UNIVERSITY OF CENTRAL FLORIDA

HOMER Users Manual

HELPER OF MY ETERNAL RETRIEVALS

Authors:
Michael D. HIMES

Supervisor:
Dr. Joseph HARRINGTON

May 21, 2025

Contents

1 Team Members

- Michael Himes[1], University of Central Florida (mhimes@knights.ucf.edu)
- Joseph Harrington, University of Central Florida
- David C. Wright, University of Central Florida
- Zacchaeus Scheffer, University of Central Florida

2 Introduction

This document describes HOMER, the Helper Of My Eternal Retrievals. HOMER is a Bayesian inverse modeling code. Given some data and uncertainties, the posterior distribution is determined for some model. While normal Bayesian modeling codes utilize some numerical forward model that implements the known process (e.g., radiative transfer for planetary atmospheres), HOMER uses a neural network (NN) approximation of the forward process, such as that trained by MARGE. HOMER uses the Large-selection Interface for Sampling Algorithms (LISA) for Markov Chain Monte Carlo (MCMC) and nested sampling (NS) Bayesian frameworks.

The detailed HOMER code documentation and User Manual [2] are provided with the package to assist users in its usage. For additional support, contact the lead author (see Section 1).

HOMER is released under the Reproducible Research Software License. For details, see https://planets.ucf.edu/resources/reproducible-research/software-license/.

The HOMER package is organized as follows:

```
HOMER
 ├── doc
 ├── example
 ├── lib
 └── modules
      ├── datasketches
      └── LISA
```

3 Installation

3.1 System Requirements

HOMER was developed on a Linux machine using the following versions of packages:

- Python 3.7.2
- Keras 2.2.4
- Numpy 1.16.2

[1] https://github.com/mdhimes/

[2] Most recent version of the manual available at https://exosports.github.io/HOMER/doc/HOMER_User_Manual.html

- Matplotlib 3.0.2
- Scipy 1.2.1
- sklearn 0.20.2
- Tensorflow 1.13.1
- CUDA 9.1.85
- cuDNN 7.5.00
- pymultinest 2.10
- ultranest 2.2.2

If installing the Datasketches library, CMake 3.12.0+ is required.

3.2 Install and Compile

To begin, obtain the latest stable version of HOMER.
First, decide on a local directory to hold HOMER. Let the path to this directory be ‘HOMER’. Now, clone the repository:

```
git clone --recursive https://github.com/exosports/HOMER HOMER/
cd HOMER/
```

HOMER contains a file to easily build a conda environment capable of executing the software. Create the environment via

```
conda env create -f environment.yml
```

Then, activate the environment:

```
conda activate marge_homer
```

Now, build the submodules:

```
make all
```

The Datasketches library is optional. If you do not want to install it, do

```
make mccubed
```

instead of ‘make all’.
You are now ready to run HOMER.

4 Example

The following script will walk a user through using HOMER for a retrieval on HD 189733 b, following from MARGE's example which trains an NN model for RT. These instructions are meant to be executed from a Linux terminal. The requirements for this example are less than the requirements for MARGE's example. Since MARGE's example is required to execute this example, we do not explicitly list all system requirements. Ensure you have at least 1 GB of free space before beginning.

To begin, copy the requisite files to a directory parallel to HOMER. Beginning from HOMER/,

```
mkdir ../run
cp -a ./example/* ../run/.
cd ../run
```

Note that, if the user is running this following the MARGE example, the 'run' directory should contain all of the files from both examples.

Now, execute HOMER for the desired sampler(s):

```
../HOMER/HOMER.py config_demc.cfg
../HOMER/HOMER.py config_snooker.cfg
../HOMER/HOMER.py config_multinest.cfg
../HOMER/HOMER.py config_ultranest.cfg
```

5 Program Inputs

The executable HOMER.py is the driver for the HOMER program. It takes a a configuration file of parameters. Once configured, HOMER is executed via the terminal as described in Section 4.

5.1 HOMER Configuration File

The HOMER configuration file is the main file that sets the arguments for a HOMER run. The arguments follow the format **`argument = value`**, where **`argument`** is any of the possible arguments described below.

The available options for a HOMER configuration file are listed below.

(Directories)

- inputdir : str. Directory containing HOMER inputs.
- outputdir : str. Directory containing HOMER outputs.

Run Parameters

- quantiles : bool. Determines whether to compute spectra quantiles. If the Datasketches library is not install, this setting has no effect.
- onlyplot : bool. Determines whether to skip the inference. Reproduces plots related to posterior.

- plot_PT : bool. Determines whether to compute the explored pressure–temperature profiles using the formulation of Line et al. (2013). Presently, this requires the model parameters to be ordered as log(kappa), log(gamma1), log(gamma2), alpha, beta, followed by any other parameters.

- credregion : bool. Determines whether to calculate the 68, 95, and 99% credible regions & uncertainties.

Data Normalization Parameters

- ilog : bool. Determines whether the NN takes the logarithm of the inputs. Alternatively, specify comma-, space-, or newline-separated integers to selectively take the log of certain inputs.

- olog : bool. Determines whether the NN predicts the log of the outputs. Alternatively, specify comma-, space-, or newline-separated integers to selectively take the log of certain outputs.

- normalize : bool. Determines whether to standardize the data by its mean and standard deviation.

- scale : bool. Determines whether to scale the data to be within a range.

- scalelims : ints. Range to scale the data to. Format: low, high

- fmean : str. Path to .NPY file of mean training input and output values. Format: [inp0, inp1, ..., outp0, outp1, ...] If relative path, assumed to be with respect to the input directory.

- fstdev : str. Path to .NPY file of standard deviation of inputs/outputs. See ‘fmean‘ for format & path description.

- fmin : str. Path to .NPY file of minima of inputs/outputs. See ‘fmean‘ for format & path description.

- fmax : str. Path to .NPY file of maxima of inputs/outputs. See ‘fmean‘ for format & path description.

Neural Network (NN) Parameters

- weight_file: str. File containing NN model and weights. NOTE: MUST end in .h5

- inD : int. Dimensionality of the input to the NN.

- outD : int. Dimensionality of the output of the NN.

Bayesian Sampler Parameters

- alg : str. Bayesian sampling algorithm. Options: demc (ter Braak 2006) snooker (ter Braak & Vrugt 2008) multinest (Feroz et al. 2008) ultranest (Buchner 2014, 2016, 2019)

- func : strs. Function and file to evaluate at each iteration of the MCMC. Format: function file or, if the file is in a different directory, function file path/to/file/ Note: omit the ’.py’ from ‘file‘.

- pnames: strs. Name of each free parameter. Can include LaTeX formatting.
- pinit : floats. Initial parameters for the MCMC.
- pmin : floats. Minima for free parameters.
- pmax : floats. Maxima for free parameters.
- pstep : floats. Step size for free parameters. This will change throughout the MCMC due to the differential evolution algorithm used.
- data : floats. Values to be fit via MCMC. Format: Separate each value by an indented newline. Alternatively, specify a .NPY file.
- uncert: floats. Uncertainties on values to be fit via MCMC. Same format (indented newlines or .NPY)

Additional Sampler Parameters Only required if your setup requires the parameter.

- filters : strs. Paths to filter bandpasses associated with each datum. Separate each by indented newlines. X-axis values must have the same type of units as ‘xvals‘, but may be separated by a constant multiplicative factor (e.g., mm vs um, ft vs inch), see ‘filtconv‘ argument.
- starspec : str. Path to .NPY file of the stellar spectrum. More generally, this is an array of scaling factors that can be used when calculating the model.
- factor : str. Path to .NPY file of scaling factor by which to modify de-normalized predictions. E.g., unit conversion.
- wn : bool. In astro context, determines whether the X-axis units are spatial frequency (True) or wavelength (False). More generally, when wn = False, the X-axis values are transformed to their inverse when plotting and band-integrating models.
- wnfact : float. Multiplication factor when computing the inverse of ‘xvals‘. E.g., if ‘xvals‘ is in cm-1 and the desired inverse units are um, then wnfact = 1e4.
- filtconv : float. Multiplication factor to convert the filter X-value units to the ‘xvals‘ units. E.g., if filter X values are in nm, but ‘xvals‘ are in um, this argument would be set to 1e-3.

MCMC Parameters

- flog : str. Path to MCMC log file. If relative, with respect to input dir.
- nchains : int. Number of parallel samplers.
- niter : int. Number of total iterations.
- burnin : int. Number of burned iterations from the beginning of chains.
- thinning : int. Thinning factor for the posterior (keep every N iterations). Example: a thinning factor of 3 will keep every third iteration. Only recommended when the computed posterior is extremely large; not recommended for any nested sampling algorithms. Default: 1

Plotting Parameters

- xvals : str. Path to .NPY file containing the x-axis values associated with a prediction. If relative, path is with respect to ‘inputdir‘.
- xlabel : str. X-axis label for plots.
- ylabel : str. Y-axis label for plots.
- fpress : str. Path to text file containing the pressures of each layer of the atmosphere, for plotting T(p) profiles. If relative, with respect to ‘inputdir‘. If plot_PT is False, this argument does nothing.
- PTargs : str. Path to .txt file containing values necessary to calculate the temperature–pressure profile. Currently, only option is Line et al. (2013) method. Format: R_star (m), T_star (K), T_int (K), sma (m), grav (cm s-2) If plot_PT is False, this argument does nothing.
- truepars : floats. The true values of the model parameters to be retrieved. If unknown (e.g., real observational data), omit this key, or set it to None or False. Format: values separated by spaces or newlines, or a .NPY file containing the values.
- postshift : floats. Shifts all values of a particular parameter by a set amount in the posterior to be compared, such as for factor-of-10 unit conversions in a log space. E.g., postshift = 1 0 0 would increase all of the first parameter’s values by 1 and leave the other parameters alone. Format: val1 val2 val3 val4 ...
- savefile : str. (optional) Prefix for MCMC plots to be saved.

Comparing Posteriors (optional)

- compost : bool. Determines whether to compare HOMER’s posterior to another.
- compfile : str. Path to posterior .NPY file to compare with HOMER.
- compname : str. Name of the other posterior for plot legends.
- compsave : str. File name prefix for the saved comparison plots.
- compshift : floats. Same as ‘postshift‘, but for the posterior being compared.
- cinit : floats. Same as ‘pinit‘, but for the posterior being compared. Only required if ‘plot_PT‘ is True and one or more of the T(p) parameters are held fixed.

6 Program Outputs

HOMER produces the following outputs:

- MCMC.log – (MCMC only) a record of the MCMC exploration, including SPEIS, ESS, and credible regions.
- output.npy – the posterior determined by the Bayesian sampler

- Posterior plots – pairwise, posterior, PT (if ‘plot_PT‘), and trace plots.

- comparison plots – if ‘compost‘, makes overplots of the 1D marginalized posteriors, 2D pairwise posteriors, and explored temperature–presure profiles.

- bhatchar.npy – if ‘compost‘, calculates the Bhattacharyya coefficients between HOMER’s posterior and the other specified posterior.

7 Be Kind

Please cite this paper if you found this package useful for your research:
Himes et al. 2022, PSJ, 3, 91

```
@ARTICLE{2022PSJ.....3...91H,
       author = {{Himes}, Michael D. and {Harrington}, Joseph and {Cobb}, A
        title = "{Accurate Machine-learning Atmospheric Retrieval via a Neu
      journal = {\psj},
     keywords = {Exoplanet atmospheres, Bayesian statistics, Posterior dist
         year = 2022,
        month = apr,
       volume = {3},
       number = {4},
          eid = {91},
        pages = {91},
          doi = {10.3847/PSJ/abe3fd},
       adsurl = {https://ui.adsabs.harvard.edu/abs/2022PSJ.....3...91H},
      adsnote = {Provided by the SAO/NASA Astrophysics Data System}
}
```

Thanks!