PyProSAIL Documentation

Release 1.0

Robin Wilson

June 06, 2016

Contents

PyProSAIL is a Python interface to the ProSAIL combined leaf optical properties (PROSPECT) and canopy bi-directional reflectance (SAIL) model. The Python code simply interfaces with the Fortran code provided here and, currently at least, does nothing more than let you run it easily from code.

(Very) Quickstart

Install it from PyPI:

```
pip install pyprosail
```

Run it with some parameters, and look at the results:

```python
>>> import pyprosail
>>> result = pyprosail.run(1.5, 40, 8, 0, 0.01, 0.009, 1, 3, 0.01, 30, 10, 0, pyprosail.Planophile)
>>> result[:,0] # These are the wavelengths (in micrometres)
array([ 0.4  ,  0.401,  0.402, ...,  2.498,  2.499,  2.5  ])
>>> result[:,1] # These are the reflectances (as a fraction)
array([ 0.02119318,  0.02121176,  0.0212236 , ...,  0.02039977,
        0.02024749,  0.02028665])
```

That's basically it! For more details on how to install it, exactly what parameters you can give the `run` method, and how to use PyProSAIL with Py6S, then read on.

Installation

PyProSAIL is available on the Python Package Index (PyPI), and can therefore be installed using easy_install or pip:

```
pip install pyprosail
# or
easy_install pyprosail
```

Or a Windows Installer can be downloaded from the PyPI page. It depends on the `numpy` and `scipy` modules - which you are likely to already have installed if you do any scientific programming with Python. If not, I would suggest installing something like the Enthought Python Distribution, or Pythonxy.

The actual installation process is a little more complicated than it is for many pure Python modules, as it involves compiling the original Fortran code of the ProSAIL model. If running the installation commands above does not give any errors then you can assume everything is working fine - but if it does, then read on below.

As the installation requires compiling some Fortran code, it will need to be able to find a Fortran compiler. Thus, the most likely reason for the install to fail is because a Fortran compiler can't be found. The installation routine is fairly good at finding Fortran compilers (searching various sensible places - depending on the operating system), so simply installing one of the compilers listed below should make it all work:

- **Windows:** The easiest way is to use the Windows installer, but if you want to compile from the Fortran source then follow the steps 1-4 here to install the GFortran compiler and set it up so that it can be used by the PyProSAIL installation procedure.

- **OS X:** Install GCC by following the instructions here.

- **Linux:** Install GFortran using the package manager for your system, for example `apt-get install gfortran`.

If you have any issues with installation, please contact me using the details in the Support section below.

Running the model

To run the model using some parameters, simply import the model:

```python
import pyprosail
```

and then call the run method, as documented below.

pyprosail.**run**(*N*, *chloro*, *caroten*, *brown*, *EWT*, *LMA*, *psoil*, *LAI*, *hot_spot*, *solar_zenith*, *solar_azimuth*, *view_zenith*, *view_azimuth*, *LIDF*)

Runs the ProSAIL model with the given parameters and returns a 2D array with two columns: wavelengths and reflectances. The model is always run for the entire wavelength range of 0.4–2.5 micrometres.

Arguments:

- `N` – Structure co-efficient
- `chloro` – Chlorophyll content (ug per cm^2)
- `caroten` – Carotenoid content (ug per cm^2)
- `brown` – Brown pigment content (arbitrary units)
- `EWT` – Equivalent Water Thickness (cm)
- `LMA` – Leaf Mass per unit area (g per cm^2)
- `soil_reflectance` – Soil reflectance: wet soil = 0, dry soil = 1
- `LAI` – Leaf Area Index
- `hot_spot` – Hot spot parameter
- `solar_zenith` – Solar zenith angle (degrees)
- `solar_azimuth` – Solar azimuth angle (degrees)
- `view_zenith` – View zenith angle (degrees)
- `view_azimuth` – View azimuth angle (degrees)
- `LIDF` – Leaf distibution function parameter(s) (see below)

The leaf distribution function parameter(s) argument can be either:

- A single number giving the average leaf angle (in degrees) for an elipsoidal distribution. 0 is planophile and 90 is erectophile.

- A tuple giving (average leaf slope, bimodality parameter). These parameters must sum to be less than 1.0. For convenience a number of predefined options are given including:

    - pyprosail.Planophile

- – `pyprosail.Erectophile`

- – `pyprosail.Plagiophile`

- – `pyprosail.Extremophile`

- – `pyprosail.Spherical`

- – `pyprosail.Uniform`

Examples of valid values for the leaf distribution function parameter include:

- `30` (a 30 degree average leaf angle)

- `(0, -1)` (an average leaf slope of 0 and a bimodality parameter of -1)

- `PyProsail.Planophile`

Using with Py6S

PyProSAIL is very easy to use with Py6S (a Python interface to the 6S Radiative Transfer Model - see here). PyPro-SAIL can be used to create a spectrum, which can then be used by Py6S as the ground reflectance for a simulation:

```python
# Make sure you have both PyProSAIL and Py6S installed
import pyprosail
from Py6S import *

spectrum = pyprosail.run(1.5, 40, 8, 0, 0.01, 0.009, 1, 3, 0.01, 30, 0, 10, 0, pyprosail.Planophile)
s = SixS()
s.ground_reflectance = GroundReflectance.HomogeneousLambertian(spectrum)
s.run()
```

It's as simple as that! For more information on the parameters that you can pass to the GroundReflectance functions, see the Py6S documentation.

Support

Py6S was developed by the author as part of his PhD (which has not yet finished). He is still developing the software, but has many other demands on his time. He will try to answer any support queries as soon as possible, but he cannot guarantee a quick response.

Email: robin@rtwilson.com

P