Land Model Testbed: Accelerating Development, Benchmarking and Analysis of Land Surface Models

Sarat Sreepathi, Min Xu, Nathan Collier, Jitendra Kumar, Jiafu Mao, and Forrest M. Hoffman

Oak Ridge National Laboratory

October 19, 2020

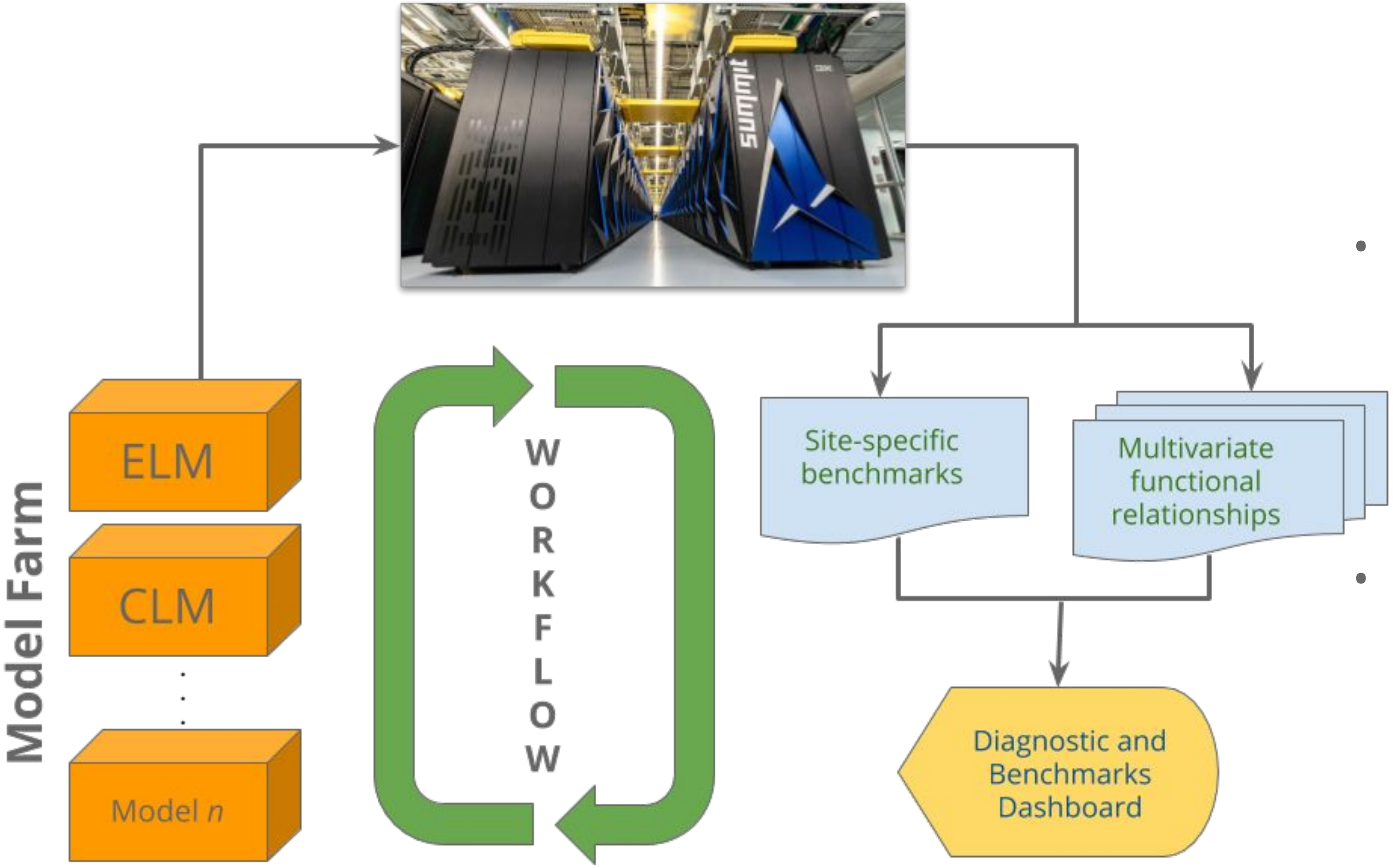
Gateways 2020 Conference

October 19–23, 2020

Land Model Testbed (LMT)

- Jupyter Notebooks for workflow prototype deployed on ORNL Cloud Computing Resources

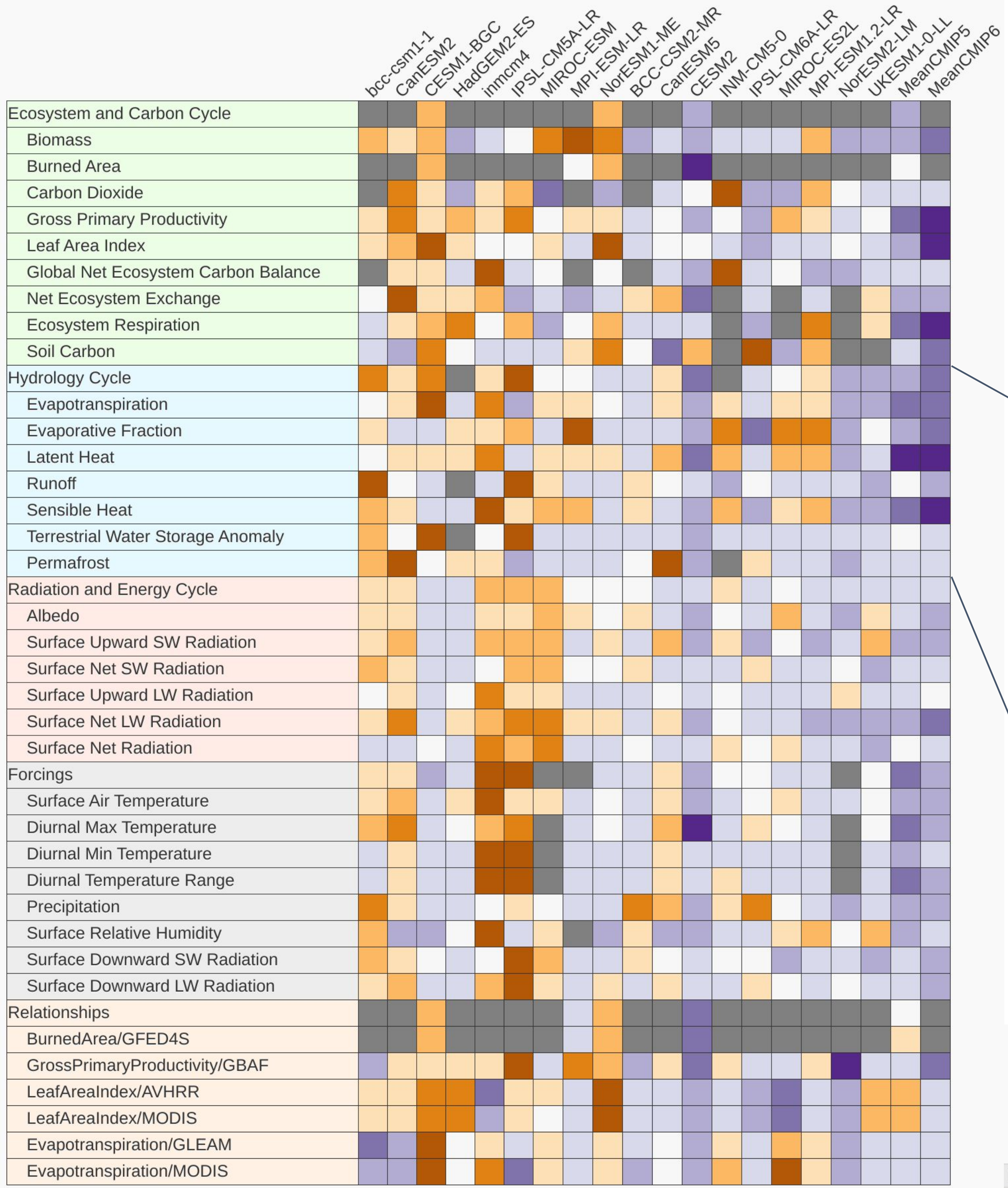
- Main workflow now tested and running on ORNL Cloud and Summit supercomputer (simulations)
- Model output post-processing tools developed

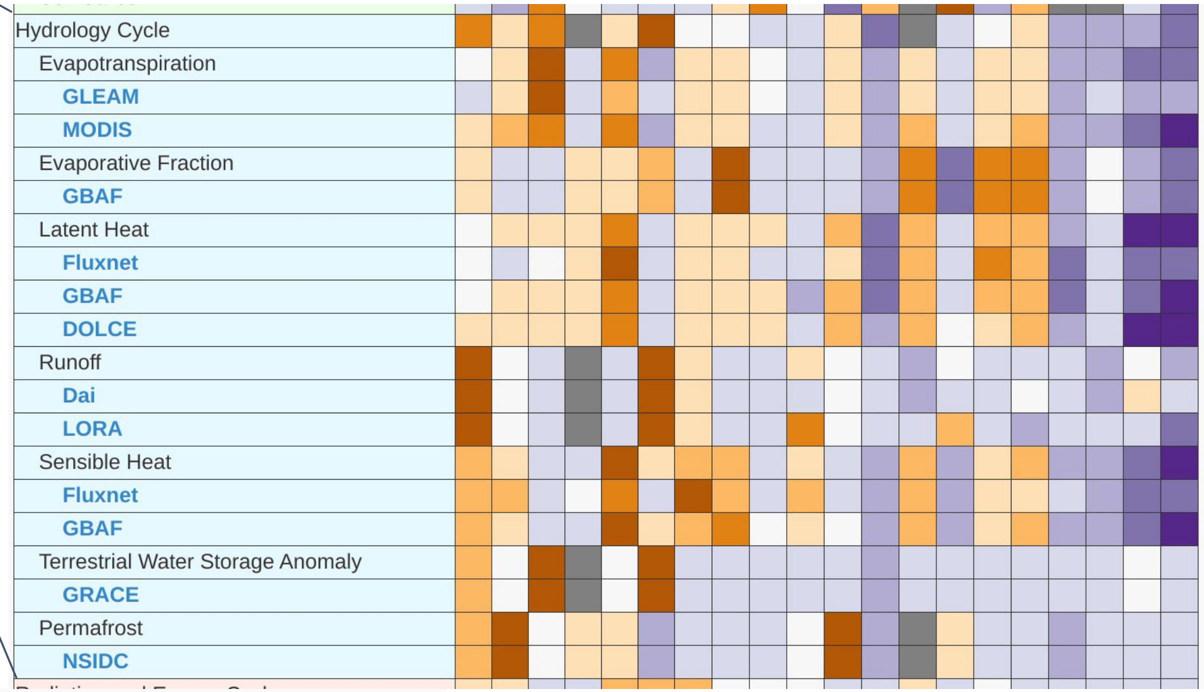


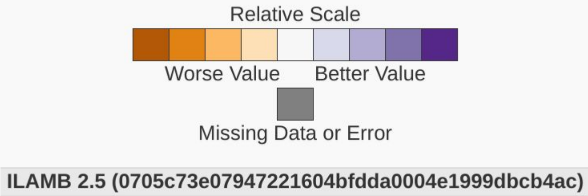
- Developing dashboard specific to climate phenomena-focused analysis
- Facilitating climate Coupled Model Intercomparison Project (CMIP) style analysis
- Evolutionary study (CMIP6 vs. CMIP5 models) to assess if complexity improved predictability

International Land Model Benchmarking (ILAMB) Package

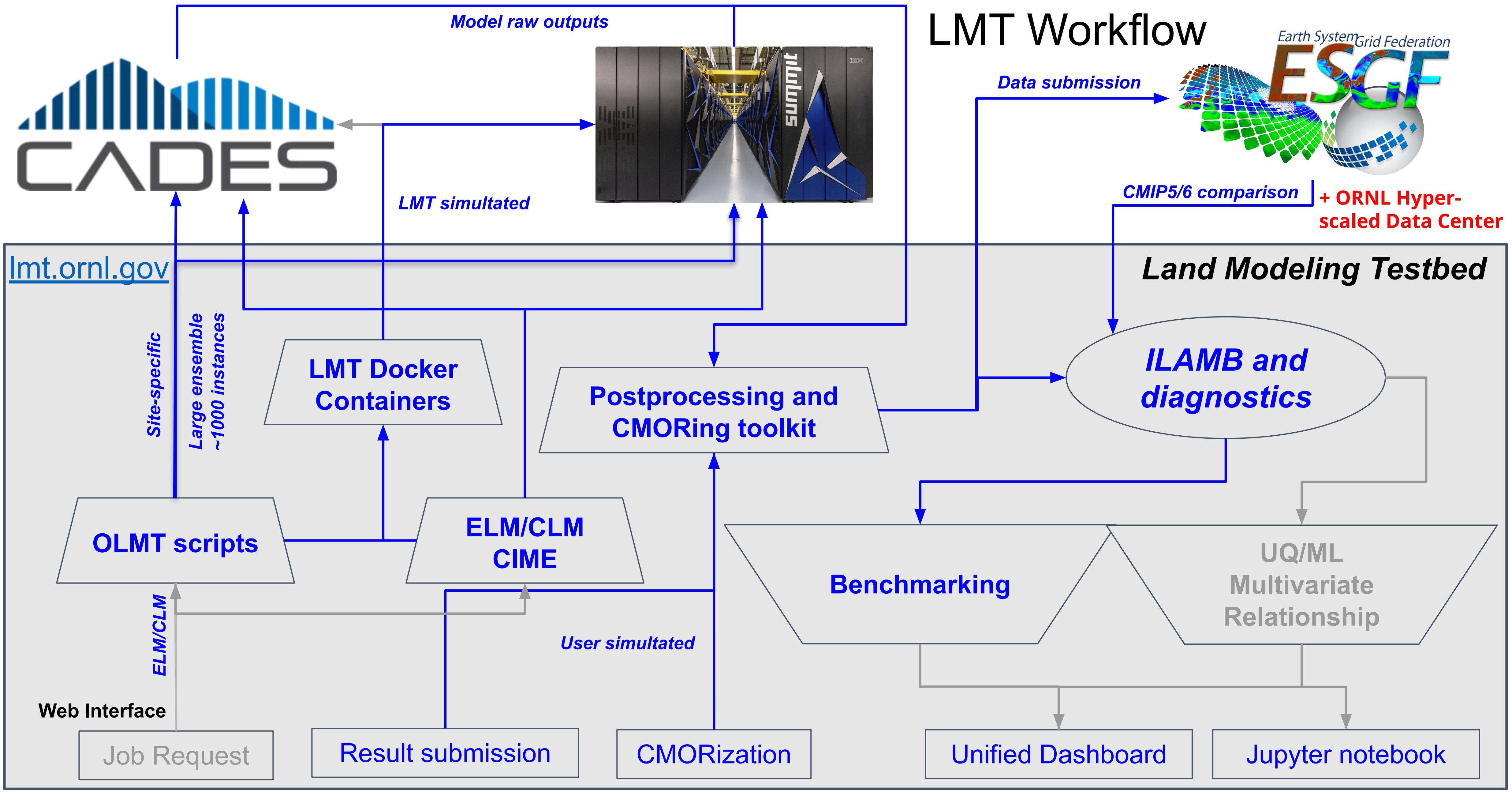
- ILAMB evaluates land model results by comparing with global-, regional-, and site-scale data
- For every variable, multiple observational datasets may be used to evaluate model results







Top-level portrait plot shows relative scores by variable for CMIP5 and CMIP6 models (Hoffman et al., in prep)



Our ensemble generation optimizations resulted in a 72× speed improvement.

Use Case: Perturbed Parameter Ensemble (PPE)

- Critical capability for model uncertainty quantification and parameter optimization
- Developed a framework with ability to launch thousands of model instances (ELM) with parameter variations
- Concurrent model execution on supercomputers like Summit
- Performance optimization to substantially improve ensemble generation workflow
- Required workflow time reduced from 12 hours to 10 minutes on a single node of Summit

JupyterLab: Interactive analysis environment

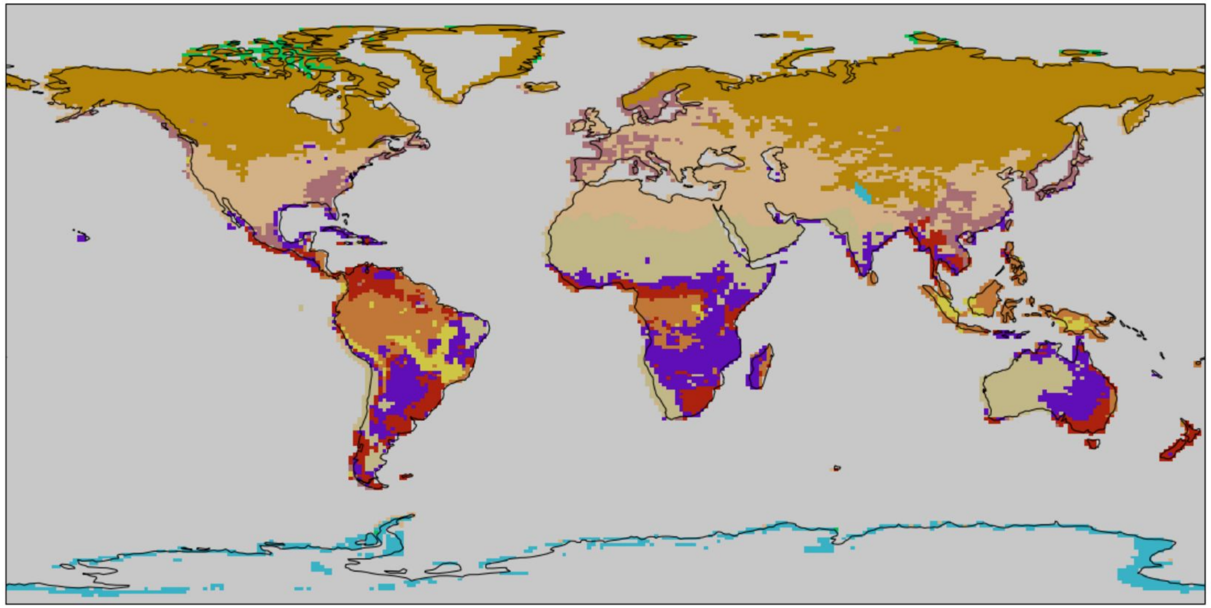
- Deployment on high memory computational nodes facilitates analysis of large datasets
- Allows reproducible and shareable analysis
- Investigating machine learning methods for analysis of model outputs and observations
- ILAMB enables computing standardized metrics and geospatial visualizations.

```
def regime_plotter(t):
    starttime = pd.to_datetime('1850-01-01')
    months_since = int((t - starttime)/np.timedelta64(1,'M'))
    print("%d months since"%(months_since))
    print(t)
    plot_cluster_at_t(months_since+1)


slider = widgets.SelectionSlider(
    options=dates,
    value=pd.to_datetime('2000-01-01'),
    description='Time [YYYYMM]',
    disabled=False,
    continuous_update=False,
    orientation='horizontal',
    readout=True
)
interact(regime_plotter, t=slider)
```

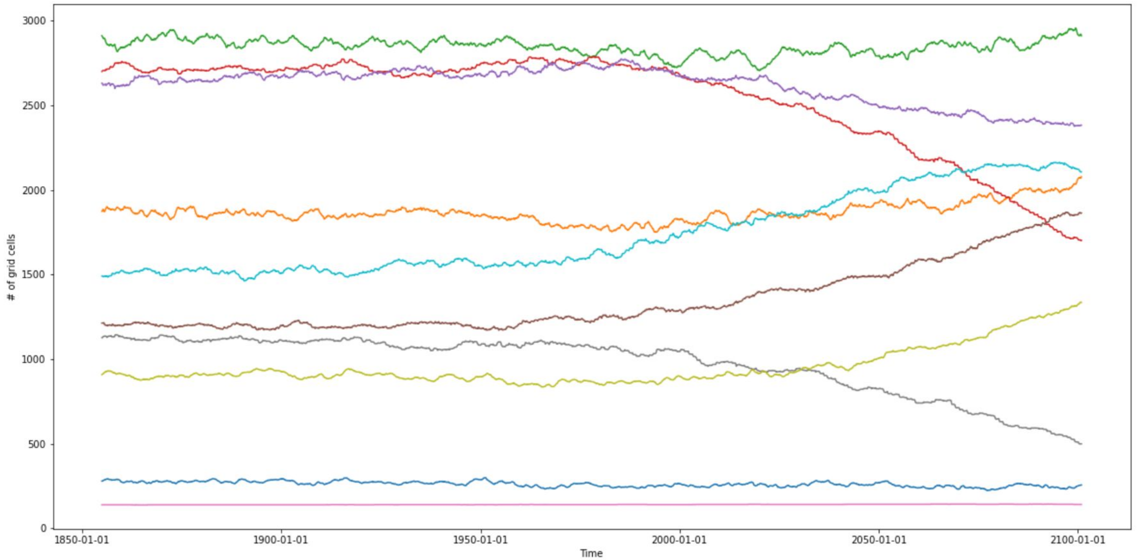
Time [YYY... 2000-01-01 00:

```
1799 months since
2000-01-01 00:00:00
```



```
[14]: <function __main__.regime_plotter(t)>
```

```
[20]: regime_extent_60, dates_60 = moving_average(regime_extent, 60)
      plot_regime_extent(regime_extent_60, dates_60)
```



Unsupervised clustering-based identification of spatial (right) and temporal (left) patterns of dynamic climate regimes under the SSP5-8.5 scenario for the future.

JupyterLab: Dynamic / Descriptive Prototyping

Prototype new functionality

Code refactoring/redesign

Ensemble analysis capability added to ILAMB

[1]:
```
from ModelResult import ModelResult
```

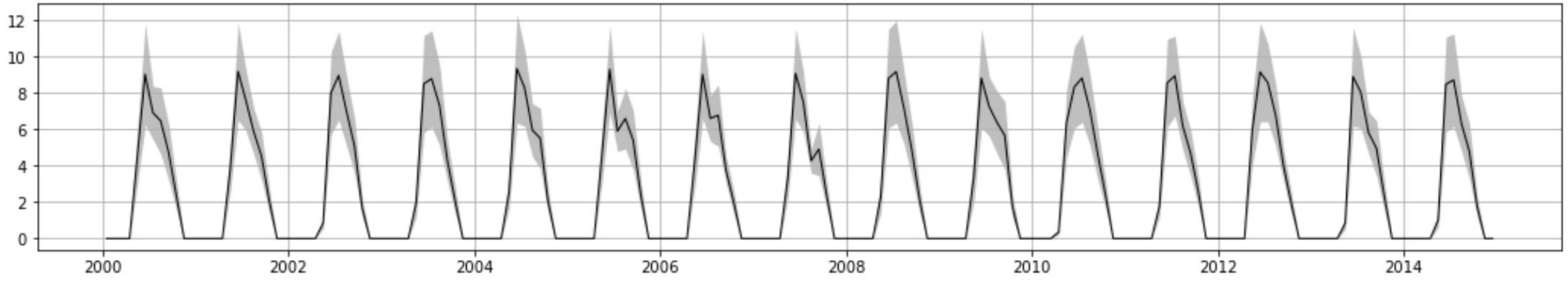
Model results can work as they did before with one small change that I do not automatically find all the files. This is because sometimes we will want a model to be a 'shell' and only really serve as a parent for child models.

[2]:
```
clm4 = ModelResult("/home/ncf/CLM/CLM4.0",name="CLM4").findFiles()
clm5 = ModelResult("/home/ncf/CLM/CLM5.0",name="CLM5").findFiles()
```

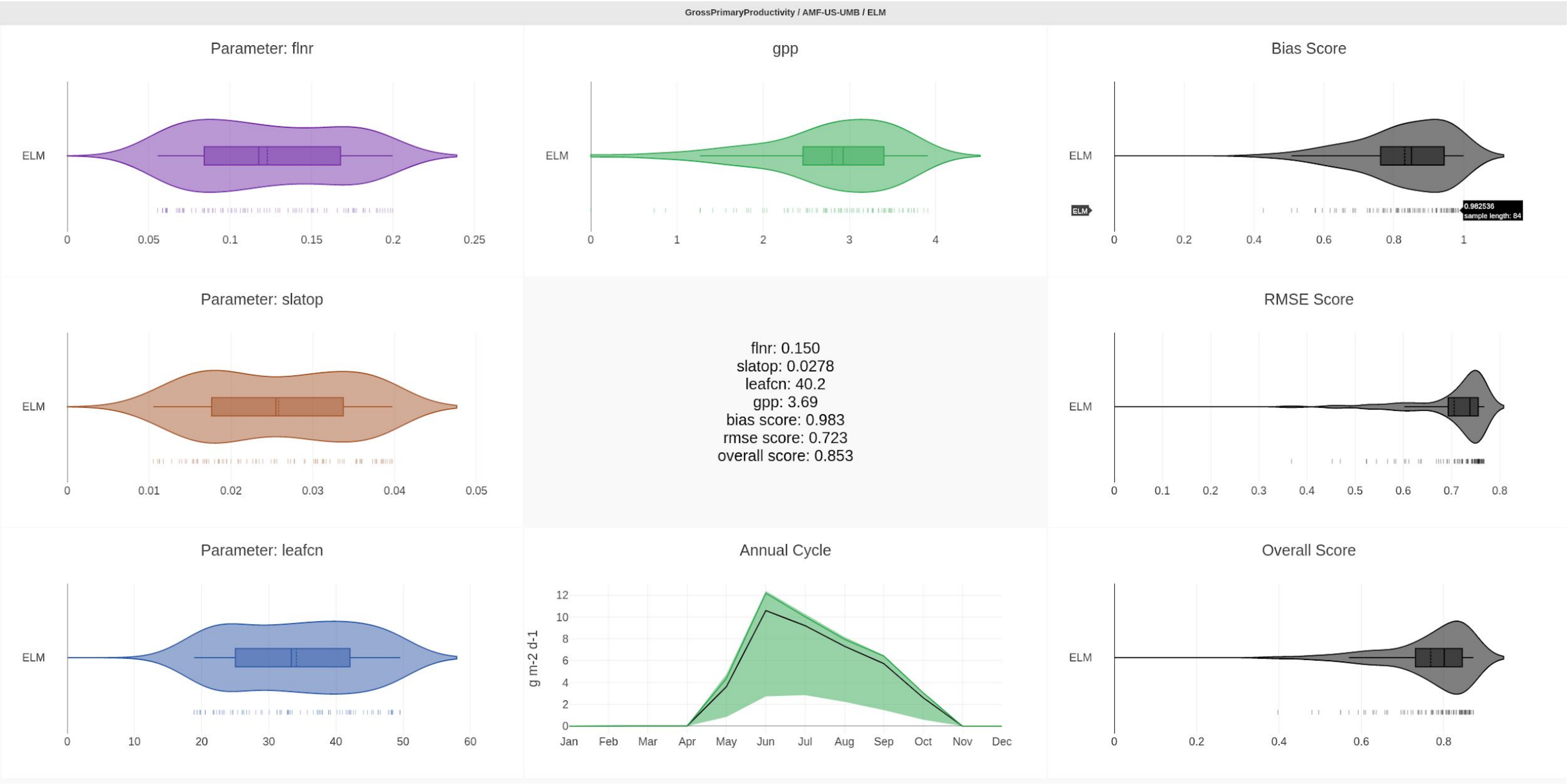
A model could also be a group / collection of models. You could use this to define a collection, like a MIP or to define an ensemble. There is a manual addition of models to the group, seen here:

[3]:
```
grp = ModelResult("/home/ncf/CLM/",name="CLM - manual")
grp.addModel(clm4)
grp.addModel([clm4,clm5]) # can add lists too, won't duplicate
print(grp)
```

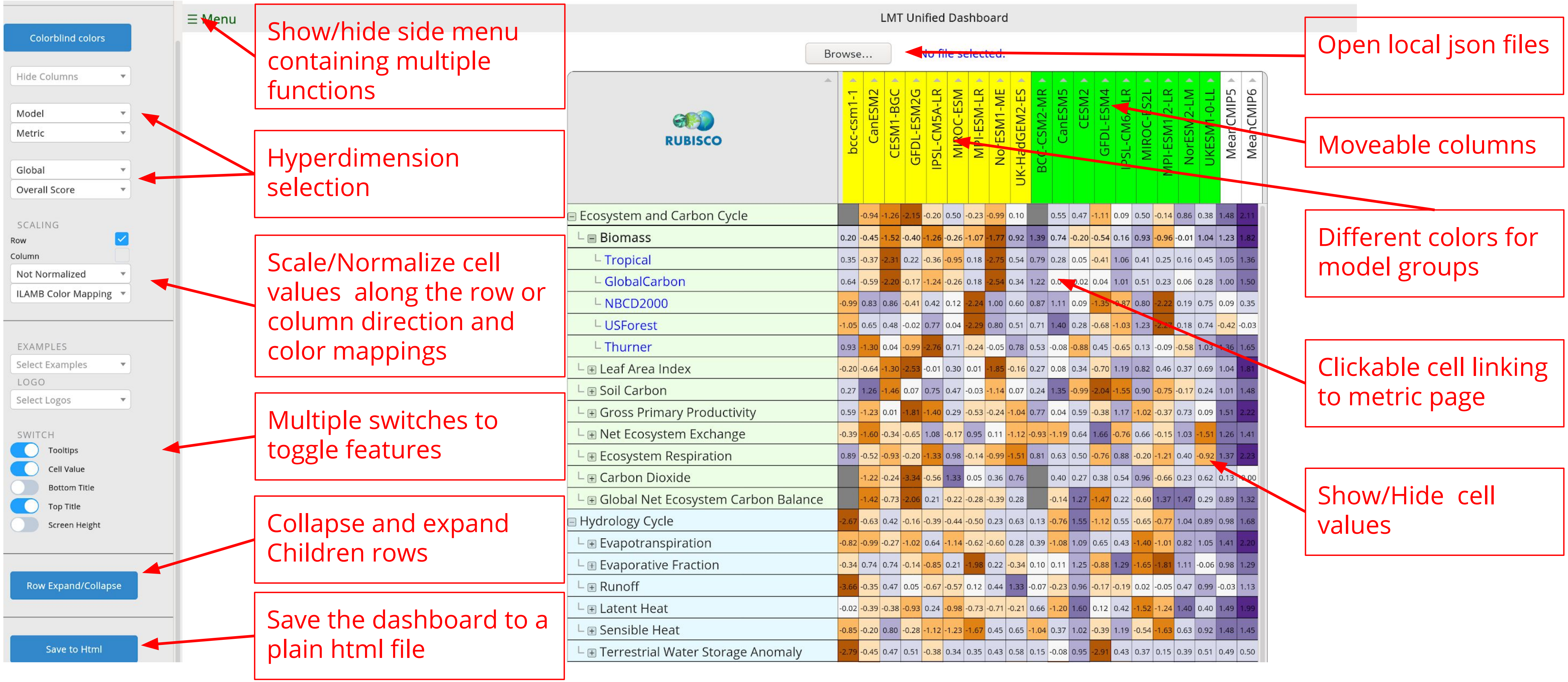
[26]:
```
import matplotlib.pyplot as plt
fig,ax = plt.subplots(figsize=(18,3))
mean_gpp.convert("g m-2 d-1")
mean_gpp.plot(ax)
plt.show()
```



Ensemble Diagnostics Prototype

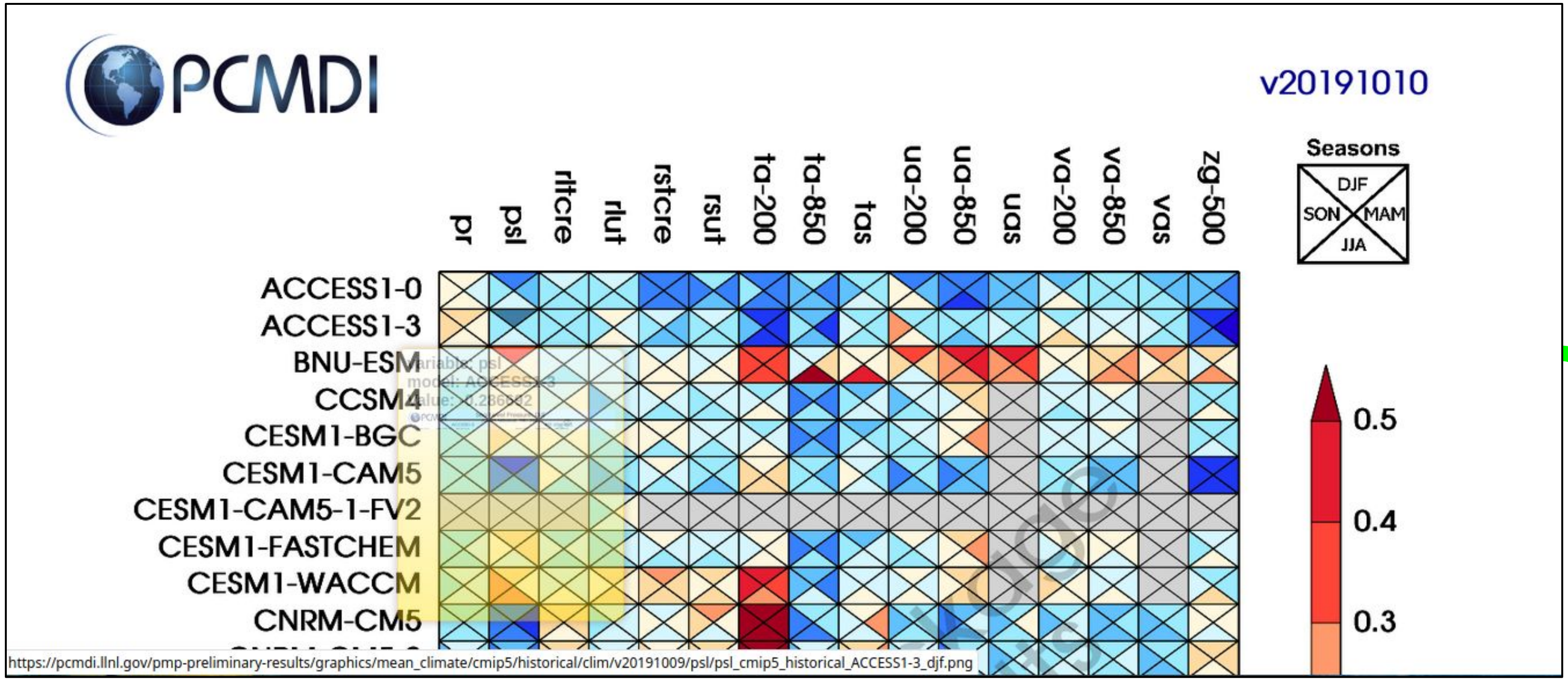


LMT Dashboard: https://lmt.ornl.gov/unified-dashboard/

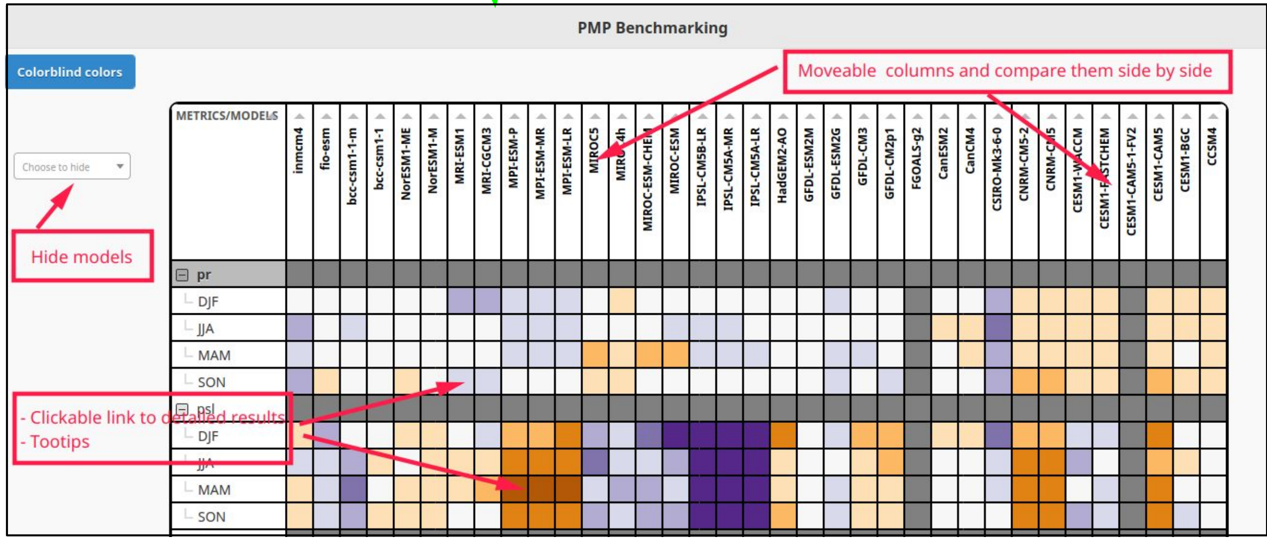


- **Tooltips:** show scores when mouse hovers the cells.
- **Column Hiding:** hide some models (columns) to focus into models of interest.
- **Column sorting:** sort the scores along the columns/models to see the best metric for the model.

Convert other diagnostic results for use in LMT dashboard



https://lmt.ornl.gov/tab_pmp



PMP: The Program for Climate Model Diagnostics and Intercomparison (PCMDI) Metrics Package (PMP)

- Clicking cell will go to maps of geographic distributions generated by PMP
- Our LMT dashboard can be used to study science questions like ENSO-BGC feedbacks

Interactive web-enabled tool (https://lmt.ornl.gov/ers4cmor/) for users to analyze and benchmark any land model results using LMT

Interactive Variable Mapping System For C4MIP

Refresh Table | E3SM variable list | C4MIP | E3SM | Lmon

Hide columns × | Search:

id	priority	long name	varname	units	relationship
Lmon (33 item)					
0	1,2	Precipitation onto Canopy	prveg	kg m-2 s-1	QINTR
1	1,2	Total Carbon Mass Flux from Litter to Soil	fLitterSoil	kg m-2 s-1	LITR1C_TO_SOI
2	1,2	Total Carbon Mass Flux from Vegetation to Litter	fVegLitter	kg m-2 s-1	LITFALL
3	1,2	Percentage of Land Which Is Anthropogenic Pasture	pastureFrac	%	
4	1,2	Total Runoff	mrro	kg m-2 s-1	QRUNOFF+QSN
5	1,2	Net Primary Production Allocated to Roots as Carbon Mass Flux [kgC m-2 s-1]	nppRoot	kg m-2 s-1	FROOTC_ALLO
6	1,2	Carbon Mass in Roots	cRoot	kg m-2	LIVECROOTC+I
7	1,2	Total Soil Moisture Content	mrso	kg m-2	SOILLIQ[0:14]
8	1,2	Carbon Mass in Leaves	cLeaf	kg m-2	LEAFC
9	1,2	Bare Soil Percentage Area Coverage	baresoilFrac	%	PCT_LANDUNIT

- Relationships and equations are saved and versioned in a repository
- Post-processing to standardize (CMORize) the model outputs.
- User defined relationships and mapping to CMIP6 variables

Machine Learning for Deriving Empirical Relationships

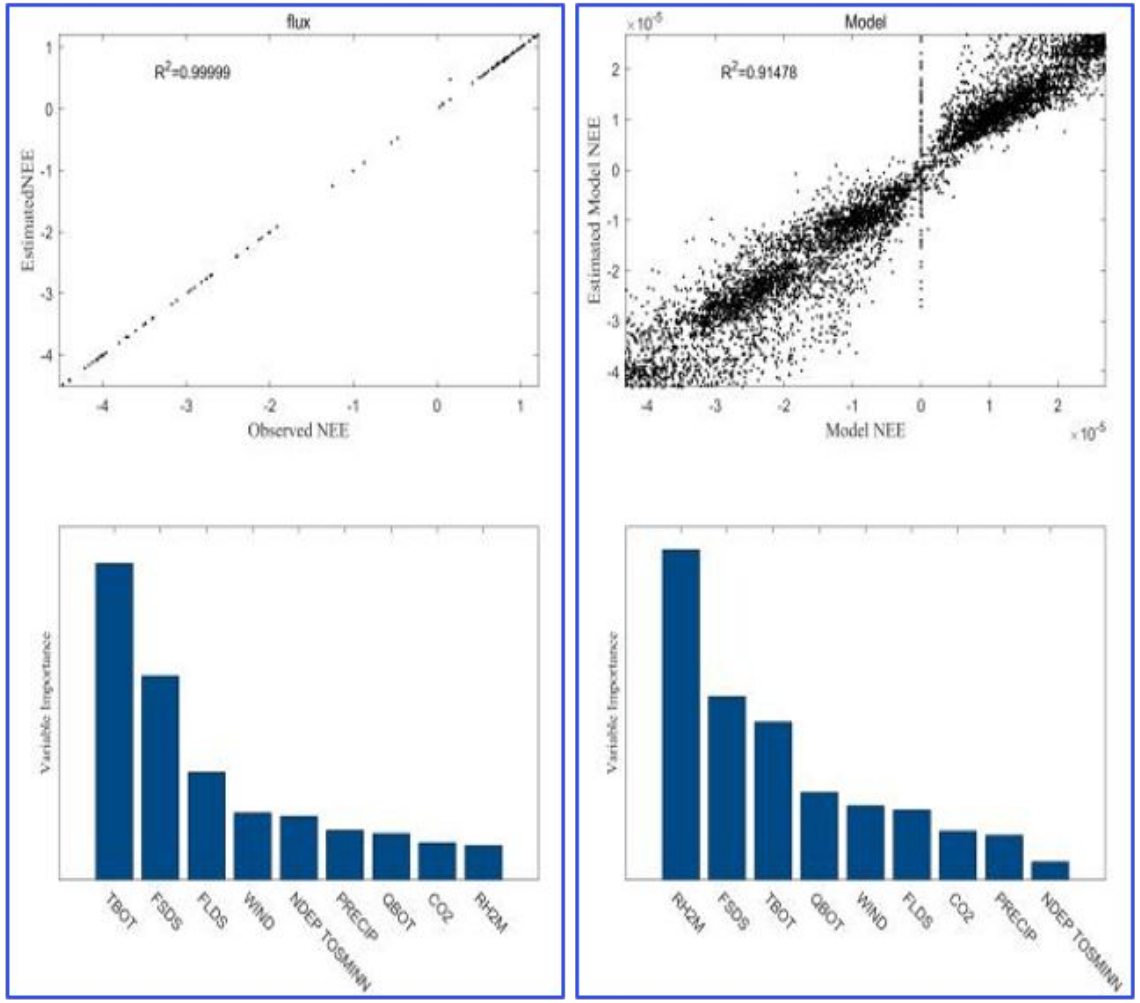


Figure: The relative importance of environmental factors in predicting (left) the observed and (right) the modeled net ecosystem exchange from a land surface model, using a random forest machine learning technique

- We employed a random forest machine learning method to quantitatively predict net carbon ecosystem exchange from observations and from land surface model output
- Various parameter perturbations were used for 84 simulations of the land surface model
- The monthly net ecosystem exchange was predicted accurately by environmental forcing variables in the observational and the model cases
- However, the importance of the environmental factors was very different between the observations and the model results
- This study indicates a need to further study driving relationships in the model

Summary and Future Work

- We developed a suite of tools and capabilities that could become part of an institute or center focused on modeling, model evaluation, and Earth system predictability
- We are extending Perturbed Parameter Ensemble (PPE) effort to test both the Community Land Model (CLM) and the Energy Exascale Earth System Model (E3SM) Land Model (ELM)
- Opportunities for future development include
  - Adding more models to the land model farm
  - Further development of ILAMB benchmarks and diagnostics for PPE
  - Collaborative extension of the Unified Dashboard across a suite of diagnostics packages through the Coordination Model Evaluation Capabilities (CMEC) activity supported by the U.S. Department of Energy
  - Library and tool development for collaborative analysis through JupyterHub
  - Additional analysis employing machine learning approaches to derive empirical relationships as another suite of methods for evaluating model performance