

---

# **Sambuca**

***Release 0.4.0***

**Jun 13, 2018**



---

## Contents

---

<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Contents</b>	<b>5</b>
<b>3</b>	<b>Indices and tables</b>	<b>15</b>
	<b>Python Module Index</b>	<b>17</b>



- Free software: CSIRO Open Source Software Licence Agreement
- Homepage: <https://github.com/csiro-aquatic-remote-sensing/sambuca>
- Documentation: <https://sambuca.readthedocs.io/en/latest/>
- Version: 0.4.0

Spatial application of the Semi-Analytical Model for Bathymetry, Un-mixing, and Concentration Assessment (SAM-BUCA).



# CHAPTER 1

---

## Installation

---

```
pip install sambuca
```

For everything else, please refer to the [project documentation](#).





## 2.1 Usage

To use Sambuca in a project:

```
import sambuca
```

Or:

```
import sambuca as sb
```

## 2.2 Tutorial

For a basic tutorial on using Sambuca, please see the IPython Notebooks.

### 2.2.1 To Do

- Write a tutorial as an IPython Notebook
- Place it somewhere
- Link to here

## 2.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

### 2.3.1 Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- The Sambuca package version.
- Detailed steps to reproduce the bug.

### 2.3.2 Documentation improvements

Sambuca could always use more documentation, whether as part of the official Sambuca docs, in docstrings, or even on the web in blog posts, articles, and such.

---

**Note:** This project uses Google-style docstrings. Contributed code should follow the same conventions. For examples, please see the [Napoleon examples](#), or the [Google Python Style Guide](#).

---

### 2.3.3 Feature requests and feedback

The best way to send feedback is to [file an issue](#)

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Or, implement the feature yourself and submit a pull request.

### 2.3.4 Development

1. Fork the *sambuca* repo on GitHub.
2. Clone your fork locally.
3. Create a feature branch.
4. When you're done making changes, run all the tests, doc builder and pylint checks:

```
py.test
tox
pylint ./src/sambuca/
sphinx-build -b html docs build/docs
```

Or, using the project makefile:

```
make clean lint tests html
tox
```

5. Commit your changes and push your branch to GitHub.
6. Submit a pull request through the GitHub website.

There is a makefile in the project root with targets for the most common development operations such as lint checks, running unit tests, building the documentation, and installing packages.

`Bumpversion` is used to manage the package version numbers. This ensures that the version number is correctly incremented in all required files. Please see the `bumpversion` documentation for usage instructions, and do not edit the version strings directly.

---

**Note:** Sphinx requires a working LaTeX installation to build the pdf documentation.

---

## 2.3.5 Pull Request Guidelines

If you need some code review or feedback while you're developing the code just make the pull request.

For merging, you should:

1. Include passing tests (`run py.test`).
2. Update documentation when there's new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

## 2.4 Authors

- Janet Anstee -
- Stephen Sagar -
- Hannelie Botha -
- Daniel Collins - [daniel.collins@csiro.au](mailto:daniel.collins@csiro.au)

## 2.5 Changelog

### 2.5.1 0.1.0 (2014-12-19)

- Minimal Sambuca model based on the cut-down Matlab code.

### 2.5.2 0.3.0 (2015-12-24)

- Working single-threaded demo of Sambuca minimisation on a raster in the *New Sambuca.ipynb* notebook.

### 2.5.3 0.4.0 (Wednesday June 13, 2018)

- First release to GitHub. No code changes, just documentation updates, config, Travis CI setup, Read the Docs setup, and stripping out test files.

## 2.6 Licence

### 2.6.1 CSIRO Open Source Software Licence Agreement (variation of the BSD / MIT License)

Copyright (c) 2014, Commonwealth Scientific and Industrial Research Organisation (CSIRO) ABN 41 687 119 230. All rights reserved. CSIRO is willing to grant you a licence to use Sambuca on the following terms, except where otherwise indicated for third party material. Redistribution and use of this software in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of CSIRO nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of CSIRO.

EXCEPT AS EXPRESSLY STATED IN THIS AGREEMENT AND TO THE FULL EXTENT PERMITTED BY APPLICABLE LAW, THE SOFTWARE IS PROVIDED “AS-IS”. CSIRO MAKES NO REPRESENTATIONS, WARRANTIES OR CONDITIONS OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY REPRESENTATIONS, WARRANTIES OR CONDITIONS REGARDING THE CONTENTS OR ACCURACY OF THE SOFTWARE, OR OF TITLE, MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, THE ABSENCE OF LATENT OR OTHER DEFECTS, OR THE PRESENCE OR ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE.

TO THE FULL EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL CSIRO BE LIABLE ON ANY LEGAL THEORY (INCLUDING, WITHOUT LIMITATION, IN AN ACTION FOR BREACH OF CONTRACT, NEGLIGENCE OR OTHERWISE) FOR ANY CLAIM, LOSS, DAMAGES OR OTHER LIABILITY HOWSOEVER INCURRED. WITHOUT LIMITING THE SCOPE OF THE PREVIOUS SENTENCE THE EXCLUSION OF LIABILITY SHALL INCLUDE: LOSS OF PRODUCTION OR OPERATION TIME, LOSS, DAMAGE OR CORRUPTION OF DATA OR RECORDS; OR LOSS OF ANTICIPATED SAVINGS, OPPORTUNITY, REVENUE, PROFIT OR GOODWILL, OR OTHER ECONOMIC LOSS; OR ANY SPECIAL, INCIDENTAL, INDIRECT, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES, ARISING OUT OF OR IN CONNECTION WITH THIS AGREEMENT, ACCESS OF THE SOFTWARE OR ANY OTHER DEALINGS WITH THE SOFTWARE, EVEN IF CSIRO HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH CLAIM, LOSS, DAMAGES OR OTHER LIABILITY.

APPLICABLE LEGISLATION SUCH AS THE AUSTRALIAN CONSUMER LAW MAY APPLY REPRESENTATIONS, WARRANTIES, OR CONDITIONS, OR IMPOSES OBLIGATIONS OR LIABILITY ON CSIRO THAT CANNOT BE EXCLUDED, RESTRICTED OR MODIFIED TO THE FULL EXTENT SET OUT IN THE EXPRESS TERMS OF THIS CLAUSE ABOVE “CONSUMER GUARANTEES”. TO THE EXTENT THAT SUCH CONSUMER GUARANTEES CONTINUE TO APPLY, THEN TO THE FULL EXTENT PERMITTED BY THE APPLICABLE LEGISLATION, THE LIABILITY OF CSIRO UNDER THE RELEVANT CONSUMER GUARANTEE IS LIMITED (WHERE PERMITTED AT CSIRO’S OPTION) TO ONE OF FOLLOWING REMEDIES OR SUBSTANTIALLY EQUIVALENT REMEDIES:

1. THE REPLACEMENT OF THE SOFTWARE, THE SUPPLY OF EQUIVALENT SOFTWARE, OR SUPPLYING RELEVANT SERVICES AGAIN;
2. THE REPAIR OF THE SOFTWARE;
3. THE PAYMENT OF THE COST OF REPLACING THE SOFTWARE, OF ACQUIRING EQUIVALENT SOFTWARE, HAVING THE RELEVANT SERVICES SUPPLIED AGAIN, OR HAVING THE SOFTWARE REPAIRED.

IN THIS CLAUSE, CSIRO INCLUDES ANY THIRD PARTY AUTHOR OR OWNER OF ANY PART OF THE SOFTWARE OR MATERIAL DISTRIBUTED WITH IT. CSIRO MAY ENFORCE ANY RIGHTS ON BEHALF OF THE RELEVANT THIRD PARTY.

## 2.7 API Reference

### 2.7.1 sambuca.AllParameters

```
sambuca.all_parameters.create_fixed_parameter_set(wavelengths, a_water, a_ph_star,
                                                    substrates, substrate_fraction=1,
                                                    chl=None, cdom=None,
                                                    nap=None, depth=None,
                                                    a_cdom_slope=0.0168052,
                                                    a_nap_slope=0.00977262,
                                                    bb_ph_slope=0.878138,
                                                    bb_nap_slope=None,
                                                    lambda0cdom=550.0,
                                                    lambda0nap=550.0,
                                                    lambda0x=546.0,
                                                    x_ph_lambda0x=0.00157747,
                                                    x_nap_lambda0x=0.0225353,
                                                    a_cdom_lambda0cdom=1.0,
                                                    a_nap_lambda0nap=0.00433,
                                                    bb_lambda_ref=550, water_refractive_index=1.33784,
                                                    theta_air=30.0, off_nadir=0.0,
                                                    q_factor=3.141592653589793)
```

Get an AllParameters tuple with Sambuca default values for use as a fixed parameter set.

Note that if the (wavelength, value) tuples returned from the sambuca\_core spectra loading functions are passed to this function, only the value arrays are stored in order to match the expectations of sambuca\_core.forward\_model.

```
class sambuca.all_parameters.AllParameters(chl, cdom, nap, depth, wavelengths, a_water,
                                           a_ph_star, num_bands, substrate_fraction,
                                           substrates, substrate_combinations,
                                           a_cdom_slope, a_nap_slope, bb_ph_slope,
                                           bb_nap_slope, lambda0cdom, lambda0nap,
                                           lambda0x, x_ph_lambda0x, x_nap_lambda0x,
                                           a_cdom_lambda0cdom, a_nap_lambda0nap,
                                           bb_lambda_ref, water_refractive_index,
                                           theta_air, off_nadir, q_factor)
```

namedtuple containing all sambuca-core forward\_model parameters.

**chl**

*float* – Concentration of chlorophyll (algal organic particulates).

**cdom**

*float* – Concentration of coloured dissolved organic particulates (CDOM).

**nap**

*float* – Concentration of non-algal particulates (NAP).

**depth**

*float* – Water column depth.

**substrates**

*array-like* – A list of benthic substrates.

**wavelengths**

*array-like* – Central wavelengths of the modelled spectral bands.

**a\_water**

*array-like* – Absorption coefficient of pure water

**a\_ph\_star**

*array-like* – Specific absorption of phytoplankton.

**num\_bands**

*int* – The number of spectral bands.

**substrate\_fraction**

*float* – Substrate proportion, used to generate a convex combination of two substrate members.

**substrate\_combinations**

*array-like* – The index in substrates for all combinations of pairs in substrates

**a\_cdom\_slope**

*float, optional* – slope of CDOM absorption

**a\_nap\_slope**

*float, optional* – slope of NAP absorption

**bb\_ph\_slope**

*float, optional* – Power law exponent for the phytoplankton backscattering coefficient.

**bb\_nap\_slope**

*float, optional* – Power law exponent for the NAP backscattering coefficient. If no value is supplied, the default behaviour is to use the `bb_ph_slope` value.

**lambda0cdom**

*float, optional* – Reference wavelength for CDOM absorption.

**lambda0nap**

*float, optional* – Reference wavelength for NAP absorption.

**lambda0x**

*float, optional* – Backscattering reference wavelength.

**x\_ph\_lambda0x**

*float, optional* – Specific backscatter of chlorophyll at `lambda0x`.

**x\_nap\_lambda0x**

*float, optional* – Specific backscatter of NAP at `lambda0x`.

**a\_cdom\_lambda0cdom**

*float, optional* – Absorption of CDOM at `lambda0cdom`.

**a\_nap\_lambda0nap**

*float, optional* – Absorption of NAP at `lambda0nap`.

**bb\_lambda\_ref**

*float, optional* – Reference wavelength for backscattering coefficient.

**water\_refractive\_index**

*float, optional* – refractive index of water.

**theta\_air**

*float, optional* – solar zenith angle in degrees.

**off\_nadir**

*float, optional* – off-nadir angle.

**q\_factor**

*float, optional* – q value for producing the R(0-) values from modelled remotely-sensed reflectance (rrs) values.

## 2.7.2 sambuca.error

**class** `sambuca.error.ErrorTerms` (*alpha, alpha\_f, f, lsq*)  
namedtuple containing the error terms.

**alpha**

*float* – TODO

**alpha\_f**

*float* – TODO

**f**

*float* – TODO

**lsq**

*float* – TODO

Sambuca Error Functions.

Used when assessing model closure during parameter estimation.

`sambuca.error.error_all` (*observed\_rrs, modelled\_rrs, nedr=None*)

Calculates all common error terms.

### Parameters

- **observed\_rrs** (*array-like*) – The observed reflectance(remotely-sensed).
- **modelled\_rrs** (*array-like*) – The modelled reflectance(remotely-sensed).
- **nedr** (*array-like*) – Noise equivalent difference in reflectance.

**Returns** The error terms.

**Return type** *ErrorTerms*

`sambuca.error.distance_alpha` (*observed\_rrs, modelled\_rrs, nedr=None*)

Calculates TODO

### Parameters

- **observed\_rrs** – The observed reflectance(remotely-sensed).
- **modelled\_rrs** – The modelled reflectance(remotely-sensed).
- **noise** – Optional spectral noise values.

Returns: TODO

`sambuca.error.distance_f` (*observed\_rrs, modelled\_rrs, nedr=None*)

Calculates TODO

### Parameters

- **observed\_rrs** – The observed reflectance(remotely-sensed).
- **modelled\_rrs** – The modelled reflectance(remotely-sensed).
- **noise** – Optional spectral noise values.

Returns: TODO

`sambuca.error.distance_alpha_f(observed_rrs, modelled_rrs, nedr=None)`

Calculates TODO

**Parameters**

- **observed\_rrs** – The observed reflectance(remotely-sensed).
- **modelled\_rrs** – The modelled reflectance(remotely-sensed).
- **noise** – Optional spectral noise values.

Returns: TODO

`sambuca.error.distance_lsq(observed_rrs, modelled_rrs, nedr=None)`

Calculates TODO

**Parameters**

- **observed\_rrs** – The observed reflectance(remotely-sensed).
- **modelled\_rrs** – The modelled reflectance(remotely-sensed).
- **noise** – Optional spectral noise values.

Returns: TODO

### 2.7.3 sambuca.FreeParameters

**class** `sambuca.free_parameters.FreeParameters(chl, cdom, nap, depth, substrate_fraction)`  
namedtuple containing the default Sambuca free parameters.

**chl**

*float* – Concentration of chlorophyll (algal organic particulates).

**cdom**

*float* – Concentration of coloured dissolved organic particulates (CDOM).

**nap**

*float* – Concentration of non-algal particulates (NAP).

**depth**

*float* – Water column depth.

**substrate\_fraction**

*float* – relative proportion of substrate1 and substrate2.

### 2.7.4 Objective Functions

**class** `sambuca.scipy_objective.SciPyObjective(sensor_filter, fixed_parameters, error_function=<function distance_f>, nedr=None)`

Configurable objective function for Sambuca parameter estimation, intended for use with the SciPy minimisation methods.

**observed\_rrs**

*array-like* – The observed remotely-sensed reflectance. This attribute must be updated when you require the objective instance to use a different value.



**id**

*integer* – The index of the substrate pair combination. This attribute must be updated when you require the objective instance to use a different substrate pair.

### 2.7.5 Pixel Result Handlers

Pixel result handlers are called by the parameter estimator with the final result for a pixel. Subclasses can implement various functionality, such as writing results to memory structures or files.

**class** `sambuca.pixel_result_handler.PixelResultHandler`  
Pixel result handler base class.

Pixel result handlers are called by the parameter estimator with the final result for a pixel. Subclasses can implement various functionality, such as writing results to memory structures or files.

**class** `sambuca.array_result_writer.ArrayResultWriter` (*width, height, sensor\_filter, nedr,*  
*fixed\_parameters*)  
Pixel result handler that writes pixel model outputs to in-memory numpy arrays.

**Note that the current implementation is writing a hard-coded set of outputs for the alpha implementation. The intention is to replace this with a data-driven system that only captures the outputs specified by the user.**

**In the short-term, this class could be modified to add additional outputs, but this is not an ideal long-term solution.**

The intent is that this class is used to capture model outputs of interest during raster processing. Once processing is complete, these outputs can then be used in various ways, such as writing to file (HDF, NetCDF, multi-band raster etc), plotting, or communication via message passing to a parallel processing manager. Each of these uses can be built on top of the basic array capture implemented in this class.

### 2.7.6 sambuca.sambuca

Sambuca module



## CHAPTER 3

---

### Indices and tables

---

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



### S

`sambuca.error`, [11](#)  
`sambuca.sambuca`, [13](#)



## A

buca.all\_parameters.AllParameters attribute),  
 10  
 a\_cdom\_slope (sambuca.all\_parameters.AllParameters  
 attribute), 10  
 a\_nap\_lambda0nap (sam-  
 buca.all\_parameters.AllParameters attribute),  
 10  
 a\_nap\_slope (sambuca.all\_parameters.AllParameters at-  
 tribute), 10  
 a\_ph\_star (sambuca.all\_parameters.AllParameters at-  
 tribute), 10  
 a\_water (sambuca.all\_parameters.AllParameters at-  
 tribute), 10  
 AllParameters (class in sambuca.all\_parameters), 9  
 alpha (sambuca.error.ErrorTerms attribute), 11  
 alpha\_f (sambuca.error.ErrorTerms attribute), 11  
 ArrayResultWriter (class in sam-  
 buca.array\_result\_writer), 13

## B

bb\_lambda\_ref (sambuca.all\_parameters.AllParameters  
 attribute), 10  
 bb\_nap\_slope (sambuca.all\_parameters.AllParameters at-  
 tribute), 10  
 bb\_ph\_slope (sambuca.all\_parameters.AllParameters at-  
 tribute), 10

## C

cdom (sambuca.all\_parameters.AllParameters attribute),  
 9  
 cdom (sambuca.free\_parameters.FreeParameters at-  
 tribute), 12  
 chl (sambuca.all\_parameters.AllParameters attribute), 9  
 chl (sambuca.free\_parameters.FreeParameters attribute),  
 12  
 create\_fixed\_parameter\_set() (in module sam-  
 buca.all\_parameters), 9

## D

depth (sambuca.all\_parameters.AllParameters attribute),  
 9  
 depth (sambuca.free\_parameters.FreeParameters at-  
 tribute), 12  
 distance\_alpha() (in module sambuca.error), 11  
 distance\_alpha\_f() (in module sambuca.error), 12  
 distance\_f() (in module sambuca.error), 11  
 distance\_lsqr() (in module sambuca.error), 12

## E

error\_all() (in module sambuca.error), 11  
 ErrorTerms (class in sambuca.error), 11

## F

f (sambuca.error.ErrorTerms attribute), 11  
 FreeParameters (class in sambuca.free\_parameters), 12

## I

id (sambuca.scipy\_objective.SciPyObjective attribute), 12

## L

lambda0cdom (sambuca.all\_parameters.AllParameters  
 attribute), 10  
 lambda0nap (sambuca.all\_parameters.AllParameters at-  
 tribute), 10  
 lambda0x (sambuca.all\_parameters.AllParameters  
 attribute), 10  
 lsqr (sambuca.error.ErrorTerms attribute), 11

## N

nap (sambuca.all\_parameters.AllParameters attribute), 9  
 nap (sambuca.free\_parameters.FreeParameters attribute),  
 12  
 num\_bands (sambuca.all\_parameters.AllParameters at-  
 tribute), 10

## O

observed\_rrs (sambuca.scipy\_objective.SciPyObjective  
 attribute), 12

off\_nadir (sambuca.all\_parameters.AllParameters attribute), [10](#)

## P

PixelResultHandler (class in sambuca.pixel\_result\_handler), [13](#)

## Q

q\_factor (sambuca.all\_parameters.AllParameters attribute), [11](#)

## S

sambuca.error (module), [11](#)

sambuca.sambuca (module), [13](#)

SciPyObjective (class in sambuca.scipy\_objective), [12](#)

substrate\_combinations (sambuca.all\_parameters.AllParameters attribute), [10](#)

substrate\_fraction (sambuca.all\_parameters.AllParameters attribute), [10](#)

substrate\_fraction (sambuca.free\_parameters.FreeParameters attribute), [12](#)

substrates (sambuca.all\_parameters.AllParameters attribute), [9](#)

## T

theta\_air (sambuca.all\_parameters.AllParameters attribute), [10](#)

## W

water\_refractive\_index (sambuca.all\_parameters.AllParameters attribute), [10](#)

wavelengths (sambuca.all\_parameters.AllParameters attribute), [10](#)

## X

x\_nap\_lambda0x (sambuca.all\_parameters.AllParameters attribute), [10](#)

x\_ph\_lambda0x (sambuca.all\_parameters.AllParameters attribute), [10](#)