Geoscientific
Model Development

CLASSIC v1.0: the open-source community successor to the Canadian Land Surface Scheme (CLASS) and the Canadian Terrestrial Ecosystem Model (CTEM) – Part 1: Model framework and site-level performance

Joe R. Melton[1], **Vivek K. Arora**[2], **Eduard Wisernig-Cojoc**[1], **Christian Seiler**[1], **Matthew Fortier**[1], **Ed Chan**[3], **and Lina Teckentrup**[4]

[1]Climate Research Division, Environment and Climate Change Canada, Victoria, B.C., Canada
[2]Canadian Centre for Climate Modelling and Analysis, Environment and Climate Change Canada, Victoria, B.C., Canada
[3]Climate Research Division, Environment and Climate Change Canada, Toronto, Ontario, Canada
[4]Max Planck Institute for Meteorology, 20146 Hamburg, Germany

Correspondence: Joe R. Melton (joe.melton@canada.ca)

Abstract. Recent reports by the Global Carbon Project highlight large uncertainties around land surface processes such as land use change, strength of CO_2 fertilization, nutrient limitation and supply, and response to variability in climate. Process-based land surface models are well suited to address these complex and emerging global change problems but will require extensive development and evaluation. The coupled Canadian Land Surface Scheme and Canadian Terrestrial Ecosystem Model (CLASS-CTEM) framework has been under continuous development by Environment and Climate Change Canada since 1987. As the open-source model of code development has revolutionized the software industry, scientific software is experiencing a similar evolution. Given the scale of the challenge facing land surface modellers, and the benefits of open-source, or community model, development, we have transitioned CLASS-CTEM from an internally developed model to an open-source community model, which we call the Canadian Land Surface Scheme including Biogeochemical Cycles (CLASSIC) v.1.0. CLASSIC contains many technical features specifically designed to encourage community use including software containerization for serial and parallel simulations, extensive benchmarking software and data (Automated Model Benchmarking; AMBER), self-documenting code, community standard formats for model inputs and outputs, amongst others. Here, we evaluate and benchmark CLASSIC against 31 FLUXNET sites where the model has been tailored to the site-level conditions and driven with observed meteorology. Future versions of CLASSIC will be developed using AMBER and these initial benchmark results to evaluate model performance over time. CLASSIC remains under active development and the code, site-level benchmarking data, software container, and AMBER are freely available for community use.

1 Introduction

Open collaboration has revolutionized software development leading to a proliferation of open-source software (OSS) projects. Notable successes include the internet browser Mozilla Firefox, office suite LibreOffice, the GNU/Linux operating system, and its derivative operating system for mobile devices, Android. OSS also has a large impact in sci-

entific computing. A Google Scholar search for "science open source software" reveals 4.7 million hits (accessed 21 June 2019) indicating a high level of activity. In the field of land surface modelling, there are several well-known large-scale community (OSS) models including the Joint UK Land Environment Simulator (JULES; Best et al., 2011; Clark et al., 2011), the Community Atmosphere Biosphere Land Exchange model (CABLE; Haverd et al., 2018), the Community Land Model (CLM; Lawrence et al., 2019), and the Noah Multi-Parameterization land surface model (Noah-MP; Niu et al., 2011). As recently stated by WIRED magazine, "Open source isn't counterculture anymore. It's the establishment" (Finley et al., 2019).

A community approach land surface modelling presents several benefits to its participants including (1) affordability, creating a new land surface scheme is a massive undertaking; (2) transparency, as the code is open to full scrutiny; (3) flexibility, as the models are designed to be both used in their present configuration and also extended to answer new science questions; (4) perpetuity, as many users across diverse institutions help protect the code against loss, deletion, or obsolescence; and (5) a collaborative community invested in developing, applying, and improving a common resource – the model. However, it is not clear that open code will necessarily lead to more open science. Easterbrook (2014) outlines several barriers to sharing of code in his commentary piece on the utility of open code for more open science. While he argues that open code should lead to better quality code, as he sees it, the main barriers to code sharing include portability (ability to run the code on different platforms), configurability (model setup to perform a simulation), entrenchment (historical reasons behind code development decisions), model–data blur (processing of model inputs impact upon model outputs), and provenance (reproduction of a model result).

Here, we present the Canadian Land Surface Scheme including Biogeochemical Cycles (CLASSIC; v.1.0), the successor to the coupled model framework of the Canadian Land Surface Scheme (CLASS; Verseghy, 2017) and the Canadian Terrestrial Ecosystem Model (CTEM; Melton and Arora, 2016). In developing the model framework for CLASSIC, careful attention has been paid to exploit the benefits of OSS, while minimizing the risks outlined by Easterbrook (2014). The CLASSIC model framework includes several key features designed to encourage collaboration and community use including (1) self-documenting code, (2) version control allowing source code management, distributed non-linear workflows, issue tracking, and wiki functionality, (3) native support for Network Common Data Format (NetCDF) input and output along with conversion tools for ASCII legacy inputs, (4) ability for code to run both serially on personal computers and using Message Passing Interface (MPI) on computing clusters, (5) output file description and metadata handled via a web interface, (6) model parameters read in from an external file, (7) containerization, and (8) extensive benchmarking of model state (via checksums) and

performance. Each of these features will be expanded upon in the following sections.

In Sect. 2, we describe the submodels, CLASS and CTEM that form the scientific basis of CLASSIC and the initial state of the model framework at the start of CLASSIC development. Section 3 details the model developments implemented along with tools to help existing users migrate to the new model framework. Section 4 outlines our model evaluation and benchmarking approach, while Sect. 5 presents the present state of CLASSIC as evaluated by the AMBER protocol. Section 6 describes future technical and scientific directions for CLASSIC.

2 Model description

2.1 Model physics: CLASS

The Canadian Land Surface Scheme (CLASS) was initiated in 1987 to produce a "second-generation" land surface scheme, characterized by more soil moisture and thermal layers with a separate treatment of the vegetation canopy (Verseghy, 2000), for inclusion in the Canadian general circulation model (GCM) and has been under continual development since. The first publications introducing CLASS described the physics calculations for movement of heat and water through the soil and snow layers (Verseghy, 1991), and the physics algorithms for energy and moisture fluxes within the vegetation canopy as radiation and precipitation cascade through it, with an explicit thermal separation of the vegetation from the underlying ground (Verseghy et al., 1993). Development of CLASS has been predominantly within Environment Canada, a federal department of the Government of Canada (later renamed Environment and Climate Change Canada; ECCC) with the exception of an organized community effort as part of the Canadian Climate Research Network (1994–1997; Verseghy, 2000) as well as ad hoc collaborations with the broader research community. CLASS is presently at version 3.6.2 (Verseghy, 2017). While CLASS includes a biogeochemical component (CTEM; described below), it is possible to turn it off and run CLASS with specified structural vegetation attributes such as tree height, plant area index, etc., which is desirable in some situations. As a land surface scheme, CLASS simulates the fluxes of energy, momentum, and water between the atmosphere and land surface (Fig. 1).

CLASS operates on a subdaily time step that varies depending on its application. When within the framework of the most recent version of the Canadian Earth System Model (CanESM5; Swart et al., 2019), it operates at a time step of 15 min, consistent with the atmospheric component. A time step of 30 min is used for offline, uncoupled simulations where it is driven with observed meteorological data. Differing measurement heights for the observed temperature and wind speed are accounted for. Since the forcing data typ-
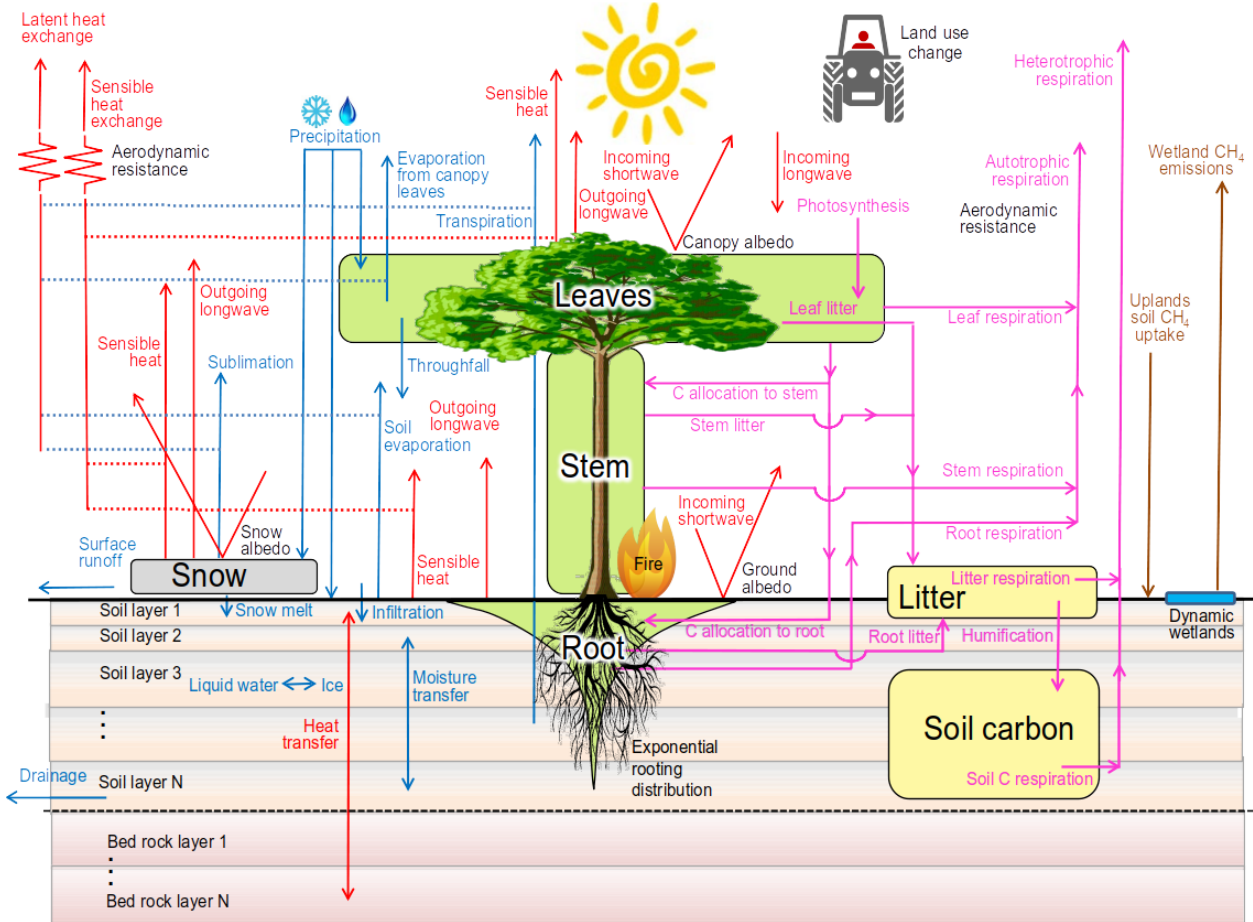
Figure 1. Schematic illustration of processes simulated by CLASSIC.

ically do not separate precipitation into its rainfall and snowfall components, three options are available in the CLASSIC modelling framework: (1) a strict threshold of 0 °C below which all precipitation is considered snowfall, (2) a linear gradual transition of rainfall into snowfall as temperature decreases from 2 to 0 °C, and (3) a non-linear transition from rainfall to snowfall as temperature decreases from 6 to 0 °C. Here, we use option 1.

Vegetation characteristics are described by rooting depth, canopy mass, leaf area index (LAI), and vegetation height, which are then used in calculations of the transfers of energy, water, and momentum with the atmosphere. The physical land surface processes in CLASS are modelled using plant functional types (PFTs). Presently, global vegetation in CLASS is represented using four PFTs – needleleaf trees, broadleaf trees, crops, and grasses. The structural vegetation characteristics of these PFTs are modelled as a function of driving meteorological data and atmospheric CO_2 concentration by CTEM, which handles the biogeochemical processes, as explained in the next section. The number of ground layers in CLASS can vary depending upon application. The

standard offline model setup currently uses 20 ground layers starting with 10 layers of 0.1 m thickness, gradually increasing to a 30 m thick layer for a total ground depth of over 61 m, while three ground layers with thicknesses of 0.1, 0.25, and 3.75 m are used in CanESM5 (the standard model setup is provided for each of the 31 FLUXNET sites; please see Appendix A). Water fluxes are calculated for soil layers within the permeable soil depth of the ground column but not the underlying bedrock layers, whereas temperatures are calculated for both soil and bedrock layers (depicted in Fig. 1). Both water fluxes and ground temperatures are calculated each time step. The permeable soil depth varies geographically and is read in from the model initialization file. Also calculated each time step are the temperature, mass, albedo, and density of the single layer snowpack (where it exists), the temperature and interception and storage of rain and snow on the vegetation canopy, and the temperature and depth of ponded water on the soil surface. Each grid cell is independent with no lateral transfers of heat or moisture between them. While runoff can be routed to estimate streamflow, such as in CanESM5 using the approach of Arora et al.

(1999) or the MEC-Surface & Hydrology System (MESH) framework (Pietroniro et al., 2007), in both cases, once the runoff leaves a grid cell, it does not modify the soil moisture of downstream grid cells.

Application and evaluation of the performance of CLASS over the course of its development has been extensive with dozens of publications using the model at point, regional, and global scales both in coupled and offline modes. CLASS has been applied in an offline context, i.e. forced with observed meteorology (e.g. Bailey et al., 2000; Kothavala et al., 2005; Roy et al., 2013; Bartlett et al., 2006; Brown et al., 2006; Wu et al., 2016; Verseghy and MacKay, 2017; Melton et al., 2019d), as the physical land surface component of regional climate models, e.g. CRCM (Ganji et al., 2015; Paquin and Sushama, 2014) and CanRCM (Scinocca et al., 2016), and integrated into each version of the Canadian Atmospheric Model (CanAM; von Salzen et al., 2013), Coupled Global Climate Model (CanCM), and Earth System Model (CanESM; Arora et al., 2011; Swart et al., 2019) since the early 1990s.

2.2   Model biogeochemistry: CTEM

Development of the Canadian Terrestrial Ecosystem Model (CTEM) began in the early 2000s at Environment Canada in response to the need for a land surface carbon cycle component for the CanESM. CTEM, which is presently at version 2.0 (Melton and Arora, 2016), couples with CLASS through the exchange of information describing the state of the physical land surface and overlying vegetation. CLASS provides CTEM with physical land surface information including soil moisture, soil temperature, and net radiation. CTEM uses this information to simulate photosynthesis and prognostically calculate carbon in its three live vegetation components (leaves, stem, and roots) and two dead carbon pools (litter and soil). The carbon amounts in these five carbon pools evolve prognostically, in the default CLASSIC configuration, for nine PFTs that map directly onto the four PFTs used by CLASS. Needleleaf trees are divided into their deciduous and evergreen types, broadleaf trees are divided into cold deciduous, drought deciduous, and evergreen types, and crops and grasses are divided based on their photosynthetic pathways into C_3 and C_4 versions. The finer distinctions between PFTs in the CTEM PFTs is required for modelling biogeochemical processes. For instance, simulating leaf phenology prognostically requires the distinction between deciduous and evergreen versions of broadleaf trees. However, once the LAI has been dynamically determined by CTEM, CLASS only needs to know that this PFT is a broadleaf tree since the physics calculations do not require information about underlying the deciduous or evergreen nature of the leaves. The prognostic carbon masses of leaf, stem, and root simulated by CTEM are used to calculate the structural vegetation characteristics required by CLASS: rooting depth (using a dynamic root distribution; Arora and Boer, 2003), canopy mass, LAI, and vege-

tation height (Arora and Boer, 2005a). Other than these structural vegetation attributes, CTEM also provides canopy conductance values to CLASS based on photosynthesis calculations at the CLASS time step, as explained in Arora and Boer (2003). The remainder of the biogeochemical process simulated by CTEM (Fig. 1) and the resulting vegetation dynamics operate on a daily time step. CTEM models all primary terrestrial ecosystem processes including maintenance and growth respiration (Arora and Boer, 2005a); heterotrophic respiration (Melton et al., 2015); tissue turnover, allocation of carbon, and phenology (Arora and Boer, 2005a); disturbance (fire; Arora and Boer, 2005b; Arora and Melton, 2018); competition for space between PFTs (Arora and Boer, 2006; Melton and Arora, 2016); and land use change (Arora and Boer, 2010).

CTEM also dynamically calculates wetland extent, methane emissions from wetlands and fires, and methane uptake by soils (Curry, 2007) as described in Arora et al. (2018). To determine the wetland extent, as the liquid soil moisture in the top soil layer increases above latitudinally dependent threshold values, the wetland fraction increases linearly up to a maximum value, equal to the flat fraction in a grid cell with slopes less than 0.2 %. Methane emissions from wetlands are calculated by scaling the heterotrophic respiration flux from the model's litter and soil carbon pools. A separate submodule simulates peatland specific processes following Wu et al. (2016).

Both CLASS and CTEM have the capability to be run in a "mosaic" configuration in which grid cells are divided into separate and independent tiles that simulate their own energy and water balance. There is no lateral transfer of energy or water between tiles that lie within a grid cell. As a result, the soil moisture in the permeable soil layers and the temperature of soil and bedrock layers in each tile evolve independently of each other, despite being driven with the same meteorological data. A grid cell may be divided into tiles using any criteria, e.g. by each PFT on its own tile, different soil textures, or peatlands vs. uplands. In contrast, with the "composite" or single-tile approach, structural vegetation attributes (LAI, rooting depth, vegetation height, canopy mass, and albedo) are averaged over all PFTs for use in energy and water balance calculations. Soil texture for permeable soil layers is also common to all PFTs within a grid cell resulting in a single liquid and frozen soil moisture for each permeable soil layer, and a single temperature for each soil and bedrock layer for the grid cell. The effect of tiling on the basis of fractional coverage of PFTs has been evaluated on estimation of the terrestrial carbon sink (Melton and Arora, 2014), competition between PFTs (Shrestha et al., 2016), and the use of soil texture clusters as a tiling criterion has been evaluated by Melton et al. (2017).

Terrestrial biogeochemical processes simulated by CTEM have been evaluated at scales from site level to global (e.g. Peng et al., 2014; Melton and Arora, 2014, 2016; Melton et al., 2015). The fire subroutine has been extensively eval-

uated as part of the Fire Model Intercomparison Project (FireMIP; Hantson et al., 2016; Forkel et al., 2019) in addition to being used to estimate carbon cycle implications of the reduction in global wildfire since the 1930s (Arora and Melton, 2018). The parameterization for competition between PFTs has been evaluated at the site level (Shrestha et al., 2016) as well as at the global scale (Melton and Arora, 2016). CLASS-CTEM has contributed to the Global Carbon Project's methane assessment by providing wetland methane emissions (Poulter et al., 2017). An assessment of natural methane emissions simulated by CTEM is presented in Arora et al. (2018) who use a one-box model of atmospheric methane together with prescribed anthropogenic and geological sources to reproduce atmospheric methane concentrations consistent with the observational record over the historical period. CLASS-CTEM has also contributed regularly to the Global Carbon Project's annual carbon budget analyses since 2016 (Le Quéré et al., 2016, 2018b, a).

# 3 Model development

3.1 Motivation

At the start of our project to transform CLASS-CTEM into CLASSIC, the model code base was not well suited to modern computing and model development practices. CLASS-CTEM was written following the Fortran 77 standard, documentation was provided in stand-alone documents that often significantly lagged model development or scientific papers, code management did not use modern version control systems, model parameters such as the number of soil layers and PFTs were hard-coded into subroutines, and the offline framework (as opposed to the version coupled into the family of CanESM models) used fixed-format ASCII text files for model inputs and outputs. While these issues are not unique to CLASS-CTEM, they made model development and evaluation challenging. For example, while ASCII inputs are reasonable for site-level simulations, they quickly become difficult to prepare and manage over large modelling domains, which entailed handling thousands of ASCII files that are poorly suited for parallel computation. Additionally, as computer hardware or compiler versions change the model results are susceptible to issues of reproducibility due to a changing computational environment. While CLASS-CTEM has been used extensively by the Canadian research community, as evidenced by the publications cited in the previous section, the modelling framework was poorly designed for new users who faced a steep learning curve in attempting to run and understand the model. In our modernization and improvement of the CLASS-CTEM code base to develop the CLASSIC model, we have addressed these concerns and made several important changes to the model code, which we describe below.

3.2 Model developments to encourage community use

3.2.1 Containerization

A virtual machine (VM) emulates a physical computer allowing different operating systems to run on different hardware, e.g. the Linux operating system on a Windows machine. A software container is a form of operating system virtualization, essentially a pared-down version of a virtual machine with only the necessary software required to complete the intended task. Both options have been used to facilitate the running of complex models. The Weather Research and Forecasting model (WRF; Hacker et al., 2016) has been implemented in both VM and container configurations, while the United Kingdom Chemistry and Aerosols (UKCA) composition–climate model has been implemented within the Met Office VM (Abraham et al., 2018). In choosing between a VM or container to provide users of CLASSIC, we chose a container due to its light computational overhead, and design geared to high-performance computing (HPC) applications (Arango et al., 2017). The advantages of containers for scientific computing are numerous. Firstly, as the environment within the container is consistent across computer systems, the reproducibility of the code is enhanced. Containers are encapsulated portable environments that can contain both the software and data dependencies, along with libraries needed by the software, ensuring all necessary environment variables are provided and consistent. Secondly, they prevent orphaning of old code due to environment changes. If a container image is retained, the container can recreate the same environment for the code each time it is run. This ability allows a simple means to rerun old code, without having to consider environment changes that would make running the code outside of the container onerous. Third, the use of a container makes it easier to get up and running with a model quicker as all dependencies are included in the container. Lastly, many HPC centres allow the use of containers on their systems, reducing the need for users to reconfigure models for different server systems.

For CLASSIC, we use a Singularity container (https://www.sylabs.io/, last access: 19 November 2019), which is portable to Linux, Mac, and Windows systems as well as HPC clusters, for which Singularity was specifically designed (Kurtzer et al., 2017). Tests of Singularity's performance on HPC benchmarks have demonstrated a negligible performance overhead (Le and Paz, 2017). Singularity is already installed on many national, university and government HPC centres including Compute Canada, Argonne National Labs, Fermilab, National Supercomputer Centre (Sweden), amongst others. The CLASSIC container image is available from our Zenodo community page (https://zenodo.org/communities/classic/, last access: 19 November 2019). Version 1.0 of our container contains all software libraries needed to run the model in serial or parallel (using MPI) as well as the capability to run our benchmarking software (de-
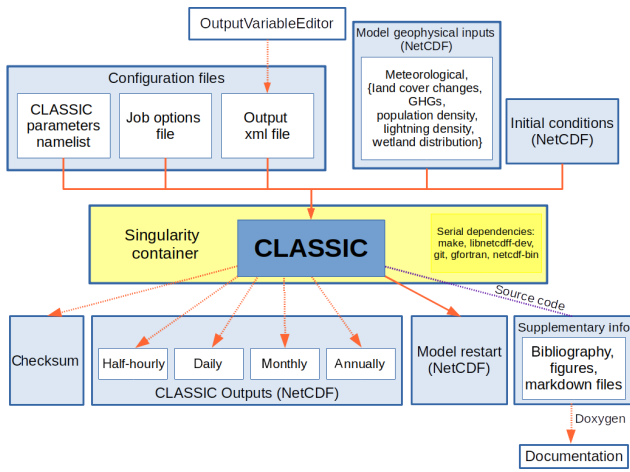
Figure 2. CLASSIC model workflow. For each CLASSIC simulation, the solid lines indicate mandatory model inputs or outputs. The OutputVariablesEditor can be used to edit the XML file that controls the model outputs and specifies their metadata (Sect. 3.2.8). Model parameters are read in from the CLASSIC parameters' namelist file, while the simulation options such as location of input files and model configuration options are read in from the CLASSIC job options file. The required model geophysical inputs vary depending on the model configuration chosen but meteorological inputs are always required. All model outputs can be switched on or off depending on the simulation except the model restart file. Checksums can be used to ensure changes to the model do not impact model performance (see Sect. 3.2.2). The CLASSIC source code is combined with other supplementary information by Doxygen to produce the model documentation (see Sect. 3.2.3).

scribed in Sect. 4.2). The general model workflow is outlined in Fig. 2.

3.2.2 Code design and management

To permit flexibility in the code, CLASSIC's parameters are read in at runtime from a Fortran namelist file. This use of a namelist permits rapid testing of model sensitivity to parameter values, as the model does not require recompilation between tests. Additionally, new PFTs can be more easily added to the code as all parameters are located in the namelist file, rather than distributed throughout the code. However, by design, new PFTs cannot be introduced into CLASSIC without due care. Within the code, case structures are used at each PFT-level branching calculation to determine the code branch a particular PFT is assigned to. These case structures have integrated error checks, whereby an unknown PFT causes the model to abort with a flag thrown informing the user where in the code the model detected an unknown PFT. This safety check helps to encourage thoughtful introduction of new PFTs to the model.

In transitioning from CLASS-CTEM to CLASSIC, the code base was ported into the distributed version control system Git (https://git-scm.com/, last access: 19 Novem-

ber 2019) and is distributed via the software development tool GitLab (https://gitlab.com/cccma/classic, last access: 19 November 2019). The GitLab issues tracker and a Nabble forum (http://classic-message-board.158658.n8.nabble.com/, last access: 19 November 2019) will be used to facilitate communication between CLASSIC users. The issues tracker can be used to submit bug reports, elaborate on new parameterizations, and discuss code changes, whereas the forum can be used to ask other users for advice and assistance.

As a further aid to development, CLASSIC can output checksums to speed model development for code changes that should not impact upon the model outputs. The checksum subroutine computes content-based checksums (summation of the flipped bits, i.e. the 1 in binary representation) of several groups of variables after a run has completed.

Checksums are an imperfect means of ensuring no logical changes, as two numbers may have different binary representations with the same number of flipped bits. However, as the number of variables checked increases, it becomes highly unlikely to render a false positive. For example, if we take a data item of bit length n, having b flipped bits in its representation, the number of same-length data items with the same checksum can be expressed through the binomial coefficient

$$\binom{n}{b}. \tag{1}$$

If we assume the worst case where $b = \frac{n}{2}$, then Eq. (1) becomes

$$\binom{n}{n/2}. \tag{2}$$

Dividing by the total number of possible values for an n-digit binary number, we find the probability of a false positive checksum to be

$$\frac{\binom{n}{n/2}}{2^n}. \tag{3}$$

So, for example, if the checksum is computed using only three 32-bit numbers, the probability of a false positive is about 2.87×10^{-7}.

3.2.3 Self-documenting code

To modernize the CLASS-CTEM code base, it was first transformed from the fixed form (Fortran 77) to a free-form structure (allowing coding constructs permitted by the Fortran 2008 standard). Best practices suggest that it is most desirable to embed documentation within the software, as this increases the likelihood that when developers update code, they will update the documentation at the same time (Wilson et al., 2014). Our model documentation was then incorporated directly into the code using the syntax of the Doxygen documentation generator (http://www.doxygen.nl/, last access: 19 November 2019). The use of Doxygen allows both the programmatic and scientific reasoning behind

the code to be detailed within the generated manual along with variable dictionaries. This ensures that the documentation for a particular model version is always included in the model source code distribution and lowers the burden for developers to maintain its currency as it can be edited as the source code is updated. The full Doxygen-generated CLASSIC documentation for version 1.0 can be found in the Zenodo archive and, for the most recent CLASSIC release, at https://cccma.gitlab.io/classic (last access: 19 November 2019). As Doxygen is included in the CLASSIC container, the user can run "doxygen Doxyfile" to generate a local version of the documentation as required.

3.2.4 Coding standards

As part of the code modernization effort, we refactored the code to follow a recently developed CLASSIC coding convention. Coding conventions help to ensure code portability, readability, and maintainability. In adopting our coding conventions, we have developed a tool (a "linter") to enforce code quality that can be used on legacy code, or new code, to either change the code to meet the coding standards or to flag suspect sections of the code for manual intervention. Our coding convention is available in the Supplement and on the CLASSIC website. The linter, which is written in Python, is available in the CLASSIC code repository.

3.2.5 Serial and parallel computations (MPI)

As CLASSIC can be run at the site level as well as over gridded domains, it is important to support both modes of operation while maintaining only one code base. The newly created CLASSIC driver allows compilation for serial or parallel processing of the model code. For site-level simulations, serial compilation and running of the model is sufficient, whereas, for speed, runs over gridded domains require parallel processing which uses MPI directives within the CLASSIC code. Pre-compiler directives, used sparingly, allow the same code for both applications. The default compiler supplied within the CLASSIC container is the GNU compiler (gfortran for serial and mpif90 as wrapper around gfortran for parallel computation) with additional Makefile scripts for Intel and Cray compilers.

3.2.6 Simulation domains

CLASSIC is designed to be easily run over simulation domains from site level to global. To run the model at a point location, the model input files can be either also site level with point-scale information or regional (two-dimensional fields). For regional domains, the model grid can be specified either with one-dimensional vectors or two-dimensional grids of longitudes and latitudes. The 1-D case includes regular grids equally spaced in longitude and latitude as well as regular Gaussian grids. The 2-D case is more general and can support any rectangular grid. The model call determines whether a simulation is run across a region or at a point. When using a regular grid, the model domain can be set by calling CLASSIC with a longitude latitude box. When the input files are provided on an irregular grid, CLASSIC uses grid cell indexes instead of geographic coordinates to delineate the domain for the simulation.

3.2.7 NetCDF input/outputs

While CLASS-CTEM used ASCII text files for model input/output (I/O), this method is cumbersome over gridded domains. The CLASSIC framework uses NetCDF files instead (https://www.unidata.ucar.edu/software/netcdf/, last access: 19 November 2019). NetCDF is a machine independent data format that is self-describing, allowing extensive metadata within the files, and which is the most common data format within the land surface modelling community. Scripts are provided in the CLASSIC code repository to convert legacy ASCII input files (meteorology and model initialization files) into the new NetCDF format. Additionally, for site-level users, a Fortran tool is included to convert Fortran namelist files, which are easier to work with than the legacy ASCII files, into a NetCDF file suitable for CLASSIC model initialization.

3.2.8 Output files (xml)

To best use NetCDF as an output format requires writing the output file metadata at the time of the file's creation. To facilitate this, CLASSIC uses extensible markup language (XML) files that are edited through a web interface and the resulting files are validated using an adjacent schema. The web interface allows a user to configure the output variables, as well as add new variables and edit metadata, using any JavaScript-compatible browser. Once the changes are complete, the user may download an updated version of the modified XML configuration file for use by CLASSIC. Where possible CLASSIC output files are Climate and Forecast (CF) compliant (http://cfconventions.org/, last access: 19 November 2019) and use variable names consistent with the data request of the Coupled Model Intercomparison Project (CMIP6; http://clipc-services.ceda.ac.uk/dreq/mipVars.html, last access: 19 November 2019).

3.2.9 Meteorological inputs

Meteorological inputs required by CLASSIC include downwelling shortwave and longwave radiation, surface precipitation rate, surface air pressure, specific humidity, wind speed, and air temperature together with a reference height at which these quantities were measured. Within the CanESM framework these meteorological variables are provided by the atmospheric component of the ESM at its time step of 15 min. For global offline simulations, reanalysis meteorological variables are typically available on either a 3 or 6 h time step, but it could be any multiple of the CLASSIC

physics time step. To convert the reanalysis meteorological data to the offline CLASSIC physics time step, CLASSIC disaggregates the coarse temporal resolution meteorology on the fly. Surface pressure, specific humidity, wind speed, and surface temperature are linearly interpolated. Longwave radiation is uniformly distributed across the reanalysis time period. Shortwave radiation is distributed diurnally using the day of year and a grid cell's latitude with the maximum value occurring at solar noon. The total reanalysis time period precipitation amount determines the number of wet half hours in each period following Arora (1997). In a conservative manner, the total time period's precipitation amount is then randomly spread across the wet physics time step periods. For example, if the reanalysis meteorology is provided in 6 h time periods, the non-zero precipitation for each 6 h is spread across the number of wet 30 min time steps as determined by the disaggregation scheme. If CLASSIC is being run with observed meteorology, such as from an eddy covariance tower, on a 15–30 min time step, the meteorology is used without modification.

4 Running and benchmarking CLASSIC

4.1 Quick-start tutorial to run and benchmark CLASSIC

Appendix A contains a tutorial that guides the reader through downloading, compiling, and running CLASSIC over a set of FLUXNET sites. FLUXNET is a global network of micrometeorological tower sites that uses the eddy covariance method to measure the exchanges of carbon dioxide, water vapour, and energy between terrestrial ecosystems and the atmosphere (https://fluxnet.ornl.gov, last access: 19 November 2019). FLUXNET data have been used extensively to improve and evaluate land surface models hydrology, energy, and carbon fluxes as well as model processes (e.g. Blyth et al., 2010; Stöckli et al., 2008; Melaas et al., 2013). The outputs of the model runs are then run through a benchmarking system described below. All figures presented here except the schematic figures (Figs. 1 and 2) are produced by the benchmarking software.

4.2 Automated Model Benchmarking (AMBER)

The increasing complexity of land surface models necessitates more advanced methods of model evaluation and assessment. Such methods have been developed in a number of collaborative projects including the Project for Intercomparison of Land Surface Parameterization Schemes (PILPS; Henderson-Sellers et al., 1993), the Global Land-Atmosphere Coupling Experiment (GLACE; Koster et al., 2006), the Protocol for the Analysis of Land Surface Models (PALS; Abramowitz, 2012), the PALS Land Surface Model Benchmarking Evaluation Project (PLUMBER; Best

et al., 2015), and the International Land Model Benchmarking (ILAMB; Collier et al., 2018).

ILAMB has developed a framework that summarizes model performance across multiple statistical metrics using a dimensionless skill score system implemented in an open-source benchmarking and diagnostics software tool for land surface model evaluation written in Python (Collier et al., 2018). The evaluation of CLASSIC presented in this study is based on ILAMB's statistical framework, which we implemented in a new R package, referred to as the Automated Model Benchmarking (AMBER) package (Seiler, 2019). The development of AMBER allowed us to tailor the ILAMB approach to CLASSIC model outputs allowing: (i) a seamless data ingestion that does not require pre-processing steps, (ii) the ability to evaluate CLASSIC in different simulation modes (i.e. global and regional simulations on differing grids, and site-level runs), and (iii) full control on how the statistical framework is implemented. AMBER uses a variety of observation-based reference data against which it evaluates CLASSIC. These data consist of global-scale remote sensing-based products, eddy covariance flux towers, and annual streamflow measurements at the mouth of major rivers. Site-level evaluations are based on eddy covariance flux tower data alone. All AMBER outputs from site-level evaluation of CLASSIC v.1.0 are in the model benchmarking archive (Table 2), with AMBER itself included in the CLASSIC software container.

4.2.1 Skill scores

In ILAMB (Collier et al., 2018) and AMBER, the performance of a model is expressed through scores that range from zero to one, where higher values imply better performance. These scores are computed for each variable in five steps: (1) computation of a statistical metric, (2) non-dimensionalization, (3) conversion to unit interval, (4) spatial integration, and (5) averaging scores computed from different statistical metrics. The statistical metrics considered are the bias, root mean square error, phase shift, interannual variability, and spatial distribution. For example, the scalar score for the bias is calculated as

$$\text{bias}(\lambda, \phi) = \overline{v_{\text{mod}}}(t, \lambda, \phi) - \overline{v_{\text{ref}}}(t, \lambda, \phi), \tag{4}$$

where $\overline{v_{\text{mod}}}(t, \lambda, \phi)$ and $\overline{v_{\text{ref}}}(t, \lambda, \phi)$ are the mean values in time (t), over a specified period of time, of a variable v, which varies geographically as a function of longitude λ and latitude ϕ for model and reference data, respectively. Non-dimensionalization is achieved by dividing the absolute bias by the standard deviation of the reference data (σ_{ref}) that corresponds to the same period over which the mean of model simulated quantities is calculated:

$$\varepsilon_{\text{bias}}(\lambda, \phi) = \frac{|\text{bias}(\lambda, \phi)|}{\sigma_{\text{ref}}(\lambda, \phi)}. \tag{5}$$

A bias score that ranges between 0 and 1 is calculated next:

$$s_{\text{bias}}(\lambda, \phi) = e^{-\varepsilon_{\text{bias}}(\lambda, \phi)}. \tag{6}$$

The spatial integration of s_{bias} leads to the scalar score:

$$S_{\text{bias}} = \overline{\overline{s_{\text{bias}}}}. \tag{7}$$

The spatially integrated score values for the root mean square error (S_{rmse}), phase shift (S_{phase}), interannual variability (S_{iav}), and spatial distribution (S_{dist}) are found in a similar way although the actual calculation of the metrics is, of course, different as shown in Table A1. To demonstrate the intermediate statistical metrics for the component scores, the components of S_{rmse} are listed in Table A2. The score values based on these metrics are then combined to derive a single overall score for each output variable:

$$S_{\text{overall}} = \frac{S_{\text{bias}} + 2S_{\text{rmse}} + S_{\text{phase}} + S_{\text{iav}} + S_{\text{dist}}}{1 + 2 + 1 + 1 + 1}. \tag{8}$$

S_{rmse} is assigned twice as much value as the other metrics since we consider it more important than the other metrics. Interpretation of benchmarking scores should be done carefully. While ILAMB uses "stoplight" colours to distinguish different thresholds, at least in a visual sense, of model performance (e.g. see Fig. 1 in Collier et al., 2018), we do not adopt that approach. The use of thresholds may be a useful visual tool when distinguishing the performance of several models (as in Fig. 1 of Collier et al., 2018) or versions of the same model; however, here we are presenting the results of a single model, and thus any thresholds chosen would be arbitrary. Following Collier et al. (2018), we also "... do not view these aggregate absolute scores as a determinant of *good* or *bad* models. We envision the scores as a tool to more quickly identify relative differences among models and model versions which the scientist must then interpret". We also agree with Collier et al. (2018) that the absolute value of score is not particularly meaningful. As well, a perfect score is not achievable due to reference datasets having measurement error and uncertainty, a lack of consistency between datasets for the same variables, and a lower score may only highlight the need for an enhanced observational sampling effort or the lack of an appropriate metric of model performance (Collier et al., 2018). Therefore, we do not use the model score as the basis to determine whether model performance is acceptable or deficient for different model outputs but rather use these scores to define the initial model performance that future model developments will be compared against.

5   Evaluation of CLASSIC

We used three separate approaches to benchmark CLASSIC. First, we used FLUXNET sites (Pastorello et al., 2017) where CLASSIC was driven with observed meteorology and the model initialization file was set up to correspond to site conditions, i.e. vegetation composition and coverage, soil texture and permeable depth, based on publications describing the tower sites. These "site-level" simulations were performed for the 31 FLUXNET sites listed in Table 1 and shown in Fig. 3. We then evaluated the CLASSIC outputs against the eddy covariance (EC)-tower-derived quantities of net ecosystem exchange (NEE), gross primary productivity (GPP), ecosystem respiration (RECO), net radiation (RNS), and ground (HFG), sensible (HFSS), and latent heat (HFLS) fluxes. The FLUXNET data are from the 3 November 2016 update to the FLUXNET2015 dataset. While these sites are limited in their spatial and temporal coverage, they do present the most realistic model setup which uses observed vegetation composition, soil textures, and driving meteorology.

There are two limitations, which apply to most land surface models, when comparing model simulated data against observation-based estimates at FLUXNET sites. First, CLASSIC represents vegetation at the level of PFTs, and not at the species level. That is, for example, at two temperate FLUXNET sites which have either oak (*Quercus*) or maple (*Acer*) deciduous broadleaf trees, CLASSIC represents these sites using the same set of model parameter values corresponding to its cold deciduous broadleaf PFT. While climate predominately determines energy, water, and carbon fluxes, in the real world, species level differences also play a role in modulating these fluxes. Second, the model simulated values are compared with observations only when the model's carbon pools reach equilibrium after being driven repeatedly with available meteorological data. The chosen FLUXNET sites in our study have between 3 and 19 years of available meteorological data (Table 1). Since these available years of meteorological data may not be representative of the mean climate at a given site, and/or a given site may be recovering from events that occurred prior to the start of the EC tower installation, the comparison between observations and model simulated fluxes is somewhat confounded. The modelled annual NEE thus always sums to zero, by construction, while we know that in the real world, at most sites, GPP is currently higher than RECO, making annual NEE positive with land acting as a sink of carbon in response to rising atmospheric CO_2 concentrations. Past episodic extreme climatic events (e.g. drought or extreme precipitation) and disturbances (e.g. prior timber harvest or tree planting such as site CZ-BK1; (Sedlák et al., 2010) or fire, e.g. at CG-Tch Merbold et al., 2009) that we are unable to take into account confound the mismatch between model and observation-based fluxes even more. In an ideal world, we would have historical meteorological data and disturbance history available at each site. All components of the carbon budget – GPP, RECO, and NEE – are affected by these issues, but since NEE is a residual of GPP and RECO, the model to observations comparison is confounded most significantly for NEE.

Table 1. FLUXNET2015 sites used in the FLUXNET site-level benchmarking (Sect. 5.2).

Site ID	Site name	Latitude	Longitude	Elevation (m)	Years	IGBP code*	DOI
AU-Tum	Tumbarumba	−35.6566	148.1517	1200	2001–2013	EBF	https://doi.org/10.18140/FLX/1440126
BR-Sa1	Santarem-Km67-Primary Forest	−2.8567	−54.9589	88	2002–2011	EBF	https://doi.org/10.18140/FLX/1440032
CA-Qfo	Quebec – E. Boreal, Mature Black Spruce	49.6925	−74.3421	382	2003–2010	ENF	https://doi.org/10.18140/FLX/1440045
CA-TPD	Ontario – Turkey Point Mature Deciduous	42.6353	−80.5577	260	2012–2014	DBF	https://doi.org/10.18140/FLX/1440112
CG-Tch	Tchizalamou	−4.2892	11.6564	82	2006–2009	SAV	https://doi.org/10.18140/FLX/1440142
CN-Dan	Dangxiong	30.4978	91.0664	4313	2004–2005	GRA	https://doi.org/10.18140/FLX/1440138
CZ-BK1	Bily Kriz forest	49.5021	18.5369	875	2004–2014	ENF	https://doi.org/10.18140/FLX/1440143
DE-Kli	Klingenberg	50.8931	13.5224	478	2004–2014	CRO	https://doi.org/10.18140/FLX/1440149
DE-Tha	Tharandt	50.9624	13.5652	385	1996–2014	ENF	https://doi.org/10.18140/FLX/1440152
DK-Sor	Sorø	55.4859	11.6446	40	1996–2014	DBF	https://doi.org/10.18140/FLX/1440155
FI-Hyy	Hyytiälä	61.8474	24.2948	181	1996–2014	ENF	https://doi.org/10.18140/FLX/1440158
FR-Fon	Fontainebleau-Barbeau	48.4764	2.7801	103	2005–2014	DBF	https://doi.org/10.18140/FLX/1440161
FR-Pue	Puéchabon	43.7413	3.5957	270	2000–2014	EBF	https://doi.org/10.18140/FLX/1440164
ES-LgS	Laguna Seca	37.0979	−2.9658	2267	2007–2009	OSH	https://doi.org/10.18140/FLX/1440225
GF-Guy	Guyaflux (French Guiana)	5.2788	−52.9249	48	2004–2014	EBF	https://doi.org/10.18140/FLX/1440165
GH-Ank	Ankasa	5.2685	−2.6942	124	2011–2014	EBF	https://doi.org/10.18140/FLX/1440229
IT-Lav	Lavarone	45.9562	11.2813	1353	2003–2014	ENF	https://doi.org/10.18140/FLX/1440169
IT-SRo	San Rossore	43.7279	10.2844	6	1999–2012	ENF	https://doi.org/10.18140/FLX/1440176
IT-Tor	Torgnon	45.8444	7.5781	2160	1999–2012	GRA	https://doi.org/10.18140/FLX/1440237
MY-PSO	Pasoh Forest Reserve	2.9730	102.3062	147	2003–2009	EBF	https://doi.org/10.18140/FLX/1440240
NL-Loo	Loobos	52.1666	5.7436	25	1996–2014	ENF	https://doi.org/10.18140/FLX/1440178
PA-SPs	Sardinilla-Pasture	9.3138	−79.6314	68	2007–2009	GRA	https://doi.org/10.18140/FLX/1440179
RU-Che	Cherski	68.6130	161.3414	6	2002–2005	WET	https://doi.org/10.18140/FLX/1440181
RU-Fyo	Fyodorovskoye	56.4615	32.9221	265	1998–2014	ENF	https://doi.org/10.18140/FLX/1440183
RU-Ha1	Hakasia steppe	54.7252	90.0022	446	2002–2004	GRA	https://doi.org/10.18140/FLX/1440184
RU-SkP	Yakutsk Spasskaya Pad larch	62.2550	129.1680	246	2012–2014	DNF	https://doi.org/10.18140/FLX/1440243
SD-Dem	Demokeya	13.2829	30.4783	500	2005–2009	SAV	https://doi.org/10.18140/FLX/1440186
US-WCr	Willow Creek	45.8059	−90.0799	520	1999–2014	DBF	https://doi.org/10.18140/FLX/1440095
US-Wkg	Walnut Gulch Kendall grasslands	31.7365	−109.9419	1531	2004–2014	GRA	https://doi.org/10.18140/FLX/1440096
ZA-Kru	Skukuza	−25.0197	31.4969	359	2000–2013	SAV	https://doi.org/10.18140/FLX/1440188
ZM-Mon	Mongu	−15.4378	23.2528	1053	2007–2009	DBF	https://doi.org/10.18140/FLX/1440189

* IGBP land classification abbreviations used include evergreen broadleaf forest (EBF), evergreen needleleaf forest (ENF), savanna (SAV), deciduous broadleaf forest (DBF), grassland (GRA), cropland (CRO), open shrubland (OSH), closed shrubland (CSH), permanent wetland (WET), and deciduous needleleaf forest (DNF).
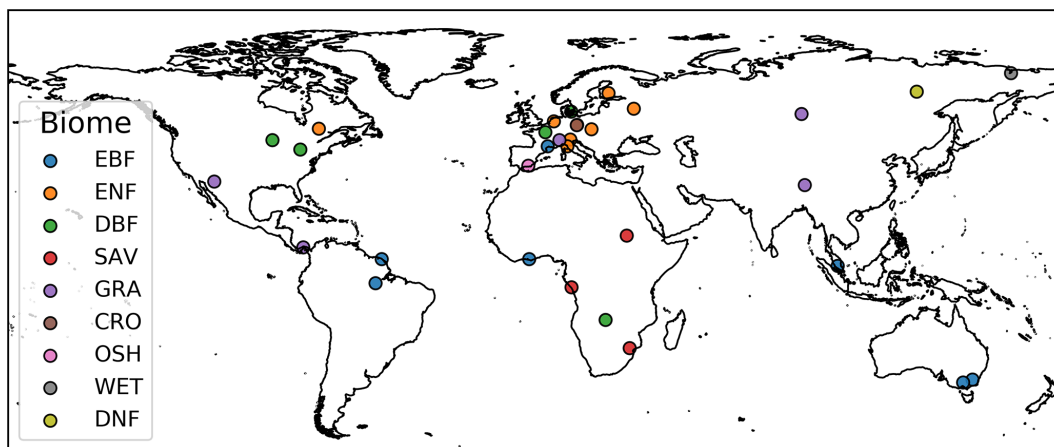
Figure 3. Locations of the 31 FLUXNET sites grouped by biome (Table 1). International Geosphere-Biosphere Programme (IGBP) land classification abbreviations used include evergreen broadleaf forest (EBF), evergreen needleleaf forest (ENF), savanna (SAV), deciduous broadleaf forest (DBF), grassland (GRA), cropland (CRO), open shrubland (OSH), closed shrubland (CSH), permanent wetland (WET), and deciduous needleleaf forest (DNF).

Table 2. Summary table of CLASSIC resources. Resources that do not list a version number will be updated with the most recent versions over time.

CLASSIC webpage	https://cccma.gitlab.io/classic (last access: 19 November 2019)
CLASSIC GitLab code repository	https://gitlab.com/cccma/classic (last access: 19 November 2019)
CLASSIC community Zenodo page	https://zenodo.org/communities/classic/ (last access: 19 November 2019)
CLASSIC code for v.1.0	https://zenodo.org/record/3522407 (last access: 19 November 2019)
CLASSIC Singularity container v.1.0 (including AMBER v.0.1.5)	https://zenodo.org/record/3525249 (last access: 19 November 2019)
Benchmarking data and outputs for CLASSIC v.1.0 (including those generated by AMBER)	https://zenodo.org/record/3525336 (last access: 19 November 2019)
AMBER CRAN page	https://cran.r-project.org/package=amber (last access: 19 November 2019)
CLASSIC discussion forum	http://classic-message-board.158658.n8.nabble.com/ (last access: 19 November 2019)

The other two benchmarking approaches drive CLASSIC with globally gridded data products of meteorological data, vegetation cover, soil permeable depth, and soil textures. In the first of these two approaches, CLASSIC is evaluated against globally gridded datasets for variables where observation-based data are available. The third and final benchmarking approach also uses gridded data products to drive CLASSIC, but the model outputs are evaluated against the entire FLUXNET2015 release of 212 sites by comparing the EC tower data against the model outputs from the CLASSIC grid cell that each tower lies within. The evaluation from these two approaches is presented in a companion paper that evaluates CLASSIC on a global scale (Seiler et al., 2020).

5.1 Further performance metrics

We present here also the statistical metrics of root mean square error and coefficient of determination (both calculated using the scikit-learn Python package; Pedregosa et al., 2011). The coefficient of determination is calculated as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}, \tag{9}$$

where $\hat{y}$ is the predicted value of the nth sample and y_i is the corresponding observed value for total n samples. $\overline{y}$ is found via

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \tag{10}$$

and

$$\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2. \tag{11}$$

While the coefficient of determination is commonly termed R^2, in this formulation, the best possible goodness of fit is 1.0, while the R^2 value can go "negative" (as can be seen from Eq. (9)). The RMSE is calculated by

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2}. \tag{12}$$

5.2 Benchmarking observation-driven CLASSIC at FLUXNET sites

The overall AMBER scores for the comparison between the observation-driven CLASSIC outputs and the 31 FLUXNET sites are shown in Fig. 4 for three energy fluxes (HFLS, HFSS, and RNS) and three carbon fluxes (GPP, NEE, and RECO). For the component scores, CLASSIC generally had higher phase and spatial distribution scores with lower scores for RMSE, which is weighted double the other metrics in calculating the overall score (see Sect. 4.2.1). Most overall scores are between 0.65 and 0.82, with the exception of NEE at 0.44. These scores are discussed more below.

5.2.1 Energy fluxes

The mean seasonal cycle of net radiation (RNS) simulated by CLASSIC compared against observation-based estimates from all FLUXNET sites is shown in Fig. 5. For each site, the plot title shows the site ID (Table 1) followed by the biome type in brackets. The average RMSE across the sites is $19\,\mathrm{W\,m^{-2}}$ with a median R^2 of 0.86 (see Table S1 in the Supplement for per site values). Sites with a pronounced seasonality, typically from the high latitudes, are better captured by CLASSIC compared to lower latitude sites where the seasonal cycle is a result of seasonality in cloudiness and precipitation. CLASSIC also generally has lower peak values than observations, which could indicate that the simulated albedo values are too high. The most poorly simulated sites include BR-Sa1 (EBF), ZM-Mon (DBF), RU-Che (WET), and RU-SkP (DNF). These sites, with the exception of BR-Sa1, have only 2–3 years of observations. The AMBER scores for RNS are all above 0.73, with an overall score of 0.82 and an RMSE score of 0.76, which are the highest scores for the energy fluxes.

The time series of the observed and simulated latent heat fluxes (HFLS) is shown in Fig. 6. Only 23 of the 31 FLUXNET sites reported latent and sensible heat fluxes. While some of the observed fluxes appear questionable, e.g. both ZM-Mon and SD-Dem appear to lose HFLS seasonality for select periods of their records, in contrast, CLASSIC simulates a more consistent seasonal cycle for all years at these sites. CLASSIC consistently simulates lower latent heat fluxes than evident in the observations. However, non-closure of the energy budget at the FLUXNET sites is a well-known issue. The residual of net radiation plus the ground heat flux minus the latent and sensible heat fluxes ranges from 10 % to 30 % of RNS across all FLUXNET sites (Wilson et al., 2002; Cui and Chui, 2019). Since CLASSIC enforces closure of the energy budget, this non-closure by the EC data complicates evaluation of simulated energy fluxes (both HFLS and HFSS). For example, the period of highest mismatch at AU-Tum also corresponds to years with the highest non-closure terms (Fig. 6, and Figs. S10 and S11 in the Supplement) The mean RMSE across sites was

$31.5\,\mathrm{W\,m^{-2}}$ with a median R^2 of 0.08 (recall the coefficient of determination, as calculated here, can go negative; see Sect. 5.1; Table S2 contains all per site values). The AMBER scores for HFLS were lowest in bias and RMSE (0.49 and 0.50, respectively), giving an overall score of 0.66.

The sensible heat fluxes (HFSS) simulated by CLASSIC were generally closer to the EC tower observations (Fig. 7) than HFLS but with a slight overestimate compared to observations. The mean RMSE across sites was $24.7\,\mathrm{W\,m^{-2}}$, with a median R^2 of 0.31 (see Table S3 for per site values). The AMBER scores for HFSS were higher than HFLS, with the lowest component score being S_{rmse} with 0.52 and an overall score of 0.66, equal to the HFLS overall score.

5.2.2 Carbon fluxes

Figure 8 compares the simulated mean seasonal cycle of GPP to FLUXNET observations. While we removed implausible values from the observations, i.e. negative GPP, we did retain other data points that we suspect could be spurious (e.g. 2005 GPP at SD-Dem in Fig. S16). Over all 31 sites, the CLASSIC outputs have an average RMSE of $0.07\,\mathrm{kg\,C\,m^{-2}\,month^{-1}}$ and a median R^2 of 0.15 (see Table S4 for per site values). The AMBER scores for GPP are lowest for S_{rmse} at 0.48 with a S_{overall} of 0.69.

For biomes which are present at multiple sites (EBF, ENF, GRA, DBF), we can draw some general conclusions about the model performance. The EBF sites are primarily located in the low latitudes, with the exception of sites in Turkey (GH-Ank) and France (FR-Pue). These midlatitude EBF sites are for *Quercus ilex* (evergreen oak), which do not correspond well to the CLASSIC formulation for evergreen broadleaf PFT, which is typically assumed to be a tropical PFT. CLASSIC generally has a small bias, simulating somewhat lower GPP for the EBF sites (excepting GH-Ank; Fig. 8) with an RMSE for those sites of $0.08\,\mathrm{kg\,C\,m^{-2}\,month^{-1}}$. This bias is not evident for the ENF sites where CLASSIC generally simulates the sites well with a mean RMSE of $0.05\,\mathrm{kg\,C\,m^{-2}\,month^{-1}}$ but with one Canadian site (CA-Qfo) significantly underestimated. The deciduous broadleaf forest biome sites indicate that CLASSIC may generally underestimate GPP for this biome. The grassland sites are generally well simulated (mean RMSE of $0.5\,\mathrm{kg\,C\,m^{-2}\,month^{-1}}$) with the exception of US-Wkg, where CLASSIC is unable to capture the peak GPP values. The Kendall grasslands (US-Wkg) is a semi-arid site in southern Arizona, USA. The mean annual rainfall is 345 mm, of which most falls in the summer months (Scott et al., 2010). Due to the extreme moisture stress caused by the low precipitation and heavy drought in 2003–2006, CLASSIC is unable to properly establish the vegetation and attain the productivity level observed by the EC tower. This site demonstrates the challenges inherent in this kind of comparison. At US-Wkg, the native vegetation cover of C_4 grasses interspersed with C_3 shrubs was intact prior to the early 2000s drought. Post-
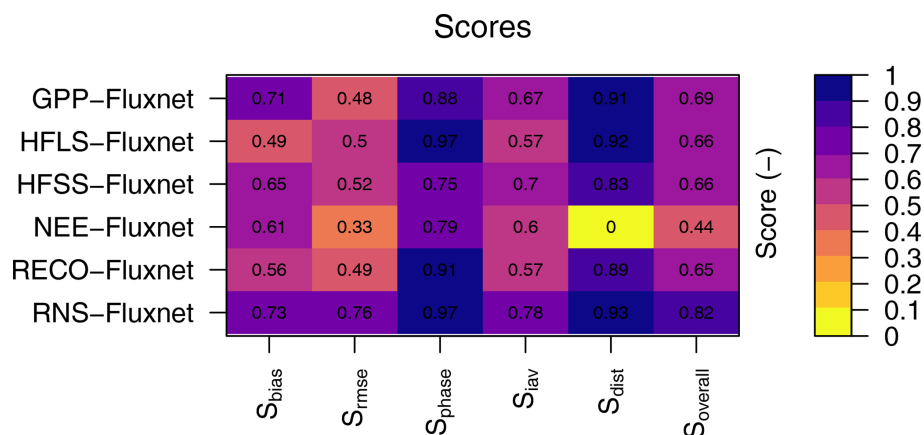
Scores

Figure 4. AMBER site-level scores comparing CLASSIC to the 31 FLUXNET sites (Table 1). Higher scores indicate better model performance. Further AMBER plots are available in the benchmarking archive listed in Table 2.

drought most of the native bunchgrasses were dead, along with high shrub mortality, leading to the establishment of an invasive grass (Scott et al., 2010). The site is also lightly to moderately grazed. Changes in vegetation such as this, along with unique drought conditions (which are only partly captured in the observation period) are difficult to capture adequately by CLASSIC if the goal is to evaluate a model across a large number of sites to capture the diversity of biomes globally. Other sites present similar difficulties to replicate the observed GPP due to complex histories or conditions. As a result, it is important to compare across several sites to reduce the importance of the site-level conditions and to observe the model performance on the whole. The other biomes have too few sites to interpret the results in detail. CLASSIC presently does not represent shrubs as separate PFTs so the shrubland biomes are simulated with trees PFTs in place of shrubs.

Seasonal plots of simulated ecosystem respiration (RECO) are compared against FLUXNET estimated values in Fig. 9. CLASSIC has generally lower variability in RECO than suggested by EC-tower-derived estimates, especially at low-latitude sites such as BR-Sa1, GF-Guy, MY-PSO, ZM-Mon, and SD-Dem. In evergreen needleleaf forest sites, the simulated RECO is generally higher than the FLUXNET values. This overestimate is consistent with high GPP values at these sites. Similarly, the low GPP simulated for the evergreen broadleaf forests yields a low biased RECO. MY-PSO is a notable exception with CLASSIC simulating smaller GPP but larger RECO than the EC-tower-derived quantities. Also, while RU-SkP (DNF) was reasonably well simulated for GPP, CLASSIC simulates a much larger RECO than the EC-tower-derived value. The mean RMSE across all sites is $0.06 \, \mathrm{kg \, C \, m^{-2} \, month^{-1}}$, with a median R^2 of 0.12 (see Table S5 for per site values). The AMBER scores for RECO are similar to the GPP scores with a slightly lower overall score (0.65) but a higher S_{rmse} of 0.49.

NEE is directly observed at eddy covariance towers. The comparison between model and observations for NEE is, however, confounded due to the issues discussed in Sect. 5. Due to these factors, the AMBER NEE scores are lower than all others presented here, with an S_{rmse} of 0.33, S_{dist} of 0, and an overall score of 0.44 (site-level plots of NEE are included in the Supplement).

6 Future directions for CLASSIC development

CLASSIC remains under active development. Scientific developments include the introduction of non-structural carbohydrates (NSCs) (Asaadi et al., 2018), which address a known issue of a delayed spring leaf out for deciduous tree species. The NSC work also lays the groundwork for incorporation of an N cycle in CLASSIC that is presently in development. Other works in progress include the incorporation of high-latitude shrub PFTs for both the physical and biogeochemical aspects of this growth form. Following on from recent work demonstrating CLASSIC's capable performance in simulating the physics of permafrost regions (Melton et al., 2019d), CLASSIC's bulk soil C pools will be replaced with an explicit tracking of soil C per soil layer that is better suited to simulate permafrost C dynamics. As part of this work, a C tracer is being incorporated to allow tracking of ^{14}C or other isotopes through the model C pools. Planned technical developments include further modularization of the physics code, greater adoption of data structures to move away from lengthy argument lists, and modifications to allow the same code to be seamlessly used offline as well as in the CanESM framework. While future releases of CanESM will include CLASSIC, CanESM5 contains older model versions (CLASS v.3.6.2 and CTEM v.1.1). Benchmarking developments include adding peatland EC tower sites to allow evaluation of our peatland module (Wu et al., 2016) as well as sites for biomes that are presently poorly represented.
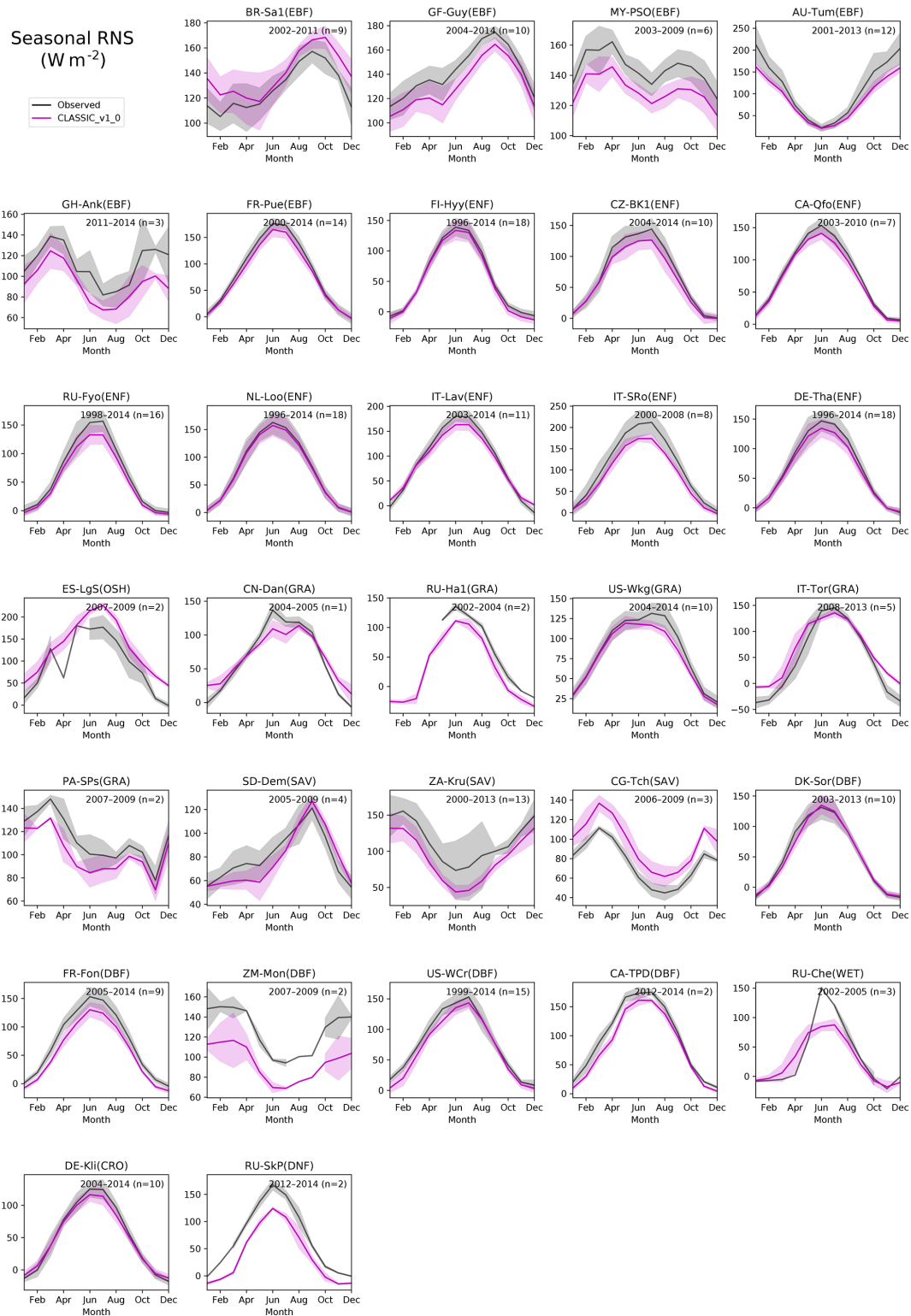
Figure 5. Mean seasonal monthly net radiation for the 31 FLUXNET sites and CLASSIC simulated values. Shaded regions indicate the standard deviation across the sample years. The years of observations for each site are listed along with the site name and IGBP biome which correspond to evergreen broadleaf forest, EBF; evergreen needleleaf forest, ENF; grassland, GRA; deciduous broadleaf forest, DBF; cropland, CRO; open shrubland, OSH; permanent wetland, WET; and deciduous broadleaf forest, DBF.
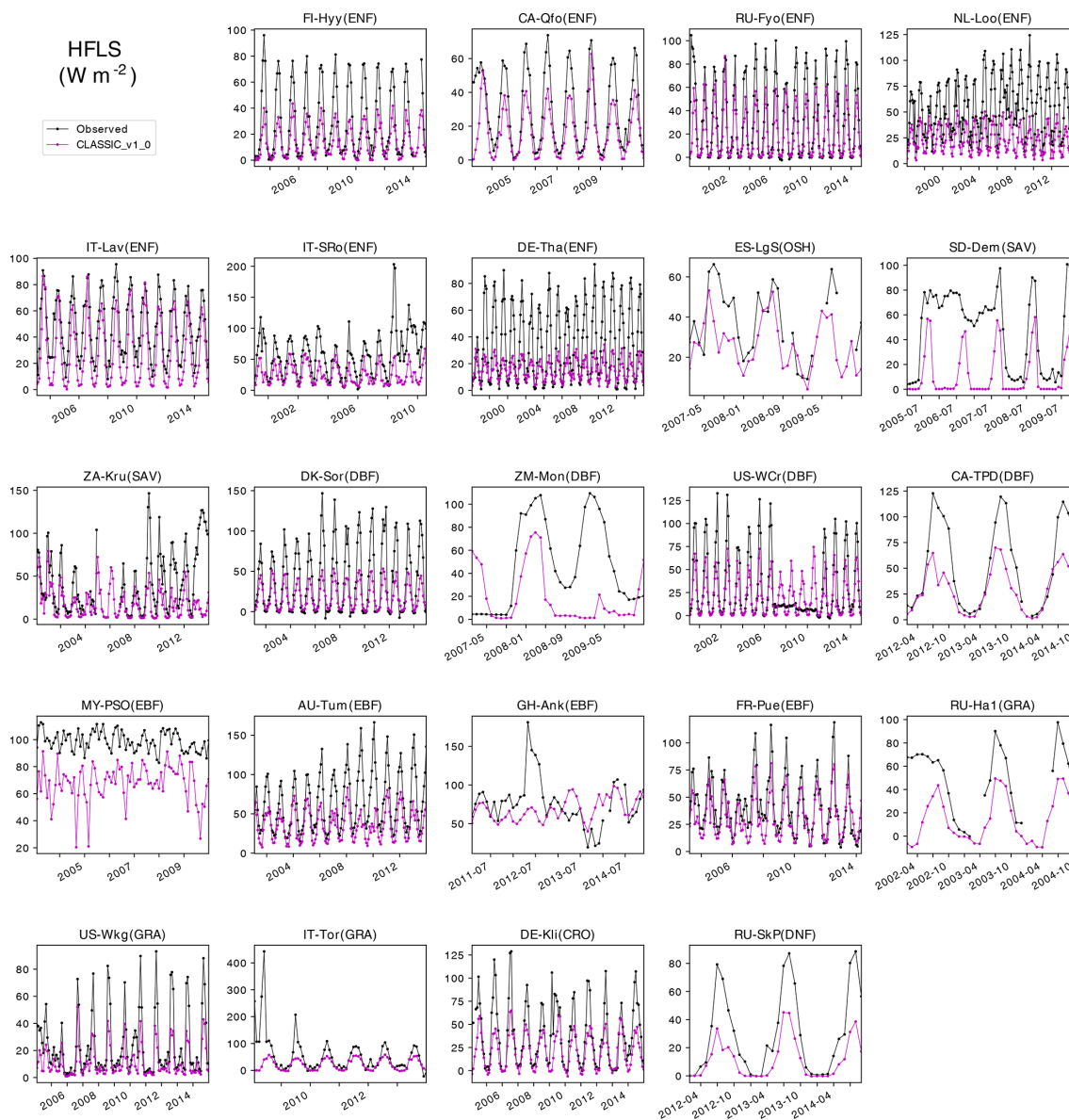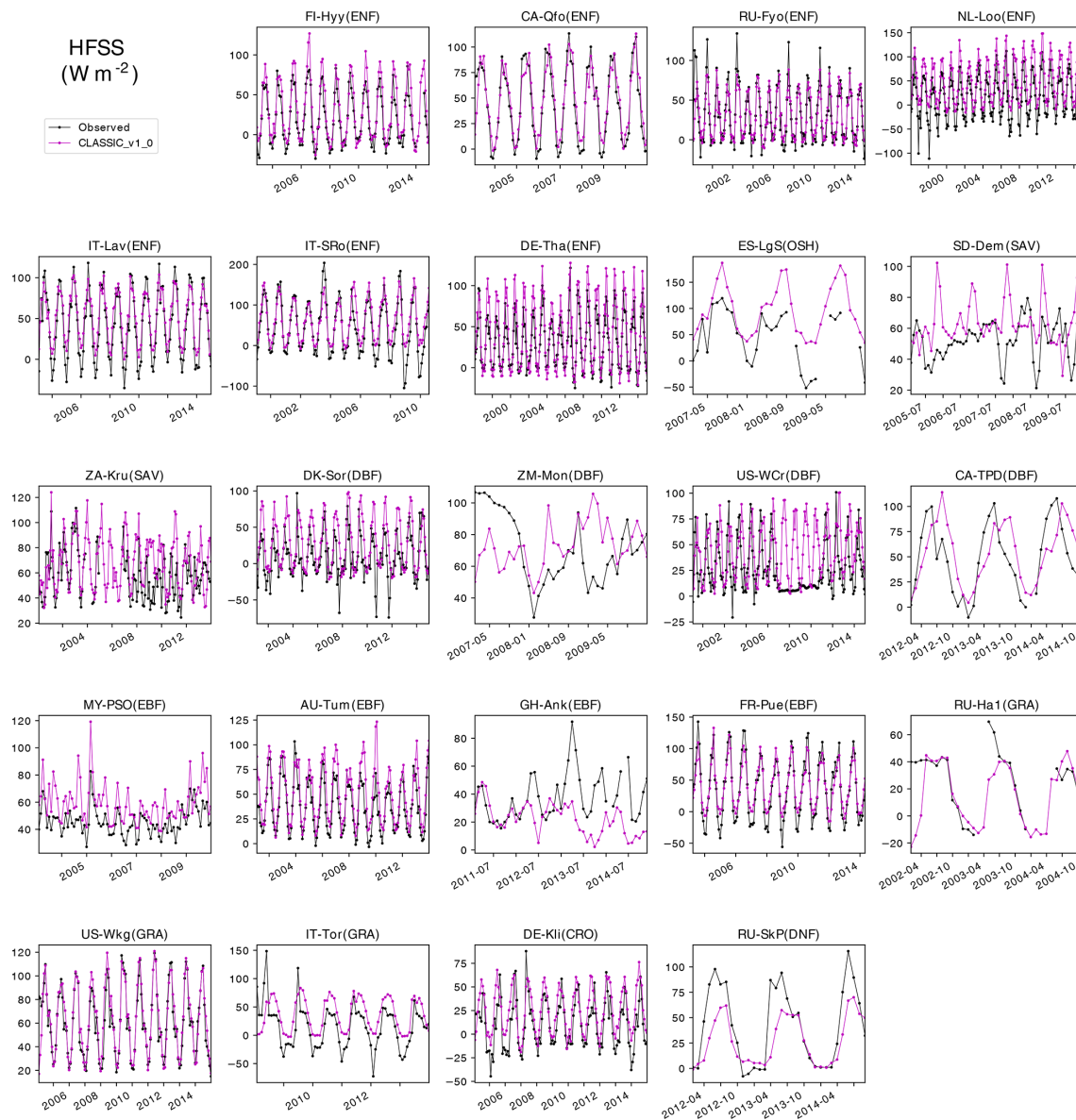
Figure 6. Monthly observed and simulated latent heat fluxes for 23 FLUXNET sites. See caption of Fig. 5 for biome names. Shaded regions indicate the standard deviation across the sample years. Table 1 lists the full site names.

7 Conclusions

CLASSIC is the open-source community successor to CLASS-CTEM. CLASS and CTEM have decades of development behind them and have been extensively evaluated and applied at scales from site level, involving evaluation against data from EC towers, to coupling to an atmosphere model as part of CanESM. While the science within CLASS-CTEM has continued to advance in response to changing scientific questions and applications, the technical aspects of CLASS-CTEM have fallen behind. Our study details the transformation of CLASS-CTEM from a primarily internally developed model to one designed to encourage community use and de-

velopment. Table 2 summarizes the URLs of CLASSIC resources.

The work detailed here to create CLASSIC addresses the barriers to code sharing identified by Easterbrook (2014): portability, configurability, entrenchment, model–data blur, and provenance. Specifically, we have addressed code portability by providing a software container for CLASSIC to run in serial or parallel modes, along with rewriting the model I/O to use community standard NetCDF format files. Configurability of CLASSIC is handled by providing extensive documentation of appropriate model setup via Doxygen, as well as benchmarking sites that can be used as examples of how to configure CLASSIC appropriately along with scripts to
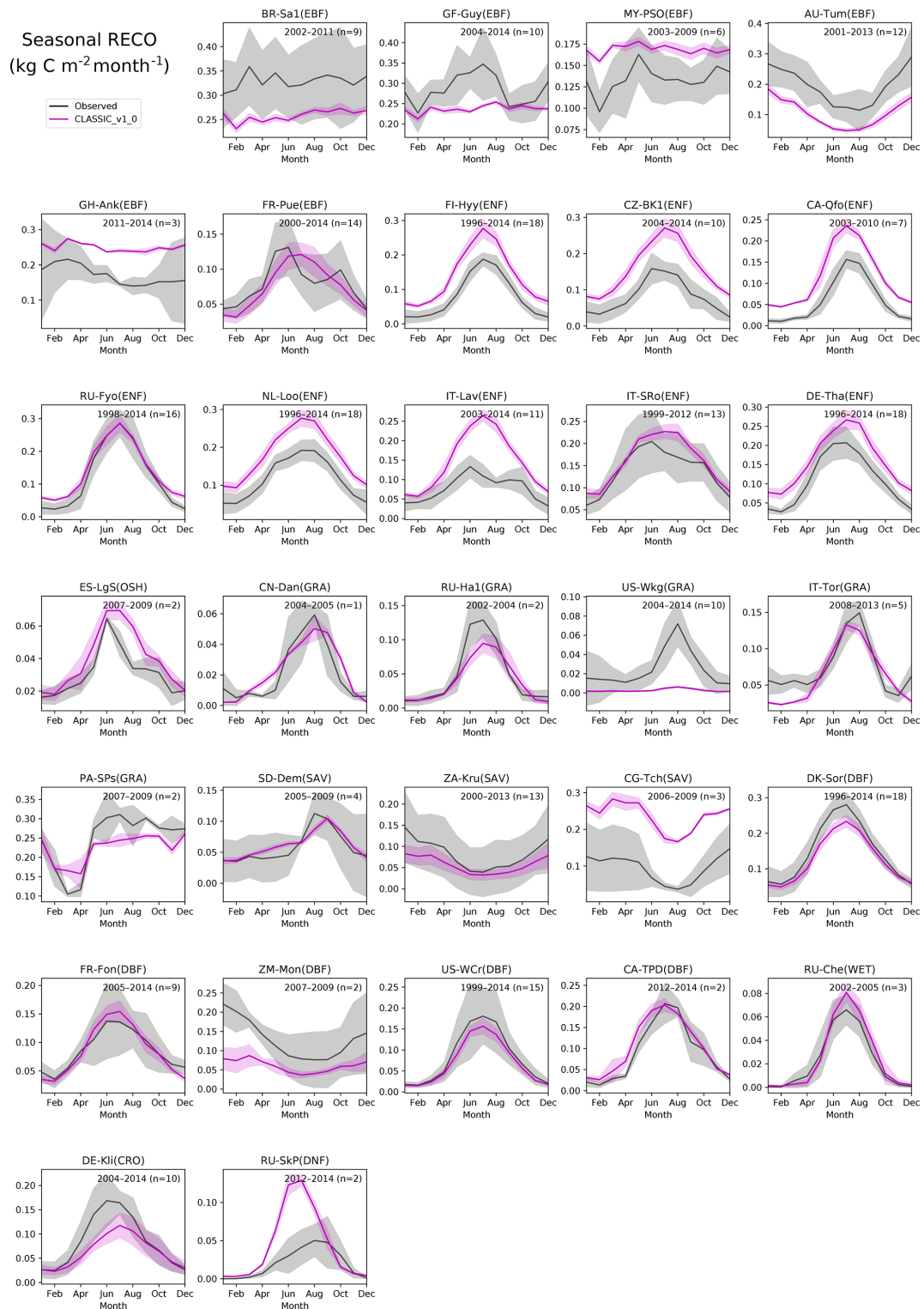
Figure 7. Monthly observed and simulated sensible heat fluxes for 23 FLUXNET sites. See caption of Fig. 5 for biome names. Shaded regions indicate the standard deviation across the sample years. Table 1 lists the full site names.

properly setup and run CLASSIC over 31 FLUXNET sites. Entrenchment is addressed most easily going forward as we adopt an open community approach to model development including the use of issue trackers and forums to outline new model developments and bug fixes. Model–data blur is reduced by using the FLUXNET sites for our site-level benchmarking, as they required minimal processing prior to use. Lastly, provenance is made possible by our use of a Singularity software container and our benchmarking pipeline (Appendix A) where any user can reproduce the site-level plots and AMBER scores presented here. AMBER is made available to the community as a means to guide model development where low scores highlight possible areas for improve-

ment. Additionally, the use of AMBER can help ensure that improvements in one area of CLASSIC are not degrading the performance of other variables. We encourage members of the scientific community to use, develop, and contribute back to CLASSIC.
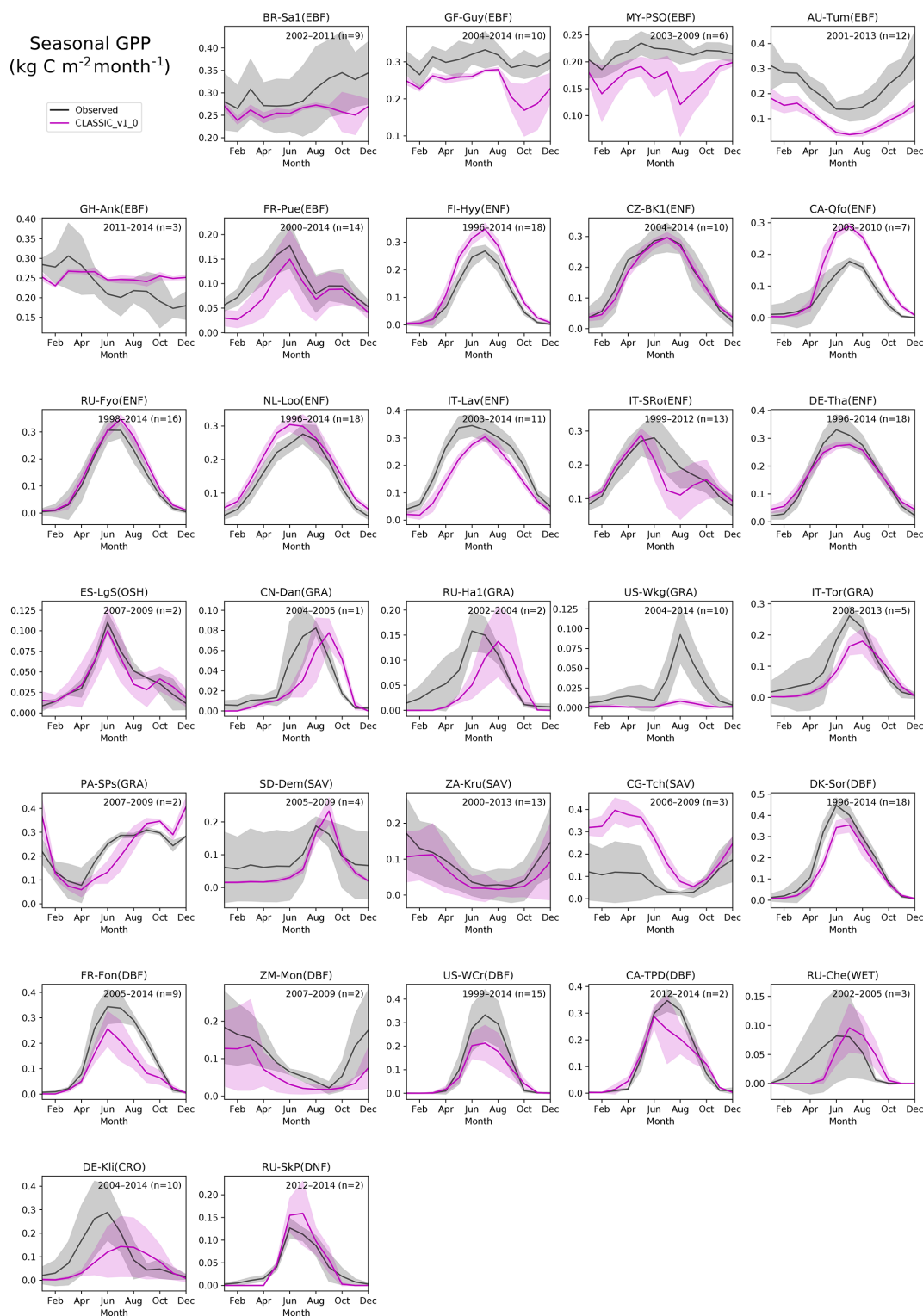
Figure 8. The mean seasonal cycle of GPP for 31 FLUXNET sites as simulated by CLASSIC. The IGBP biome of each site is listed alongside its FLUXNET site ID (see Table 1). Shaded regions indicate the standard deviation across the sample years. See caption of Fig. 5 for biome names.

Figure 9. The mean seasonal cycle of RECO for 31 FLUXNET sites as simulated by CLASSIC. The IGBP biome of each site is listed alongside its FLUXNET name (see Table 1). Shaded regions indicate the standard deviation across the sample years. See caption of Fig. 5 for biome names.

Appendix A: Quick-start guide to running CLASSIC at FLUXNET sites

This document will walk through the basic setup, compilation, and running of CLASSIC at FLUXNET sites.

A1 Requirements

This guide assumes that the reader is using a Linux machine, and that they have Singularity and tar installed. The installation guide for Singularity can be found at https://sylabs.io/docs/#singularity (last access: 19 November 2019). Tar can be installed from https://www.gnu.org/software/tar/ (last access: 19 November 2019). If using a Windows or Mac machine, see instructions on setting up Singularity at https://sylabs.io/guides/3.3/user-guide/installation.html#install-on-windows-or-mac (last access: 19 November 2019).

A2 Obtaining the source code

The first step is setting up a directory structure. For the sake of this guide, we will work out of a directory located at "/home/eccc". So first we navigate to our chosen directory location:

```
cd /home/eccc
```

The CLASSIC source code is hosted on GitLab and can be cloned with

```
git clone https://gitlab.com/cccma/
CLASSIC.git
```

If you are familiar with Git's workflow and wish to contribute to the code base in the future, then it is a good idea to first fork a copy to your own GitLab account. Once you have a fork, you can clone the repository with

```
git clone https://gitlab.com/
**your_gitlab_username**/CLASSIC.git
```

Once the cloning process is complete, there should be a directory titled CLASSIC located in your directory:

```
/home/eccc/CLASSIC
```

A3 Obtaining other necessary files

All other necessary files are found on Zenodo in the form of compressed packages. We will use the "tar" command to unpack them beside the CLASSIC repository. First, navigate to the CLASSIC community Zenodo page at https://zenodo.org/communities/classic (last access: 19 November 2019) and download the FLUXNET, and CLASSIC_container tar files. Your directory should now contain the following:

```
/home/eccc/CLASSIC
/home/eccc/CLASSIC_container.tar.gz
/home/eccc/FLUXNET.tar.gz
```

From the "/home/eccc" directory, we will decompress the tarballs and move the decompressed files to their correct locations. The following list of commands will accomplish this:

```
tar xzvf CLASSIC_container.tar.gz
tar xzvf FLUXNET.tar.gz

mv CLASSIC_container.simg CLASSIC/

mv FLUXNET/TRENDY_CO2_1700_2018.nc
CLASSIC/inputFiles/
mv FLUXNET/FLUXNETsites CLASSIC/
inputFiles/
mv FLUXNET/observationalDataFLUXNET
CLASSIC/inputFiles/
mv FLUXNET/benchmark_CLASSIC_output
CLASSIC/benchmark/
mv FLUXNET/benchmark_CLASSIC_plots
CLASSIC/benchmark/
mv FLUXNET/benchmark_CLASSIC_AMBER
CLASSIC/benchmark/
rm -rf FLUXNET/
```

A4 Using the container

Navigate to the CLASSIC root directory. The rest of the tutorial will assume that you stay in this working directory:

```
cd /home/eccc/CLASSIC
```

Since much of CLASSIC's functionality relies on the libraries contained within the container, we execute commands through the container with this syntax:

```
singularity exec CLASSIC_container.simg
[commands here]
```

Most of the calls like this are contained within the automated run scripts so their direct use is minimized. However, if you wish to deviate from the automated run scripts, that command will be of great use.

First, we will make sure we have a clean working directory.

```
singularity exec CLASSIC_container.simg
make mode=serial clean
```

We are now ready to compile the source code for serial simulation. This is done with the following command:

```
singularity exec CLASSIC_container.simg
make mode=serial
```

The compilation process can take several minutes. If you get errors, check that you have the latest stable version of CLASSIC pulled from the repository.

Table A1. Equations used for computing spatially integrated scalar scores for each variable. The four steps refer to (1) the definition of the statistical metric, (2) non-dimensionalization, (3) conversion to unit interval, and (4) spatial integration. The integration limits t_0 and t_f are the initial and final time step, respectively. The variables c_{mod} and c_{ref} refer to the monthly mean annual cycle of the model and reference data, respectively. "iav" is the interannual variability. "crmse" is the "centralized RMSE", which is the RMSE of the anomalies. This ensures that the RMSE score excludes also considering the bias, which is captured by the bias score.

Score	Step	Equation
S_{bias}	(1)	$\text{bias}(\lambda, \phi) = \overline{v_{mod}}(\lambda, \phi) - \overline{v_{ref}}(\lambda, \phi)$
	(2)	$\varepsilon_{bias} = |\text{bias}(\lambda, \phi)|/\sigma_{ref}(\lambda, \phi)$, where σ_{ref} is the standard deviation of the reference data
	(3)	$s_{bias}(\lambda, \phi) = e^{-\varepsilon_{bias}(\lambda, \phi)}$
	(4)	$S_{bias} = \overline{s_{bias}}$
S_{rmse}	(1)	$\text{rmse}(\lambda, \phi) = \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} (v_{mod}(t, \lambda, \phi) - v_{ref}(t, \lambda, \phi))^2 dt}$
	(2)	$\varepsilon_{rmse}(\lambda, \phi) = \text{crmse}(\lambda, \phi)/\sigma_{ref}(\lambda, \phi)$, where $\text{crmse}(\lambda, \phi)$
		$= \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} [(v_{mod}(t, \lambda, \phi) - \overline{v_{mod}}(\lambda, \phi)) - (v_{ref}(t, \lambda, \phi) - \overline{v_{ref}}(\lambda, \phi))]^2 dt}$
	(3)	$s_{rmse}(\lambda, \phi) = e^{-\varepsilon_{rmse}(\lambda, \phi)}$
	(4)	$S_{rmse} = \overline{\overline{s_{rmse}}}$
S_{phase}	(1)	$\theta(\lambda, \phi) = \max(c_{mod}(t, \lambda, \phi)) - \max(c_{ref}(t, \lambda, \phi))$
	(2)	not applicable, as units are consistent across all variables
	(3)	$s_{phase}(\lambda, \phi) = \frac{1}{2}[1 + \cos(\frac{2\pi\theta(\lambda, \phi)}{365})]$
	(4)	$S_{phase} = \overline{s_{phase}}$
S_{iav}	(1)	$\text{iav}_{ref}(\lambda, \phi) = \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} (v_{ref}(t, \lambda, \phi) - c_{ref}(t, \lambda, \phi))^2 dt}$,
		$\text{iav}_{mod}(\lambda, \phi) = \sqrt{\frac{1}{t_f - t_0} \int_{t_0}^{t_f} (v_{mod}(t, \lambda, \phi) - c_{mod}(t, \lambda, \phi))^2 dt}$
	(2)	$\varepsilon_{iav} = |(\text{iav}_{mod}(\lambda, \phi) - \text{iav}_{ref}(\lambda, \phi))|/\text{iav}_{ref}(\lambda, \phi)$
	(3)	$s_{iav}(\lambda, \phi) = e^{-\varepsilon_{iav}(\lambda, \phi)}$
	(4)	$S_{iav} = \overline{\overline{s_{iav}}}$
S_{dist}	(1)	$\sigma = \sigma_{\overline{v_{mod}}}/\sigma_{\overline{v_{ref}}}$
	(2)	not applicable
	(3)	not applicable
	(4)	$S_{dist} = 2(1 + R)/(\sigma + \frac{1}{\sigma})^2$,
		where R is the spatial correlation coefficient of $\overline{v_{ref}}(\lambda, \phi)$ and $\overline{v_{mod}}(\lambda, \phi)$

A5 Setting up FLUXNET site-level runs

With the FLUXNET sites in place and CLASSIC compiled, we can now setup the job options file(s) to run over the FLUXNET sites. To do this, run the job options script by invoking

```
tools/siteLevelFLUXNET/prep_jobopts.sh
```

This script puts a new job options file in every FLUXNET site directory tailored to that particular site. The "singularity exec" command is not necessary for this particular script.

A6 Running CLASSIC over FLUXNET sites

This guide will cover two methods for running the CLASSIC binary on FLUXNET sites. They can be run individually or through the batch-run script.

A6.1 Running all sites with the batch script (recommended)

A script is provided in the CLASSIC repository to run all FLUXNET sites, provided the directory structure of this document is followed. The script will run for all sites it finds within the inputFiles/FLUXNETsites directory and is invoked by

```
tools/siteLevelFLUXNET/run_sites.sh
```

All model restart files are provided in a "pre-spunup" state. Still, running all sites can be a lengthy process on older machines, and sometimes you may not want or need all of them. To reduce the number of sites being run, remove the undesired site directories from inputFiles/FLUXNETsites. For example, if you do not need to process "DK-Sor", you could do

```
mv inputFiles/FLUXNETsites/DK-Sor
inputFiles
```

Table A2. Site-level statistical metrics for calculating S_{rmse} at the 31 FLUXNET sites. "crmse" is the "centralized RMSE", which is the RMSE of the anomalies. This ensures that the RMSE score excludes also considering the bias, which is captured by the bias score.

Variable	Unit	RMSE	crmse	σ_{ref}	ϵ_{rmse} (−)	S_{rmse} (−)
GPP	$gC\,m^{-2}\,d^{-1}$	2.26	1.88	2.74	0.78	0.48
HFLS	$W\,m^{-2}$	32.37	22.31	31.50	0.72	0.50
HFSS	$W\,m^{-2}$	25.10	19.55	32.03	0.70	0.52
NEE	$gC\,m^{-2}\,d^{-1}$	1.95	1.56	1.48	1.24	0.33
RECO	$gC\,m^{-2}\,d^{-1}$	1.90	1.29	1.82	0.75	0.49
RNS	$W\,m^{-2}$	19.34	12.32	48.56	0.28	0.76

This would simply move the site out of the way, and it could be placed back into the FLUXNETsites directory when needed.

A6.2 Running individual sites (advanced)

Individual sites are run through the Singularity container directly referencing the job options file for that particular site. The command takes the following form:

```
singularity exec
[Location of CLASSIC container]
[location of CLASSIC binary]\
[Location of job options file
for that site]
[Longitude/Latitude of the site]
```

Since this is a site-level simulation, the shorthand 0/0 or 0/0/0/0 can be used in place of the actual longitude and latitude of the site. Piecing it all together, if we want to run on the site AU-Tum, the command would be

```
singularity exec CLASSIC_container.simg
bin/CLASSIC_serial\inputFiles/
FLUXNETsites/AU-Tum/job_options_file.txt
0/0/0/0
```

If you decide to shell into the singularity container (recommended if you are familiar with Singularity and terminal commands), then this becomes

```
bin/CLASSIC_serial inputFiles/
FLUXNETsites/AU-Tum/job_options_file.txt
0/0/0/0
```

More information on running CLASSIC can be found in the CLASSIC manual at https://cccma.gitlab.io/classic/index.html (last access: 19 November 2019).

A7 Processing output

Whether you ran multiple sites with the batch job or a single site individually, the output can be found in "outputFiles/FLUXNETsites/sitename/". These outputs are in the form of NetCDF files. More information on NetCDF can be found at https://www.unidata.ucar.edu/software/netcdf/ (last access: 19 November 2019), but briefly, it is a machine-agnostic array-oriented form of data storage.

A script is provided to convert these data to csv format, which may be easier to work with if unfamiliar with NetCDFs, as well as generate plots of several output variables against observational values of those variables. The script is run with

```
tools/siteLevelFLUXNET/
process_outputs.sh
```

Note that if not all sites are run with the batch script, errors will be seen in the output of this script. This is expected and does not mean that the processing has failed.

Once the output plots have been generated, the "process_outputs" script will run the AMBER tool. More information on AMBER can be found at https://cran.rstudio.com/web/packages/amber/amber.pdf (last access: 19 November 2019).

After the script completes, PDF copies of the plots are found in "outputFiles/plots", while AMBER results are in "outputFiles/AMBER".

References

Abraham, N. L., Archibald, A. T., Cresswell, P., Cusworth, S., Dalvi, M., Matthews, D., Wardle, S., and Whitehouse, S.: Using a virtual machine environment for developing, testing, and training for the UM-UKCA composition-climate model, using Unified Model version 10.9 and above, Geosci. Model Dev., 11, 3647–3657, https://doi.org/10.5194/gmd-11-3647-2018, 2018.

Abramowitz, G.: Towards a public, standardized, diagnostic benchmarking system for land surface models, Geosci. Model Dev., 5, 819–827, https://doi.org/10.5194/gmd-5-819-2012, 2012.

Arango, C., Dernat, R., and Sanabria, J.: Performance Evaluation of Container-based Virtualization for High Performance Computing Environments, Revista UIS Ingenierías, 18, 31–42, https://doi.org/10.18273/revuin.v18n4-2019003, 2017.

Arora, V.: Land surface modelling in general circulation models: A hydrological perspective, Ph.D. thesis, Department of Civil and Environmental Engineering, University of Melbourne, 1997.

Arora, V. K. and Boer, G. J.: A Representation of Variable Root Distribution in Dynamic Vegetation Models, Earth Interact., 7, 1–19, https://doi.org/10.1175/1087-3562(2003)007<0001:AROVRD>2.0.CO;2, 2003.

Arora, V. K. and Boer, G. J.: A parameterization of leaf phenology for the terrestrial ecosystem component of climate models, Global Chang. Biol., 11, 39–59, https://doi.org/10.1111/j.1365-2486.2004.00890.x, 2005a.

Arora, V. K. and Boer, G. J.: Fire as an interactive component of dynamic vegetation models, J. Geophys. Res., 110, G02008, https://doi.org/10.1029/2005JG000042, 2005b.

Arora, V. K. and Boer, G. J.: Simulating Competition and Coexistence between Plant Functional Types in a Dynamic Vegetation Model, Earth Interact., 10, 1–30, https://doi.org/10.1175/EI170.1, 2006.

Arora, V. K. and Boer, G. J.: Uncertainties in the 20th century carbon budget associated with land use change, Global Chang. Biol., 16, 3327–3348, https://doi.org/10.1111/j.1365-2486.2010.02202.x, 2010.

Arora, V. K. and Melton, J. R.: Reduction in global area burned and wildfire emissions since 1930s enhances carbon uptake by land, Nat. Commun., 9, 1326, https://doi.org/10.1038/s41467-018-03838-0, 2018.

Arora, V. K., Chiew, F. H. S., and Grayson, R. B.: A river flow routing scheme for general circulation models, J. Geophys. Res., 104, 14347–14357, https://doi.org/10.1029/1999JD900200, 1999.

Arora, V. K., Scinocca, J. F., Boer, G. J., Christian, J. R., Denman, K. L., Flato, G. M., Kharin, V. V., Lee, W. G., and Merryfield, W. J.: Carbon emission limits required to satisfy future representative concentration pathways of greenhouse gases, Geophys. Res. Lett., 38, L05805, https://doi.org/10.1029/2010GL046270, 2011.

Arora, V. K., Melton, J. R., and Plummer, D.: An assessment of natural methane fluxes simulated by the CLASS-CTEM model, Biogeosciences, 15, 4683–4709, https://doi.org/10.5194/bg-15-4683-2018, 2018.

Asaadi, A., Arora, V. K., Melton, J. R., and Bartlett, P.: An improved parameterization of leaf area index (LAI) seasonality in the Canadian Land Surface Scheme (CLASS) and Canadian Terrestrial Ecosystem Model (CTEM) modelling framework, Biogeosciences, 15, 6885–6907, https://doi.org/10.5194/bg-15-6885-2018, 2018.

Bailey, W. G., Saunders, I. R., Bowers, J. D., and Verseghy, D. L.: Application of the Canadian land surface scheme to a full canopy crop during a drying cycle, Atmos.-Ocean, 38, 57–80, https://doi.org/10.1080/07055900.2000.9649640, 2000.

Bartlett, P. A., MacKay, M. D., and Verseghy, D. L.: Modified snow algorithms in the Canadian land surface scheme: Model runs and sensitivity analysis at three boreal forest stands, Atmos.-Ocean, 44, 207–222, https://doi.org/10.3137/ao.440301, 2006.

Best, M. J., Pryor, M., Clark, D. B., Rooney, G. G., Essery, R. L. H., Ménard, C. B., Edwards, J. M., Hendry, M. A., Porson, A., Gedney, N., Mercado, L. M., Sitch, S., Blyth, E., Boucher, O., Cox, P. M., Grimmond, C. S. B., and Harding, R. J.: The Joint UK Land Environment Simulator (JULES), model description – Part 1: Energy and water fluxes, Geosci. Model Dev., 4, 677–699, https://doi.org/10.5194/gmd-4-677-2011, 2011.

Best, M. J., Abramowitz, G., Johnson, H. R., Pitman, A. J., Balsamo, G., Boone, A., Cuntz, M., Decharme, B., Dirmeyer, P. A., Dong, J., Ek, M., Guo, Z., Haverd, V., van den Hurk, B. J. J., Nearing, G. S., Pak, B., Peters-Lidard, C., Santanello, J. A., Stevens, L., and Vuichard, N.: The Plumbing of Land Surface Models: Benchmarking Model Performance, J. Hydrometeorol., 16, 1425–1442, 2015.

Blyth, E., Gash, J., Lloyd, A., Pryor, M., Weedon, G. P., and Shuttleworth, J.: Evaluating the JULES Land Surface Model Energy Fluxes Using FLUXNET Data, J. Hydrometeorol., 11, 509–519, https://doi.org/10.1175/2009jhm1183.1, 2010.

Brown, R., Bartlett, P., MacKay, M., and Verseghy, D.: Evaluation of snow cover in CLASS for SnowMIP, Atmos.-Ocean, 44, 223–238, https://doi.org/10.3137/ao.440302, 2006.

Clark, D. B., Mercado, L. M., Sitch, S., Jones, C. D., Gedney, N., Best, M. J., Pryor, M., Rooney, G. G., Essery, R. L. H., Blyth, E., Boucher, O., Harding, R. J., Huntingford, C., and Cox, P. M.: The Joint UK Land Environment Simulator (JULES), model description – Part 2: Carbon fluxes and vegetation dynamics, Geosci. Model Dev., 4, 701–722, https://doi.org/10.5194/gmd-4-701-2011, 2011.

Collier, N., Hoffman, F. M., Lawrence, D. M., Keppel-Aleks, G., Koven, C. D., Riley, W. J., Mu, M., and Randerson, J. T.: The International Land Model Benchmarking (ILAMB) System: Design, Theory, and Implementation, J. Adv. Model. Earth Syst., 10, 2731–2754, https://doi.org/10.1029/2018MS001354, 2018.

Cui, W. and Chui, T. F. M.: Temporal and spatial variations of energy balance closure across FLUXNET research sites, Agr. Forest Meteorol., 271, 12–21, https://doi.org/10.1016/j.agrformet.2019.02.026, 2019.

Curry, C. L.: Modeling the soil consumption of atmospheric methane at the global scale, Global Biogeochem. Cy., 21, GB4012, https://doi.org/10.1029/2006GB002818, 2007.

Easterbrook, S. M.: Open code for open science?, Nat. Geosci., 7, 779–781, https://doi.org/10.1038/ngeo2283, 2014.

Finley, K., Barber, G., Lapowsky, I., WIRED Staff, Simonite, T., Vogelstein, F., and Tiku, N.: The WIRED Guide to Open Source Software, Wired, 2019.

Forkel, M., Andela, N., Harrison, S. P., Lasslop, G., van Marle, M., Chuvieco, E., Dorigo, W., Forrest, M., Hantson, S., Heil, A., Li, F., Melton, J., Sitch, S., Yue, C., and Arneth, A.: Emergent relationships with respect to burned area in global satellite observations and fire-enabled vegetation models, Biogeosciences, 16, 57–76, https://doi.org/10.5194/bg-16-57-2019, 2019.

Ganji, A., Sushama, L., Verseghy, D., and Harvey, R.: On improving cold region hydrological processes in the Canadian Land Surface Scheme, Theor. Appl. Climatol., 127, 45–59, https://doi.org/10.1007/s00704-015-1618-4, 2015.

Hacker, J. P., Exby, J., Gill, D., Jimenez, I., Maltzahn, C., See, T., Mullendore, G., and Fossell, K.: A Containerized Mesoscale Model and Analysis Toolkit to Accelerate Classroom Learning, Collaborative Research, and Uncertainty Quantification, B. Am. Meteorol. Soc., 98, 1129–1138, https://doi.org/10.1175/BAMS-D-15-00255.1, 2016.

Hantson, S., Arneth, A., Harrison, S. P., Kelley, D. I., Prentice, I. C., Rabin, S. S., Archibald, S., Mouillot, F., Arnold, S. R., Artaxo, P., Bachelet, D., Ciais, P., Forrest, M., Friedlingstein, P., Hickler, T., Kaplan, J. O., Kloster, S., Knorr, W., Lasslop, G., Li, F., Mangeon, S., Melton, J. R., Meyn, A., Sitch, S., Spessa, A., van der Werf, G. R., Voulgarakis, A., and Yue, C.: The status and challenge of global fire modelling, Biogeosciences, 13, 3359–3375, https://doi.org/10.5194/bg-13-3359-2016, 2016.

Haverd, V., Smith, B., Nieradzik, L., Briggs, P. R., Woodgate, W., Trudinger, C. M., Canadell, J. G., and Cuntz, M.: A new version of the CABLE land surface model (Subversion revision r4601) incorporating land use and land cover change, woody vegetation demography, and a novel optimisation-based approach to plant coordination of photosynthesis, Geosci. Model Dev., 11, 2995–3026, https://doi.org/10.5194/gmd-11-2995-2018, 2018.

Henderson-Sellers, A., Yang, Z.-L., and Dickinson, R. E.: The Project for Intercomparison of Land-surface Parameterization Schemes, B. Am. Meteorol. Soc., 74, 1335–1350, 1993.

Koster, R. D., Sud, Y. C., Guo, Z., Dirmeyer, P. A., Bonan, G., Oleson, K. W., Chan, E., Verseghy, D., Cox, P., Davies, H., Kowalczyk, E., Gordon, C. T., Kanae, S., Lawrence, D., Liu, P., Mocko, D., Lu, C.-H., Mitchell, K., Malyshev, S., McAvaney, B., Oki, T., Yamada, T., Pitman, A., Taylor, C. M., Vasic, R., and Xue, Y.: GLACE: The Global Land–Atmosphere Coupling Experiment. Part I: Overview, J. Hydrometeorol., 7, 590–610, 2006.

Kothavala, Z., Arain, M. A., Black, T. A., and Verseghy, D.: The simulation of energy, water vapor and carbon dioxide fluxes over common crops by the Canadian Land Surface Scheme (CLASS), Agr. For. Meteorol., 133, 89–108, 2005.

Kurtzer, G. M., Sochat, V., and Bauer, M. W.: Singularity: Scientific containers for mobility of compute, PLoS One, 12, e0177459, https://doi.org/10.1371/journal.pone.0177459, 2017.

Lawrence, D. M., Fisher, R. A., Koven, C. D., Oleson, K. W., Swenson, S. C., Bonan, G., Collier, N., Ghimire, B., Kampenhout, L., Kennedy, D., Kluzek, E., Lawrence, P. J., Li, F., Li, H., Lombardozzi, D., Riley, W. J., Sacks, W. J., Shi, M., Vertenstein, M., Wieder, W. R., Xu, C., Ali, A. A., Badger, A. M., Bisht, G., Broeke, M., Brunke, M. A., Burns, S. P., Buzan, J., Clark, M., Craig, A., Dahlin, K., Drewniak, B., Fisher, J. B., Flanner, M., Fox, A. M., Gentine, P., Hoffman, F., Keppel-Aleks, G., Knox, R., Kumar, S., Lenaerts, J., Leung, L. R., Lipscomb, W. H., Lu, Y., Pandey, A., Pelletier, J. D., Perket, J., Randerson, J. T., Ricciuto, D. M., Sanderson, B. M., Slater, A., Subin, Z. M., Tang, J., Thomas, R. Q., Val Martin, M., and Zeng, X.: The Community Land Model version 5: Description of new features, benchmarking, and impact of forcing uncertainty, J. Adv. Model. Earth Syst., 11, 4245–4287, https://doi.org/10.1029/2018MS001583, 2019.

Le, E. and Paz, D.: Performance Analysis of Applications Using Singularity Container on SDSC Comet, in: Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact, PEARC17, 66:1–66:4, ACM, New York, NY, USA, https://doi.org/10.1145/3093338.3106737, 2017.

Le Quéré, C., Andrew, R. M., Canadell, J. G., Sitch, S., Korsbakken, J. I., Peters, G. P., Manning, A. C., Boden, T. A., Tans, P. P., Houghton, R. A., Keeling, R. F., Alin, S., Andrews, O. D., Anthoni, P., Barbero, L., Bopp, L., Chevallier, F., Chini, L. P., Ciais, P., Currie, K., Delire, C., Doney, S. C., Friedlingstein, P., Gkritzalis, T., Harris, I., Hauck, J., Haverd, V., Hoppema, M., Klein Goldewijk, K., Jain, A. K., Kato, E., Körtzinger, A., Landschützer, P., Lefèvre, N., Lenton, A., Lienert, S., Lombardozzi, D., Melton, J. R., Metzl, N., Millero, F., Monteiro, P. M. S., Munro, D. R., Nabel, J. E. M. S., Nakaoka, S.-I., O'Brien, K., Olsen, A., Omar, A. M., Ono, T., Pierrot, D., Poulter, B., Rödenbeck, C., Salisbury, J., Schuster, U., Schwinger, J., Séférian, R., Skjelvan, I., Stocker, B. D., Sutton, A. J., Takahashi, T., Tian, H., Tilbrook, B., van der Laan-Luijkx, I. T., van der Werf, G. R., Viovy, N., Walker, A. P., Wiltshire, A. J., and Zaehle, S.: Global Carbon Budget 2016, https://doi.org/10.5194/essd-8-605-2016, 2016.

Le Quéré, C., Andrew, R. M., Friedlingstein, P., Sitch, S., Hauck, J., Pongratz, J., Pickers, P. A., Korsbakken, J. I., Peters, G. P., Canadell, J. G., Arneth, A., Arora, V. K., Barbero, L., Bastos, A., Bopp, L., Chevallier, F., Chini, L. P., Ciais, P., Doney, S. C., Gkritzalis, T., Goll, D. S., Harris, I., Haverd, V., Hoffman, F. M., Hoppema, M., Houghton, R. A., Hurtt, G., Ilyina, T., Jain, A. K., Johannessen, T., Jones, C. D., Kato, E., Keeling, R. F., Goldewijk, K. K., Landschützer, P., Lefèvre, N., Lienert, S., Liu, Z., Lombardozzi, D., Metzl, N., Munro, D. R., Nabel, J. E. M. S., Nakaoka, S., Neill, C., Olsen, A., Ono, T., Patra, P., Peregon, A., Peters, W., Peylin, P., Pfeil, B., Pierrot, D., Poulter, B., Rehder, G., Resplandy, L., Robertson, E., Rocher, M., Rödenbeck, C., Schuster, U., Schwinger, J., Séférian, R., Skjelvan, I., Steinhoff, T., Sutton, A., Tans, P. P., Tian, H., Tilbrook, B., Tubiello, F. N., van der Laan-Luijkx, I. T., van der Werf, G. R., Viovy, N., Walker, A. P., Wiltshire, A. J., Wright, R., Zaehle, S., and Zheng, B.: Global Carbon Budget 2018, Earth Syst. Sci. Data, 10, 2141–2194, https://doi.org/10.5194/essd-10-2141-2018, 2018a.

Le Quéré, C., Andrew, R. M., Friedlingstein, P., Sitch, S., Pongratz, J., Manning, A. C., Korsbakken, J. I., Peters, G. P., Canadell, J. G., Jackson, R. B., Boden, T. A., Tans, P. P., Andrews, O. D., Arora, V. K., Bakker, D. C. E., Barbero, L., Becker, M., Betts, R. A., Bopp, L., Chevallier, F., Chini, L. P., Ciais, P., Cosca, C. E., Cross, J., Currie, K., Gasser, T., Harris, I., Hauck, J., Haverd, V., Houghton, R. A., Hunt, C. W., Hurtt, G., Ilyina, T., Jain, A. K., Kato, E., Kautz, M., Keeling, R. F., Klein Goldewijk, K., Körtzinger, A., Landschützer, P., Lefèvre, N., Lenton, A., Lienert, S., Lima, I., Lombardozzi, D., Metzl, N., Millero, F., Monteiro, P. M. S., Munro, D. R., Nabel, J. E. M. S., Nakaoka, S., Nojiri, Y., Padin, X. A., Peregon, A., Pfeil, B., Pierrot, D., Poulter, B., Rehder, G., Reimer, J., Rödenbeck, C., Schwinger, J., Séférian, R., Skjelvan, I., Stocker, B. D., Tian, H., Tilbrook, B., Tubiello, F. N., van der Laan-Luijkx, I. T., van der Werf, G. R., van Heuven, S., Viovy, N., Vuichard, N., Walker, A. P., Watson, A. J., Wiltshire, A. J., Zaehle, S., and Zhu, D.:

Global Carbon Budget 2017, Earth Syst. Sci. Data, 10, 405–448, https://doi.org/10.5194/essd-10-405-2018, 2018b.

Melaas, E. K., Richardson, A. D., Friedl, M. A., Dragoni, D., Gough, C. M., Herbst, M., Montagnani, L., and Moors, E.: Using FLUXNET data to improve models of springtime vegetation activity onset in forest ecosystems, Agr. For. Meteorol., 171–172, 46–56, https://doi.org/10.1016/j.agrformet.2012.11.018, 2013.

Melton, J. R. and Arora, V. K.: Sub-grid scale representation of vegetation in global land surface schemes: implications for estimation of the terrestrial carbon sink, Biogeosciences, 11, 1021–1036, https://doi.org/10.5194/bg-11-1021-2014, 2014.

Melton, J. R. and Arora, V. K.: Competition between plant functional types in the Canadian Terrestrial Ecosystem Model (CTEM) v.2.0, Geosci. Model Dev., 9, 323–361, https://doi.org/10.5194/gmd-9-323-2016, 2016.

Melton, J. R., Shrestha, R. K., and Arora, V. K.: The influence of soils on heterotrophic respiration exerts a strong control on net ecosystem productivity in seasonally dry Amazonian forests, Biogeosciences, 12, 1151–1168, https://doi.org/10.5194/bg-12-1151-2015, 2015.

Melton, J. R., Sospedra-Alfonso, R., and McCusker, K. E.: Tiling soil textures for terrestrial ecosystem modelling via clustering analysis: a case study with CLASS-CTEM (version 2.1), Geosci. Model Dev., 10, 2761–2783, https://doi.org/10.5194/gmd-10-2761-2017, 2017.

Melton, J. R., Arora, V., Wisernig-Cojoc, E., Seiler, C., Fortier, M., Chan, E., and Teckentrup, L.: The Canadian Land Surface Scheme including Biogeochemical Cycles, Zenodo, https://doi.org/10.5281/zenodo.3522407, 2019a.

Melton, J. R., Seiler, C., and Fortier, M.: Singularity software container for the Canadian Land Surface Scheme including Biogeochemical Cycles (CLASSIC), Zenodo, https://doi.org/10.5281/zenodo.3525249, 2019b.

Melton, J. R., Teckentrup, L., and Fortier, M.: Benchmarking data and outputs for CLASSIC v. 1.0, Zenodo, https://doi.org/10.5281/zenodo.3525336, 2019c.

Melton, J. R., Verseghy, D. L., Sospedra-Alfonso, R., and Gruber, S.: Improving permafrost physics in the coupled Canadian Land Surface Scheme (v.3.6.2) and Canadian Terrestrial Ecosystem Model (v.2.1) (CLASS-CTEM), Geosci. Model Dev., 12, 4443–4467, https://doi.org/10.5194/gmd-12-4443-2019, 2019d.

Merbold, L., Ardö, J., Arneth, A., Scholes, R. J., Nouvellon, Y., de Grandcourt, A., Archibald, S., Bonnefond, J. M., Boulain, N., Brueggemann, N., Bruemmer, C., Cappelaere, B., Ceschia, E., El-Khidir, H. A. M., El-Tahir, B. A., Falk, U., Lloyd, J., Kergoat, L., Le Dantec, V., Mougin, E., Muchinda, M., Mukelabai, M. M., Ramier, D., Roupsard, O., Timouk, F., Veenendaal, E. M., and Kutsch, W. L.: Precipitation as driver of carbon fluxes in 11 African ecosystems, Biogeosciences, 6, 1027–1041, https://doi.org/10.5194/bg-6-1027-2009, 2009.

Niu, G.-Y., Yang, Z.-L., Mitchell, K. E., Chen, F., Ek, M. B., Barlage, M., Kumar, A., Manning, K., Niyogi, D., Rosero, E., Tewari, M., and Xia, Y.: The community Noah land surface model with multiparameterization options (Noah-MP): 1. Model description and evaluation with local-scale measurements, J. Geophys., 116, D12, https://doi.org/10.1029/2010jd015139, 2011.

Paquin, J.-P. and Sushama, L.: On the Arctic near-surface permafrost and climate sensitivities to soil and snow model for-

mulations in climate models, Clim. Dynam., 44, 203–228, https://doi.org/10.1007/s00382-014-2185-6, 2014.

Pastorello, G., Papale, D., Chu, H., Trotta, C., Agarwal, D., Canfora, E., Baldocchi, D., and Torn, M.: A new data set to keep a sharper eye on land-air exchanges, Eos Trans. Amer. Geophys. Union, 98, 28–32, 2017.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, J. Mach. Learn. Res., 12, 2825–2830, 2011.

Peng, Y., Arora, V. K., Kurz, W. A., Hember, R. A., Hawkins, B. J., Fyfe, J. C., and Werner, A. T.: Climate and atmospheric drivers of historical terrestrial carbon uptake in the province of British Columbia, Canada, Biogeosciences, 11, 635–649, https://doi.org/10.5194/bg-11-635-2014, 2014.

Pietroniro, A., Fortin, V., Kouwen, N., Neal, C., Turcotte, R., Davison, B., Verseghy, D., Soulis, E. D., Caldwell, R., Evora, N., and Pellerin, P.: Development of the MESH modelling system for hydrological ensemble forecasting of the Laurentian Great Lakes at the regional scale, Hydrol. Earth Syst. Sci., 11, 1279–1294, https://doi.org/10.5194/hess-11-1279-2007, 2007.

Poulter, B., Bousquet, P., Canadell, J. G., Ciais, P., Peregon, A., Saunois, M., Arora, V. K., Beerling, D. J., Brovkin, V., Jones, C. D., Joos, F., Gedney, N., Ito, A., Kleinen, T., Koven, C. D., McDonald, K., Melton, J. R., Peng, C., Peng, S., Prigent, C., Schroeder, R., Riley, W. J., Saito, M., Spahni, R., Tian, H., Taylor, L., Viovy, N., Wilton, D., Wiltshire, A., Xu, X., Zhang, B., Zhang, Z., and Zhu, Q.: Global wetland contribution to 2000–2012 atmospheric methane growth rate dynamics, Environ. Res. Lett., 12, 094013, https://doi.org/10.1088/1748-9326/aa8391, 2017.

Roy, A., Royer, A., Montpetit, B., Bartlett, P. A., and Langlois, A.: Snow specific surface area simulation using the one-layer snow model in the Canadian LAnd Surface Scheme (CLASS), The Cryosphere, 7, 961–975, https://doi.org/10.5194/tc-7-961-2013, 2013.

Scinocca, J. F., Kharin, V. V., Jiao, Y., Qian, M. W., Lazare, M., Solheim, L., Flato, G. M., Biner, S., Desgagne, M., and Dugas, B.: Coordinated Global and Regional Climate Modeling*, J. Climate, 29, 17–35, https://doi.org/10.1175/JCLI-D-15-0161.1, 2016.

Scott, R. L., Hamerlynck, E. P., Darrel Jenerette, G., Susan Moran, M., and Barron-Gafford, G. A.: Carbon dioxide exchange in a semidesert grassland through drought-induced vegetation change, J. Geophys. Res., 115, G3, https://doi.org/10.1029/2010jg001348, 2010.

Sedlák, P., Aubinet, M., Heinesch, B., Janouš, D., Pavelka, M., Potužníková, K., and Yernaux, M.: Night-time airflow in a forest canopy near a mountain crest, Agr. For. Meteorol., 150, 736–744, https://doi.org/10.1016/j.agrformet.2010.01.014, 2010.

Seiler, C.: amber: Automated Model Benchmarking Package for the Canadian Land Surface Scheme, available at: https://CRAN.R-project.org/package=amber (last access: 19 November 2019), r package version 0.1.5, 2019.

Seiler, C., Melton, J. R., Arora, V. K., and Wang, L.: CLASSIC v1.0: the open-source community successor to the Canadian Land Surface Scheme (CLASS) and the Canadian Terrestrial Ecosystem Model (CTEM) – Part 2: Global benchmarking, in preparation, 2020.

Shrestha, R. K., Arora, V. K., and Melton, J. R.: The sensitivity of simulated competition between different plant functional types to sub-grid-scale representation of vegetation in a land surface model, J. Geophys. Res.-Biogeosci., 121, 2015JG003234, https://doi.org/10.1002/2015JG003234, 2016.

Stöckli, R., Lawrence, D. M., Niu, G.-Y., Oleson, K. W., Thornton, P. E., Yang, Z.-L., Bonan, G. B., Denning, A. S., and Running, S. W.: Use of FLUXNET in the Community Land Model development, J. Geophys. Res., 113, G01025, https://doi.org/10.1029/2007JG000562, 2008.

Swart, N. C., Cole, J. N. S., Kharin, V. V., Lazare, M., Scinocca, J. F., Gillett, N. P., Anstey, J., Arora, V., Christian, J. R., Hanna, S., Jiao, Y., Lee, W. G., Majaess, F., Saenko, O. A., Seiler, C., Seinen, C., Shao, A., Sigmond, M., Solheim, L., von Salzen, K., Yang, D., and Winter, B.: The Canadian Earth System Model version 5 (CanESM5.0.3), Geosci. Model Dev., 12, 4823–4873, https://doi.org/10.5194/gmd-12-4823-2019, 2019.

Verseghy, D.: CLASS – The Canadian land surface scheme (v.3.6.2), Climate Research Division, Science and Technology Branch, Environment Canada, 2017.

Verseghy, D. L.: CLASS – A Canadian land surface scheme for GCMs. I. Soil model, Int. J. Climatol., 11, 111–133, https://doi.org/10.1002/joc.3370110202, 1991.

Verseghy, D. L.: The Canadian land surface scheme (CLASS): Its history and future, Atmos.-Ocean, 38, 1–13, https://doi.org/10.1080/07055900.2000.9649637, 2000.

Verseghy, D. L. and MacKay, M. D.: Offline implementation and evaluation of the Canadian Small Lake Model with the Canadian Land Surface Scheme over western Canada, J. Hydrometeorol., 18, 1563–1582, https://doi.org/10.1175/JHM-D-16-0272.1, 2017.

Verseghy, D. L., McFarlane, N. A., and Lazare, M.: CLASS – A Canadian land surface scheme for GCMs, II. Vegetation model and coupled runs, Int. J. Climatol., 13, 347–370, https://doi.org/10.1002/joc.3370130402, 1993.

von Salzen, K., Scinocca, J. F., McFarlane, N. A., Li, J., Cole, J. N. S., Plummer, D., Verseghy, D., Reader, M. C., Ma, X., Lazare, M., and Solheim, L.: The Canadian Fourth Generation Atmospheric Global Climate Model (CanAM4). Part I: Representation of Physical Processes, Atmos.-Ocean, 51, 104–125, https://doi.org/10.1080/07055900.2012.755610, 2013.

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., and Wilson, P.: Best practices for scientific computing, PLoS Biol., 12, e1001745, https://doi.org/10.1371/journal.pbio.1001745, 2014.

Wilson, K., Goldstein, A., Falge, E., Aubinet, M., Baldocchi, D., Berbigier, P., Bernhofer, C., Ceulemans, R., Dolman, H., Field, C., Grelle, A., Ibrom, A., Law, B. E., Kowalski, A., Meyers, T., Moncrieff, J., Monson, R., Oechel, W., Tenhunen, J., Valentini, R., and Verma, S.: Energy balance closure at FLUXNET sites, Agr. For. Meteorol., 113, 223–243, https://doi.org/10.1016/S0168-1923(02)00109-0, 2002.

Wu, Y., Verseghy, D. L., and Melton, J. R.: Integrating peatlands into the coupled Canadian Land Surface Scheme (CLASS) v3.6 and the Canadian Terrestrial Ecosystem Model (CTEM) v2.0,

Geosci. Model Dev., 9, 2639–2663, https://doi.org/10.5194/gmd-9-2639-2016, 2016.