

---

# **Bayesian inference and forecast of COVID-19**

***Release 0.0.10***

**Jan 12, 2023**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>First Steps</b>	<b>5</b>
<b>3</b>	<b>Disclaimer</b>	<b>7</b>
<b>4</b>	<b>Models</b>	<b>9</b>
<b>5</b>	<b>Data Retrieval</b>	<b>13</b>
<b>6</b>	<b>Plotting</b>	<b>15</b>
<b>7</b>	<b>Indices and tables</b>	<b>17</b>
	<b>Python Module Index</b>	<b>19</b>
	<b>Index</b>	<b>21</b>



**Warning:** Important: This is the documentation of code no longer in active development. This is the link to the current docs: [covid19-inference](#)

**Warning:** Important: This is the documentation of code no longer in active development This is the link to the current docs: [covid19-inference](#)



# CHAPTER 1

---

## Installation

---

There exists three different possibilities to run the models:

1. Clone the repository:

```
git clone https://github.com/Priesemann-Group/covid19_inference_forecast
```

2. Install the module via pip

```
pip install git+https://github.com/Priesemann-Group/covid19_inference_forecast.git
```

3. Run the notebooks directly in Google Colab. At the top of the notebooks files there should be a symbol which opens them directly in a Google Colab instance.



## CHAPTER 2

---

### First Steps

---

To get started, we recommend to look at one of the currently two example notebooks:

1. **SIR model with changes points** This model is extensively discussed in our paper: [Inferring COVID-19 spreading rates and potential change points for case number forecasts](#)
2. **SEIR-like model with extensions** The second notebook runs a probably more realistic SEIR-like model. Details are found in the reference guide [Models](#) .

**Warning:** Important: This is the documentation of code no longer in active development. This is the link to the current docs: [covid19-inference](#)



## CHAPTER 3

---

### Disclaimer

---

We evaluate the data provided by the John Hopkins University [link](#). We exclude any liability with regard to the quality and accuracy of the data used, and also with regard to the correctness of the statistical analysis. The evaluation of the different growth phases represents solely our personal opinion.

The number of cases reported may be significantly lower than the number of people actually infected. Also, we must point out that week-ends and changes in the test system may lead to fluctuations in reported cases that have no equivalent in actual case numbers.

Certainly, at this stage all statistical predictions are subject to great uncertainty because the general trends of the epidemic are not yet clear. In any case, the statistical trends that we interpret from the data are only suitable for predictions if the measures taken by the government and authorities to contain the pandemic remain in force and are being followed by the population. We must also point out that, even if the statistics indicate that the epidemic is under control, we may at any time see a resurgence of infection figures until the disease is eradicated worldwide.

**Warning:** Important: This is the documentation of code no longer in active development. This is the link to the current docs: [covid19-inference](#)



```
covid19_inference.SIR_with_change_points(new_cases_obs, change_points_list,
                                          date_begin_simulation, num_days_sim,
                                          diff_data_sim, N, priors_dict=None,
                                          weekends_modulated=False, week-
                                          end_modulation_type='step')
```

### Parameters

- **new\_cases\_obs** (*list or array*) – Timeseries (day over day) of newly reported cases (not the total number)
- **change\_points\_list** (*list of dicts*) – List of dictionaries, each corresponding to one change point.

Each dict can have the following key-value pairs. If a pair is not provided, the respective default is used.

- **pr\_mean\_date\_begin\_transient** : *datetime.datetime*, NO default
- **pr\_median\_lambda** : *number*, same as default priors, below
- **pr\_sigma\_lambda** : *number*, same as default priors, below
- **pr\_sigma\_date\_begin\_transient** : *number*, 3
- **pr\_median\_transient\_len** : *number*, 3
- **pr\_sigma\_transient\_len** : *number*, 0.3
- **date\_begin\_simulation** (*datetime.datetime*) – The begin of the simulation data
- **num\_days\_sim** (*integer*) – Number of days to forecast into the future
- **diff\_data\_sim** (*integer*) – Number of days that the simulation-begin predates the first data point in *new\_cases\_obs*. This is necessary so the model can fit the reporting delay. Set this parameter to a value larger than what you expect to find for the reporting delay.
- **N** (*number*) – The population size. For Germany, we used 83e6

- **priors\_dict** (*dict*) – Dictionary of the prior assumptions

**Possible key-value pairs (and default values) are:**

- pr\_beta\_I\_begin : number, default = 100
- pr\_median\_lambda\_0 : number, default = 0.4
- pr\_sigma\_lambda\_0 : number, default = 0.5
- pr\_median\_mu : number, default = 1/8
- pr\_sigma\_mu : number, default = 0.2
- pr\_median\_delay : number, default = 8
- pr\_sigma\_delay : number, default = 0.2
- pr\_beta\_sigma\_obs : number, default = 10
- week\_end\_days : tuple, default = (6,7)
- pr\_mean\_weekend\_factor : number, default = 0.7
- pr\_sigma\_weekend\_factor : number, default = 0.17
- **weekends\_modulated** (*bool*) – Whether to add the prior that cases are less reported on week ends. Multiplies the new cases numbers on weekends by a number between 0 and 1, given by a prior beta distribution. The beta distribution is parametrised by pr\_mean\_weekend\_factor and pr\_sigma\_weekend\_factor
- **weekend\_modulation\_type** (*'step' or 'abs\_sine':*) – whether the week-ends are modulated by a step function, which only multiplies the days given by week\_end\_days by the week\_end\_factor, or whether the whole week is modulated by an abs(sin(x)) function, with an offset with flat prior.

**Returns** *pymc3.Model* – Returns an instance of pymc3 model with the change points

```
covid19_inference.SEIR_with_extensions (new_cases_obs,          change_points_list,
                                         date_begin_simulation,  num_days_sim,
                                         diff_data_sim,          N,      priors_dict=None,
                                         with_random_walk=True,   week-
                                         ends_modulated=False,     week-
                                         end_modulation_type='step')
```

**This model includes 3 extensions to the *SIR\_model\_with\_change\_points*:**

1. The SIR model now includes a incubation period during which infected people are not infectious, in the spirit of an SEIR model. In contrast to the SEIR model, the length of incubation period is not exponentially distributed but has a lognormal distribution.
2. People that are infectious are observed with a delay that is now lognormal distributed. In the *SIR\_model\_with\_change\_points* we assume a fixed delay between infection and observation.
3. *lambda\_t* has an additive term given by a Gaussian random walk. Thereby, we want to fit any deviation in *lambda\_t* that is not captured by the change points. If the change points are wisely chosen, and the rest of the model captures the dynamics well, one would expect that the amplitude of the random walk is small. In this case, the posterior distribution of *sigma\_random\_walk* will be small.

**Parameters**

- **new\_cases\_obs** (*list or array*) – Timeseries (day over day) of newly reported cases (not the total number)

- **change\_points\_list** (*list of dicts*) – List of dictionaries, each corresponding to one change point

Each dict can have the following key-value pairs. If a pair is not provided, the respective default is used.

- pr\_mean\_date\_begin\_transient: datetime.datetime, NO default
- pr\_median\_lambda: float, default: 0.4
- pr\_sigma\_lambda: float, default: 0.5
- pr\_sigma\_begin\_transient: float, default: 3
- pr\_median\_transient\_len: float, default: 3
- pr\_sigma\_transient\_len: float, default: 0.3
- **date\_begin\_simulation** (*datetime.datetime.*) – The begin of the simulation data
- **num\_days\_sim** (*integer*) – Number of days to forecast into the future
- **diff\_data\_sim** (*integer*) – Number of days that the simulation-begin predates the first data point in *new\_cases\_obs*. This is necessary so the model can fit the reporting delay. Set this parameter to a value larger than what you expect to find for the reporting delay.
- **N** (*number*) – The population size. For Germany, we used 83e6
- **priors\_dict** (*dict*) – Dictionary of the prior assumptions

**Possible key-value pairs (and default values) are:**

- pr\_beta\_I\_begin : number, default: 100
- pr\_beta\_E\_begin\_scale : number, default: 10
- pr\_median\_lambda\_0 : number, default: 2
- pr\_sigma\_lambda\_0 : number, default: 0.7
- pr\_median\_mu : number, default: 1/3
- pr\_sigma\_mu : number, default: 0.3
- pr\_median\_delay : number, default: 5
- pr\_sigma\_delay : number, default: 0.2
- scale\_delay : number, default: 0.3
- pr\_beta\_sigma\_obs : number, default: 10
- pr\_sigma\_random\_walk : number, default: 0.05
- **pr\_mean\_median\_incubation** [number, default: 5] <https://www.ncbi.nlm.nih.gov/pubmed/32150748> <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7014672/> about -1 day compared to the sources day because persons likely become infectious before.
- **pr\_sigma\_median\_incubation** [number, default: 1] The error from the sources above is smaller, but as the -1 day is a very rough estimate, we take here a larger error.
- **sigma\_incubation** [number, default: 0.418] <https://www.ncbi.nlm.nih.gov/pubmed/32150748>
- **with\_random\_walk** (*boolean*) – whether to add a Gaussian walk to *lambda\_t*. computationally expensive

**Returns** *pymc3.Model* – Returns an instance of pymc3 model with the change points

**Warning:** Important: This is the documentation of code no longer in active development. This is the link to the current docs: [covid19-inference](#)

---

## Data Retrieval

---

`covid19_inference.data_retrieval.get_jhu_confirmed_cases()`

Attempts to download the most current data from the online repository of the Coronavirus Visual Dashboard operated by the Johns Hopkins University and falls back to the backup provided with our repo if it fails. Only works if the module is located in the repo directory.

**Returns** *confirmed\_cases* – pandas table with confirmed cases

`covid19_inference.data_retrieval.get_jhu_deaths()`

Attempts to download the most current data from the online repository of the Coronavirus Visual Dashboard operated by the Johns Hopkins University and falls back to the backup provided with our repo if it fails. Only works if the module is located in the repo directory.

**Returns** *deaths* – pandas table with reported deaths

`covid19_inference.data_retrieval.filter_one_country(data_df, country, begin_date, end_date)`

Returns the number of cases of one country as a np.array, given a dataframe returned by *get\_jhu\_confirmed\_cases* :param data\_df: :type data\_df: pandas.dataframe :param country: :type country: string :param begin\_date: :type begin\_date: datetime.datetime :param end\_date: :type end\_date: datetime.datetime

**Returns** *array*

**Warning:** Important: This is the documentation of code no longer in active development. This is the link to the current docs: [covid19-inference](#)



```
covid19_inference.plotting.plot_cases (trace, new_cases_obs, date_begin_sim, diff_data_sim,
                                       start_date_plot=None, end_date_plot=None,
                                       ylim=None, week_interval=None, colors=('tab:blue',
                                       'tab:orange'), country='Germany')
```

Plots the new cases, the fit, forecast and lambda\_t evolution

### Parameters

- **trace** (*trace returned by model*) –
- **new\_cases\_obs** (*array*) –
- **date\_begin\_sim** (*datetime.datetime*) –
- **diff\_data\_sim** (*float*) – Difference in days between the begin of the simulation and the data
- **start\_date\_plot** (*datetime.datetime*) –
- **end\_date\_plot** (*datetime.datetime*) –
- **ylim** (*float*) – the maximal y value to be plotted
- **week\_interval** (*int*) – the interval in weeks of the y ticks
- **colors** (*list with 2 colornames*) –

**Returns** *figure, axes*

```
covid19_inference.plotting.plot_hist (model, trace, ax, varname, colors=('tab:blue',
                                       'tab:orange'), bins=50)
```

Plots one histogram of the prior and posterior distribution of the variable varname.

### Parameters

- **model** (*pm.Model instance*) –
- **trace** (*trace of the model*) –
- **ax** (*matplotlib.axes instance*) –

- **varname** (*string*) –
- **colors** (*list with 2 colornames*) –
- **bins** (*number or array*) – passed to np.hist

**Returns** *None*

`covid19_inference.plotting.get_all_free_RVs_names(model)`

Returns the names of all free parameters of the model

**Parameters** `model` (*pm.Model instance*) –

**Returns** *list of variable names*

`covid19_inference.plotting.get_prior_distribution(model, x, varname)`

Given a model and variable name, returns the prior distribution evaluated at x. :param model: :type model: pm.Model instance :param x: :type x: list or array :param varname: :type varname: string

**Returns** *array*

## CHAPTER 7

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



### C

`covid19_inference`, [9](#)  
`covid19_inference.data_retrieval`, [13](#)  
`covid19_inference.plotting`, [15](#)



## C

`covid19_inference` (*module*), 1, 9  
`covid19_inference.data_retrieval` (*module*), 13  
`covid19_inference.plotting` (*module*), 15

## F

`filter_one_country()` (*in module covid19\_inference.data\_retrieval*), 13

## G

`get_all_free_RVs_names()` (*in module covid19\_inference.plotting*), 16  
`get_jhu_confirmed_cases()` (*in module covid19\_inference.data\_retrieval*), 13  
`get_jhu_deaths()` (*in module covid19\_inference.data\_retrieval*), 13  
`get_prior_distribution()` (*in module covid19\_inference.plotting*), 16

## P

`plot_cases()` (*in module covid19\_inference.plotting*), 15  
`plot_hist()` (*in module covid19\_inference.plotting*), 15

## S

`SEIR_with_extensions()` (*in module covid19\_inference*), 10  
`SIR_with_change_points()` (*in module covid19\_inference*), 9