Supplement of Geosci. Model Dev., 12, 1119–1137, 2019
https://doi.org/10.5194/gmd-12-1119-2019-supplement

Geoscientific
Model Development
Open Access

Supplement of

Realized ecological forecast through an interactive Ecological Platform for Assimilating Data (EcoPAD, v1.0) into models

Yuanyuan Huang et al.

Correspondence to: Yuanyuan Huang (yuanyuanhuang2011@gmail.com) and Yiqi Luo (yiqi.luo@nau.edu)

Supplementary information

Supplement 1

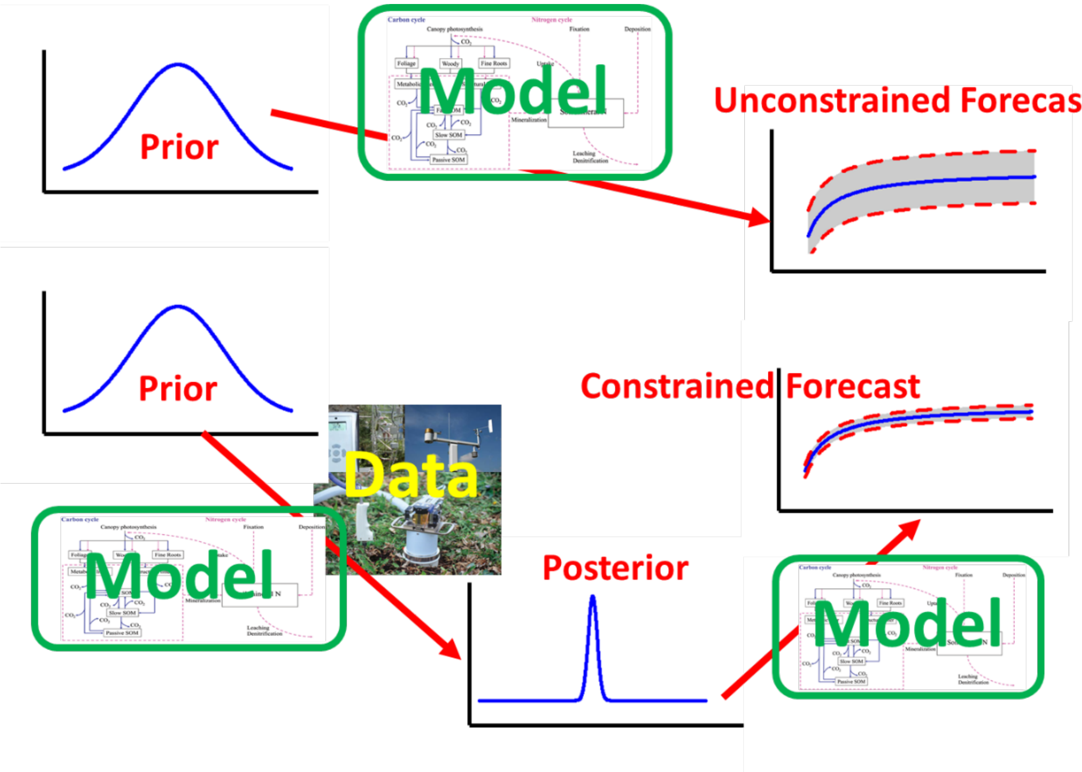


Figure S1. Conceptual demonstration of how data assimilation that updates models through observations constrains forecasting. The grey shading area corresponds to forecasting uncertainties.

Supplement 2. Scientific functionality

Scientific functionality of EcoPAD includes web-based model simulation, estimating model parameters or state variables, quantifying uncertainty of estimated parameters and projected states of ecosystems, evaluating model structures, assessing sampling strategies and conducting ecological forecasting. These functions can be organized to answer various scientific questions. In addition to the general description in this section, the scientific functionality of EcoPAD is also illustrated through a few case studies in the following sections.

EcoPAD is designed to perform web-based model simulation, which greatly reduces the workload of traditional model simulation through manual code compilation and execution. This functionality opens various new opportunities for modellers, experimenters and the general public. Model simulation and result analysis are automatically triggered after a click on the web-embedded button (Figures S2, S3 S6). Users are freed from repeatedly compiling code, running code and writing programs to analyse and display model results. Such ease of use has great potential to popularize complex modelling studies that are difficult or inaccessible for experimenters and the general public. As illustrated through the outreach activities from the TreeWatch.Net [*Steppe et al.*, 2016], the potential functionality of such web-based model simulation goes beyond its scientific value as its societal and educational impacts are critical in solving ecological issues. The web-based model simulation also frees users from model running environment, platform and software. Users can conduct model simulation and do analysis as long as they have internet access. For example, ecologists can conduct model simulation and diagnose the underlying reasons for a sudden increase in methane fluxes while they are making measurements in the field. Non-ecologists, such as youngsters, can study ecological dynamics through their phones or tablets while they are waiting for the bus. Resource managers can make timely assessment of different resource utilization strategies on spot of a meeting.

EcoPAD is backed up by data assimilation techniques, which facilitate inference of model parameters and states based on observations. Ecology have witnessed a growing number of studies focusing on parameter estimation using inverse modelling or data assimilation as large volumes of ecological measurements become available. To satisfy the growing need of model parameterization through observations, EcoPAD streamlines parameter estimations and updates. Researchers can review and download files that record parameter values from EcoPAD result repository. Since these parameters may have different biological, physical or chemical meanings, the functionality of EcoPAD related to parameter estimations can potentially embrace diverse subareas in ecology. For example, soil scientists can study the acclimation of soil respiration to manipulative warming through shifts in the distribution of the decomposition rate parameter from EcoPAD. The threshold parameter beyond which further harvesting of fish might cause a crash of fish stocks can be extracted through fish stock assessment models and observations if mounted to EcoPAD.

EcoPAD promotes uncertainty analysis, model structure evaluation and error identification. One of the advantages of the Bayesian statistics is its capacity in uncertainty analysis compared to other optimization techniques [*Xu et al.*, 2006; *Wang et al.*, 2009; *Zhou et al.*, 2012]. Bayesian data assimilation (e.g., MCMC) takes into account observation uncertainties (errors), generates distributions of model parameters and enables tracking of prediction uncertainties from different sources[*Ellison*, 2004; *Bloom et al.*, 2016; *Jiang et al.*, 2018]. Uncertainty analysis through data assimilation applied to areas such as ecosystem phenology, fish life cycle and species migration [*Clark et al.*, 2003; *Cook et al.*, 2005; *Crozier et al.*, 2008; *Luo et al.*, 2011b], can potentially take advantage of EcoPAD platform to provide critical information for well informed decisions in face of pressing global change challenges. In addition, the archive capacity of EcoPAD facilitates future inter-comparisons among different models or different versions of the same model to evaluate model structures and to disentangle structure uncertainties and errors.

The realization of both the near-time and long-term ecological forecast is one of the key innovations of EcoPAD. Forecasting capability of EcoPAD is supported by process based ecological models, multiple observational or experimental data, inverse parameter estimation and uncertainty quantification through data assimilation, and forward simulation under future external conditions. The systematically constrained forecast from EcoPAD is accompanied by uncertainty/confidence estimates to quantify the amount of information that can actually be utilized from a study. The automated near time forecast, which is constantly adjusted once new observational data streams are available, provides experimenters advanced and timely information to assess and adjust experimental plans. For example, with forecasted and displayed biophysical and biochemical variables, experimenters could know in advance what the most likely biophysical conditions are. Knowing if the water table may suddenly go aboveground in response to a high rainfall forecast in the coming week, could allow researcher to emphasize measurements associated with methane flux. In such a way, experimenters can not only rely on historical ecosystem dynamics, but also refer to future predictions. Experimenters will benefit especially from variables that are difficult to track in field due to situations such as harsh environment, shortage in man power or on instrument limitation.

Equally important, EcoPAD creates new avenues to answer classic and novel ecological questions, for example, the frequently reported acclimation phenomena in ecology. While growing evidence points to altered ecological functions as organisms adjust to the rapidly changing world [*Medlyn et al.*, 1999; *Luo et al.*, 2001; *Wallenstein and Hall*, 2012], traditional ecological models treat ecological processes less dynamical, as the governing biological parameters or mechanisms fails to explain such biological shifts. EcoPAD facilitates the shift of research paradigm from a fixed process representation to a more dynamic description of ecological mechanisms with constantly updated and archived parameters constrained by observations under different conditions. Specifically to acclimation, EcoPAD promotes quantitatively evaluations while previous studies remain mostly qualitative [*Wallenstein and Hall*, 2012; *Shi et al.*, 2015]. We will further illustrate how EcoPAD can be used to address different ecological questions in the case studies of the SPRUCE project.

Supplement 3. EcoPAD-SPRUCE web portal

We assimilate multiple streams of data from the SPRUCE experiment to the TECO model using the MCMC algorithm, and forecast ecosystem dynamics in both near time and for the next 10 years. Our forecasting system for SPRUCE is available at https://ecolab.nau.edu/ecopad_portal/ (the new portal) or http://ecolab.cybercommons.org/ecopad_portal_up/ (the older portal). From the web portal, users can check our current near and long term forecasting results, conduct model simulation, data assimilation and forecasting runs, and analyze/visualize model results (Username: test00 and password:test01 for the new portal; Username: chris and password:chris for the old portal if login information is required). The login account we created for the new portal is limited to Simulation only and registration is required for more functionalities.

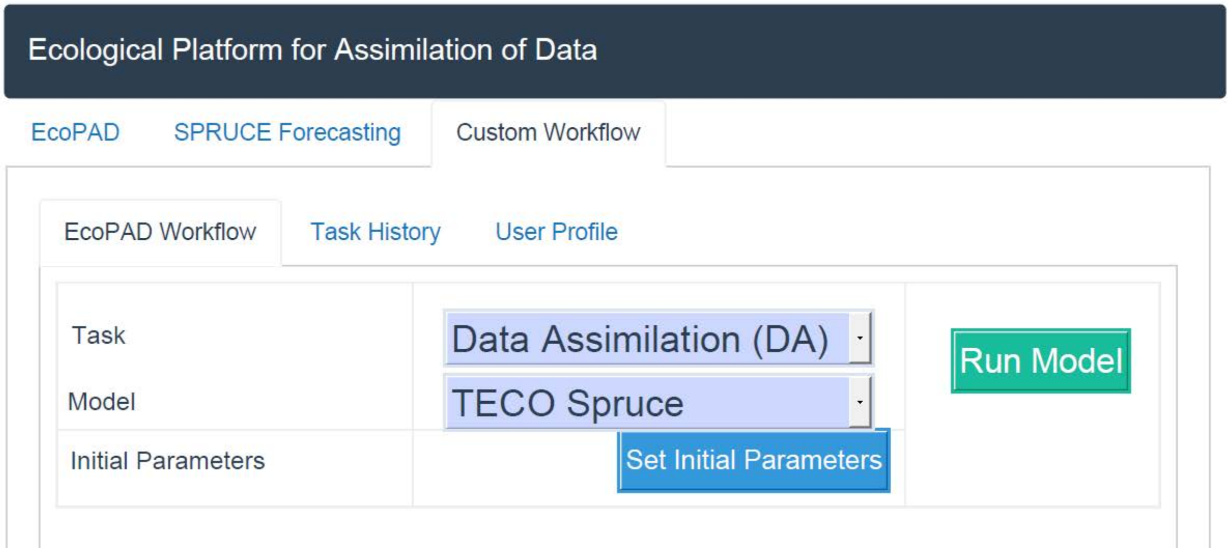
The main page of the EcoPAD-SPRUCE portal includes animation demos and a brief description of the system. The animation demos display the dynamic change of gross primary productivity (GPP), ecosystem respiration (ER), foliage carbon (foliage C), wood carbon (wood C), root carbon (root C) and soil carbon (soil C) under 10 manipulative warming and elevated atmospheric CO_2 treatments. Each animation shows observations in data assimilation period during which parameters are constrained (2011-2014) as well as model results (with uncertainty) from data assimilation and 10 years forecasting from an ensemble of model runs. Warming generally increase GPP, ER and different carbon pools. Users can also get a sense on how uncertainties in forcing variables, such as light, temperature, and precipitation that drive carbon fluxes in terrestrial ecosystem, and limited observations affect uncertainty of GPP prediction.

Under the Custom Workflow menu, users can choose different modes to run TECO model from the task dropdown box: Simulation, Data Assimilation (DA) and Forecasting (Figure S2).

In the Simulation mode, users are allowed change the initial parameters through “Set Initial Parameters” button. TECO-SPRUCE currently allows 33 key parameters to be adjustable by end-users. These 33 parameters include parameters that control soil water dynamics, plant growth, photosynthesis, carbon allocation among different plant organs, turnover rates of different pools, temperature sensitivity, and plant phenology. Researchers can choose other parameters according to their models and specific needs. The simulation runs TECO one time with user supplied initial parameters and the run normally takes several minutes in the background. Each requested task from the user is assigned a unique task ID. Users can check information such as task id, timestamp, parameters, result status, result URL from a web-enabled report once the task is submitted under the “Task History” tab. If the task status shows “SUCESS” (Figure S3), users can check datasets relevant to model simulation from the result URL (for example, http://ecolab.oscer.ou.edu/ecopad_tasks/8b4bcd9b-172c-4031-94b7-4b080e459025, where “8b4bcd9b-172c-4031-94b7-4b080e459025” is the unique task ID for this example). The URL directs users to the location (result repository) where information related to model simulation is stored. Result repository stores parameters supplied to the model run in .txt format. Yearly and daily simulation results for carbon fluxes and pools are also written in .txt file format. It also contains .png file format plots of simulated carbon fluxes and pools compared to observations (Figure S4). Users can check the results from the Task History any time with the right task ID. With several “Simulation” runs, users can easily get a sense on the sensitivity of the SPRUCE peatland carbon cycle to different parameters and what are the key processes regulate the northern peatland carbon dynamics.

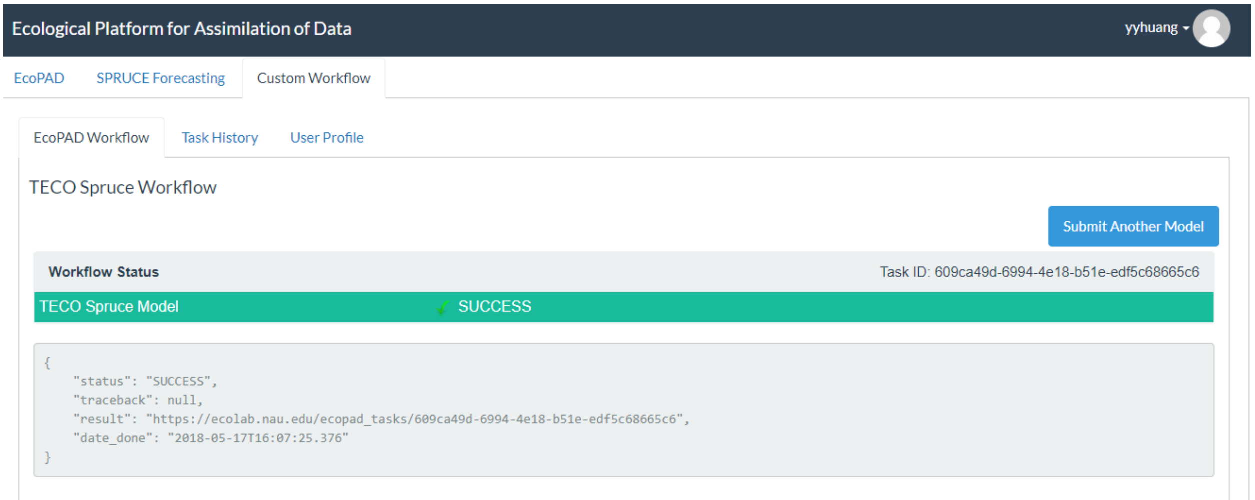
Data Assimilation mode enables users to conduct data-model fusion research through a web portal. A unique feature of the data assimilation portal is that users can pick whatever parameters to be constrained among the pool of 18 parameters which are important in ecosystem carbon cycling (Figure S5). Users can change the range of a parameter they are interested in and modify the initial values of parameters supplied to MCMC. Similarly as in Simulation mode, user can easily check data assimilation results through the result URL. Results from data assimilation contain parameter ranges and initial values supplied by users, parameter values accepted in MCMC, histograms of posterior distribution of parameters (Figure S5), and simulations of carbon fluxes and pools with 500 randomly chosen accepted parameters. Data assimilation results are also written into the universal .txt format which makes further utilization of the result convenient. For example, researchers interested in the pattern and uncertainty in GPP simulation can quickly get a handle on GPP with an ensemble of easily readable model results.

From the Forecasting mode, users are enabled to set up parameters, or choose posterior parameters from previous data assimilation results, specify forecast starting and ending dates, and select warming (0-9 degree Celsius) and CO_2 (380-900 ppm) treatments (Figure S6). If a specific data assimilation result was chosen as input for forecasting simulation, TECO-SPRUCE would read the constrained posterior parameter file, match the name of constrained parameters to the whole parameter pool, and then randomly choose 100 sets of constrained parameters to run forecast. Results from forecast store carbon fluxes and pools from simulations based on the 100 randomly chosen parameters and projected 10 years into the future at the daily time scale. Users can analyze forecasting dynamics and uncertainties based on stored results. EcoPAD-SPRUCE result repositories also provide figures that combine observation in data assimilation period, simulation results in data assimilation as well as forecasting periods, and simulation uncertainty (Figure S7) to speed up the post-processing of model results. S



Task: Data Assimilation (DA)
Model: TECO Spruce

Figure S2. The Custom Workflow web portal of the EcoPAD applied for the SPRUCE project. Users can select among “Simulation”, “Data Assimilation (DA)” and “Forecasting” modes from the task drop-down box to run ecological models in the background. In each mode, users are allowed to customize the model run, such as set the initial parameter values for “Simulation” and “Data Assimilation (DA)”, choose the updated parameters from “Data Assimilation (DA)” to conduct “Forecasting” or change the “Forecasting” periods.



```
{
    "status": "SUCCESS",
    "traceback": null,
    "result": "https://ecolab.nau.edu/ecopad_tasks/609ca49d-6994-4e18-b51e-edf5c68665c6",
    "date_done": "2018-05-17T16:07:25.376"
}
```

Figure S3. An example of a successful model simulations. In EcoPAD, each task is assigned a unique task ID. The input, output, report and plot relevant to a model task are archived and easy to tack through the unique web link based on the task ID.

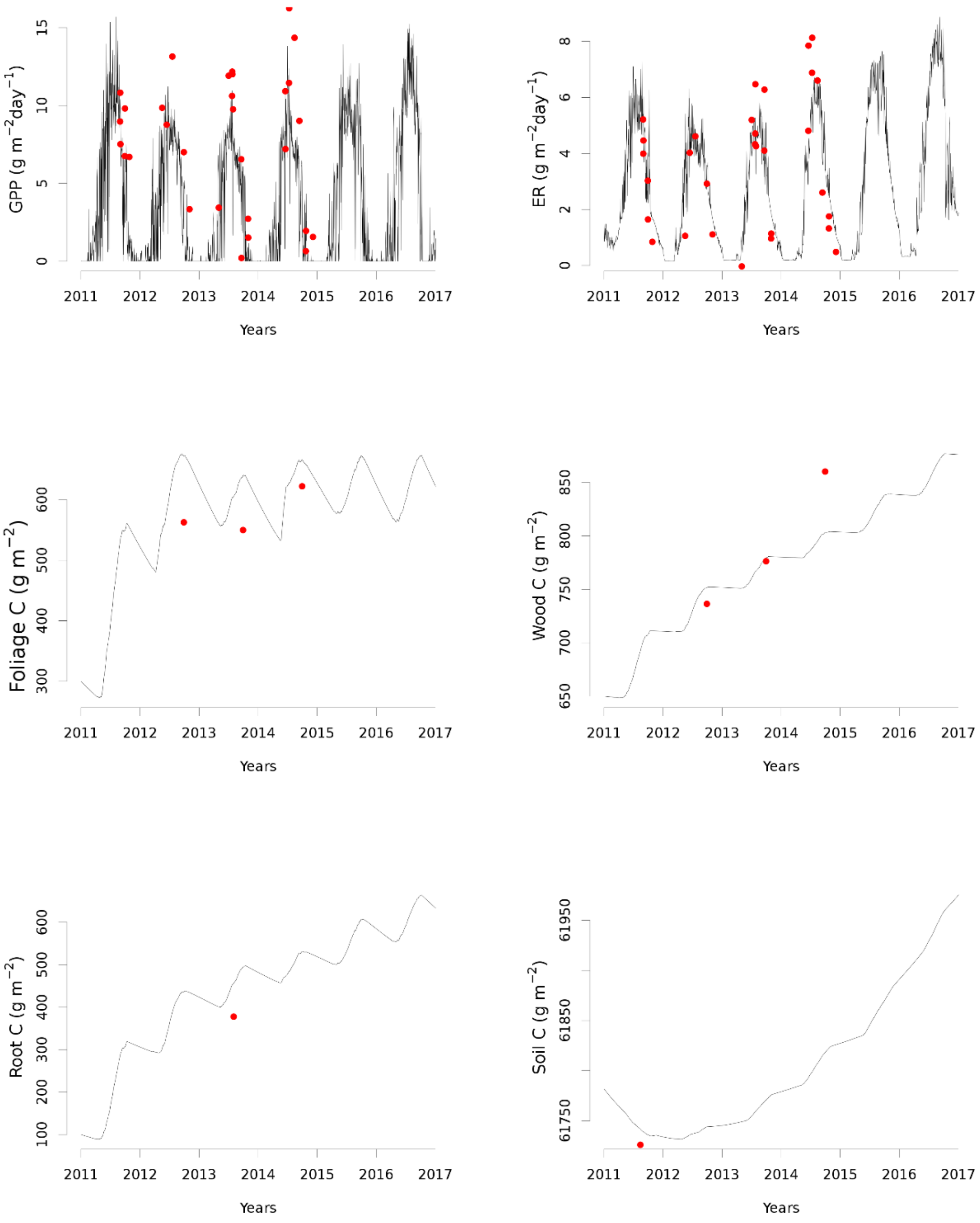
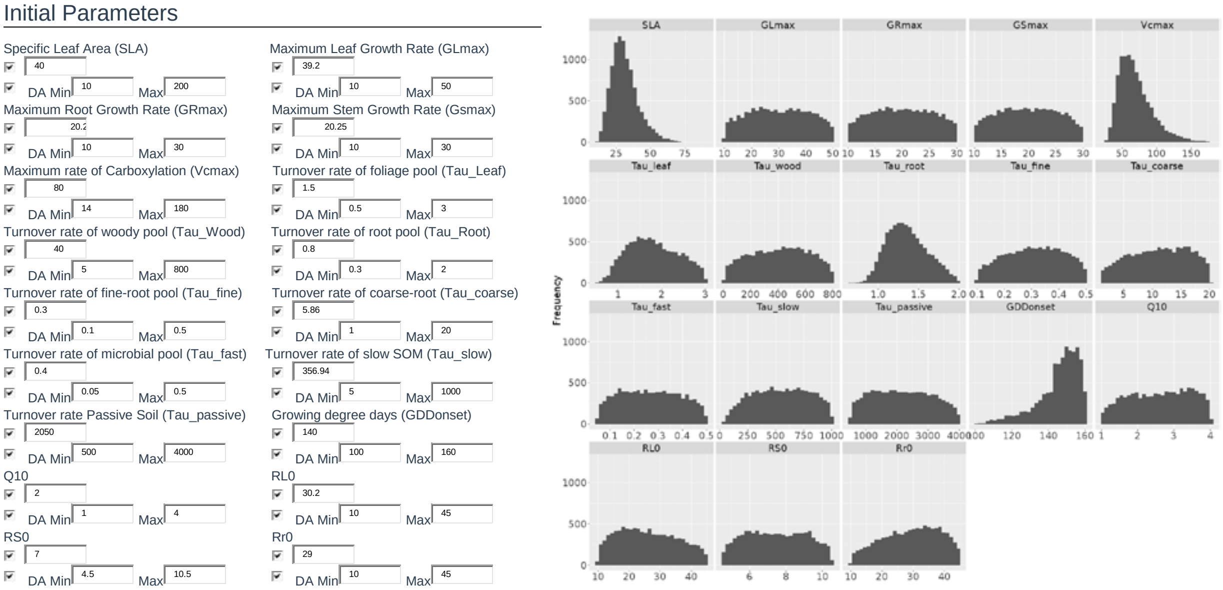

GPP (g m−2day−1)
0
5
10
15
ER (g m−2day−1)
0
2
4
6
8
Foliage C (g m−2)
300
400
500
600
Wood C (g m−2)
650
700
750
800
850
Root C (g m−2)
100
200
300
400
500
600
Soil C (g m−2)
61750
61850
61950
2011
2012
2013
2014
2015
2016
2017
Years

Figure S4. An example of the carbon flux and pool size produced from the “Simulation” mode in EcoPAD-SPRUCE. Red dots indicate available observations and gray lines correspond to model simulation results. The upper two panels display carbon fluxes: gross primary productivity (GPP, left panel) and ecosystem respiration (ER, right panel). The lower four panels show result for foliage carbon (foliage C), wood carbon (wood C), root carbon (root C) and soil carbon (soil C).



Initial Parameters

Specific Leaf Area (SLA)
☑ 40
☑ DA Min 10 Max 200

Maximum Root Growth Rate (GRmax)
☑ 20.2
☑ DA Min 10 Max 30

Maximum rate of Carboxylation (Vcmax)
☑ 80
☑ DA Min 14 Max 180

Turnover rate of woody pool (Tau_Wood)
☑ 40
☑ DA Min 5 Max 800

Turnover rate of fine-root pool (Tau_fine)
☑ 0.3
☑ DA Min 0.1 Max 0.5

Turnover rate of microbial pool (Tau_fast)
☑ 0.4
☑ DA Min 0.05 Max 0.5

Turnover rate Passive Soil (Tau_passive)
☑ 2050
☑ DA Min 500 Max 4000

Q10
☑ 2
☑ DA Min 1 Max 4

RS0
☑ 7
☑ DA Min 4.5 Max 10.5

Maximum Leaf Growth Rate (GLmax)
☑ 39.2
☑ DA Min 10 Max 50

Maximum Stem Growth Rate (Gsmax)
☑ 20.25
☑ DA Min 10 Max 30

Turnover rate of foliage pool (Tau_Leaf)
☑ 1.5
☑ DA Min 0.5 Max 3

Turnover rate of root pool (Tau_Root)
☑ 0.8
☑ DA Min 0.3 Max 2

Turnover rate of coarse-root (Tau_coarse)
☑ 5.86
☑ DA Min 1 Max 20

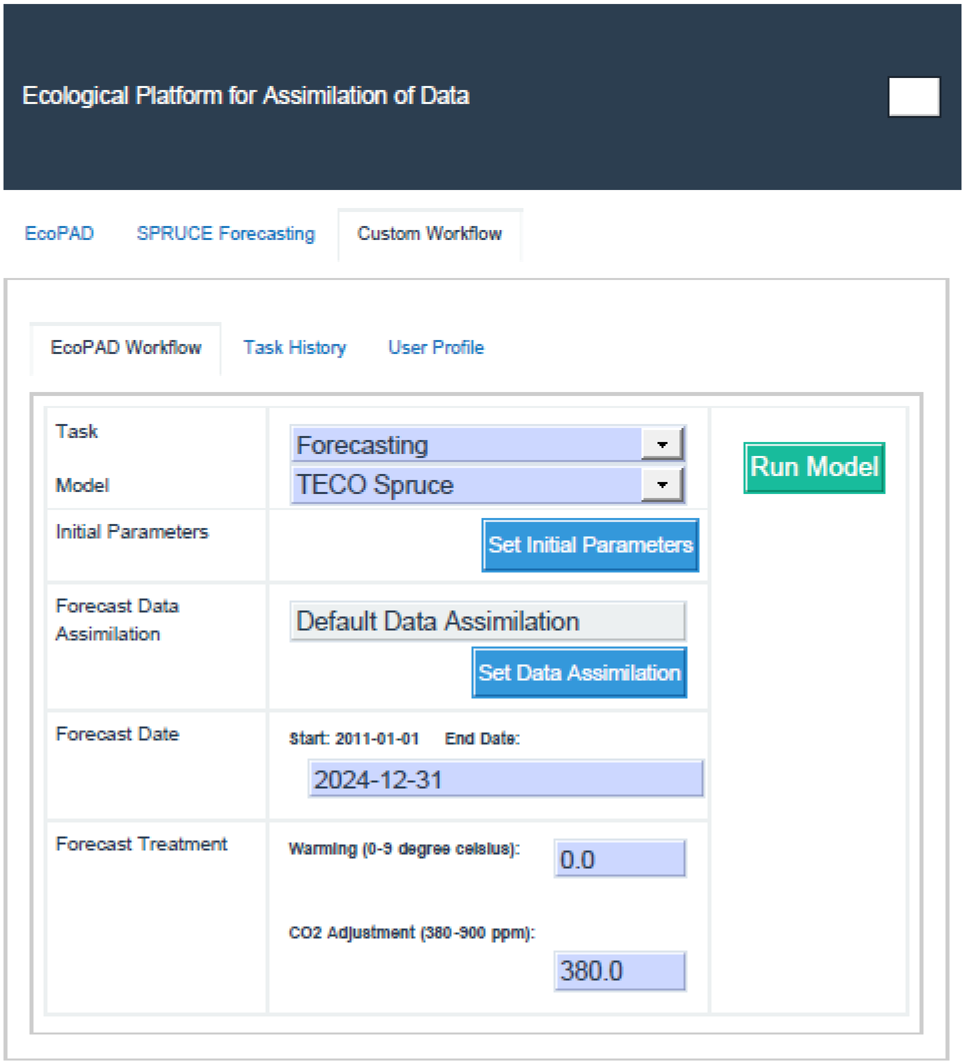
Turnover rate of slow SOM (Tau_slow)
☑ 356.94
☑ DA Min 5 Max 1000

Growing degree days (GDDonset)
☑ 140
☑ DA Min 100 Max 160

RL0
☑ 30.2
☑ DA Min 10 Max 45

Rr0
☑ 29
☑ DA Min 10 Max 45

Figure S5 Parameters that are allowed to modify in EcoPAD-SPRUCE. The left panel shows the user interface where users can change the initial parameter value and its range supplied to “Data Assimilation (DA)”. The right panel shows the histogram of the posterior distribution of each parameter that participated in the “Data Assimilation (DA)”. The right panel is automatically generated and archived for each “Data Assimilation (DA)” task.



Task: Forecasting

Model: TECO Spruce

Initial Parameters

Forecast Data Assimilation: Default Data Assimilation

Forecast Date: Start: 2011-01-01 End Date: 2024-12-31

Forecast Treatment:

Warming (0-9 degree celsius): 0.0

CO2 Adjustment (380-900 ppm): 380.0

Figure S6 An example user interface of the “Forecasting” mode in EcoPAD-SPRUCE.

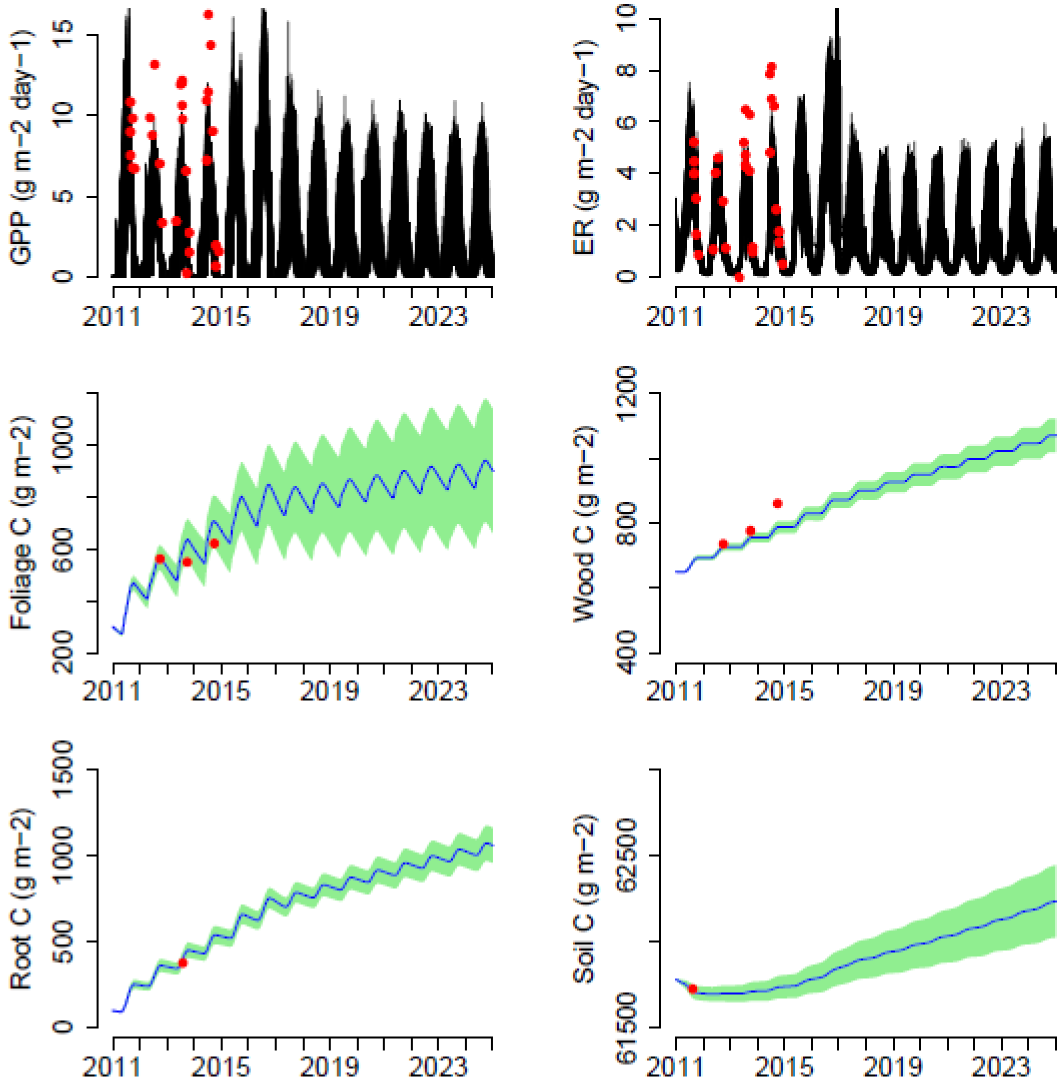


Figure S7. An example figure produced from the "Forecasting" mode in EcoPAD-SPRUCE. Red dots indicate observations used in the data assimilation period (2011-2014). Forecasting runs from 2015-2024. The upper two panels display dynamic changes of carbon fluxes: gross primary productivity (GPP, left panel) and ecosystem respiration (ER, right panel). The lower four panels show result for foliage carbon (foliage C), wood carbon (wood C), root carbon (root C) and soil carbon (soil C). Blue lines indicate the mean and green shading areas corresponding to simulation uncertainties for carbon pools generated from an ensemble of model simulations with randomly chosen parameters from their posterior distributions.

Supplement 4. Details on adding a new model or data assimilation approach

The framework of the system is established through using cookiecutter to install a microservice architecture that provides a RESTful API (Django REST Framework), Data Catalog(MongoDB), and Asyncronous workflow system (Celery) (https://github.com/cybercommons/cybercom-cookiecutter) (Figures 3 and S8). Cookiecutter creates projects from project templates for open source python libraries (https://cookiecutter.readthedocs.io/en/latest/readme.html). Figure S8 shows the file structure created through cookiecutter for EcoPAD.

An additional model can be added to the system through creating a docker image for the model and adding a new task to task.py (see code below). Task.py controls the functionalities or tasks the system is setup to realize. A task can be as simple as adding two numbers together or conduct complex process-based model simulation, data assimilation or forecasting. For a complex task like process-based model simulation, we make the core of the task relatively independent through wrapping the process-based model simulation into a docker container. Task.py takes charge of passing the path of input data or parameters required by the process-based model simulation, initializing model simulation and providing the path of simulating results for other tasks. The docker container wraps the file system and environment that are needed to conduct a model simulation. The model execution is confined within a docker container which is relatively independent of the workflow. The docker container is an instance of a docker image which can be triggered or executed from different systems. A new docker image can be built through the code of a new model and triggered by the workflow. Data assimilation and forecasting work similarly as model simulation. The differences lie in external forcing, initial conditions and model parameters passed to trigger the process-based model simulation. Addition of a new data assimilation algorithm corresponds to add a new docker image into the system. We currently wrap the data assimilation algorithm and the process based ecological model code inside one docker container. The interface (e.g., parameters generated from data assimilation algorithm that are passed to model simulation, and model simulation results that are passed to data assimilation algorithm to evaluate the objective function) between process-based model simulation and data assimilation is wrapped inside the docker container through function calls. One alternative way is to separate the data assimilation algorithm and the process-based modeling into two images, and setup the task.py to take charge of the interface between these two images. In the forecasting task, external forcing is fetched from forecasted forcing. The initial condition is the previous modeling results with real climate forcing and model parameters are passed from the data assimilation task where model parameters are adjusted to observed carbon state variables. The inclusion of multiple models or data assimilation techniques correspond to mounting multiple docker images into the system and an extension of the tasks.

Figure S8. File structures created through Cookiecutter.

```
├── api_code (Github Repository https://github.com/cybercommons/cybercom-api)
├── config
│   ├── addmongouser
│   ├── api_config.py
│   ├── celery
│   │   └── code
│   │       ├── celeryconfig.py
│   │       ├── celeryconfig.pyc
│   │       └── requirements.txt
│   ├── config.sh
│   ├── db.sqlite3 (Database of API users, permissions, and group settings)
│   ├── nginx (Configuration related files)
│   └── ssl
│       ├── backend
│       │   ├── ca.conf
│       │   ├── client (Generated client side certificates)
│       │   ├── generate
│       │   ├── keystoresecret
│       │   ├── server (Generated server side certificates)
│       │   └── testca (Generated CA files used to sign client and server side certifictes)
│       └── nginx
│           ├── generate
│           ├── keys (generated dhparam and self signed certificates)
│           ├── letsencrypt
│           │   ├── dockerfiles (Files used to generate Docker container)
│           │   ├── etc (Let's Encrypt working files and generated certificates)
│           │   ├── nginx-bootstrap.conf
│           │   └── www (Working path for Let's Encrypt's .well-known/acme-challenge)
│           └── runLetsEncrypt
├── data
│   ├── local
│   │   └── static
│   │       └── api (REST API CSS Content)
│   ├── mongo (MongoDB data folder)
│   └── static (Web Accessible Directory)
│       └── portal (Github Repository https://github.com/cybercommons/cybercom-portal)
├── log
│   ├── api.log
│   └── celery.log
└── run
    ├── appContainerKill
    ├── cybercom_up
    ├── genSSLKeys
    └── resetDBCreds
```