**Library for retrieving radiative transfer model (RTM) variables from Hamiltonian Monte Carlo (HMC)** **under a Bayesian framework**

**Version: 0.1 February 2022**

This repository contains the library created for estimating radiative transfer model variables (LAI, chlorophyll content, etc.) from satellite reflectance observations. A full explanation of the methodology and validation is given in Wang et al.

The repository contains an example scripts RunHMC\_estimation.py that creates and run a HMC estimation experiment for PROSPECT+SAILH models from a small set of reflectance values from different sensors (taken from Wang et al.). The script uses the configuration file ‘Maize\_HMC\_Example.xlsx’ to generate and run an experiment. This configuration file (see description below) contains all the necessary information (observation, RTM variables, etc.) to run the experiment. Please note that the example runs only a single chain. Normally, at least four parallel chains are needed to test that the algorithm convergence has been reached (computing the R hat parameter, see Wang et al.).

The library HamiltonianRTM.py creates and run the variable estimation experiment. The method proposed divides the RTM variables in two groups (see Wang et al.): crop-specific variables (hierarchical level 0) which have the same distribution for all the observations belonging to the same crop; and observation-specific distribution (hierarchical level 1), which have independent distributions for each observations.

Once the HMC Experiment is run, the posterior distribution for each variable (level 0 or 1) is described from the values sampled by the algorithm at each iteration of the experiment (see RunHMC\_estimation.py).

The HamiltonianRTM uses model emulators to improve cost-efficiency, rather than running the original models. The emulators for leaf optical properties model, canopy reflectance model must be available in the ‘emulators/’ folder in shelve format. Example scripts are given for creating such emulators. To further improve computing efficiency, those RTM emulators provide reflectance in a selected number of wavelengths, not on the full optical spectrum. For that reason, emulators (neural networks) of the spectral response functions for each sensor (e.g. LANDSAT, S2, etc.) must be available in a single shelve file in the ‘emulators/’ folder. An example script to generate spectral response are given in that folder as well.

Here follows a description of the configuration file necessary to configure and run an experiment of variables estimation.

# Experiment configuration Excel File (the excel worksheets):

The config excel file is composed of five basic worksheets:

## *RunConfig* sheet

This sheet contains the necessary information on the radiative transfer models and data composition needed to create the Experiment object by the HamiltonianRTM.py library. This includes:

* the number of iterations to run
* the number of sensors to simulate
* the number of hierarchical levels for RTM variables. A value of 2 means that two levels will be considered: level 0 for groups of observations, and level 1 for individual observations.
* leaf optical model: the name of the shelve (in the ‘emulators’ folder) file containing the leaf optical properties model emulator used. See example in the folder ‘emulators’ to generate such emulator.
* soil optical model: the name of the soil reflectance model used (soilspec is embedded in the HamiltonianRTM.py library).
* direct beam model: the name of the model describing the proportion of direct light (skylspec is embedded in the RTMHamiltonian.py library).
* canopy reflectance model: the name of the shelve file (in the ‘emulators’ folder) containing the emulator of the canopy reflectance model. See example in the folder ‘emulators’ to generate such emulator.
* epsilon\_start: initial value for step size leapfrog iterations.
* NNs for band convolution: the name of the shelve file containing the neural networks (in the ‘emulators’ folder) to aggregate reflectance from individual wavelengths simulated by the emulators to sensor bands. A script to generate such NN is also given in the ‘emulators’ folder.
* sensor\_list: the name of the different sensors used. These names have to be the same used in the obs\_reflectance sheets (see below) and in the NN for bands convolution to name the different sensors used in the experiment.

## *obs\_reflectance* sheets

These sheets contain the sensor-specific reflectance observations sheets. The syntax for the sheet name is: obs\_reflectance*name-of-the-sensor* (e.g. obs\_reflectanceLANDSAT8). The name of the sensor has to be equal to any of those in the sensor\_list row of the RunConfig sheet.

Each sheet comprises the bands names, band reflectance and the level id. In the example, the level\_0 refers to the crop-specific level while level\_1 refers to the observation-specific level. Hence, for the level0 sampling, all reflectance (at different sites/ times) will be used to together, regarded as with the same id (level\_0=0), to derive the joint distribution of crop-specific variables. In contrast, for level\_1 sampling, each reflectance corresponds to an individual site, with sequential id, and the HMC will independently sample the distributions of observation-specific variables.

## *obs\_prior* sheet

This sheet contains parameters of prior probability distributions of one or more RTM variables for each observation.

In the example, GAI observations are used as priors for each reflectance observation contained in the sensor-specific reflectance sheets. The order of GAI values must be the same as the level\_1 column of the reflectance sheets. The value in the GAI column expresses the mean GAI of the prior distribution (most probable value for each observation). The standard deviation is given in the parameter sheet.

## *paramvalues* sheet

This sheets specifies the intervals and hyperparameters for all the RTM variables included in the inversion + sigma (the model error between observed and modelled reflectance):

* start\_value: the value for that variable in the first iteration. If the cell has the value ‘random’ it means that it will be chosen randomly between the min and max value when generating the Experiment object.
* min\_value: the lower physical limit for each variable.
* max\_value: the upper physical limit for each variable.
* Minv\_HMC (the mass parameter) and Phi\_HMC (the velocity parameter) are hyper parameters of the HMC algorithm. They have to be chose accordingly to the expected variance of each variable. These values do not change the posterior distribution results, just the number of iterations required to converge. For a better explanation of both hyperparameters, see *Bayesian Data Analysis (Gelman et al.).*
* level: the hierarchical level of each variable. If level= 0 a posterior distribution will be calculated for all the observations with the same level\_0 id in the obs\_reflectance sheets.
* prior\_dist: the prior distribution of each variable. Four types of distribution are considered: uniform, normal, n-i inverse gamma (non-informative inverse gamma), and inverse gamma. Normal and inverse gamma distribution require two (μ and σ) or three parameters (a, loc and scale, refer to the scipy Python package for more details), respectively, which can be specified in columns param\_distrib\_a and param\_distrib\_b (and param\_distrib\_c).

## *fixedparameters* sheet

This sheet includes columns with those parameters that are assumed to be known for each observation. In the example, the sun zenith angle, view zenith angle and relative azimuth angle are specified for each observation. The order of the observations have to be the same as in the obs\_reflectance sheets.

# How to cite this library:

If you use or modify this library, please cite the reference publication:

Wang, J., Lopez-Lozano, R., Weiss, M., Buis, S., Li, W., Liu, S., Baret, F. and Zhang, J. (2022): Crop specific inversion of PROSAIL to retrieve green area index (GAI) from several decametric satellites using a Bayesian framework. *Submitted to Remote Sensing of the Environment*.