

ExoSpec

Generated by Doxygen 1.8.13

Contents

1	ExoSpec	1
1.1	Introduction	1
1.2	Installation	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Class Documentation	9
5.1	exospec.lc_class.DifferentFileSizes Class Reference	9
5.2	exospec.lc_class.DifferentParamNum Class Reference	9
5.3	exospec.read_input.EmptyFile Class Reference	10
5.4	exospec.lc_class.EmptyFile Class Reference	10
5.5	exospec.lc_class.EmptyFolder Class Reference	10
5.6	exospec.lc_class.IncorrectNameFormat Class Reference	11
5.7	exospec.lc_class.LightCurve Class Reference	11
5.7.1	Member Function Documentation	12
5.7.1.1	LC_dic()	12
5.7.1.2	new_wave_length()	12
5.7.1.3	store_transit_model()	12

5.7.1.4	<code>wave_length()</code>	12
5.8	<code>exospec.lc_class.LightCurveData</code> Class Reference	13
5.8.1	Detailed Description	13
5.8.2	Member Function Documentation	14
5.8.2.1	<code>ferr()</code>	14
5.8.2.2	<code>flux()</code>	14
5.8.2.3	<code>len_file()</code>	14
5.8.2.4	<code>new_time_bin()</code>	14
5.8.2.5	<code>param_list()</code>	15
5.8.2.6	<code>param_name()</code>	15
5.8.2.7	<code>param_num()</code>	15
5.8.2.8	<code>plot_flux_param()</code>	15
5.8.2.9	<code>plot_flux_time()</code>	16
5.8.2.10	<code>time()</code>	16
5.9	<code>exospec.mcmc.MCMC</code> Class Reference	16
5.9.1	Constructor & Destructor Documentation	17
5.9.1.1	<code>__init__()</code>	17
5.9.2	Member Function Documentation	18
5.9.2.1	<code>get_mean_acceptance_fraction()</code>	18
5.9.2.2	<code>get_median_and_errors()</code>	18
5.9.2.3	<code>light_curve_plot()</code>	18
5.9.2.4	<code>run()</code>	19
5.9.2.5	<code>save_chain()</code>	19
5.9.2.6	<code>triangle_plot()</code>	20
5.9.2.7	<code>walker_plot()</code>	20
5.10	<code>exospec.read_input.NoInput</code> Class Reference	21
5.11	<code>exospec.read_input.read_input</code> Class Reference	21
5.11.1	Member Function Documentation	22
5.11.1.1	<code>param_dic()</code>	22
5.12	<code>exospec.TransitModel.TransitModel</code> Class Reference	22

5.12.1	Detailed Description	24
5.12.2	Constructor & Destructor Documentation	24
5.12.2.1	__init__()	24
5.12.3	Member Function Documentation	24
5.12.3.1	kernelnc()	24
5.12.3.2	Inlike_gp()	25
5.12.3.3	Inprior_base()	25
5.12.3.4	Inprior_gp()	25
5.12.3.5	Inprob_gp()	26
5.12.3.6	Inprob_mcmc()	26
5.12.3.7	meannc()	26
5.12.3.8	model()	27
5.12.3.9	read_errors_data()	27
5.12.3.10	read_limb_dark_params()	27
5.12.3.11	sample_conditional()	28
5.12.3.12	set_values()	28
5.12.3.13	update_data()	28
5.12.3.14	update_kernel_params()	29
5.12.3.15	update_transit_params()	29
5.12.3.16	updateTransitMode()	29
6	File Documentation	31
6.1	deliverables.py File Reference	31
6.1.1	Function Documentation	31
6.1.1.1	get_median_and_errors()	31
6.1.1.2	latex_table()	32
6.1.1.3	plot_single_wavelength()	32
6.1.1.4	plot_transmission_spec()	33
6.1.1.5	post_processing_all_wl()	33
6.1.1.6	simple_table()	33
6.2	exospec_main.py File Reference	34
6.2.1	Variable Documentation	34
6.2.1.1	chain	34
6.3	fitting_single_lc.py File Reference	34
6.3.1	Function Documentation	35
6.3.1.1	run_mcmc_single_wl()	35
	Index	37

Chapter 1

ExoSpec

1.1 Introduction

Exospec is a python tool for fitting your multi-wavelength transit light-curves. It can accept an arbitrary number of wavelength channels and an arbitrary number of auxiliary measurements. Currently the fitting has two Gaussian Process kernel options: the kernel outlined in Gibson 2011 which incorporates auxiliary measurements made over the course of transit observation and a more general squared exponential. The latest version is available at <https://github.com/ckreisch/ExoplanetSpectra/>.

1.2 Installation

Dependencies: In addition to standard Python libraries, ExoSpec requires the following packages to run: numpy, scipy, matplotlib, virtualenv, mpi4py, emcee, george, batman-package, corner, and pandas. These packages are automatically installed when installing ExoSpec. To run tests with the setup.py file, the nose package is also required and installs automatically.

Runs with Python version 2.7.

- Before installing ExoSpec, dependencies for george and mpi4py must be installed since these dependencies are not Python packages. You must install the following:
 - Eigen3:
 - * On linux: `sudo apt-get install libeigen3-dev`
 - * On mac: `brew install eigen`
 - * On Windows: the developers of george say they did not test george on Windows, so it may not work but you can still try. We have not tested ExoSpec on Windows
 - OpenMPI:
 - * On linux: `sudo apt-get install openmpi-bin openmpi-common openssh-client openssh-server libopenmpi1.3 libopenmpi-dbg libopenmpi-dev`
 - * On mac: `brew install openmpi`
 - Batman issues: If after moving to step 1 and running the setup.py file you receive an error from batman, you can install it from the source file instead
 - * Download the stable release [here](https://github.com/ckreisch/Batman-package/), and then run `sudo python setup.py install`
- To install ExoSpec, download (<https://github.com/ckreisch/ExoplanetSpectra/>) and unpack the source file. Then run `python setup.py install` and all dependencies and packages will be built.
- A suite of tests are included in `exospec/tests/`

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Exception	
exospec.lc_class.DifferentFileSizes	9
exospec.lc_class.DifferentParamNum	9
exospec.lc_class.EmptyFile	10
exospec.lc_class.EmptyFolder	10
exospec.lc_class.IncorrectNameFormat	11
exospec.read_input.EmptyFile	10
exospec.read_input.NoInput	21
exospec.lc_class.LightCurve	11
exospec.lc_class.LightCurveData	13
object	
exospec.mcmc.MCMC	16
exospec.TransitModel.TransitModel	22
exospec.read_input.read_input	21

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

exospec.lc_class.DifferentFileSizes	Raise when one the light curve file does not have the same size with light curve file for the lowest wavelength (this file serves as reference)	9
exospec.lc_class.DifferentParamNum	Raise when one the light curve file does not have the same number of parameters with light curve file for the lowest wavelength (this file serves as reference)	9
exospec.read_input.EmptyFile	Raise exception when the input file is empty or only has comments	10
exospec.lc_class.EmptyFile	Raise when the light curve file is empty	10
exospec.lc_class.EmptyFolder	Raise exception when the light curve folder is empty	10
exospec.lc_class.IncorrectNameFormat	Raise when the light curve file name is not under the expected format sample_lc_↵<wavelength>.txt	11
exospec.lc_class.LightCurve	List all the light curve files in the indicated folder if the user decided to use another wavelength resolution than the one provided with the files, the code compute the resulting number of new wavelength to be used loop over the light curve files to get their file names and original wavelengths if the user decided to use a new wavelength resolution, the code computes the new wavelength to be used instantiate the new lc objects (with new wave_bin_size) Warning: The code is not able yet to handle situation where files_num is not proportional to wave_bin_size	11
exospec.lc_class.LightCurveData	This class is used by the class LightCurve to extract, store and process the data from the light curve files	13
exospec.mcmc.MCMC	Class to run MCMC to fit curve and produce diagnostic plots and statistics Uses emcee (to run MCMC) and corner (to produce triangle plots)	16
exospec.read_input.NoInput	Raise exception when the input entry is not properly set	21
exospec.read_input.read_input	Reads the input file and stores the input entries	21
exospec.TransitModel.TransitModel	Class to estimate the Transit Model with the customized kernel	22

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

deliverables.py	Deliverables module to parse MCMC data into latex code and plots	31
exospec_main.py	Fits the transits for multiple wavelengths to produce the transmission spectrum	34
fitting_single_lc.py	Fits a transit lightcurve for a single wavelength	34

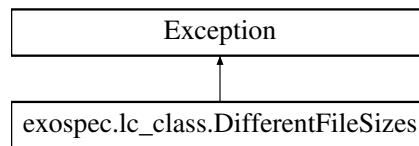
Chapter 5

Class Documentation

5.1 `exospec.lc_class.DifferentFileSizes` Class Reference

Raise when one the light curve file does not have the same size with light curve file for the lowest wavelength (this file serves as reference)

Inheritance diagram for `exospec.lc_class.DifferentFileSizes`:



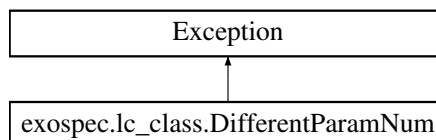
The documentation for this class was generated from the following file:

- `lc_class.py`

5.2 `exospec.lc_class.DifferentParamNum` Class Reference

Raise when one the light curve file does not have the same number of parameters with light curve file for the lowest wavelength (this file serves as reference)

Inheritance diagram for `exospec.lc_class.DifferentParamNum`:



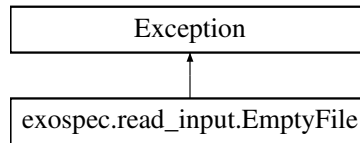
The documentation for this class was generated from the following file:

- `lc_class.py`

5.3 `exospec.read_input.EmptyFile` Class Reference

Raise exception when the input file is empty or only has comments.

Inheritance diagram for `exospec.read_input.EmptyFile`:



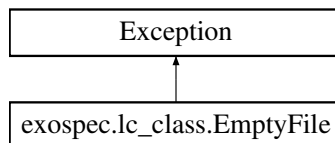
The documentation for this class was generated from the following file:

- `read_input.py`

5.4 `exospec.lc_class.EmptyFile` Class Reference

Raise when the light curve file is empty.

Inheritance diagram for `exospec.lc_class.EmptyFile`:



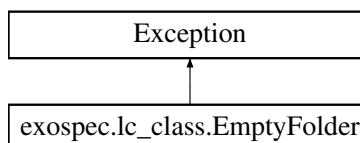
The documentation for this class was generated from the following file:

- `lc_class.py`

5.5 `exospec.lc_class.EmptyFolder` Class Reference

Raise exception when the light curve folder is empty.

Inheritance diagram for `exospec.lc_class.EmptyFolder`:



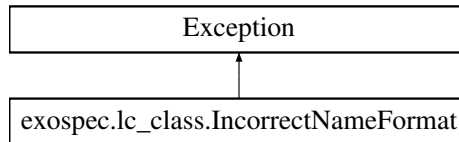
The documentation for this class was generated from the following file:

- `lc_class.py`

5.6 exospec.lc_class.IncorrectNameFormat Class Reference

Raise when the light curve file name is not under the expected format sample_lc_<wavelength>.txt.

Inheritance diagram for exospec.lc_class.IncorrectNameFormat:



The documentation for this class was generated from the following file:

- lc_class.py

5.7 exospec.lc_class.LightCurve Class Reference

List all the light curve files in the indicated folder if the user decided to use another wavelength resolution than the one provided with the files, the code compute the resulting number of new wavelength to be used loop over the light curve files to get their file names and original wavelengths if the user decided to use a new wavelength resolution, the code computes the new wavelength to be used instantiate the new lc objects (with new wave_bin_size) Warning: The code is not able yet to handle situation where files_num is not proportional to wave_bin_size.

Public Member Functions

- def **__init__** (self, PathToLC, wave_bin_size)
- def **LC_dic** (self)
Return LC_dic.
- def **wave_length** (self)
Return wave_length.
- def **new_wave_length** (self)
Return new_wave_length.
- def **store_transit_model** (self, transit_model)
Stores transit_model.

Public Attributes

- **files_list**
- **files_num**
- **wave_length**
- **new_wave_length**
- **LC_dic**
- **obj_mcmc**
- **obj_chain**
- **obj_mcmcGP**
- **obj_chainGP**
- **transit_model**

5.7.1 Member Function Documentation

5.7.1.1 LC_dic()

```
def exospec.lc_class.LightCurve.LC_dic (
    self )
```

Returns

LC_dic the dictionary that contains the light curve objects and which are references by their wavelength as keys

5.7.1.2 new_wave_length()

```
def exospec.lc_class.LightCurve.new_wave_length (
    self )
```

Returns

new_wave_length a list containing the new user defined wavelength of the lighth curve objects

5.7.1.3 store_transit_model()

```
def exospec.lc_class.LightCurve.store_transit_model (
    self,
    transit_model )
```

Parameters

<i>transit_model</i>	
----------------------	--

5.7.1.4 wave_length()

```
def exospec.lc_class.LightCurve.wave_length (
    self )
```

Returns

wave_length a list containing the original wavelength of the lighth curve files

The documentation for this class was generated from the following file:

- lc_class.py

5.8 exospec.lc_class.LightCurveData Class Reference

this class is used by the class [LightCurve](#) to extract, store and process the data from the light curve files

Public Member Functions

- def **__init__** (self, Path_to_files)
- def **len_file** (self)
Returns len_file.
- def **time** (self)
Return time.
- def **flux** (self)
Returns flux.
- def **ferr** (self)
Returns ferr.
- def **param_num** (self)
Returns param_num.
- def **param_name** (self)
Returns param_name.
- def **param_list** (self)
Returns param_list.
- def **new_time_bin** (self, bin_size)
Change the time resolution of a light curve object enables the user to use a new time resolution.
- def **plot_flux_time** (self, bin_size)
Plot the flux against the time plot the flux with a new time resolution using function new_time_bin.
- def **plot_flux_param** (self, param_index)
Plot the flux against a parameter plot the flux against the parameter indicated by the user.

Public Attributes

- **len_file**
- **time**
- **flux**
- **ferr**
- **param_num**
- **param_name**
- **param_list**

5.8.1 Detailed Description

Parameters

<i>Path_to_files</i>	A list that contains the path(s) to the file(s) that will be used to create a light curve object (multiple files can be lumped together to create a ligh curve object)
----------------------	--

5.8.2 Member Function Documentation**5.8.2.1 ferr()**

```
def exospec.lc_class.LightCurveData.ferr (
    self )
```

Returns

ferr the error on the flux

5.8.2.2 flux()

```
def exospec.lc_class.LightCurveData.flux (
    self )
```

Returns

flux the flux vector

5.8.2.3 len_file()

```
def exospec.lc_class.LightCurveData.len_file (
    self )
```

Returns

len_file the length of the light curve file

5.8.2.4 new_time_bin()

```
def exospec.lc_class.LightCurveData.new_time_bin (
    self,
    bin_size )
```

Parameters

<i>bin_size</i>	number of time points to lump together
-----------------	--

5.8.2.5 param_list()

```
def exospec.lc_class.LightCurveData.param_list (
    self )
```

Returns

param_list a list containing the parameters

5.8.2.6 param_name()

```
def exospec.lc_class.LightCurveData.param_name (
    self )
```

Returns

param_name a list containing the names of the parameters

5.8.2.7 param_num()

```
def exospec.lc_class.LightCurveData.param_num (
    self )
```

Returns

param_num the number of parameters defined in the light curve file

5.8.2.8 plot_flux_param()

```
def exospec.lc_class.LightCurveData.plot_flux_param (
    self,
    param_index )
```

Parameters

<i>param_index</i>	the index of the param selected by the user in param_list
--------------------	---

5.8.2.9 plot_flux_time()

```
def exospec.lc_class.LightCurveData.plot_flux_time (
    self,
    bin_size )
```

Parameters

<i>bin_size</i>	number of time points to lump together
-----------------	--

5.8.2.10 time()

```
def exospec.lc_class.LightCurveData.time (
    self )
```

Returns

time the time vector

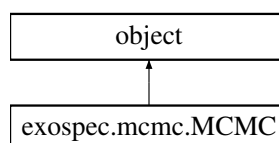
The documentation for this class was generated from the following file:

- lc_class.py

5.9 exospec.mcmc.MCMC Class Reference

Class to run [MCMC](#) to fit curve and produce diagnostic plots and statistics Uses emcee (to run [MCMC](#)) and corner (to produce triangle plots)

Inheritance diagram for exospec.mcmc.MCMC:



Public Member Functions

- def `__init__` (self, t, val, err, ln_prob_fn, transit_params, hyper_params, num_walkers, num_threads)
The constructor.
- def `run` (self, pos, burnin_steps, production_run_steps)
Runs the MCMC should run emcee given a log probability function result is the MCMC chains which are saved as an object attribute.
- def `save_chain` (self, filename)
Saves the chain as a numpy array.
- def `get_mean_acceptance_fraction` (self)
Allows the user to access the mean acceptance fraction The mean acceptance fractions should be between about 0.25 and 0.5.
- def `get_median_and_errors` (self)
Best fit parameters and 1 sigma errors.
- def `triangle_plot` (self, extra_burnin_steps=0, theta_true=None, plot_transit_params=True, plot_hyper_params=True, save_as_dir=".", save_as_name="triangle.png")
Makes a triangle plot If an error is encountered the function returns 1 but does not raise an exception.
- def `walker_plot` (self, extra_burnin_steps=0, theta_true=None, plot_transit_params=True, plot_hyper_params=True, save_as_dir=".", save_as_name="walkers.png")
Plots the chains of each walker and a histogram showing how each parameter was sampled If an error is encountered the function returns 1 but does not raise an exception.
- def `light_curve_plot` (self, model, extra_burnin_steps=0, theta_true=None, save_as_dir=".", save_as_name="light_curve")
Plots the best-fit curve and the data points with errors If the true parameters are given then the true curve is also plotted If an error is encountered the function returns 1 but does not raise an exception.

5.9.1 Constructor & Destructor Documentation

5.9.1.1 `__init__()`

```
def exospec.mcmc.MCMC.__init__ (
    self,
    t,
    val,
    err,
    ln_prob_fn,
    transit_params,
    hyper_params,
    num_walkers,
    num_threads )
```

Parameters

<i>self</i>	The object pointer
<i>t</i>	A numpy array of the independent variable for the data
<i>val</i>	A numpy array of the dependent variable for the data
<i>err</i>	A numpy array of the errors on the dependent variable
<i>ln_prob_fn</i>	The log probability function to be sampled
<i>transit_params</i>	A list of the names of the curve's parameters
<i>hyper_params</i>	A list of the names of noise parameters
<i>num_walkers</i>	Integer giving the number of walkers for the MCMC run
<i>num_threads</i>	Integer giving the number of threads to use per core

5.9.2 Member Function Documentation

5.9.2.1 `get_mean_acceptance_fraction()`

```
def exospec.mcmc.MCMC.get_mean_acceptance_fraction (
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Mean acceptance fraction

5.9.2.2 `get_median_and_errors()`

```
def exospec.mcmc.MCMC.get_median_and_errors (
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Three numpy arrays giving the median and one sigma errors for each parameter

5.9.2.3 `light_curve_plot()`

```
def exospec.mcmc.MCMC.light_curve_plot (
    self,
    model,
    extra_burnin_steps = 0,
    theta_true = None,
    save_as_dir = ".",
    save_as_name = "light_curve" )
```

These plots are useful for visualization but should not cause the code to crash, as the main purpose is to create and save the [MCMC](#) chains

Parameters

<i>self</i>	The object pointer
<i>model</i>	A function that returns the lightcurve shape given (parameters, t, val, err)
<i>extra_burnin_steps</i>	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
<i>theta_true</i>	Numpy array of true parameter values if known (used for test data)
<i>plot_transit_params</i>	Boolean value specifying whether or not to plot the transit parameters
<i>plot_hyper_params</i>	Boolean value specifying whether or not to plot the hyper parameters
<i>save_as_dir</i>	Directory where plot should be saved. Default is current working Directory
<i>save_as_name</i>	Name under which plot should be saved

Return values

0	if successful
1	on failure

5.9.2.4 run()

```
def exospec.mcmc.MCMC.run (
    self,
    pos,
    burnin_steps,
    production_run_steps )
```

Parameters

<i>self</i>	The object pointer
<i>pos</i>	A 2D numpy array giving the initial positions of the walkers in parameter space
<i>burnin_steps</i>	An integer giving the number of initial steps to take to start exploring the parameter space before starting to save the chains
<i>production_run_steps</i>	The number of steps to take for each walker after the burnin phase

Returns

A 2D numpy array with all the samples for each of the transit and hyper parameters

5.9.2.5 save_chain()

```
def exospec.mcmc.MCMC.save_chain (
    self,
    filename )
```

Parameters

<i>self</i>	The object pointer
<i>filename</i>	The filename including path where the chains will be saved

Return values

0	if successful
1	if an IO error occurs or if the chain is empty

5.9.2.6 triangle_plot()

```
def exospec.mcmc.MCMC.triangle_plot (
    self,
    extra_burnin_steps = 0,
    theta_true = None,
    plot_transit_params = True,
    plot_hyper_params = True,
    save_as_dir = ".",
    save_as_name = "triangle.png" )
```

These plots are useful for visualization but should not cause the code to crash, as the main purpose of the code is to create and save the [MCMC](#) chains.

Parameters

<i>self</i>	The object pointer
<i>extra_burnin_steps</i>	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
<i>theta_true</i>	Numpy array of true parameter values for the parameters to be plotted (only used for test data)
<i>plot_transit_params</i>	Boolean value specifying whether or not to plot the transit parameters
<i>plot_hyper_params</i>	Boolean value specifying whether or not to plot the hyper parameters
<i>save_as_dir</i>	Directory where plot should be saved. Default is current working Directory
<i>save_as_name</i>	Name under which plot should be saved

Return values

0	if successful
1	on failure

5.9.2.7 walker_plot()

```
def exospec.mcmc.MCMC.walker_plot (
    self,
```

```

extra_burnin_steps = 0,
theta_true = None,
plot_transit_params = True,
plot_hyper_params = True,
save_as_dir = ".",
save_as_name = "walkers.png" )

```

These plots are useful for visualization but should not cause the code to crash, as the main purpose is to create and save the [MCMC](#) chains

Parameters

<i>self</i>	The object pointer
<i>extra_burnin_steps</i>	Number of steps (in addition to burnin_steps from run) at the start of each chain to neglect
<i>theta_true</i>	Numpy array of true parameter values if known (used for test data)
<i>plot_transit_params</i>	Boolean value specifying whether or not to plot the transit parameters
<i>plot_hyper_params</i>	Boolean value specifying whether or not to plot the hyper parameters
<i>save_as_dir</i>	Directory where plot should be saved. Default is current working Directory
<i>save_as_name</i>	Name under which plot should be saved

Return values

0	if successful
1	on failure

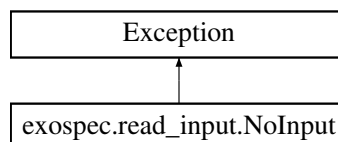
The documentation for this class was generated from the following file:

- mcmc.py

5.10 exospec.read_input.NoInput Class Reference

Raise exception when the input entry is not properly set.

Inheritance diagram for exospec.read_input.NoInput:



The documentation for this class was generated from the following file:

- read_input.py

5.11 exospec.read_input.read_input Class Reference

reads the input file and stores the input entries

Public Member Functions

- `def __init__(self, input_file)`
- `def param_dic(self)`
Returns the parameters dictionary.
- `def is_float(self, string)`
Check if the string contains a float.

Public Attributes

- `param_dic`

5.11.1 Member Function Documentation

5.11.1.1 param_dic()

```
def exospec.read_input.read_input.param_dic (
    self )
```

Returns

`param_dic` the parameters dictionary

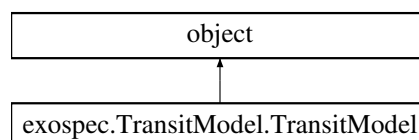
The documentation for this class was generated from the following file:

- `read_input.py`

5.12 exospec.TransitModel.TransitModel Class Reference

Class to estimate the Transit Model with the customized kernel.

Inheritance diagram for `exospec.TransitModel.TransitModel`:



Public Member Functions

- def `__init__` (self, kwargs)
The constructor.
- def `set_values` (self, dict_of_values, kwargs)
Set the parameters of the model based on the values provided.
- def `read_limb_dark_params` (self, kwargs)
Forms a list of the limb darkening parameters from the user input In case some values are missed, the default ones will be taken.
- def `read_errors_data` (self, kwargs)
Collects the data about errors that was passed In case some values are missed, the default ones will be taken.
- def `update_data` (self, time=None, obs=None, kwargs)
Updates the data for the given parameters.
- def `update_transit_params` (self, rp_new, u_new)
Updates the parameters of the model.
- def `update_kernel_params` (self, a_new=None, gamma_new=None, variance_new=None)
Updates the hyperparameters of the kernel function.
- def `updateTransitMode` (self)
Updates the transit model parameters.
- def `model` (self)
Returns the flux values array.
- def `model` (self, params)
- def `meanfnc` (self, t)
Mean function for the kernel meaan estimation.
- def `kernelfnc` (self, x1, x2, p=None)
Computes the kernel function for the arbitrary sources of errors in the observations.
- def `lnlike_gp` (self)
Computes the log likelihood from gaussian process.
- def `lnprior_base` (self)
Checks if the batman parameters are within the predefined prior ranges.
- def `lnprior_gp` (self)
Checks if the kernel parameters are within the predefined prior ranges.
- def `lnprob_gp` (self)
Computes the log probability of the parameters for the given data.
- def `sample_conditional` (self, p, t, y, yerr)
For a given set of parameters get predicted y values at t, and separate this into the transit signal component and the noise component.
- def `lnprob_mcmc` (self, p, t, y, yerr)
MCMC API for the Transit Model object.

Public Attributes

- `batman_default_params`
- `transit_default_priors`
- `kernel_default_params`
- `kernel_default_priors`
- `data_defaults`
- `n_errors`
- `err_names`
- `errors_list`
- `params`

- **batman_model**
- **model**
- **data_dict**
- **model_initialized**
- **kernel_type**
- **Inprob**

5.12.1 Detailed Description

More details.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 `__init__()`

```
def exospec.TransitModel.TransitModel.__init__ (
    self,
    kwargs )
```

Takes the disctionary of the parameters and data to customize the object. In case some values are missed, the default one are used.

Parameters

<i>self</i>	The object pointer
<i>**kwargs</i>	Accepts the dictionary of data, transit and kernel parameters

5.12.3 Member Function Documentation

5.12.3.1 `kernelnc()`

```
def exospec.TransitModel.TransitModel.kernelnc (
    self,
    x1,
    x2,
    p = None )
```

Parameters

<i>self</i>	The object pointer
<i>x1</i>	First time coordinate
<i>x2</i>	Second time coordinate
<i>p(=None)</i>	Kernel auxiliary parameters

Returns

Covariance between two points in time

5.12.3.2 Inlike_gp()

```
def exospec.TransitModel.TransitModel.Inlike_gp (  
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Log likelihood of a set of observations under the Gaussian process model.

5.12.3.3 Inprior_base()

```
def exospec.TransitModel.TransitModel.Inprior_base (  
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Returns 0 in case transit parameters within the prior range and -inf otherwise

5.12.3.4 Inprior_gp()

```
def exospec.TransitModel.TransitModel.Inprior_gp (  
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Returns -inf in case parameters out of the range and 0.0 if within the prior range

5.12.3.5 Inprob_gp()

```
def exospec.TransitModel.TransitModel.lnprob_gp (
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Log probability of the parameters

5.12.3.6 Inprob_mcmc()

```
def exospec.TransitModel.TransitModel.lnprob_mcmc (
    self,
    p,
    t,
    y,
    yerr )
```

Parameters

<i>self</i>	The object pointer
<i>p</i>	Parameters of the transit
<i>t</i>	Time data
<i>y</i>	Observations data
<i>yerr</i>	Errors data

Returns

Log probability of the chosen parameters

5.12.3.7 meanfnc()

```
def exospec.TransitModel.TransitModel.meanfnc (
    self,
    t )
```


Parameters

<i>self</i>	The object pointer
<i>t</i>	The time data

5.12.3.8 model()

```
def exospec.TransitModel.TransitModel.model (
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

Returns

Model-generated observation for the given transit parameters

5.12.3.9 read_errors_data()

```
def exospec.TransitModel.TransitModel.read_errors_data (
    self,
    kwargs )
```

Parameters

<i>self</i>	The object pointer
<i>**kwargs</i>	Dictionary of the parameters to pass

5.12.3.10 read_limb_dark_params()

```
def exospec.TransitModel.TransitModel.read_limb_dark_params (
    self,
    kwargs )
```

Parameters

<i>self</i>	The object pointer
<i>**kwargs</i>	Dictionary of the parameters to pass

5.12.3.11 sample_conditional()

```
def exospec.TransitModel.TransitModel.sample_conditional (
    self,
    p,
    t,
    y,
    yerr )
```

Parameters

<i>self</i>	The object pointer
<i>p</i>	Parameters of the transit
<i>t</i>	Time data
<i>y</i>	Observations data
<i>yerr</i>	Errors data

Returns

Predicted observations

5.12.3.12 set_values()

```
def exospec.TransitModel.TransitModel.set_values (
    self,
    dict_of_values,
    kwargs )
```

In case some values are missed, the default ones will be taken.

Parameters

<i>self</i>	The object pointer
<i>dict_of_values</i>	Dictionary of the parameters to pass

5.12.3.13 update_data()

```
def exospec.TransitModel.TransitModel.update_data (
    self,
    time = None,
    obs = None,
    kwargs )
```

Parameters

<i>self</i>	The object pointer
<i>time(=None)</i>	Time data
<i>obs(=None)</i>	Observations data
<i>**kwargs</i>	Handles arbitrary number of the errors that was passed

5.12.3.14 update_kernel_params()

```
def exospec.TransitModel.TransitModel.update_kernel_params (
    self,
    a_new = None,
    gamma_new = None,
    variance_new = None )
```

Parameters

<i>self</i>	The object pointer
<i>a_new</i>	New value of the kernel_a
<i>gamma_new</i>	New value of the kernel_gamma

5.12.3.15 update_transit_params()

```
def exospec.TransitModel.TransitModel.update_transit_params (
    self,
    rp_new,
    u_new )
```

Parameters

<i>self</i>	The object pointer
<i>rp_new</i>	New value of the rp parameter
<i>u_new</i>	New list of values for the limb darkening

5.12.3.16 updateTransitMode()

```
def exospec.TransitModel.TransitModel.updateTransitMode (
    self )
```

Parameters

<i>self</i>	The object pointer
-------------	--------------------

The documentation for this class was generated from the following file:

- TransitModel.py

Chapter 6

File Documentation

6.1 deliverables.py File Reference

Deliverables module to parse MCMC data into latex code and plots.

Functions

- def `exospec.deliverables.latex_table` (LC_dic, transit_params, hyper_params, separate_flag, confidence, filename)
Writes the best fit transit & hyper parameters from the MCMC fit as a latex table.
- def `exospec.deliverables.simple_table` (LC_dic, filename)
Writes the best fit transit parameters from MCMC fit to an output file.
- def `exospec.deliverables.plot_single_wavelength` (wl_id, mcmc_obj, model, extra_burnin_steps=0, theta_↵ true=None, plot_transit_params=True, plot_hyper_params=True, save_as_dir="")
Produces triangle, walker and lightcurve plots for the MCMC results for a single wavelength.
- def `exospec.deliverables.plot_transmission_spec` (LC_dic, save_as_dir="")
Produces a plot of the best-fit radius as a function of wavelengths.
- def `exospec.deliverables.get_median_and_errors` (flatchain)
Obtains the median parameter values and 1 sigma errors from the MCMC flatchain.
- def `exospec.deliverables.post_processing_all_wl` (input_param_dic, LC_dic)
Runs the post processing once all wavelength mcmc chains are finished This includes saving a simple table, latex table, and transmission spectrum.

6.1.1 Function Documentation

6.1.1.1 `get_median_and_errors()`

```
def exospec.deliverables.get_median_and_errors (
    flatchain )
```

Parameters

<i>flatchain</i>	A 2D numpy array with all the samples for each of the transit and hyper parameters
------------------	--

Returns

median Array of median value for each parameter
 err_plus Array of upper error for each parameter
 err_minus Array of lower error for each parameter

6.1.1.2 latex_table()

```
def exospec.deliverables.latex_table (
    LC_dic,
    transit_params,
    hyper_params,
    separate_flag,
    confidence,
    filename )
```

Rows correspond to the light curve with a specified wavelength, and each column corresponds to a given parameter. The use has an option of outputting a single table with transit and hyper parameters, or outputting two separate tables.

Parameters

<i>LC_dic</i>	A light curve dictionary with finished chains stored.
<i>transit_params</i>	List of transit parameter names.
<i>hyper</i>	List of hyper parameter names.
<i>separate_flag</i>	Flag to toggle if combine parameters in a single table or separate them into two tables
<i>confidence</i>	The confidence level desired for the parameter bounds.
<i>filename</i>	The name of the file to write the table in.

6.1.1.3 plot_single_wavelength()

```
def exospec.deliverables.plot_single_wavelength (
    wl_id,
    mcmc_obj,
    model,
    extra_burnin_steps = 0,
    theta_true = None,
    plot_transit_params = True,
    plot_hyper_params = True,
    save_as_dir = "" )
```

Parameters

<i>wl_id</i>	The wavelength being processed
<i>mcmc_obj</i>	An object of the MCMC class that has been run for the wavelength <i>wl_id</i>
<i>model</i>	A function that takes a set of parameters and produces plots of the transit's lightcurve
<i>extra_burnin_steps</i>	Number of steps (in addition to <i>burnin_steps</i> from run) at the start of each chain to neglect
<i>theta_true</i>	Numpy array of true parameter values if known (used for test data)
<i>plot_transit_params</i>	Boolean value specifying whether or not to plot the transit parameters
<i>plot_hyper_params</i>	Boolean value specifying whether or not to plot the hyper parameters
<i>save_as_dir</i>	Directory where plot should be saved. Default is current working Directory

6.1.1.4 plot_transmission_spec()

```
def exospec.deliverables.plot_transmission_spec (
    LC_dic,
    save_as_dir = "" )
```

Parameters

<i>LC_dic</i>	A lightcurve dictionary with a lightcurve object for each wavelengths
<i>save_as_dir</i>	Directory where plot should be saved. Default is current working Directory

6.1.1.5 post_processing_all_wl()

```
def exospec.deliverables.post_processing_all_wl (
    input_param_dic,
    LC_dic )
```

Parameters

<i>input_param_dic</i>	
<i>LC_dic</i>	

6.1.1.6 simple_table()

```
def exospec.deliverables.simple_table (
    LC_dic,
    filename )
```

Output file columns are: wavelength, radius of planet, first limb darkening parameter, second limb darkening parameter, followed by lower bound of corresponding confidence intervals then upper bound of corresponding confidence intervals. Note that this currently only works for the quadratic limb darkening model.

Parameters

<i>LC_dic</i>	A light curve dictionary with finished chains stored.
<i>filename</i>	The name of the file to write the table in.

6.2 exospec_main.py File Reference

Fits the transits for multiple wavelengths to produce the transmission spectrum.

Variables

- `exospec_main.input_file` = `read_input(sys.argv[1])`
- `exospec_main.input_param_dic` = `input_file.param_dic`
- `exospec_main.mpi_flag` = `input_param_dic['mpi_flag']`
- `exospec_main.lc_path` = `input_param_dic['lc_path']`
- `exospec_main.wave_bin_size` = `input_param_dic['wave_bin_size']`
- `exospec_main.LC` = `exospec.lc_class.LightCurve(lc_path, wave_bin_size)`
- `exospec_main.LC_dic` = `LC.LC_dic`
- `exospec_main.comm` = `MPI.COMM_WORLD`
- `exospec_main.rank` = `comm.Get_rank()`
- `int exospec_main.j` = 0
- `exospec_main.jmax` = `len(LC_dic) / comm.Get_size()`
- `exospec_main.dest`
- `exospec_main.tag`
- `exospec_main.chain`
- `exospec_main.source`
- `exospec_main.i`
- `exospec_main.obj_chain`

6.2.1 Variable Documentation

6.2.1.1 chain

`exospec_main.chain`

Initial value:

```
1 = np.empty(
2     np.shape(LC_dic[LC_dic.keys()[0]].obj_chain))
```

6.3 fitting_single_lc.py File Reference

Fits a transit lightcurve for a single wavelength.

Functions

- def `exospec.fitting_single_lc.run_mcmc_single_wl` (input_param_dic, LC_dic, wl_id)
Definition which carries out the procedure for fitting a single wavelength.

6.3.1 Function Documentation

6.3.1.1 run_mcmc_single_wl()

```
def exospec.fitting_single_lc.run_mcmc_single_wl (
    input_param_dic,
    LC_dic,
    wl_id )
```

This follows the steps of: initializing a TransitModel object with correct data for that wavelength, initializing an mcmc object and running it, saving the results of mcmc run, and making walker and triangle plots for that mcmc run.
input_param_dic A dictionary of input parameter values
LC_dic A light curve dictionary containing the wavelength data
wl_id The key for the wavelength to be fit

Return values

0	if successful
---	---------------

Index

- `__init__`
 - `exospec::TransitModel::TransitModel`, 24
 - `exospec::mcmc::MCMC`, 17
- `chain`
 - `exospec_main.py`, 34
- `deliverables.py`, 31
 - `get_median_and_errors`, 31
 - `latex_table`, 32
 - `plot_single_wavelength`, 32
 - `plot_transmission_spec`, 33
 - `post_processing_all_wl`, 33
 - `simple_table`, 33
- `exospec.lc_class.DifferentFileSizes`, 9
- `exospec.lc_class.DifferentParamNum`, 9
- `exospec.lc_class.EmptyFile`, 10
- `exospec.lc_class.EmptyFolder`, 10
- `exospec.lc_class.IncorrectNameFormat`, 11
- `exospec.lc_class.LightCurve`, 11
- `exospec.lc_class.LightCurveData`, 13
- `exospec.mcmc.MCMC`, 16
- `exospec.read_input.EmptyFile`, 10
- `exospec.read_input.NoInput`, 21
- `exospec.read_input.read_input`, 21
- `exospec.TransitModel.TransitModel`, 22
- `exospec::TransitModel::TransitModel`
 - `__init__`, 24
 - `kernelfnc`, 24
 - `Inlike_gp`, 25
 - `Inprior_base`, 25
 - `Inprior_gp`, 25
 - `Inprob_gp`, 26
 - `Inprob_mcmc`, 26
 - `meanfnc`, 26
 - `model`, 27
 - `read_errors_data`, 27
 - `read_limb_dark_params`, 27
 - `sample_conditional`, 28
 - `set_values`, 28
 - `update_data`, 28
 - `update_kernel_params`, 29
 - `update_transit_params`, 29
 - `updateTransitMode`, 29
- `exospec::lc_class::LightCurve`
 - `LC_dic`, 12
 - `new_wave_length`, 12
 - `store_transit_model`, 12
 - `wave_length`, 12
- `exospec::lc_class::LightCurveData`
 - `ferr`, 14
 - `flux`, 14
 - `len_file`, 14
 - `new_time_bin`, 14
 - `param_list`, 15
 - `param_name`, 15
 - `param_num`, 15
 - `plot_flux_param`, 15
 - `plot_flux_time`, 16
 - `time`, 16
- `exospec::mcmc::MCMC`
 - `__init__`, 17
 - `get_mean_acceptance_fraction`, 18
 - `get_median_and_errors`, 18
 - `light_curve_plot`, 18
 - `run`, 19
 - `save_chain`, 19
 - `triangle_plot`, 20
 - `walker_plot`, 20
- `exospec::read_input::read_input`
 - `param_dic`, 22
- `exospec_main.py`, 34
 - `chain`, 34
- `ferr`
 - `exospec::lc_class::LightCurveData`, 14
- `fitting_single_lc.py`, 34
 - `run_mcmc_single_wl`, 35
- `flux`
 - `exospec::lc_class::LightCurveData`, 14
- `get_mean_acceptance_fraction`
 - `exospec::mcmc::MCMC`, 18
- `get_median_and_errors`
 - `deliverables.py`, 31
 - `exospec::mcmc::MCMC`, 18
- `kernelfnc`
 - `exospec::TransitModel::TransitModel`, 24
- `LC_dic`
 - `exospec::lc_class::LightCurve`, 12
- `latex_table`
 - `deliverables.py`, 32
- `len_file`
 - `exospec::lc_class::LightCurveData`, 14
- `light_curve_plot`
 - `exospec::mcmc::MCMC`, 18
- `Inlike_gp`

- exospec::TransitModel::TransitModel, 25
- Inprior_base
 - exospec::TransitModel::TransitModel, 25
- Inprior_gp
 - exospec::TransitModel::TransitModel, 25
- Inprob_gp
 - exospec::TransitModel::TransitModel, 26
- Inprob_mcmc
 - exospec::TransitModel::TransitModel, 26
- meanfnc
 - exospec::TransitModel::TransitModel, 26
- model
 - exospec::TransitModel::TransitModel, 27
- new_time_bin
 - exospec::lc_class::LightCurveData, 14
- new_wave_length
 - exospec::lc_class::LightCurve, 12
- param_dic
 - exospec::read_input::read_input, 22
- param_list
 - exospec::lc_class::LightCurveData, 15
- param_name
 - exospec::lc_class::LightCurveData, 15
- param_num
 - exospec::lc_class::LightCurveData, 15
- plot_flux_param
 - exospec::lc_class::LightCurveData, 15
- plot_flux_time
 - exospec::lc_class::LightCurveData, 16
- plot_single_wavelength
 - deliverables.py, 32
- plot_transmission_spec
 - deliverables.py, 33
- post_processing_all_wl
 - deliverables.py, 33
- read_errors_data
 - exospec::TransitModel::TransitModel, 27
- read_limb_dark_params
 - exospec::TransitModel::TransitModel, 27
- run
 - exospec::mcmc::MCMC, 19
- run_mcmc_single_wl
 - fitting_single_lc.py, 35
- sample_conditional
 - exospec::TransitModel::TransitModel, 28
- save_chain
 - exospec::mcmc::MCMC, 19
- set_values
 - exospec::TransitModel::TransitModel, 28
- simple_table
 - deliverables.py, 33
- store_transit_model
 - exospec::lc_class::LightCurve, 12
- time
 - exospec::lc_class::LightCurveData, 16
- triangle_plot
 - exospec::mcmc::MCMC, 20
- update_data
 - exospec::TransitModel::TransitModel, 28
- update_kernel_params
 - exospec::TransitModel::TransitModel, 29
- update_transit_params
 - exospec::TransitModel::TransitModel, 29
- updateTransitMode
 - exospec::TransitModel::TransitModel, 29
- walker_plot
 - exospec::mcmc::MCMC, 20
- wave_length
 - exospec::lc_class::LightCurve, 12