

# Package ‘epiworldRcalibrate’

February 18, 2026

**URL** <https://sima-njf.github.io/epiworldRcalibrate/>,  
<https://github.com/sima-njf/epiworldRcalibrate>

**BugReports** <https://github.com/sima-njf/epiworldRcalibrate/issues>

**Title** Fast and Effortless Calibration of Agent-Based Models using  
Machine Learning

**Version** 0.1.2

**Description** Provides tools and pre-trained Machine Learning [ML] models for calibration of Agent-Based Models [ABMs] built with the R package 'epiworldR'. Implements methods described in Najafzadehkhoi, Vega Yon, Modenesi, and Meyer (2025) <[doi:10.48550/arXiv.2509.07013](https://doi.org/10.48550/arXiv.2509.07013)>. Users can automatically calibrate ABMs in seconds with pre-trained ML models, effectively focusing on simulation rather than calibration. Bridges a gap by allowing public health practitioners to run their own ABMs without the advanced technical expertise often required by calibration.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Suggests** testthat (>= 3.0.0), epiworldR

**Config/testthat/edition** 3

**Imports** reticulate (>= 1.2), utils

**Depends** R (>= 3.5)

**LazyData** true

**NeedsCompilation** no

**Author** Sima Najafzadehkhoi [aut, cre] (ORCID:  
<<https://orcid.org/0009-0002-6253-2910>>),  
George Vega Yon [aut] (ORCID: <<https://orcid.org/0000-0002-3171-0844>>),  
Bernardo Modenesi [aut],  
Centers for Disease Control and Prevention [fnd] (Award number  
1U01CK000585; 75D30121F00003)

**Maintainer** Sima Najafzadehkhoi <[sima.njf@utah.edu](mailto:sima.njf@utah.edu)>

**Repository** CRAN

**Date/Publication** 2026-02-18 18:00:02 UTC

## Contents

abc_calibration_params . . . . .	2
calibrate_sir . . . . .	3
check_model_status . . . . .	4
check_python_setup . . . . .	4
cleanup_model . . . . .	4
estimate_sir_parameters . . . . .	5
init_bilstm_model . . . . .	5
setup_python_deps . . . . .	6
utah_covid_data . . . . .	7
<b>Index</b>	<b>8</b>

---

abc\_calibration\_params

*ABC calibration results for COVID-19 SIR model*

---

### Description

Results from Approximate Bayesian Computation (ABC) calibration of an SIR network model fitted to Utah COVID-19 incidence data.

### Usage

abc\_calibration\_params

### Format

A named list with the following elements:

**contact\_rate** Posterior median of the contact rate.

**recovery\_rate** Posterior median of the recovery rate.

**transmission\_prob** Posterior median of the transmission probability.

**R0** Basic reproduction number computed from posterior medians.

**contact\_rate\_ci** 95 percent credible interval for the contact rate.

**recovery\_rate\_ci** 95 percent credible interval for the recovery rate.

**transmission\_prob\_ci** 95 percent credible interval for the transmission probability.

**calibration\_time\_seconds** Total runtime of the ABC calibration (seconds).

**n\_samples** Number of MCMC samples used in calibration.

**burnin** Number of burn-in iterations discarded.

**epsilon** ABC tolerance parameter.

**seed** Random seed used for reproducibility.

**posterior\_samples** Matrix of post-burn-in accepted parameter samples.

**acceptance\_rate** Acceptance rate of the ABC-MCMC algorithm (percent).

**Source**

Generated internally using the script `data-raw/process_covid_calibration.R`.

**Examples**

```
data("abc_calibration_params")
str(abc_calibration_params)
```

---

 calibrate\_sir

*Calibrate SIR Parameters (one-step convenience wrapper)*


---

**Description**

Optionally initializes the BiLSTM model and then calls `estimate_sir_parameters` on the provided data.

**Usage**

```
calibrate_sir(
  daily_cases,
  population_size,
  recovery_rate,
  model_dir = NULL,
  auto_init = TRUE
)
```

**Arguments**

<code>daily_cases</code>	Numeric vector of length 61 containing daily incidence counts (day 0 to day 60).
<code>population_size</code>	Single numeric value giving the total population size.
<code>recovery_rate</code>	Single numeric value giving the recovery rate parameter.
<code>model_dir</code>	Optional path to the directory containing the trained BiLSTM model and scaler files. If NULL, the package's bundled assets are used.
<code>auto_init</code>	Logical; if TRUE (default), automatically calls <code>init_bilstm_model</code> when the model is not yet loaded.

**Value**

Named numeric vector: `ptran`, `crate`, `R0`.

check\_model\_status      *Check model status*

---

**Description**

Check model status

**Usage**

check\_model\_status()

**Value**

A list describing model load status and Python config.

---

check\_python\_setup      *Check Python and package installation status*

---

**Description**

Reports whether Python is available, which virtual environment is in use, and whether each required Python package is importable.

**Usage**

check\_python\_setup()

**Value**

A list with Python installation details and package status.

---

cleanup\_model            *Unload the model from memory*

---

**Description**

Unload the model from memory

**Usage**

cleanup\_model()

**Value**

Invisibly TRUE on success.

---

`estimate_sir_parameters`*Estimate SIR Parameters from 61-day incidence*

---

**Description**

Estimate SIR Parameters from 61-day incidence

**Usage**

```
estimate_sir_parameters(daily_cases, population_size, recovery_rate)
```

**Arguments**

`daily_cases` Numeric vector of length 61 containing daily incidence counts for days 0 to 60.  
`population_size` Single numeric value giving the total population size used in the SIR model.  
`recovery_rate` Single numeric value giving the recovery rate parameter of the SIR model.

**Value**

Named numeric vector: `ptran`, `crate`, `R0`.

---

`init_bilstm_model`*Initialize BiLSTM Model for SIR Parameter Estimation*

---

**Description**

Loads the trained BiLSTM model and associated scaler objects into memory. Requires that Python dependencies have been set up via [setup\\_python\\_deps](#).

**Usage**

```
init_bilstm_model(model_dir = NULL, force_reload = FALSE)
```

**Arguments**

`model_dir` Optional path to model directory. Defaults to the package's bundled model files.  
`force_reload` Logical; reload even if already loaded.

**Value**

Invisibly returns TRUE on success.

---

setup\_python\_deps      *Set up Python dependencies for epiworldRcalibrate*

---

### Description

This function creates a dedicated virtual environment and installs all required Python packages (numpy, scikit-learn, joblib, PyTorch). Run this once after installing the package. This is kept separate from model functions so that package installation never happens automatically during normal use.

### Usage

```
setup_python_deps(force = FALSE)
```

### Arguments

force	Logical; if TRUE, removes and recreates the virtual environment from scratch. Default is FALSE.
-------	---

### Details

The function performs the following steps:

1. Finds a suitable Python installation (or uses RETICULATE\_PYTHON if set).
2. Creates a virtual environment named "epiworldRcalibrate".
3. Installs numpy, scikit-learn, joblib, and torch (CPU version) into the virtual environment.
4. Verifies all packages can be imported.

### Value

Invisibly returns TRUE on success.

### Examples

```
## Not run:  
# First-time setup (run once after installing the package)  
setup_python_deps()  
  
# Force reinstall if something went wrong  
setup_python_deps(force = TRUE)  
  
## End(Not run)
```

---

utah_covid_data	<i>Utah COVID-19 epidemic trends</i>
-----------------	--------------------------------------

---

**Description**

Daily COVID-19 epidemic indicators for the state of Utah, obtained from the Utah Department of Health COVID-19 dashboard.

**Usage**

```
utah_covid_data
```

**Format**

A data frame with 365 rows and 5 variables:

**Date** Date of the observation (Date).

**Daily.Cases** Number of newly reported COVID-19 cases (numeric).

**Smoothed.3.Day.Moving.Average** Smoothed daily cases using a 3-day moving average (numeric).

**X3.Day.Moving.Average** Alternative 3-day moving average of daily cases (numeric).

**Status** Indicator of reporting or epidemic status (character or factor).

**Source**

Utah Department of Health COVID-19 Dashboard.

**Examples**

```
data("utah_covid_data")  
head(utah_covid_data)
```

# Index

## \* datasets

- abc\_calibration\_params, [2](#)
- utah\_covid\_data, [7](#)

abc\_calibration\_params, [2](#)

calibrate\_sir, [3](#)

check\_model\_status, [4](#)

check\_python\_setup, [4](#)

cleanup\_model, [4](#)

estimate\_sir\_parameters, [3](#), [5](#)

init\_bilstm\_model, [3](#), [5](#)

setup\_python\_deps, [5](#), [6](#)

utah\_covid\_data, [7](#)