of photons *blocked* by the transiting planet/atmosphere. For a standard

transit of a planet with no atmosphere, this fraction of photons blocked would simply be given by the transit depth  $D$ , with the remaining unblocked photons can then be considered to be our noise. However, for transmission spectroscopy, we are specifically interested in the photons blocked by the planet atmosphere; it is this wavelength-dependent absorption which gives us insight into its composition. We can construct a simple model of an annulus around the planetary disk with a non-negligible area.

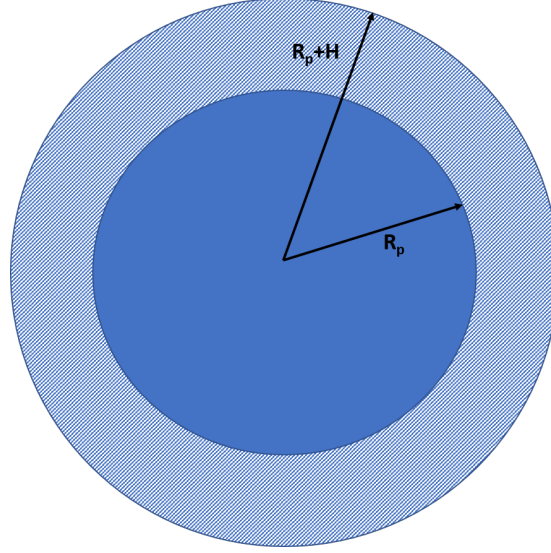


Figure 2.2: Illustration of a planet (solid blue disc) and atmosphere (blue hatched area).

This is illustrated in Figure 2.2, where  $H$  is the scale height of the atmosphere. This is a simple approximation based on treating the atmospheric annulus as consisting of an ideal gas, at approximately equilibrium temperature:

$$\begin{aligned}
 PV &= Nk_B T_{\text{atm}}, \\
 PV &= \frac{Mk_B T_{\text{atm}}}{\mu_m} \simeq \frac{Mk_B T_{\text{eq}}}{\mu_m}, \\
 \rho g H \frac{M}{\rho} &\simeq \frac{Mk_B T_{\text{eq}}}{\mu_m}, \\
 H &\simeq \frac{k_B T_{\text{eq}}}{\mu_m g}.
 \end{aligned} \tag{2.19}$$

We see that  $H \propto T_{\text{eq}}$ , so  $\gamma = 1$ ; the strength of the atmospheric signal increases linearly with temperature.  $H$  is also dependent on the planet surface gravity  $g$ , but this is only listed for a few hundred objects in the TEPcat catalogue. However, we can still explore its effects on a planet's atmosphere. From Newtonian dynamics, and for scale heights much smaller than the planet radius  $R_p$ , we can say:

$$\begin{aligned}\delta &\propto \left(\frac{R_p}{R_*}\right)^2, \\ g &= \frac{GM_p}{R_p^2} \propto \delta^{-1}.\end{aligned}\tag{2.20}$$

We know that  $H \propto g^{-1}$ , therefore  $H \propto R_p^2$ ; a large (and more massive) planet is capable of hosting a larger atmosphere, and will produce a deeper transit. This in turn argues for  $d = 1$ . However, this comes with some serious caveats; the mass-radius relation for exoplanets is poorly constrained, with dependencies on bulk composition, planet age, tidal effects and (particularly for hot planets) irradiation by the host star. Indeed, for close-in hot Jupiters, this relation breaks completely, with planet radius dictated entirely by the extreme environment in which it resides. Additionally, the mass of the planet is not always known; this cannot be determined from transits alone, and typically needs a radial velocity (RV) follow-up to provide this. Hence, we are only able to reasonably say:

$$H \propto T_{\text{eq}} \cdot R_p^{2-n},\tag{2.21}$$

where  $n$  is some power describing the mass-radius relation,

$$n = \begin{cases} 0 & (R_p > 0.8R_J) \\ 2 & (R_p < 0.8R_J). \end{cases}\tag{2.22}$$

A two-part mass-radius relation is necessary to adequately describe both bloated Jupiters and smaller worlds such as super-Earths. Previously, only  $n=0$  (for large planets) was used, but this resulted in some well-studied smaller planets such as GJ 1214b being excluded. A two-part relation holds to the scores from known masses reasonably well, as shown in Figure 2.3, albeit with significant scatter. Hence, it is as good a starting point as can be made, in the absence of known masses. A factor of  $1/0.8R_J$  is also required in order to prevent a discontinuity at  $0.8R_J$ .

The approximation made here has other consequences, as we will see later. In any case, we are interested specifically in atmospheric effects, rather than simply the transit itself, and so we must return to Figure 2.2 and our annulus calculation:

$$\begin{aligned}A_{\text{ann}} &= \pi(R_p + H)^2 - \pi R_p^2, \\ &= \pi(R_p^2 + 2R_p H + H^2) - \pi R_p^2, \\ &= \pi(2R_p H + H^2)\end{aligned}\tag{2.23}$$

In the limit of  $R_p \gg H$ , we recover:

$$A_{\text{ann}} \propto 2R_p H\tag{2.24}$$

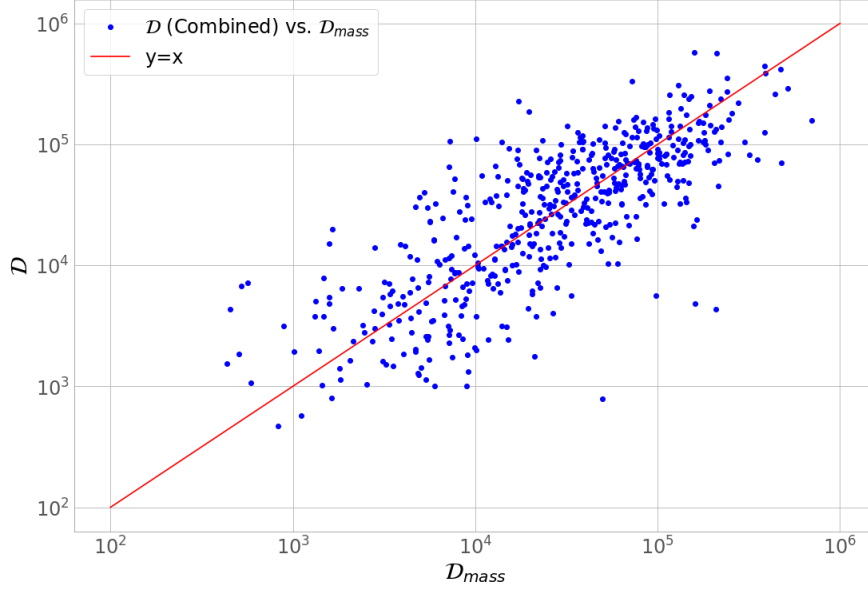


Figure 2.3:  $\mathcal{D}$  vs.  $\mathcal{D}_{\text{mass}}$  plot, using all planets with known masses and both  $n = 0$  and  $n = 2$ . A perfect relation would have the blue points trace the red line exactly ( $\mathcal{D} = \mathcal{D}_{\text{mass}}$ ).

Hence, our atmospheric signal per exposure goes as:

$$\begin{aligned}
 S &= N_* \cdot \frac{A_{\text{ann}}}{R_*^2} \\
 &= t_{\text{exp}} \cdot 10^{0.4(m_{\text{zp}} - m_*)} \cdot \frac{2R_p H}{R_*^2} \\
 &= t_{\text{exp}} \cdot 10^{0.4(m_{\text{zp}} - m_*)} \cdot \frac{2\delta H}{R_p} \\
 &\propto 10^{0.4(m_{\text{zp}}(\lambda) - m_*(\lambda))} t_{\text{exp}} T_{\text{eq}} \frac{R_p^{1-n}}{0.8R_J} \delta.
 \end{aligned} \tag{2.25}$$

Neglecting limb darkening, our Poisson noise (from host flux not blocked during transit) during a single exposure will subsequently be:

$$\varepsilon_* = [n_*(1 - \delta)]^{1/2} \propto 10^{0.2(m_{\text{zp}} - m_*)} t_{\text{exp}}^{1/2} (1 - \delta)^{1/2}. \tag{2.26}$$

Our signal-to-noise ratio per exposure will then be

$$S/N_* \equiv S_*/\varepsilon_* \propto 10^{0.2(m_{\text{zp}} - m_*)} t_{\text{exp}}^{1/2} T_{\text{eq}} \frac{R_p^{1-n} \delta}{0.8R_J(1 - \delta)^{1/2}}. \tag{2.27}$$

As we have already seen, the ability to fit a transit model will depend not just on the quality of each observation but also on the number of exposures obtained during the transit. The total signal-to-noise ratio summed over all  $n_{\text{exp}}$  exposures will be

$$\begin{aligned} \mathcal{S}/\mathcal{N}_{*,\text{tot}} &= (\mathcal{S}/\mathcal{N}_*)n_{\text{exp}}^{1/2} = \mathcal{S}/\mathcal{N}_* \left( \frac{t_{14}}{t_{\text{exp}} + t_{\text{over}}} \right)^{1/2} \\ &\propto 10^{0.2(m_{\text{zp}} - m_*)} \frac{T_{\text{eq}} R_{\text{p}}^{1-n} \delta}{0.8 R_{\text{J}} (1 - \delta)^{1/2}} \left( \frac{t_{14} t_{\text{exp}}}{t_{\text{exp}} + t_{\text{over}}} \right)^{1/2}. \end{aligned} \quad (2.28)$$

In the regime of  $R_{\text{p}} H \ll R_*^2$ , we can safely approximate  $(1 - \delta)^{-1/2}$  as unity, though this factor may remain relevant for scenarios with hot-Jupiters around M-dwarfs, or in instances where the planet is undergoing atmospheric escape.

We can now define our detectability metric,  $\mathcal{D} \propto \mathcal{S}/\mathcal{N}_*$ , based upon the expected signal-to-noise contribution from an opaque atmosphere:

$$\mathcal{D} = C_{\text{T}} 10^{-0.2m_*} t_{14}^{1/2} T_{\text{eq}} \frac{R_{\text{p}}^{1-n}}{0.8 R_{\text{J}}} \delta \quad (R_{\text{p}} \ll R_*), \quad (2.29)$$

where  $C_{\text{T}}$  contains telescope-dependent factors:

$$C_{\text{T}} = N_{\lambda}^{-1/2} 10^{0.2m_{\text{zp}}} \left( \frac{t_{\text{exp}}}{t_{\text{exp}} + t_{\text{over}}} \right)^{1/2}. \quad (2.30)$$

The factor  $N_{\lambda}$  specifies the number of spectral bins spanning the observed wavelength range. It is included to allow both for photometric observations, where  $N_{\lambda} = 1$ , or for spectroscopic observations where  $N_{\lambda} \gg 1$ .

### Additional Noise Sources

In addition to Eq. 2.17, an exposure of some duration will also capture photons from the sky background. This will comprise a second noise contribution of

$$N_{\text{sky}} = 10^{0.4(m_{\text{zp}}(\lambda) - m_{\text{sky}}(\lambda))} t_{\text{exp}}, \quad (2.31)$$

where  $m_{\text{sky}}(\lambda)$  is the area-dependent apparent magnitude, given by

$$m_{\text{sky}}(\lambda) = -2.5 \log \left( \frac{\pi \theta_{\text{see}}^2}{4} \right) + \mu_{\text{sky}}(\lambda), \quad (2.32)$$

where  $\mu_{\text{sky}}(\lambda)$  is the sky background in units of mag/arcsec<sup>2</sup>. Both of these parameters will depend on the chosen observing site.

Hence, the total error on a single exposure taken during transit will be

$$\epsilon_{\text{exp}} = \sqrt{N_* + N_{\text{sky}}}. \quad (2.33)$$

Extending to all observations in transit, the total time-integrated error on the transit exposures is

$$\varepsilon_{\text{trans}} = \varepsilon_{\text{exp}} \sqrt{\frac{t_{14}}{(t_{\text{exp}} + t_{\text{over}})}}, \quad (2.34)$$

Eq. 2.33 also describes the Poisson noise in an exposure taken out of transit to form baselines, essential for recovering the depth of a transit curve. For a single observation, this will be of duration  $t_{\text{base}}$  in units of seconds. In this paper, these are composed of an hour before the transit begins and an hour after it ends. Defining  $N_{\text{obs}}$  as the number of exposures taken as part of the baseline observations, the total error in fitting between the transit depth and baseline level will then be

$$\varepsilon_{\text{base}} = \frac{\varepsilon_{\text{exp}}}{\sqrt{N_{\text{obs}}}} \quad (2.35)$$

$$= \varepsilon_{\text{exp}} \sqrt{\frac{(t_{\text{exp}} + t_{\text{over}})}{t_{\text{base}}}}. \quad (2.36)$$

It is important to note that Eq. 2.36 scales as  $1/\sqrt{N_{\text{obs}}}$ ; more out-of-transit observations will drive down the error on the mean baseline level, improving the accuracy of the fit. Conversely, Eq. 2.34 scales linearly with  $N_{\text{obs}}$ ; more noise is collected with more observations, but this itself scales as  $\sqrt{S}$ , and so the quality of the light curve is improved.

The total error on the light curve will be

$$\varepsilon_{\text{tot}} = \sqrt{\varepsilon_{\text{trans}}^2 + \varepsilon_{\text{base}}^2} \quad (2.37)$$

$$\begin{aligned} &= \sqrt{\left( \varepsilon_{\text{exp}} \sqrt{\frac{t_{14}}{(t_{\text{exp}} + t_{\text{over}})}} \right)^2 + \left( \varepsilon_{\text{exp}} \sqrt{\frac{(t_{\text{exp}} + t_{\text{over}})}{t_{\text{base}}}} \right)^2} \\ &= \sqrt{(N_* + N_{\text{sky}}) \left( \frac{t_{14}}{t_{\text{exp}} + t_{\text{over}}} + \frac{t_{\text{exp}} + t_{\text{over}}}{t_{\text{base}}} \right)} \\ &= \left[ t_{\text{exp}} \left( 10^{0.4(m_{\text{zp}}(\lambda) - m_*(\lambda))} + 10^{0.4(m_{\text{zp}}(\lambda) - m_{\text{sky}}(\lambda))} \right) \right]^{1/2} \\ &\quad \left( \frac{t_{14}}{t_{\text{exp}} + t_{\text{over}}} + \frac{t_{\text{exp}} + t_{\text{over}}}{t_{\text{base}}} \right)^{1/2}. \end{aligned} \quad (2.38)$$

The ratio of Eq. (2.25) and (2.38) gives

$$\begin{aligned} \mathcal{S}/\mathcal{N} &\propto \frac{2(10^{(0.2m_{\text{zp}}(\lambda) - 0.4m_*(\lambda))})}{10^{-0.2m_*(\lambda)} + 10^{-0.2m_{\text{sky}}(\lambda)}} t_{14} \\ &\quad \left[ \frac{t_{\text{exp}} t_{\text{base}}}{(t_{\text{exp}} + t_{\text{over}})(t_{14} t_{\text{base}} + (t_{\text{exp}} + t_{\text{over}})^2)} \right]^{1/2} T_{\text{eq}} \frac{R_p^{1-n}}{0.8R_J} \delta. \end{aligned} \quad (2.39)$$

By re-casting Eq. (2.39), our new metric is

$$\mathcal{D} \propto C_T(\lambda) \frac{10^{-0.4m_*(\lambda)}}{10^{-0.2m_*(\lambda)} + 10^{-0.2m_{\text{sky}}(\lambda)}} t_{14} T_{\text{eq}} \frac{R_p^{1-n}}{0.8R_J} \delta. \quad (2.40)$$

As before, the term  $C_T(\lambda)$  contains all of the telescope-dependent factors,

$$C_T(\lambda) = N_\lambda^{-1/2} 10^{0.2m_{zp}(\lambda)} \left[ \frac{t_{\text{exp}} t_{\text{base}}}{(t_{\text{exp}} + t_{\text{over}})(t_{14} t_{\text{base}} + (t_{\text{exp}} + t_{\text{over}})^2)} \right]^{1/2}. \quad (2.41)$$

As a final check, in the limit of long baselines (second term of Eq. 2.38 tends to 0) and negligible sky backgrounds ( $10^{-0.2m_{\text{sky}}} \approx 0$ ), Eqs. 2.40 and 2.41 reduce to

$$\mathcal{D} \approx C_T(\lambda) 10^{-0.2m_*(\lambda)} t_{14}^{1/2} T_{\text{eq}} \frac{R_p^{1-n}}{0.8R_J} \delta, \quad (R_p \ll R_*) \quad (2.42)$$

$$C_T(\lambda) = N_\lambda^{-1/2} 10^{0.2m_{zp}(\lambda)} \left( \frac{t_{\text{exp}}}{t_{\text{exp}} + t_{\text{over}}} \right)^{1/2}. \quad (2.43)$$

We demonstrate in the metric paper that both of those assumptions are valid, and that Eq. 2.5 is valid for general use. However, the software pipeline also contains Eq. 2.40 to use if the inclusion of these noise sources is desired.

### 2.3.2 Extending the Metric: Spectroscopic, Habitability & Long-Term Studies

Eq. 2.5 is actually the first of 4 versions of the metric available to PREFACE. This first version only considers observations made over the course of a single transit. However, it is common practice for photometric observers to capture as many transit events for a planet as possible, and then fold the light curves together to boost the overall SNR. For such a campaign, short-period planets will be preferred, which necessitates an extension to our metric for multiple transits:

$$\mathcal{D}_{\text{multi}} = \mathcal{D} P^{-1/2} \quad (2.44)$$

where  $P$  is the orbital period in days.

Another useful modification is to consider using the metric for habitability studies, i.e. targeting only planets where liquid water could exist on the planet surface. This is only possible for a comparatively narrow range of temperatures, and so

$$\mathcal{D}_{\text{habit}} = \mathcal{D} \quad (T_{\text{min,hab}} \leq T_{\text{eq}} \leq T_{\text{max,hab}}), \quad (2.45)$$

We can then combine the two constraints used in Eqs. 2.44 and 2.45 to yield:

$$\mathcal{D}_{\text{multi.habit}} = \mathcal{D}_{\text{habit}} (P)^{-1/2} \quad (T_{\text{min,hab}} \leq T_{\text{eq}} \leq T_{\text{max,hab}}). \quad (2.46)$$

### 2.3.3 Planets with Known Masses

The mass of a given transiting planet must be recovered from follow-up observations. As more such radial velocity observations are undertaken, the subset of transiting planets with known masses will become large, to the point where statistically significant samples can be recovered without the need for a mass substitution. We can return to Eqs. (2.20) & (2.21) and apply our scaling for the surface gravity  $g$  using the mass directly, yielding

$$S \propto 2 \times 10^{0.4(m_{zp}(\lambda) - m_*(\lambda))} t_{14} \left( \frac{t_{\text{exp}}}{t_{\text{exp}} + t_{\text{over}}} \right) \frac{T_{\text{eq}} R_p \delta}{M_p}. \quad (2.47)$$

Our overall metric using known planet masses will then be

$$\mathcal{D}_{\text{mass}} \propto C_T(\lambda) 10^{-0.2m_*} t_{14} \frac{T_{\text{eq}} R_p \delta}{M_p}. \quad (2.48)$$

In the current form of PREFACE, if the planet mass is known to within 40% or better, the metric in Equation 2.48 is always used. A previous incarnation of the pipeline had the ability to toggle the use of masses on or off.

## 2.4 Completing the Loop: Atmospheric Modelling

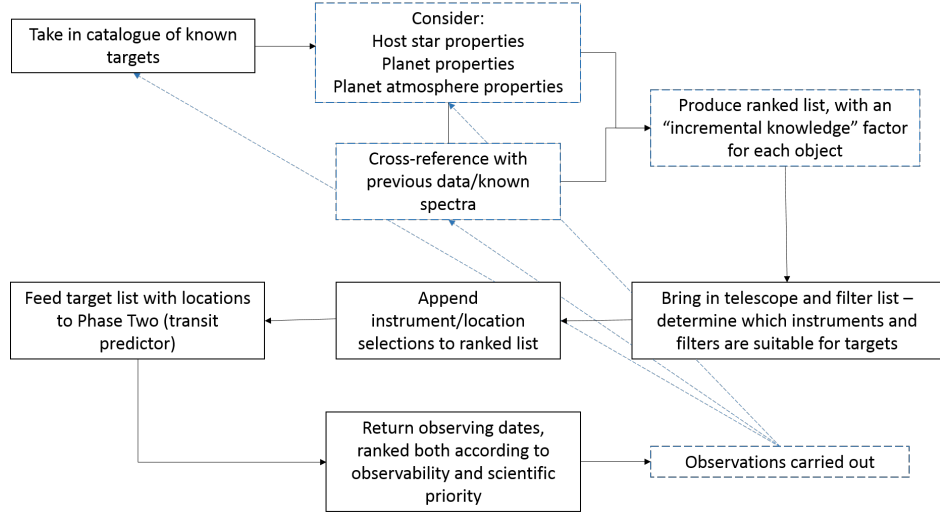


Figure 2.4: The closed flowchart for finished pipeline, featuring the completed feedback loop (blue dashes).

Once PREFACE has run and a ranked list of target events has been generated, the next step is to carry out those observations and recover a spectrum, through using several broad-band filters or a spectrograph. All of our targets thus far are known to be viable and/or interesting for atmospheric study, but despite ongoing observations[16],



our understanding of their atmospheric constituents and dynamics remains incomplete. Indeed, for the vast majority of known transiting planets, we have no information on their atmospheres whatsoever.

The result of this is that the possible parameter space for modelling a planet's atmosphere is usually vast, with few (if any) well-known priors. As such, software must be capable of exploring this space to generate a grid/library of possible forward models, which can then be compared to observed spectra. One example of such a library is ATMO, a grid of models for 117 observationally significant planets. [17]

Another development is that of Exo-Transmit, a publicly-available software package written in C, published in November 2016 [8] for the calculation of exoplanet transmission spectra. Although initially written for use on super-Earth-type objects, the program extends well to giant planets/objects of diverse composition, and has been well-received by exoplanet researchers. As mentioned in Section 1.3, Exo-Transmit calculates forward models - it uses prior knowledge (or failing that, assumptions) about chemical abundances and temperature to generate model spectra by solving the equation of radiative transfer through the atmosphere along the line of sight. The user can specify various system parameters which go into these models, which are as follows:

- Temperature-pressure (T-P) profile - an isothermal 1-D T-P profile must be specified in order to generate spectra. In their paper, Kempton et al. argue that transmission spectra are comparatively insensitive to vertical temperature gradients, but the temperature itself is important to determine the atmosphere scale height and for controlling gas composition and opacity. Exo-Transmit contains profiles for temperatures from 300 to 1500K, in steps of 100K, stored as .dat files.
- Eq.-of-State file - the choice here controls the abundances of different absorbing species and atmosphere constituents, as a function of the previously-specified T-P profile. The choices here include:
  - Solar-based, with metallicities ranging from 0.1 to 1000 times the solar value.
  - Solar-based but with varying C/O ratios, ranging from 0.2 to 1.2.
  - Single-component atmospheres, eg. atmospheres of pure methane, carbon dioxide, etc. Exo-Transmit contains opacities for 30 different species, which are detailed in Table 1 of the Nov. 2016 paper. The presence of different gases and processes in the calculations can be toggled in one of the input files.

Each of the solar-based files comes in two or three versions; a model with just gas phase chemistry, a model also including condensation and rain-out, and for high-metallicity/carbon-rich atmospheres, a model also including condensation and rain-out of carbon as graphite.

## 2.4: COMPLETING THE LOOP: ATMOSPHERIC MODELLING

- Cloud deck location - the user can specify a pressure in Pa at which clouds become optically thick. Exo-Transmit will then only run for pressures lower than this, ie. for atmosphere that is above the cloud deck. This has the effect of generally reducing the strength of absorption features. This value can simply be left as 0 for a cloud-free atmosphere.
- Strength of Rayleigh scattering - by changing this value from its default of 1, the scattering cross-section can be increased (to simulate aerosols and atmospheric hazes) or turned off by setting this value to 0. A high value here will weaken the observed strength of atomic spectral features and produce a classic ‘Rayleigh slope’ in the optical/NIR.
- Planet surface gravity in  $\text{ms}^{-2}$ , calculable from:

$$g = \frac{G \cdot M}{R^2}, \quad (2.49)$$

where  $M$  and  $R$  are the planet mass and radius respectively.

- Planet radius in m - this is the radius at the base of the planet atmosphere (for cloud-free models), or the radius at the top of the cloud deck if clouds are present.
- Stellar radius in m - required for calculation of transit depth.

By generating a grid of forward model spectra and comparing these to new data, the atmospheric properties of planets can be constrained, feeding back into the wider data set/body of knowledge, as illustrated in Figure 2.4. Planets for which a wide range of models could conceivably fit (also referred to as “high-ignorance” planets) will be priority targets for observers, as a comparatively large amount of new science could be done by targeting them, and subsequently ruling out many possible models from the forward grid. Similarly, planets for which spectra have already been taken [16] will be comparatively “low-ignorance” planets; these have already been studied in some detail, and so new observations may not be useful (not able to rule out many models from the prior grid), unless they are of particularly high quality or able to access some particular spectral feature.

Building such grids for a potentially large set of planets is a significant undertaking, and a PhD project in its own right. This task of building what will effectively be “Phase Three” has been taken up by Josh Hayes; when complete, the loop will be closed, and model spectra will be able to inform both the initial choice of targets and the nature of observed atmospheres.

# Chapter 3

## Using PREFACE

### 3.1 Requirements

PREFACE is primarily built around the following Python modules:

- astropy (min. version 1.2.1)
- datetime
- matplotlib
- multiprocessing
- numpy
- pandas
- os
- scipy

In addition, the non-standard ephem module is needed, although this may be deprecated at some point in the future. Access to a script editor such as spyder or gedit is also recommended. In order to update the **TEPCat.csv** files, a stable Internet connection is needed, but this is not otherwise required.

### 3.2 Pipeline Architecture

#### 3.2.1 **MasterShell**

Because of the pipeline's size, it is divided into 11 scripts, with the **MasterShell** script **importing** the first nine as modules in sequence <sup>1</sup>, and firing the main function associ-

---

<sup>1</sup>At one stage, the pipeline branches, using one script or another depending on the instrument type selected.

ated with that script. Each main function can then draw on smaller functions defined in each individual script.

The MasterShell is also important for setting desired user parameters, as follows:

- Local paths to .csv files used in the pipeline, as well as output paths for .csvs produced by the pipeline.
- The choice of instrument and filter. It is recommended to consult Scope.csv (Section 2.2) to inspect the filters available to each instrument and to make sure they are named correctly for the pipeline to find.
- The pipeline’s running mode - this controls how the optimum exposure time for a given instrument/planet is chosen. This is typically done as described in Section 2.3.1, but other options are available, such as fixed-cadence observing for a set exposure time or desired SNR.
- The presence or absence of the additional noise sources described in Section 2.3.1.
- Toggling telescope defocus on or off. (Currently available for photometers only.)
- The pipeline’s “metric mode” - which one of the four metrics defined in Sections 2.3.1 and 2.3.2 should be used for calculations.
- A “sensitivity cut-off” - this is some fraction of the cumulative distribution of planet metrics for a chosen calibration instrument. This process is detailed in Section 3.2.2.
- A time cut - the pipeline will run for transit events occurring between some start and end datetime, eg. an observing season from November 2017 to May 2018. These should be specified in UTC, as the pipeline is not time zone-dependent.
- The number of cores to draw on for multiprocessing purposes. By default, the pipeline will use all but one of the total available cores on the machine. Using all of them is not advised, as your machine will likely lock up when given this (typically large) job.

#### 3.2.2 Phase One

##### ModCheck

Before any ranking can take place, the initial catalogue of planets to be ranked must first be assembled. As already mentioned, this is drawn from the TEPcat website, but this initially exists as three separate .csv files:

- allplanets-csv.csv - This contains information and column headers for well-studied objects.

- `keplanets-csv.csv` - This contains information for less-studied objects, which are typically Kepler planets around faint stars, as the filename suggests.
- `observables.csv` - This contains co-ordinates, stellar magnitudes, transit times, durations, depths and ephemerides for all planets in both of the previous two files, as well as column headers.

The first order of business is to update these files (if necessary) and then put them together into a coherent whole. The script is set to check when the local copy of each of these three files was created/last modified; if this is longer than 7 days ago, new versions are downloaded from the TEPcat website and stored as Core Files. These are then read in using the pandas module and concatenated together using the **axis** argument, first by row and then by column to build up a complete, legible pandas DataFrame with sensible column headers. This is illustrated in Figure 3.1. Some data types must be specified at read-in, and any values of -1 in `allplanets-csv.csv` and `keplanets-csv.csv` are wiped - TEPcat uses -1 to denote null/missing values, and pandas must be told this explicitly.



Figure 3.1: Illustration of the construction of the FullTEPSet.csv.

With this done, typos can be fixed (of which there are a few), and all planet entries which are missing either  $t_{14}$  or  $D$  are dropped, as they will not be usable without this data. The catalogue also contains a few brown dwarves (marked as 'BD' in the Type column), which must also be excluded. The cleaned DataFrame is then saved as FullTEPSet.csv as a Core File.

### Impact Parameter Recovery

As mentioned in Section 2.1, TEPcat does not hold inclinations or impact parameters in the three files used in the previous section. However, as Phase Two developed further in order to become part of the wider pipeline, rather than a stand-alone script, finding/recovering  $b$ -values became necessary. Section 3.2.3 explores why this is the case, but for now we will only explore the technicalities of recovering this data.

Fortunately, two other catalogues do hold  $b$ -values; the one at exoplanets.eu and the Exoplanet Orbit Database at exoplanets.org/ [18] [19]. Snapshots of these databases were taken (as these are not automated to the same extent as TEPcat) and stored as Core .csvs. As Exoplanet.org was found to hold more  $b$ -values, it is passed first along with the FullTEPSet file from previous, and after some massaging to ensure both catalogues have identical planet naming conventions, planets that appear in both catalogues (i.e. objects with known  $b$ -values) receive their impact parameters, and a new file is written out, FullTEPSetWithExoOrgImpacts.csv, in order to be passed with exoplanets.eu to repeat the process. This catalogue holds fewer  $b$ -values, but is fairly effective for more recent discoveries such as WASP-127. Planets which are still missing  $b$  and which appear in both of these catalogues receive theirs in this second pass, and the file is written out as FullTEPSetWithAllImpacts.csv, as a Core File to be passed to the next segment of PREFACE.

When this solution was first implemented at the end of September 2017, 1213 out of 1451 values were successfully recovered from literature in this manner - a success rate of 83.6%. However, this may fall over time as TEPcat is updated, while the snapshots from the other catalogues remain static. It is unknown as to what extent this will be detrimental to pipeline operation. Targets for which  $b$  cannot be recovered pass to a further step in the next section.

### Working\_TEPSetBuilder

At this stage, the catalogue is still not yet ready to be passed to the metric; there are further crucial parameters to be recovered internally. Foremost among these are stellar magnitudes; TEPcat holds  $V$  and  $K$ -magnitudes for its planets, but many other filters might be used, such as Johnson-Cousins  $R$  and  $I$  (for PROMPT-8 and similar telescopes), Sloan  $ugriz$  (for the TNT), as well as any of the specialist filters and grisms available to instruments such as the Liverpool Telescope (LT) and VLT.

Magnitudes for each of these wavebands must be recovered. Unfortunately no stellar identifiers are included in TEPcat, so the catalogue cannot be simply passed to something like Simbad, which in any case may not be complete. Instead, a table of spectral conversions [20] [21] (stored as a Core .csv) is used; extrapolating from the star's known effective temperature and  $V$ -band magnitude, magnitudes in  $U$ ,  $B$ ,  $R$ ,  $I$  and  $K$  can be recovered reasonably well. Two entries have no stated  $K$ -magnitude, so the calculated ones can be used instead. The M-dwarf TRAPPIST-1 is too cool for  $U$ -band magnitudes to be reliably recovered - no conversion values for  $U$ - $B$  are listed. Hence, its  $U$ -band magnitude must be set as NaN.

To recover magnitudes in the Sloan bands, the conversions set out by Jordi et al. [22] are used, to retrieve magnitudes for  $g$ ,  $r$ ,  $i$  and  $z$  ( $u$  is not routinely recovered at this time). For other longpass filters, a more general approach is used. By way of example, RISE on the Liverpool Telescope uses a 720nm filter which can be approximated as  $(I + Z)$ :

$$\begin{aligned} F_I &= 10^{-0.4 \cdot (m_*(I) - m_{zp}(I))}, \\ F_Z &= 10^{-0.4 \cdot (m_*(Z) - m_{zp}(Z))}, \\ F_{\text{RISE}} &= F_I + F_Z, \\ m_{\text{RISE}} &= m_I + 2.5 \log_{10} \left( \frac{F_I}{F_{\text{RISE}}} \right) \end{aligned} \quad (3.1)$$

Other calculable quantities include the transit depth and scale height (both seen previously), as well as the semi-major axis in AU and planet equilibrium temperature, via:

$$a = \sqrt[3]{\frac{P^2}{M_*}} \quad (M_p \ll M_*), \quad (3.2)$$

where  $P$  is the planet's orbital period in years and  $M_*$  is the stellar mass in solar masses, and:

$$T_{\text{eq}} = T_{\text{eff}} \cdot \sqrt{\frac{R_*}{2 \cdot a}} \quad (\text{Albedo} = 0), \quad (3.3)$$

where  $T_{\text{eff}}$  is the stellar effective temperature,  $R_*$  is the solar radius in AU and  $a$  is the semi-major axis. These values typically agree well with literature values (in cases where they exist), and so are used from this point onwards in the pipeline for consistency.

At this point, we must return to the issue of impact parameters, and ensure that all entries that do not yet have a  $b$ -value receive one. We can attempt to calculate  $b$  for these remaining planets using the model of Seager & Mallén-Ornelas (ORM model), as follows [14] [23] :

$$\begin{aligned}
\frac{a}{R_*} &= \sqrt{\frac{\left(1 + \frac{R_p}{R_*}\right)^2 - b^2 \cdot \left(1 - \sin^2\left(\frac{\pi \cdot t_{14}}{P}\right)\right)}{\sin^2\left(\frac{\pi \cdot t_{14}}{P}\right)}}, \\
\left(\frac{a}{R_*} \cdot \sin\left(\frac{\pi \cdot t_{14}}{P}\right)\right)^2 &= \left(1 + \frac{R_p}{R_*}\right)^2 - b^2 \cdot \left(1 - \sin^2\left(\frac{\pi \cdot t_{14}}{P}\right)\right), \\
\left(1 + \frac{R_p}{R_*}\right)^2 - \left(\frac{a}{R_*} \cdot \sin\left(\frac{\pi \cdot t_{14}}{P}\right)\right)^2 &= b^2 \cdot \left(1 - \sin^2\left(\frac{\pi \cdot t_{14}}{P}\right)\right), \\
\frac{\left(1 + \frac{R_p}{R_*}\right)^2}{\cos^2\left(\frac{\pi \cdot t_{14}}{P}\right)} - \left(\frac{a}{R_*} \cdot \tan\left(\frac{\pi \cdot t_{14}}{P}\right)\right)^2 &= b^2, \\
b &= \sqrt{\left(\frac{\left(1 + \frac{R_p}{R_*}\right)^2}{\cos^2\left(\frac{\pi \cdot t_{14}}{P}\right)} - \left(\frac{a}{R_*} \cdot \tan\left(\frac{\pi \cdot t_{14}}{P}\right)\right)^2\right)}.
\end{aligned} \tag{3.4}$$

This model makes the following assumptions:

- The planet orbit is circular (eccentricity  $e \simeq 0$ ).
- $M_p \ll M_*$  and the companion is dark compared to the central star.
- The stellar mass-radius relation is known.
- The light comes from a single star, rather than from two or more blended stars. (No circumbinary objects.)

However, it does not assume  $R_p \ll R_*$ , or that  $a \gg R_*$ . This is important, as these approximations are not valid for our data set, which contains many large planets in close-in orbits. Unfortunately this formula does return unphysical solutions for some targets, which the pipeline handles through a try/except block. This may be because these objects are on eccentric orbits (violating our first assumption), or that any/all of the parameters which go into the calculation are erroneous in some way. These parameters do have errors associated with them, sometimes significant, and  $b$  is normally retrieved from a raw transit light curve as part of a box least-squares (BLS) or Markov-Chain Monte Carlo (MCMC) analysis, rather than calculated numerically as done here. If the model fails for a particular planet,  $b$  is instead assigned an arbitrary value of 0.5 - all possible avenues of retrieval have been exhausted, and we cannot say anything meaningful about what it might be.

Once all this is done and safely stored, the new DataFrame is saved as a new Core File, WorkingTEPSet.csv. All of the information needed to run the ranking is now in place.



### RankMaker

This segment of code is where the ranks for each planet are calculated and assigned, and where instrument-dependency first comes in. Using Eq. 2.11 for photometry or Eq. 2.12 for spectroscopy, optimum exposure times are recovered for each planet, as well as an approximate signal-to-noise ratio and the number of exposures capturable over the course of a full transit. Please note that these exposure times and SNR values are only good for using relative to each other, and are not a substitute for using an exposure-time calculator.

Metric scores are generated for each of the 4 metrics in Section 2.3 (also using masses where available), and the DataFrame is sorted by the standard rank score, highest to lowest. Sorting by whichever running mode for the pipeline would compromise the general nature of the data product, so is avoided at this stage. The output is a ranked .csv titled `RankedTEPSet_[Instrument]_[Filter]-band_for_[Running Mode]_[Add_Noise?][Defocus?][Masses?].csv`, which is then passed to the next segment.

### Sensitivity Cut-Off & ViabilitySplitter

With all planets now ranked for the chosen instrument, some sensible cut-off must now be made; which targets are viable to observe, and which are not? Unfortunately due to Eq. 2.21 being a proportionality, a fixed, physically-informed cut is not possible. This problem is discussed in greater detail in the SPEARNET paper detailing the metric’s development. [24]

The next-best solution, then, is to look at the cumulative distribution of planet rank scores for both the chosen telescope and some calibration instrument, and see if some sensible cut can be placed.

Figure 3.2 is an example of this, generated for the TNT using the metric in Eq. 2.5. The distribution has a ‘knee’ containing a few hundred planets, and then a long tail containing  $\approx 1000$  planets that will likely not be viable to observe. The process of making a “sensitivity cut”, then, is as follows:

1. Run the RankMaker (detailed in the previous section) for some comparison/calibration instrument - for spectrographic set-ups, this is the VLT using the FORS2 spectrograph package with the 600RI+19 grism. For photometry, TNT/ULTRASPEC is the calibration instrument. This choice is arbitrary; a different calibration instrument could be used, but as these are the largest of the main instruments used by PREFACE so far, it stands to reason that they should be able to access the largest sub-sets of viable targets.
2. Impose a cut, specified by the `ViableCut` global variable in the MasterShell script, to capture all of the planets in the ‘knee’ of the cumulative distribution of the calibration scope. This is handled in either `PhaseOne_SpecCutter` or `PhaseOne_Cutter` for spectrographs or photometry respectively.

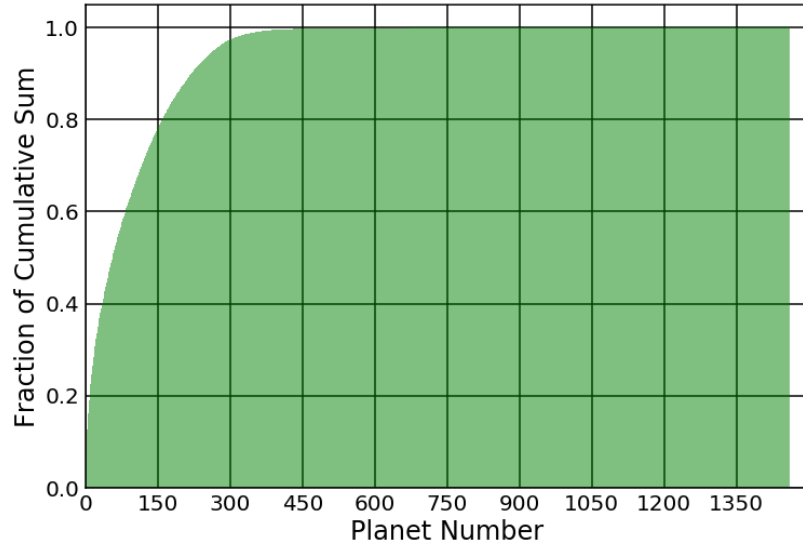


Figure 3.2: Cumulative distribution of rank scores for all planets, observed with the TNT’s ULTRASPEC camera in Sloan r-band (as of 10/11/17).

3. Having made this cut, return the minimum absolute rank score that makes it into this cut.
4. For our instrument chosen for the running of PREFACE, take all planets ranked equal to or greater than this minimum rank score. This becomes our viable subset for the instrument. Figure 3.3 explores how the sample of planets returned for each instrument varies as the cut widens; a very high cut ( $\approx 99\%$  of the cumulative distribution) is needed to capture all planets in the “knee” of the graph in Figure 3.2.

At this point, the data set for the chosen instrument is split; those planets that make the cut go into a ranked .csv titled `TopTEPSet_[Instrument]_[Filter]-band_for_[Running Mode],[Add_Noise?],[Defocus?],[Metric Mode],[Masses?],[ViableCut].Cut.csv`, while those that don’t go into `UnusableTargets_[Instrument]_[Filter]-band_for_[Running Mode],[Add_Noise?],[Defocus?],[Metric Mode],[Masses?],[ViableCut].Cut.csv`. Only these top targets are subsequently fed to the second phase of the pipeline; the transit predictor. Figure 3.4 shows the sample size for each instrument, taking a 99% cut using the method described previously.

### 3.2.3 Phase Two

With the viable targets now selected for the desired instrument, this list can now be passed (as a .csv) to Phase Two of the pipeline, in order to return all transit dates and

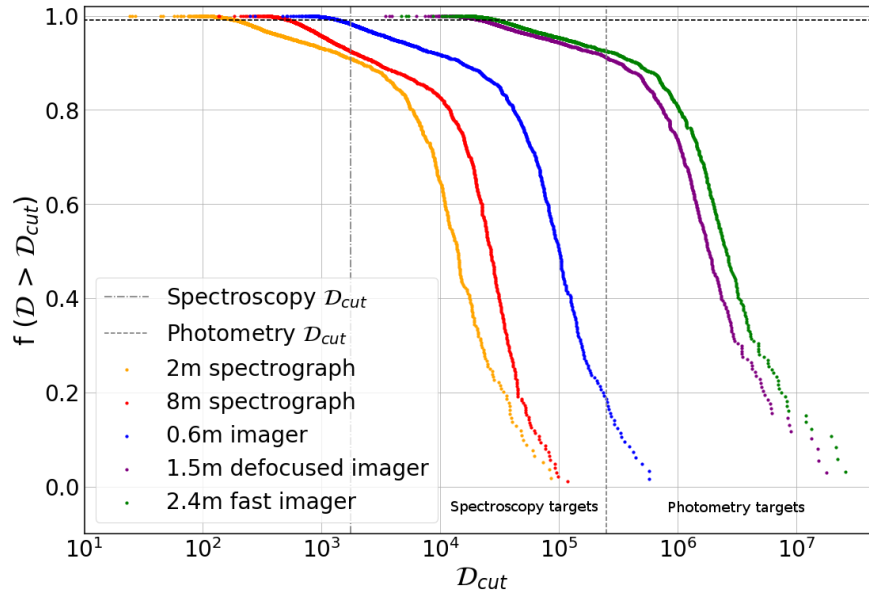


Figure 3.3: Planet samples for five of the primary instruments used by the pipeline thus far, for a 99% cut ( $f(D > D_{cut})=0.99$ ). All planets to the right of the grey vertical dashed lines are chosen.

times for this subset of planets, and to determine how well they can be seen from the chosen observing location.

### P2TimeSplitter

In order for the transit predictor to predict upcoming events, it must have a starting ephemeris (time of mid-transit) and a known orbital period to extrapolate forwards from. At this stage of the pipeline, these ephemerides are Barycentric Julian Dates in Barycentric Dynamical Time (BJD\_TDB), in order to be absolute relative to the rest of the solar system. These can differ from the standard Julian date in UTC (JD\_UTC) by up to 10 minutes, so carrying out this correction is important, both to account for Relativity and to get the time stamp into a form usable by observers. It is also important to account for the different timescales, as TDB and UTC differ by:

$$\text{Diff} = 32.184s + N, \quad (3.5)$$

where  $N$  is the number of leap seconds that have elapsed since 1961 (37, as of January 2017).

An existing online calculator [25] based on IDL code can perform this calculation, but it is not practical to call this for every planet in a set of hundreds (or in future, possibly thousands) of targets. In keeping with the vision of PREFACE being a self-contained, end-to-end pipeline, an internal handling method is needed. Unfortunately, a single command to carry out this calculation is not present in astropy; one can move

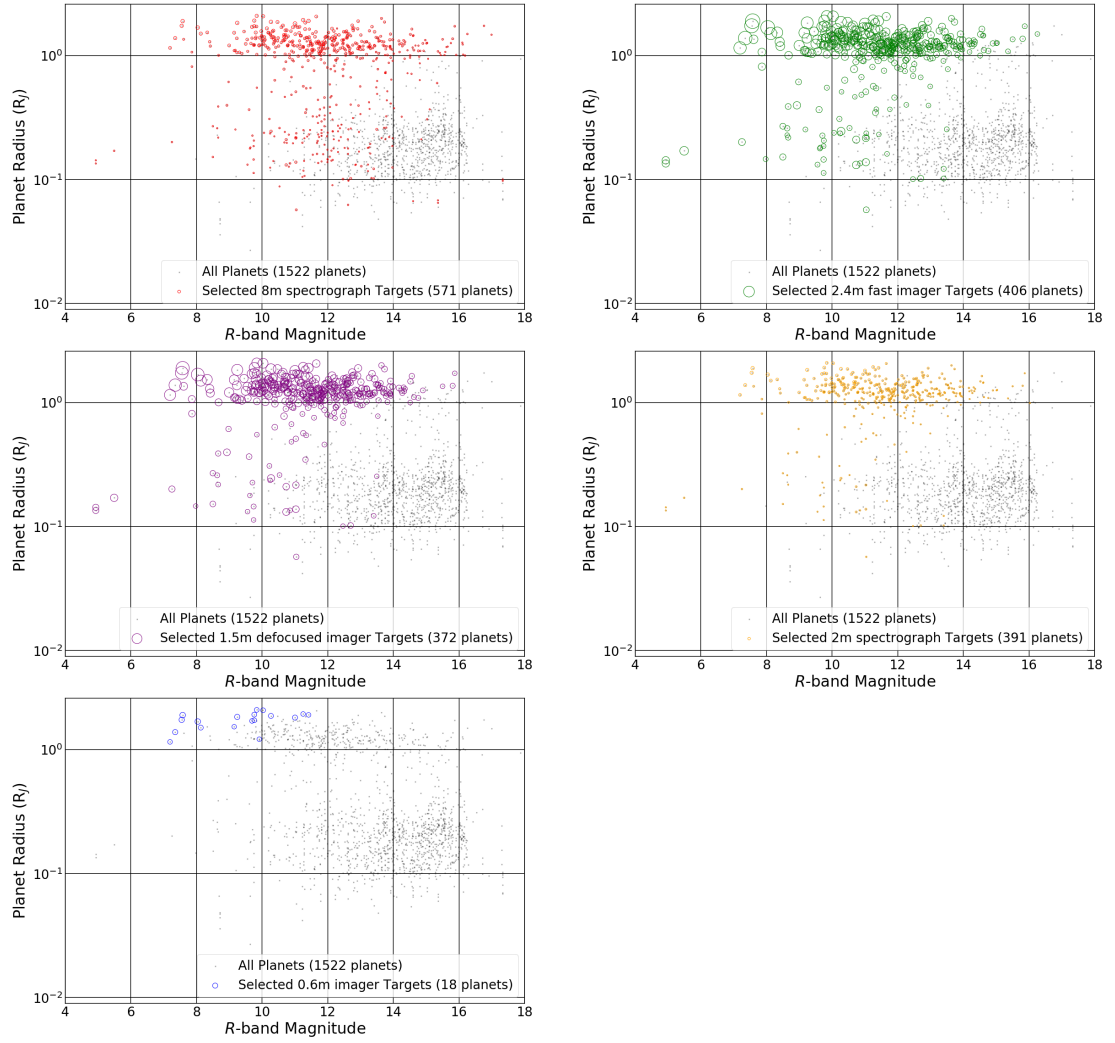


Figure 3.4:  $R_p$  vs.  $R_{\text{mag}}$  plot using the planet samples for a 99% cut, for five of the primary instruments used by the pipeline thus far. The marker size represents the metric score of that target.

from JD.UTC to BJD.TDB reasonably well, but not back. However, using this, we can reverse-engineer a solution; if we know the JD and BJD are within 10 minutes of each other, we can construct an array of possible JDs (around the known BJD) and solve each element for a BJD. The JD that returns the BJD closest to the actual (TEPCat) value is the JD solution we need.

Hence, the TimeSplitter script takes in the top set of targets (contained in the TopTEPSet .csv produced in the previous section) and runs this calculation. It is comparatively expensive ( $\approx 3$  seconds to solve for a single target) computationally, which is why it has been 'back-ended' to this point in the pipeline, to only run for planets that will be handled by Phase Two (and thus, will need a JD.UTC time-stamp). The user can check the progress of this script by examining the printed output; a wrapper func-

tion counts the number of targets solved for out of the wider set. Once this is complete, the top set is split into many .csv files, one for each planet, and stored in order to be handled by Phase Two proper.

### PhaseTwoCSV

PhaseTwoCSV is the last main script to fire as part of PREFACE, and is also the largest and most complex. It handles the job of finding transits for each planet, assigning each event a metric detailing its goodness to observe, producing the plots as seen in Figure 1.1 and bundling up these data outputs for final processing by the MasterShell script. By this point, the pipeline is telescope, filter, location and time-dependent. As the script with the largest number of co-dependent ‘moving parts’, it also has the largest capacity for things going wrong, and so the working of the code itself is detailed in more depth here than for other sections.

As of May 2018, this section of the pipeline employs multiprocessing to speed up running. Using a glob mask, all relevant planets to be handled are found via a glob mask and gathered into a list, which is then split into evenly-sized chunks. Each subset is then manually assigned to a core (setting the core affinity for the job), and each planet in that set joins the list of multiprocessing jobs to execute. The target of these jobs is the central function in PhaseTwoCSV. Once all processes are gathered, the list is joined and the run begins. Each core subsequently handles many planets simultaneously, rather than sequentially as was previously the case. However, stress-testing with a single core handling 144 planets found no ill effects or loss of performance; the simultaneous processes are handled as expected for a single-core run (as such, these tests showed no performance gain). In order to see this, more cores are needed, with an increase from one to 11 cores producing a factor six improvement in run time.

With the multiprocessing jobs configured, the first check is simple - to see if the target is ever observable from the chosen site. The equation used is:

$$\text{Min. Alt} \geq 90^\circ - |\text{Lat} - \text{Dec (DD:MM:SS)}|, \quad (3.6)$$

where Min. Alt is the minimum observing altitude of the instrument (typically  $30^\circ$ , corresponding to air mass 2), Lat is the latitude of the instrument (North taken as positive), and Dec is the declination of the target in degrees, as given in TEPcat. If this inequality is not satisfied, the target in question will never be observable from the location, and so the pipeline can safely skip over it and move to the next target. This ensures that computing power is not wasted on targets that will never produce viable events.

In order to understand why we go through the process of giving each planet its own file, we must consider the shapes of the DataFrames we are using. First we consider a process in which no splitting occurs; all of the planets are in one DataFrame, as shown in Figure 3.5. Each planet has a  $t_0$  (now in JD) associated with it, corresponding to the midpoint of a transit. To recover the transit start time  $t_1$ , we can simply subtract half of the known duration,  $t_{14}$ :

$$t_1 = t_0 - (0.5 \cdot t_{14}), \quad (3.7)$$

where  $t_0$  is an astropy Time object (so that the Julian format can be easily handled) expressed as a DateTime, and  $t_{14}$  is a TimeDelta object. This is important, as astropy Times and standard DateTimes (from the module of the same name) are not compatible with each other; a standard convention must be adopted for a particular process or calculation. Fortunately, converting between the two is trivial.

With this start time in hand, we can now extrapolate forwards to find future start times:

$$t_1(\text{future}) = t_1 + (n \cdot P), \quad (3.8)$$

where  $P$  is the orbital period in days (as a TimeDelta object) and  $n$  is a series of integers chosen such that the observing window specified in the MasterShell is completely covered, thus catching all events in the time window. In older code versions,  $n$  was just an array from 0 to some large number, but this approach was both inefficient and prone to breaking if  $n$  was insufficiently large to cover the entirety of the chosen observation window.

The list of recovered  $t_1$  values is initially held as a list; if the length of this list is 0, the pipeline stops for this target - it does not transit at all in the chosen time frame. If it is non-zero, it is concatenated column-wise to the existing DataFrame as a column of  $t_1$  values.

Planet	...	...	...	T0 (JD)	T1 (JD)
1				X	1
2					1
3					1
4					1
5					1
6					1
7					1
8					1
9					1

Figure 3.5: An illustration of the shape problems encountered when keeping all planets in a single frame.

Figure 3.5 shows this, with all  $t_1$  values for Planet 1 recovered and appended (the blue cells). At this point, the problem is thrown up; if the DataFrame is read row-wise at this point, subsequent planets 2-9 will have  $t_1$  values meant for Planet 1. To fix this, the row for Planet 1 could be duplicated in place many times such that all entries have the correct  $t_1$  value, but this is both computationally inefficient and not how the pandas module is meant to be used. Instead, by assigning each planet its own .csv, the scenario presented in Figure 3.6 is encountered.

Planet	...	...	...	T0 (JD)	T1 (JD)
1	Data	Data	Data	Time	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1
NaN	NaN	NaN	NaN	NaN	1

Figure 3.6: For a single-planet file, many NaN values are generated.

Once the  $t_1$  values are generated, the blank space in the rows below is filled with null, or NaN values. The **fillna** command can then be used to automatically fill these, making sure each transit event has all the relevant planet data to hand for row-wise reading. Overall, this solution is much more elegant.

With all transit start times recovered for all targets, other important times must also be calculated for later use. These include the midpoint and end of each transit,

$$\begin{aligned} t_0 &= t_1 + (0.5 \cdot t_{14}), \\ t_4 &= t_1 + t_{14} \end{aligned} \quad (3.9)$$

the start and end of the 1-hour baselines either side of the transit,

$$\begin{aligned} \text{Base Start} &= t_1 - 1 \text{ hour}, \\ \text{Base End} &= t_1 + 1 \text{ hour}, \end{aligned} \quad (3.10)$$

and  $t_{23}$ , the duration of the transit at full depth, i.e. from the end of ingress to the start of egress:

$$t_{23} = \frac{P}{\pi} \cdot \arcsin \left( \frac{\sqrt{(R_* - R_p)^2 - (b \cdot R_*)^2}}{a} \right), \quad (3.11)$$

where  $a$  carries units of  $R_\odot$  and  $b$  is the impact parameter, which describes the projected distance between the centre of the star and the centre of the planet during transit. It is given by:

$$b = \frac{a}{R_*} \cdot \cos(i), \quad (3.12)$$

where  $i$  is the system inclination in degrees and  $a$  carries units of solar radii. For  $i = 90^\circ$  (a perfectly edge-on system),  $b = 0$ . However, as  $b$  approaches unity, full transit

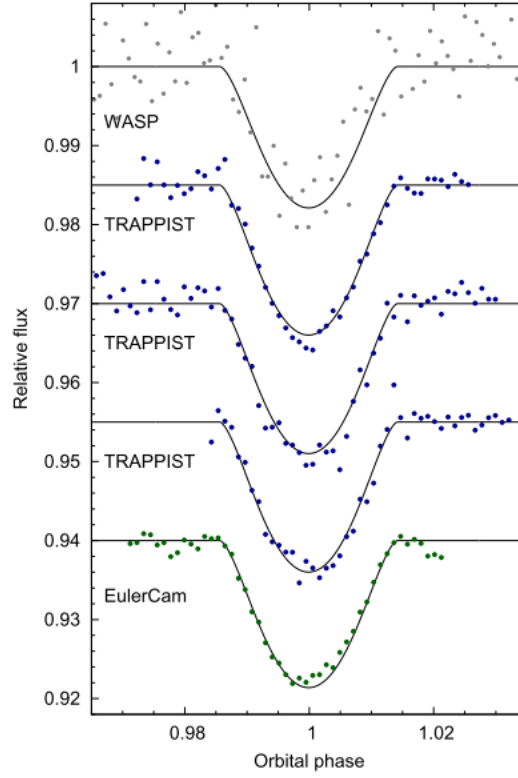


Figure 3.7: A grazing transit for WASP-140 ( $b = 0.93$ ), with a distinctive triangular profile. [26]

becomes geometrically impossible, and the observer will instead see a “grazing” event with a triangular profile, an example of which is shown in Figure 3.7.

Returning to Eq. 3.11, for sufficiently large  $b$ ,  $t_{23}$  will become unphysical, representing the shift from the trapezoidal to the triangular transit profile. Hence, these calculations are handled through a try-except block. For physical  $b$ :

$$\begin{aligned} t_{12} &= \frac{|t_{14} - t_{23}|}{2}, \\ t_2 &= t_1 + t_{12}, \\ t_3 &= t_4 - t_{12}, \end{aligned} \tag{3.13}$$

where  $t_{12}$  is the duration of ingress/egress (specified as a TimeDelta object),  $t_2$  is the end-of-ingress time and  $t_3$  is the start-of-egress time. The abs value in the equation for  $t_{12}$  is necessary as a coercive measure, as cases have been encountered where the  $t_{23}$  value returned is physical, but also greater than  $t_{14}$ . Such cases might be caused by one or more poorly-constrained system parameters.

For unphysical  $b$ :



$$\begin{aligned} t_{23} &= 0, \\ t_2 &= t_0, \\ t_3 &= t_0, \end{aligned} \tag{3.14}$$

to represent that the full transit depth is never achieved. All of these times are then written to the DataFrame for subsequent use.

Now that individual events have been found for the planet being handled, each event must be assigned a score, representing its viability to observe. To accomplish this, we must check that two key conditions are both satisfied:

1. At least some part of the transit must happen at night.
2. The parent star is above the minimum altitude of the telescope during the transit.

Hence, we must plot the path of the star and determine how well this coincides with both the night at the observing location and the start time and duration of transit. (As an aside, this is why it is advantageous to keep all calculations in UTC - no time zone conversions for different observing locations are needed.) Using astropy's coordinates module, with the instrument's location, the target's celestial co-ordinates and a 30-hour time window containing both the individual transit event in question and the nearest midnight, the paths (made up of alt-az co-ordinate pairs) of the target, the Sun and Moon are recovered. At this stage, the separation between the target and the Moon is recovered at 21 points along the path length, in order for the user to gauge if observations of the event in question might be affected by the Moon. The Moon's phase is also calculated at this point through the ephemeris module (which may be deprecated in future). The seven key times for each event (Start of baseline,  $t_1$ ,  $t_2$ ,  $t_0$ ,  $t_3$ ,  $t_4$  and end of baseline) are converted to be relative to midnight for calculation and plotting purposes.

In order to impose the conditions outlined previously, we must determine when the night is, defined as being when the Sun is at least  $18^\circ$  below the horizon (astronomical twilight). On our altitude-time axes, the Sun's path over time is compared to a straight line at  $y = -18^\circ$ ; the times where these two cross yield the start and end times of the night.

With these recovered, we can finally start imposing our set of key conditions. This consists of a general-purpose classification to assign each event into one of four categories, and label them appropriately with an Internal Rank Marker. These categories are defined by three conditions:

$$|t_0 - t_{N0}| \leq \frac{t_{14} + T_N}{2}, \tag{3.15}$$

$$t_{14} + t = t_N, \tag{3.16}$$

$$\frac{t_N - t_{14}}{2} - |t_0 - t_{N0}| \geq 1 \text{ hour}. \tag{3.17}$$

In Eq. 3.15,  $t_0$  is the transit midpoint time,  $t_{N0}$  is the night midpoint time,  $t_{14}$  is the transit duration and  $t_N$  is the night duration. If this condition is satisfied, at least some of the transit takes place at night, and the event is thus potentially observable. In Eq. 3.16,  $t = t_N - t_{14}$ . If this condition is satisfied, the whole of the transit falls within one observing night. Finally, if Eq. 3.17 is satisfied, the entirety of the transit plus 1-hour baselines before and after is observable. It is important to note that if Eq. 3.17 is met, then it follows that Eqs. 3.16 and 3.15 are also met. Hence, the algorithm first checks if Eq. 3.17 is satisfied, and then moves to Eqs. 3.16 and 3.15, taking a ‘top-down’ approach.

In addition to these, several “spot-checks” are employed; even if an event satisfies all of the criteria above, it may not actually be at an observable altitude. Hence, altitude checks are also needed to catch such events. Three of these are used, running over different regions of relevant parameter space:

- **SC** - moves through the transit + baselines over 1000 steps to generate a range of datetimes.
- **NC** - moves through the relevant night over 1000 steps to generate a range of datetimes.
- **KC** - finds the overlap between the transit event (no baselines) and the night, and iterates through this to generate datetimes.

These are then transformed into sets of observing frames to yield alt-az pairs, which can then be inspected with Python’s **any** and **all** logical operators. For example, a height spot check can be performed by asking **any**( $a \geq 30^\circ$ ), where  $a$  is the parameter space to walk through. If the target satisfies this at any point, this statement will be True and the check will be passed. Using the **all** statement here demands that the target is always above  $30^\circ$ . These are combined with Eqs. 3.15 to 3.17 to form a series of logic conditions, each one corresponding to a different internal marker. Each event is assigned a label of 3, 2, 1 or X, with the latter being used if the event does not satisfy any of the three equations above, and is thus unobservable.

- **03\_F** - Eq. 3.17 holds, event + baselines always above observable height, target visible at some point in the night.
- **03\_P** - Eq. 3.17 holds, event + baselines at observable height during transit or during night, transit is observable at some point during night.
- **02\_F** - Eq. 3.16 holds, event + baselines always above observable height, target visible at some point in the night.
- **02\_P** - Eq. 3.16 holds, event + baselines at observable height during transit or during night, transit is observable at some point during night.

- **01\_F** - Eq. 3.15 holds, event + baselines always above observable height, target visible at some point in the night.
- **01\_P** - Eq. 3.15 holds, event + baselines at observable height during transit or during night, transit is observable at some point during night.
- **X** - none of the above conditions are satisfied.

For each of the conditions above, all points must be satisfied before the marker can be assigned; if this does not occur, the event is passed down to the next condition. At first glance, this conditional web may seem overly complex, but it is necessary in order to prevent cases such as those in Figure 3.8 from slipping through the net. For this event, Eq. 3.15 holds and the event is visible during transit after +16 hours, satisfying the first two statements. However, the region in which the transit can be observed does not coincide with night-time, meaning a cross-check (**KC**) is necessary.

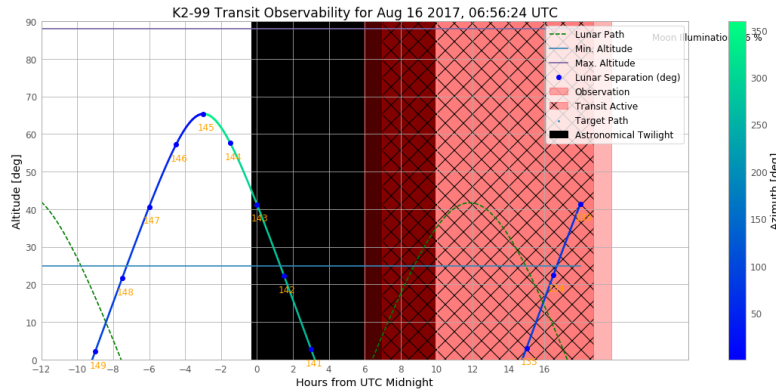


Figure 3.8: Events such as this transit of K2-99 demonstrate the need for stringent cross-checks.

While a useful classifier, this is only a qualitative approach; a quantitative test is needed in order to determine if one event is more valuable to observe than another. Previously, PREFACE's measure of individual event viability was the size of the area bounded by the altitude path of the target as it moved across the night sky, as illustrated in Figure 1.1. Here, the area is bounded by the minimum observing altitude (the line at  $y=25^\circ$ ), the start and end of the transit event itself, and the altitude-time path taken by the system over the course of the event (the blue/turquoise parabola). Targets which stayed at a high altitude (and thus, at a low observing air mass) for the duration of their transit, as well as targets with a long transit time would have a large bounded area, recovered by an integration, and thus would score highly in P2. However, when we consider the metric now employed by Phase One (Eq. 2.5), the behaviour of long transit durations being desirable is now expressed there (as the  $t_{14}$  dependency), and so had to be removed from P2 in order to avoid long transits being unfairly favoured. The event weighting must favour targets consistently observable at low air mass (behaviour

which is not accounted for in Phase One), while being independent of the time base used to carry out these observations.

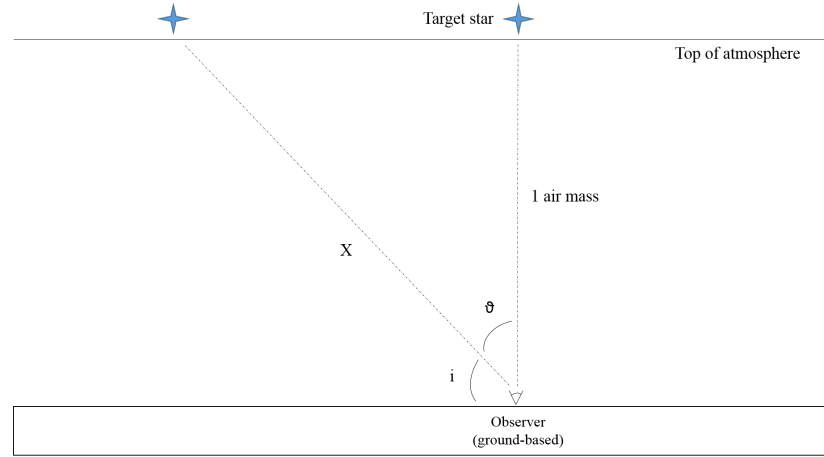


Figure 3.9: A diagram illustrating the variation of air mass as one moves through the angle  $\theta$  away from the zenith, drawn by the author.

Figure 3.9 illustrates the planar air-mass model. It shows an illustration of a ground-based observer, studying a celestial target at the zenith. In this model, we must assume that the atmosphere is of uniform density, and we must also neglect the atmosphere's curvature (stemming from the curvature of the Earth). Looking through the entirety of the atmosphere to see a target at the zenith is defined as looking through 1 air mass. Note that this quantity does not have the dimensions of mass, but of length; it is the atmospheric distance we must look through to observe the target.

What if our target is some angle  $\theta$  away from the zenith? Returning to Figure 3.9, the air mass  $X$  we look through is related to the zenith angle  $\theta$  by the following:

$$\begin{aligned}\cos(\theta) &= \frac{1 \text{ air mass}}{X}, \\ X &= \frac{1 \text{ air mass}}{\cos(\theta)} = \sec(\theta) = \operatorname{cosec}(i).\end{aligned}\tag{3.18}$$

Hence, at  $\theta=60^\circ$  (or an altitude angle  $i$  of  $30^\circ$ ), an observer will have to look through 2 air masses to see their target, and so will be more susceptible to noise associated with atmospheric effects.

This model is a relatively simple one, and will break down at low altitudes; as the zenith angle  $\theta$  approaches  $90^\circ$ , the number of air masses an observer must look through tends to infinity. Obviously, the Earth's atmosphere is not infinite, but this model is effective up until zenith angles of around  $70^\circ$ . In reality,  $X$  will cap out at around 40 air masses for a celestial target at sea level, for a zenith angle of  $90^\circ$ . As we will not be observing targets above zenith angles of  $65^\circ$ , this model is reasonable for our purposes.

From this discussion, it is clear that air mass is tied to astronomical seeing; the more atmosphere an observer has to look through, the more severe the atmosphere's effects will be. This relation is quantified through the Fried parameter,  $r_0$ :

$$r_0 = \left( 0.423 \cdot k^2 \cdot (\sec\vartheta) \int dh \cdot C_N^2(h) \right)^{-3/5}, \quad (3.19)$$

where  $k$  is the wave number:

$$k = \frac{2\pi}{\lambda}, \quad (3.20)$$

$\vartheta$  is the zenith angle seen previously,  $h$  is the observer's height above sea level and  $C_N$  is a term describing the strength of the atmospheric activity. [27] From this, we can see that  $r_0$  has the following scaling with wavelength and air mass:

$$\begin{aligned} r_0 &\propto (k^2)^{-3/5} = \left( \left( \frac{2\pi}{\lambda} \right)^2 \right)^{-3/5} \propto \lambda^{6/5}, \\ r_0 &\propto (\sec\vartheta)^{-3/5} = X^{-3/5}, \end{aligned} \quad (3.21)$$

where  $X$  is the planar air mass from previous. Hence, our seeing point spread function  $\Theta$  will go as:

$$\Theta \propto \frac{1.2 \cdot \lambda}{r_0} \propto \lambda^{-1/5} \cdot X^{3/5}. \quad (3.22)$$

We now have a relation between the quality of seeing and air mass; as air mass increases, the size of our PSF rises, leading to increased blurring/distortion of our image. [28] Returning to the event weighting for Phase Two, we must describe how the air mass changes over the course of the observation, which can be initially expressed as an integral:

$$\int_{L_1}^{L_2} \text{cosec}(i)^{3/5} dt, \quad (3.23)$$

where  $L_1$  and  $L_2$  are the start and end times of the observation. In the limiting case of a target always at an altitude of  $90^\circ$ , this integral solves as  $(L_2 - L_1)$ . The value of the solution will rise as the altitude decreases, so for weighting purposes, we must take the inverse of our integral and add in a normalisation factor:

$$\text{Weight}_{\text{AM}} = \frac{(L_2 - L_1)}{\int_{L_1}^{L_2} \text{cosec}(i)^{3/5} dt}. \quad (3.24)$$

By normalising by a factor of  $(L_2 - L_1)$ , the metric's dependence on transit duration is eliminated. For constant  $i$  of  $90^\circ$ , the metric returns a value of 1, which will decrease as the altitude of observations falls, and will tend to 0 for very low altitudes. It was felt

necessary to make this weighting more stringent, and so it was amended to its current form:

$$\text{Weight}_{\text{AM}} = 3 \cdot \left( \frac{(L_2 - L_1)}{\int_{L_1}^{L_2} \text{cosec}(i)^{3/5} dt} - \frac{2}{3} \right). \quad (3.25)$$

This weighting will still return a value of 1 for targets at the zenith, but will decay more steeply as altitude decreases, such that a target at a constant altitude of 30° will have a weight of 0, down from the value of 0.66 returned by Eq. 3.24. Initially, the two values in this equation were 2 and 0.5, but this relation was found to be not strong enough in suppressing low-altitude events, as illustrated in Figure 3.10.

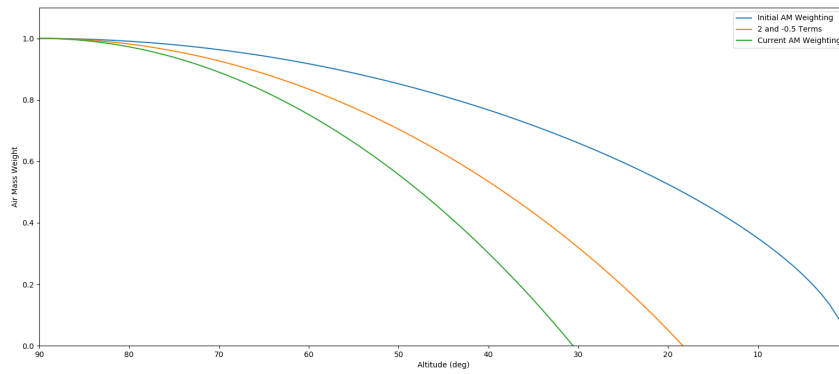


Figure 3.10: Various air mass functions for a target at constant altitude, shown with no suppression (blue), 2 and -0.5 suppression (orange) and 3 and -2/3 suppression (green).

How can the limits  $L_1$  and  $L_2$  be determined? At this stage, the pipeline branches, depending on if the event being handled is a Full (F tag, denoting always above 30°) or Partial (P tag, meaning there is at least one crossing point where the path goes through 30°) event. Events marked X are not handled in this manner; they are not observable, and so their weight will be 0. Full events are comparatively simple to handle:

$$L_1 = \begin{cases} \text{Start of Night,} & \text{if Baseline Start Time} < \text{Start of Night} \\ \text{Baseline Start Time,} & \text{otherwise} \end{cases} \quad (3.26)$$

$$L_2 = \begin{cases} \text{Baseline End Time,} & \text{if Baseline End Time} < \text{End of Night} \\ \text{End of Night,} & \text{otherwise} \end{cases} \quad (3.27)$$

If a crossing point occurs, there are many more possible scenarios. The first order of business is to determine the crossing time(s) relative to midnight, which is a simple

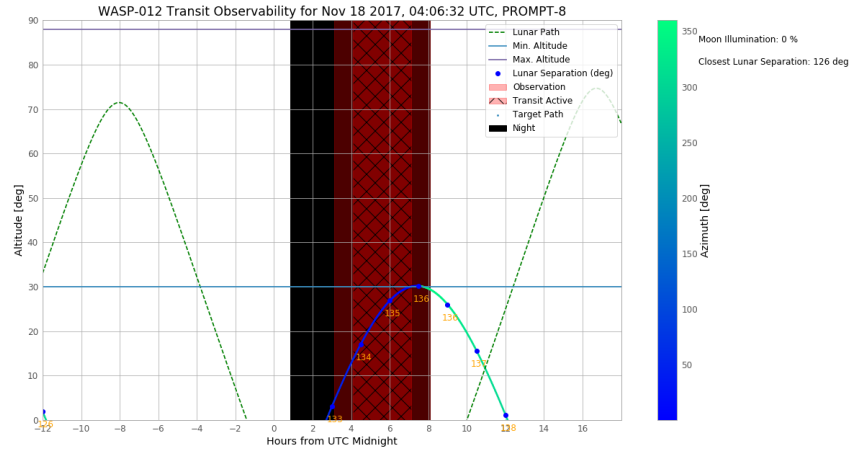


Figure 3.11: Seen from CTIO, WASP-12 barely makes it above the minimum altitude, and so crosses twice.

matter of finding where the target path and the solid line at  $+30^\circ$  cross. This is called hereafter as *LTC1*.

If two crossing points are returned, such as for the transit of WASP-12 in Figure 3.11, the set of conditions is actually very simple. Using *SC* as the time base, no crossing points will be returned outside of the event itself. Therefore, in an event with two crossing points, the only scenario in which either one will be unusable is the point falling outside of the night. (If both crossing points are outside of the night, the event is unobservable, and will have already been flagged with an X.) Hence:

$$L_1 = \begin{cases} \text{Start of Night,} & \text{if First Crossing Point} < \text{Start of Night} \\ \text{First Crossing Point,} & \text{otherwise} \end{cases} \quad (3.28)$$

$$L_2 = \begin{cases} \text{End of Night,} & \text{if End of Night} < \text{Second Crossing Point} \\ \text{Second Crossing Point,} & \text{otherwise} \end{cases} \quad (3.29)$$

For single-crossing events, a much more complex conditional web is needed. Python's **any** and **all** operators only accept one argument, so it is necessary to nest the logic conditions in order to build the gateways we need. For the lower integration limit, these gates are as follows:

$$\text{PGen2} = \begin{cases} \text{Baseline Start Time} < \text{Start of Night,} \\ \text{Baseline End Altitude} < 30^\circ \end{cases} \quad (3.30)$$

$$\text{PGen12} = \begin{cases} t_1 < \text{Start of Night} < \text{LTC1} < t_4 \text{ End of Night,} \\ \text{Baseline End Altitude} < 30^\circ \end{cases} \quad (3.31)$$

$$PGen15 = \begin{cases} t_1 < \text{Start of Night} < LTC1 < \text{End of Night} < t_4, \\ \text{Baseline End Altitude} < 30^\circ \end{cases} \quad (3.32)$$

$$PGen3 = \begin{cases} \mathbf{all}(PGen2), \\ \mathbf{all}(PGen12), \\ \mathbf{all}(PGen15), \\ \text{Start of Night} > LTC1 \end{cases} \quad (3.33)$$

$$PGen4 = \begin{cases} \text{Start of Night} < LTC1 < \text{Baseline End Time}, \\ \text{Start of Night} > \text{Baseline Start Time} \end{cases} \quad (3.34)$$

$$PGen5 = \begin{cases} \text{Baseline Start Time} < \text{Start of Night} < LTC1, \\ \text{Start of Night} < LTC1 < \text{End of Night} \end{cases} \quad (3.35)$$

$$PGen6 = \begin{cases} \text{Start of Night} < LTC1 < \text{Baseline End Time}, \\ \text{End of Night} > \text{Baseline Start Time}, \\ \text{Baseline End Altitude} > 30^\circ \end{cases} \quad (3.36)$$

$$PGen7 = \begin{cases} \text{Baseline Start Time} < \text{Start of Night} < LTC1 < \text{Baseline End Time}, \\ \text{Baseline End Altitude} > 30^\circ \end{cases} \quad (3.37)$$

$$PGen14 = \begin{cases} \mathbf{all}(PGen4), \\ \mathbf{all}(PGen6), \\ \mathbf{all}(PGen7), \end{cases} \quad (3.38)$$

These are brought together in:

$$L_1 = \begin{cases} \text{Start of Night}, & \text{if } \mathbf{any}(PGen3) \\ LTC1, & \text{elif } \mathbf{any}(PGen14) \\ \text{Baseline Start Time}, & \text{elif } \mathbf{any}(PGen5) \end{cases} \quad (3.39)$$

For the upper limit, the initial gates are:

$$PGen8 = \begin{cases} \text{Baseline Start Time} \leq LTC1 < \text{Baseline End Time} < \text{End of Night}, \\ \text{Baseline End Altitude} > 30^\circ \end{cases} \quad (3.40)$$

$$PGen9 = \begin{cases} LTC1 < \text{End of Night} < \text{Baseline End Time}, \\ LTC1 < \text{Baseline End Time} \end{cases} \quad (3.41)$$



$$\text{PGen10} = \begin{cases} \text{Baseline Start Time} \leq LTC1 < \text{End of Night} < \text{Baseline End Time}, \\ \text{Baseline End Altitude} > 30^\circ \end{cases} \quad (3.42)$$

$$\text{PGen13} = \begin{cases} \text{Start of Night} < t_1 < LTC1 < \text{End of Night} < t_4, \\ \text{Baseline End Altitude} > 30^\circ \end{cases} \quad (3.43)$$

$$\text{PGen11} = \begin{cases} \mathbf{all}(\text{PGen10}), \\ \mathbf{all}(\text{PGen13}), \\ LTC1 < \text{End of Night} \end{cases} \quad (3.44)$$

They are compiled in:

$$L_2 = \begin{cases} \text{Baseline End Time}, & \text{if } \mathbf{all}(\text{PGen8}) \\ \text{End of Night}, & \text{elif } \mathbf{any}(\text{PGen11}) \\ LTC1, & \text{elif } \mathbf{any}(\text{PGen9}) \end{cases} \quad (3.45)$$

The newest conditions (12 and 13) are to handle events that fall partially outside of the night at the start or end, but are generally comparable in duration to the night as a whole. This list may be updated in future if it becomes necessary to modify these gates further.

With  $L_1$  and  $L_2$  recovered, we can now perform the integration described in Eq. 3.25. However, being observable at a high altitude is not enough by itself; the observer will want to capture as much of the transit profile as possible, including baselines before and after, which are essential for fitting the transit curve in analysis. Of particular interest are the ingress and egress of the planet, for determining limb darkening parameters and hunting for transit timing and duration variations (TTVs and TDVs, respectively). It might also be the case that an observer is missing some part of an existing transit profile, and so needs an event in which the remainder can be captured and folded with existing data to complete the profile.

Therefore, it is useful to know what percentage of each of these regions an observer can potentially capture for a given transit, and assign a weight accordingly. This was previously handled by a logistic function looking at the total transit percentage captured, but was changed to a linear 5-part approach on Supachai's suggestion, for the reasons just outlined. The 5 transit "segments" that are inspected are as follows:

1. Baseline before transit - 1 hour before  $t_1$ .
2. Ingress -  $t_1$  to  $t_2$ .
3. Full transit -  $t_2$  to  $t_3$ .

4. Egress -  $t_3$  to  $t_4$ .
5. Baseline after transit - 1 hour after  $t_4$ .

The times at which these various points happen have already been recovered (Eqs. 3.7 to 3.14), but now we must see if they are visible for the event in question, i.e. what percentage of each segment falls within our observing window, bounded by  $L_1$  and  $L_2$ . There are four basic scenarios which may be encountered:

1. Both the start and end of the segment fall between  $L_1$  and  $L_2$ , in which case the segment is observable in its entirety.
2. Either the start or the end of the segment falls within the observing window - partial capture is achieved. This might be because the observing night ends, or the target might be below the minimum altitude when the segment starts.
3. Both the start and end of the segment fall outside of  $L_1$  and  $L_2$ , but on either side of  $L_1$  and  $L_2$  so that some middle region can be captured.
4. Both the start and end of the segment fall outside of  $L_1$  and  $L_2$ , such that no part of the segment is observable.

If an event falls through the conditional web outlined in this section, a System Exception is thrown stating the location of the fault. The guidelines outlined in Appendix C should be followed in this case.

With the various segment percentages recovered, an event weight can be calculated:

$$\text{Event Weight} = \left( \frac{\text{Duration of baseline captured}}{2hr} \right) \cdot \left( \frac{\text{Duration of } t_{23} \text{ captured}}{t_{23}} \right) \cdot \left( \frac{\text{Duration of ingress \& egress captured}}{t_{12} + t_{34}} \right) \quad (3.46)$$

This returns a number between 0 and 1, with incomplete events strongly suppressed; if exactly the first half of a transit is captured, Eq. 3.46 will give  $0.5 \cdot 0.5 \cdot 0.5 = 0.125$ . If any of these three components are missing completely, the event will be zero-weighted. This ensures that fully capturable transits (of most interest to observers) are prioritised.

At this stage, the plot for the event (like that shown in Figures 1.1 and 3.8) is generated and saved in a separate image output folder. The following data products are read out and written to the DataFrame at this point:

- The lunar phase (percentage illuminated).
- The smallest separation between the Moon and the target.

- The internal rank marker.
- The start and end times of observation, from  $L_1$  and  $L_2$ .
- The air mass weight calculated by Eq. 3.25.
- Each of the three terms in Eq. 3.46.
- The final product given by Eq. 3.46.

The final weight for a given transit, of a given planet from a given location can now be recovered, as the product of whichever of the four Phase One metrics is being used, Eq. 3.25 and Eq. 3.46:

$$\text{Final Weighting} = \mathcal{D}_{\text{version}} \cdot \left[ 3 \cdot \left( \frac{(L_2 - L_1)}{\int_{L_1}^{L_2} \text{cosec}(i)^{3/5} dt} - \frac{2}{3} \right) \right] \cdot \left( \frac{\text{Baseline capturable}}{2hr} \right) \cdot \left( \frac{t_{23} \text{ capturable}}{t_{23}} \right) \cdot \left( \frac{\text{Ingress \& egress capturable}}{t_{12} + t_{34}} \right) \quad (3.47)$$

With this done for all events for a particular planet, the complete DataFrame is saved as a .csv to an Output.Parts folder. The script then takes in the next .csv (corresponding to the next planet) and repeats the process. Phase Two is the most computationally-expensive part of the pipeline, at  $\simeq 6\text{s/event}$ , and a set of even a couple of dozen planets can generate 2000 events. Hence, the use of multiprocessing at this stage is important.

### 3.2.4 Final Outputs

With a collection of output .csv files generated, they must be merged back together again to get a final ranked list. This is handled by the MasterShell itself as the final step in the running of PREFACE; all of the relevant files are read in using pandas and a glob mask, merged together, cleaned such that all unobservable events (those with an Internal Rank of X) are dropped, and saved as a Full Event List to the Final\_Outputs folder. In future, this file may feed into a website, which can then be updated from this final data product.

For an extended program of observing, it is also important to consider how many times a planet transits, in order to maximise the chances of building up a good set of light curves. Hence, cumulative observing scores are recovered. Considering an individual planet, all events for which Eq. 3.46 is greater than 0.5 (corresponding to full transit capture and at least 50% baselines) are found and their rank scores summed. These cumulative scores and the number of “good” events available are written to a separate .csv, allowing observers to identify planets with multiple viable transits or

targets of opportunity with a single highly-ranked event. This can be cross-referenced with the Full Event List .csv file to produce observing calendars.

It is even possible to repeat this exercise for multiple telescopes and cut their cumulative scores together, to find the best targets accessible to elements of a heterogeneous, worldwide telescope network.

# Appendix A

## Initial Targets

Some key parameters for each of the 6 initial planets of interest are gathered here.

### **WASP-127b**

- Star Name - BD-03 2978
- RA (J2000) - 10:42:14.08
- Dec (J2000) - -03:50:06.3
- $V_{\text{mag}}$  - 10.15
- P (days) - 4.178062
- arXiv link -1607.07859

### **HAT-P-26b**

- Star Name - GSC 0320-01027
- RA (J2000) - 14:12:37.44
- Dec (J2000) - +04:03:36.0
- $V_{\text{mag}}$  - 11.744
- Orbital period (days) - 4.234516
- arXiv link - 1010.1008

### **HAT-P-47b**

- Star Name - GSC 2324-00031
- RA (J2000) - 02:33:13.97

- Dec (J2000) - +30:21:37.8
- $V_{\text{mag}}$  - 10.7
- Orbital period (days) - 4.7322
- arXiv link - 1606.04556

#### **HAT-P-48b**

- Star Name - GSC 2326-00214
- RA (J2000) - 02:57:53.03
- Dec (J2000) - +30:37:32.5
- $V_{\text{mag}}$  - 12.2
- Orbital period (days) - 4.4087
- arXiv link - 1606.04556

#### **HAT-P-10b**

- Star Name - GSC 02340-01714
- RA (J2000) - 03:09:28.55
- Dec (J2000) - +30:40:24.9
- $V_{\text{mag}}$  - 11.89
- Orbital period (days) - 3.7224690
- arXiv link - 0808.4925

#### **WASP-107b**

- Star Name - TYC 5530-1795-1
- RA (J2000) - 12:33:32.84
- Dec (J2000) - -10:08:46.1
- $V_{\text{mag}}$  - 11.6
- Orbital period (days) - 5.72149
- arXiv link - 1701.03776

## Appendix B

# Defocused Photometry, Or How I Learned to Stop Worrying and Love 2D Gaussians

A technique pioneered by Dr. John Southworth of Keele University is that of defocused photometry; deliberate defocusing of an optical telescope such that its image is smeared out over many pixels. This allows the targeting of bright stars that would otherwise quickly saturate, and allows longer exposure times to boost SNR. Good examples of this process in the literature were not forthcoming, so I have illustrated the process here.

Consider a circular 2D Gaussian, projected onto our CCD detector. Under our metric, it will have a maximum height of  $N_{\text{HWD}}$  and a full width at half maximum (FWHM) of  $\Theta_{\text{see}}$ . When defocused, the image will be smeared out into a top hat distribution, which we approximate here as a rectangle. In some fixed exposure time  $t_{\text{exp}}$ , defocusing will increase the width of the distribution (14'' is chosen for this example) and reduce its height to some new level,  $y$ . This is illustrated in Figure X.

By defocusing, what new level does our peak count fall to? We can consider our starting Gaussian to be normalised such that its FWHM is  $\Theta_{\text{see}}$ . What volume is contained within the FWHM? To recover this, we start from the standard equation describing a 2D Gaussian:

$$f(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad (\text{B.1})$$

where  $\sigma$  is the standard deviation,

$$\text{FWHM} = 2\sqrt{2\ln(2)}\sigma \approx 2.355\sigma \quad (\text{B.2})$$

Transforming to cylindrical polar co-ordinates,

$$f(r, \theta) = \iint \frac{r}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}} dr d\theta \quad (\text{B.3})$$

Integrating  $d\theta$  from 0 to  $2\pi$  and making the substitution  $R = r/\sigma$ , we find

$$f(R) = \int_0^{2\sqrt{2\ln(2)}} Re^{\frac{-R^2}{2}} dR \quad (\text{B.4})$$

Executing this in Python yields an enclosed volume of 0.9375. Alternatively, we can say that 6.25% of the target flux is lost in the wings of the distribution.

When we defocus to a top-hat, we recover all of the target flux (assuming no leakage). Hence, its enclosed volume will be  $1/0.9375 = 107\%$  of that of our Gaussian. The height (new count number)  $y$  will subsequently be given by

$$y = \frac{A}{0.9375 \cdot 14''}, \quad (\text{B.5})$$

where  $A$  is the area of a unit Gaussian, described by

$$A = \sqrt{2\pi}ac, \quad (\text{B.6})$$

where  $a = N_{\text{HWD}}$  and  $c = \sigma = \Theta_{\text{see}}/2.355$ . Using values for MuSCAT2,

$$y = \frac{\sqrt{2\pi} \cdot N_{\text{HWD}} \cdot \Theta_{\text{see}}}{2.355 \cdot 0.9375 \cdot 14''} = \frac{\sqrt{2\pi} \cdot 32000 \cdot 0.8''}{2.355 \cdot 0.9375 \cdot 14''} = 2076.1. \quad (\text{B.7})$$

This now means that  $t_{\text{exp}}$  can now be increased by a factor of around 15 to recover the original half well depth.



# Appendix C

## How To Isolate a Fault in PREFACE Phase Two

Here's what to do if Phase Two throws an exception:

1. Type 'Name' into the console - this will show the input .csv file that threw the exception. If a NameError is thrown, this will be displayed as the output from sys.exit automatically.
2. Make sure that .csv is the only file being read in (the only one in the input folder).
3. Switch NextTransitTime and Internal Rank printing on and re-run to find the specific event that threw the exception.
4. Adjust ObsStart and ObsEnd in the MasterShell to run for just the broken event to speed things along.
5. Look at the error message - this will let you trace the specific section where the exception occurred. Switch off the EventPercent section and the broken section to produce the event plot - this can help you make a diagnosis. Also switch on all limit printing for the same reason. Note that Phase Two uses a non-interactive plotting backend - this must be changed to display plots in-line/in a Figure window. Don't forget plt.show()!
6. Trace and fix the fault - has the event had the wrong Internal Rank assigned to it? Are your integration limits being chosen correctly? Has a new scenario come up that the generators cannot account for? Or has something else happened?
7. Re-run for the event in question, then for the original observing window, to ensure the fix has taken and not thrown out any other events in the process.

Remember, errors should never pass silently - unless explicitly silenced.

# Bibliography

- [1] G. R. Ricker, J. N. Winn, and R. Vanderspek et al. The Transiting Exoplanet Survey Satellite. *Journal of Astronomical Telescopes, Instruments, and Systems*, 2014.
- [2] ESO. Science goals. <http://sci.esa.int/plato/42277-science/>, Jan 2017 2017.
- [3] The Extrasolar Planet Encyclopedia. Available at <http://exoplanet.eu/>, February 1995.
- [4] J. Southworth. TEPcat: catalogue of the physical properties of transiting planetary systems. <http://www.astro.keele.ac.uk/jkt/tepcat/tepcat.html>.
- [5] D. R. Anderson et al. The discoveries of WASP-91b, WASP-105b and WASP-107b: two warm Jupiters and a planet in the transition region between ice giants and gas giants. *Astronomy & Astrophysics*, 2017.
- [6] J. E. Rodriguez et al. A Multi-Planet System Transiting the  $V = 9$  Rapidly Rotating F-Star HD 106315. *AAS Journals*, 2017.
- [7] Sara Seager, editor. *Exoplanets*. The University of Arizona Press, 2010.
- [8] E. M.-R. Kempton, R. E. Lupu, A. Owusu-Asare, P. Slough, and B. Cale. Exo-Transmit: An Open-Source Code for Calculating Transmission Spectra for Exoplanet Atmospheres of Varied Composition. *Publications of the Astronomical Society of the Pacific*, 2016.
- [9] Kreidberg et al. Water, Methane Depletion, and High-Altitude Condensates in the Atmosphere of the Warm Super-Neptune WASP-107b. *ApJL*, 2017.
- [10] Wenger et al. The SIMBAD astronomical database. The CDS reference database for astronomical objects. *A&AS*, 2000.
- [11] R. Heller. The nature of the giant exomoon candidate Kepler-1625 b-i. *Astronomy & Astrophysics*, 2017.
- [12] Degrees      Minutes      Seconds      to/from      Decimal      Degrees.  
<https://www.fcc.gov/media/radio/dms-decimal>, 2017.
- [13] FORS2, FOcal Reducer/low dispersion Spectrograph 2 Overview.  
<http://www.eso.org/sci/facilities/paranal/instruments/fors/overview.html>,  
September 2014.
- [14] Michael Perryman. *The Exoplanet Handbook*. Cambridge University Press, 2011.

- [15] V. Dhillon et al. ULTRASPEC: a high-speed imaging photometer on the 2.4-m Thai National Telescope. *MNRAS*, 2014.
- [16] Palle et al. A feature-rich transmission spectrum for WASP-127b: Cloud-free skies for the puffiest known super-Neptune? *A&A*, 2017.
- [17] Goyal et al. A library of ATMO forward model transmission spectra for hot Jupiter exoplanets. *MNRAS*, 2017.
- [18] Exoplanets Data Explorer. <http://www.exoplanets.org/>, 2011.
- [19] Wright et al. The Exoplanet Orbit Database. *Publications of the Astronomical Society of the Pacific*, 2011.
- [20] E. Mamajek. A Modern Mean Dwarf Stellar Color and Effective Temperature Sequence. [http://www.pas.rochester.edu/~emamajek/EEM\\_dwarf\\_UBVIJHK\\_colors\\_Teff.txt](http://www.pas.rochester.edu/~emamajek/EEM_dwarf_UBVIJHK_colors_Teff.txt), October 2017.
- [21] M. J. Pecaut & E. Mamajek. Intrinsic Colors, Temperatures, and Bolometric Corrections of Pre-Main Sequence Stars. *ApJ*, 2013.
- [22] K. Jordi, E. K. Grebel, and K. Ammon. Empirical Color Transformations Between SDSS Photometry and Other Photometric Systems. *A&A*, 2007.
- [23] S. Seager and G. Mallén-Ornelas. A Unique Solution of Planet and Star Parameters from an Extrasolar Planet Transit Light Curve. *ApJ*, 2003.
- [24] J. Morgan, E. Kerins, S. Awiphan, I. McDonald, and et al. Exoplanetary atmosphere target selection in the era of comparative planetology. *arXiv:1802.05645*, 2018.
- [25] J. Eastman. Julian date calculator. Hosted at <http://astroutils.astronomy.ohio-state.edu/time/bjd2utc.html>, May 2010.
- [26] Hellier et al. WASP-South transiting exoplanets: WASP-130b, WASP-131b, WASP-132b, WASP-139b, WASP-140b, WASP-141b & WASP-142b. *MNRAS*, 2016.
- [27] A. Quirrenbach. The Effects of Atmospheric Turbulence on Astronomical Observations. In *Adaptive Optics for Vision Science and Astronomy ASP Conference Series*, 1999.
- [28] P. Martinez, J. Kolb, M. Sarazin, and A. Tokovinin. *On the Difference between Seeing and Image Quality: When the Turbulence Outer Scale Enters the Game*. ESO, 2010.