RESEARCH WORKFLOW

ZACHARY LABE (@ZLABE)

https://github.com/zmlabe/ClimatePython

IS THERE A REPRODUCIBILITY CRISIS?

BAKER, 2016; NATURE NEWS

“Today’s weather or climate scientist is far more likely to be debugging code written in Python… than to be poring over satellite images or releasing radiosondes.”

D. Irving | *Bulletin of the American Meteorological Society* | 2016

ESSAY

A MINIMUM STANDARD FOR PUBLISHING COMPUTATIONAL RESULTS IN THE WEATHER AND CLIMATE SCIENCES

BY DAMIEN IRVING

A procedure for publishing reproducible computational research is described that could be adopted as a minimum standard by journals in the weather and climate sciences.

The rise of computational science has led to unprecedented opportunities in the weather and climate sciences. Ever more powerful computers enable experiments that would have been considered impossible only a decade ago, while new hardware technologies allow data collection in even the most inaccessible places. To analyze the vast quantities of data now available to them, modern practitioners—most of whom are not computational experts—use an increasingly diverse set of software tools and packages. Today's weather or climate scientist is far more likely to be found debugging code written in Python, MATLAB, Interactive Data Language (IDL), NCAR Command Language (NCL), or R than to be poring over satellite images or releasing radiosondes.

This computational revolution is not unique to the weather and climate sciences and has led to something of a reproducibility crisis in published research (e.g., Peng 2011). Most papers do not make the data and code underpinning key findings available, nor do they adequately specify the software packages and libraries used to execute that code. This means it is impossible to replicate and verify most of the computational results presented in journal articles today.

D. Irving | *Bulletin of the American Meteorological Society* | 2016

GROWING DATA

WCRP
World Climate Research Programme

ESGF
Earth System Grid Federation

NCAR
UCAR

CISL
Computational & Information Systems Lab

ESGF
Earth System Grid Federation

WCRP
World Climate Research Programme

NCAR
UCAR

CISL
Computational & Information Systems Lab

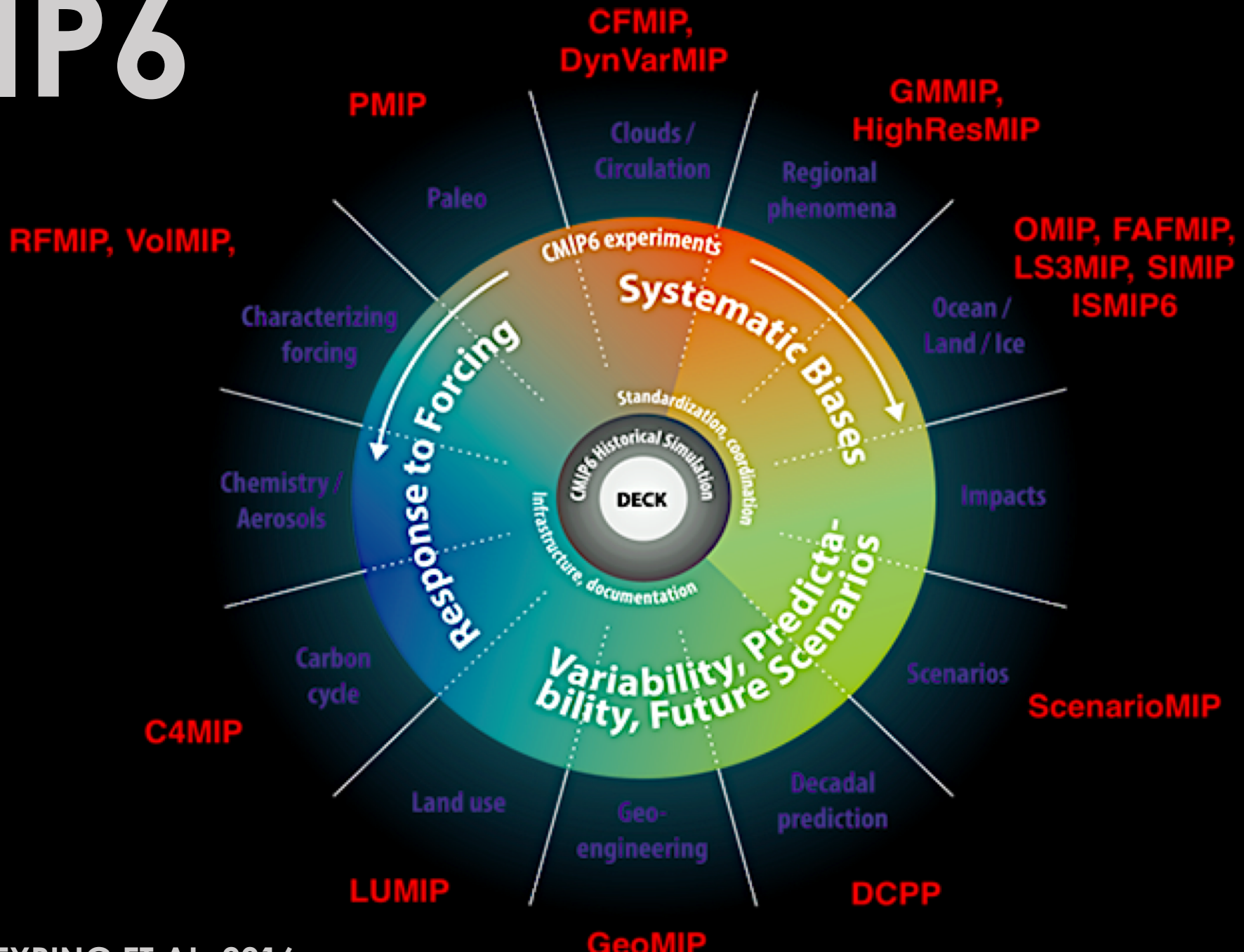
GROWING TOOLS


jupyter
ANACONDA
NCL
zenodo
figshare
R
python™
arXiv.org
open access
to the world's
science

“It helps to reduce the shame many scientists have about the quality of their code (i.e., they see that their peers are no ”better” at coding than they are)…”

D. Irving | *Bulletin of the American Meteorological Society* | 2016

CMIP6



ADAPTED FROM EYRING ET AL. 2016

WE HAVE ALL DONE IT…

Data is available from the corresponding author upon request

WE HAVE ALL DONE IT…

Code is available from the corresponding author upon request

GEOLOGY & GEOPHYSICS

Editors' Vox

"Do You Expect Me to Just Give Away My Data?"

The Editor-in-Chief of *JGR: Oceans* explains why the new AGU data policy is important for the rigor and long-term security of scientific research.

Increasing number of code and data policies

GEOLOGY & GEOPHYSICS

Editors' Vox

"Do You Expect Me to Just Give Away My Data?"

The Editor-in-Chief of *JGR: Oceans* explains why the new AGU data policy is important for the rigor and long-term security of scientific research.

GEOLOGY & GEOPHYSICS

Editors' Vox

"Do You Expect Me to Just Give Away My Data?"

This is a good thing

The Editor-in-Chief of *JGR: Oceans* explains why the new AGU data policy is important for the rigor and long-term security of scientific research.

So what can we do?

Pre-process data using NCO, CDO, or NCL

```
// global attributes:
                :history = "Fri Feb 16 18:08:22 2018: ncrename -v lon,longitude SLP_1900-2000.nc\n",
                           "Fri Feb 16 18:08:12 2018: ncrename -d lon,longitude SLP_1900-2000.nc\n",
                           "Fri Feb 16 18:08:02 2018: ncrename -v lat,latitude SLP_1900-2000.nc\n",
                           "Fri Feb 16 18:07:52 2018: ncrename -d lat,latitude SLP_1900-2000.nc" ;
```

Download observational or reanalysis data

Download or conduct climate model simulation

Advanced analysis of data using Python, NCL, etc

Create science figures

Store processed data on remote server

WORKFLOW

WORKFLOW

Create a working directory and a backup directory for project

Download observational or reanalysis data

Download or conduct climate model simulation

Advanced analysis of data using Python, NCL, etc

Create science figures

Use of GitHub for version control with 1 repository per project

WORKFLOW

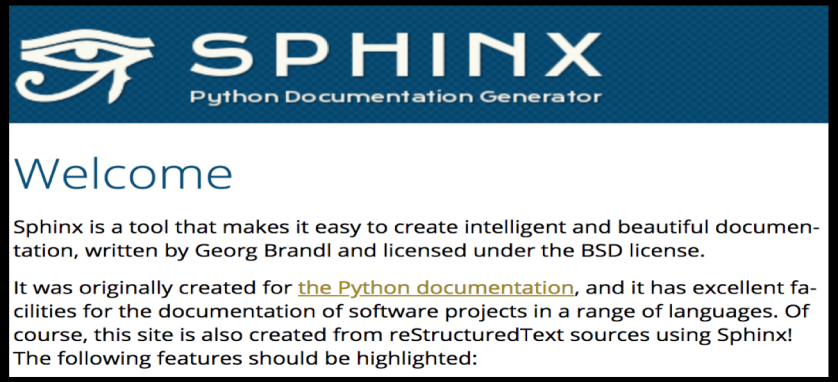
Download observational or reanalysis data

Download or conduct climate model simulation

Advanced analysis of data using Python, NCL, etc

Create science figures

Assess code documentation



Use Zenodo to create project DOI

Journal of
open research software

Dawson, A 2016 eofs: A Library for EOF Analysis of Meteorological, Oceanographic, and Climate Data. *Journal of Open Research Software*, 4: e14, DOI: http://dx.doi.org/10.5334/jors.122

SOFTWARE METAPAPER

eofs: A Library for EOF Analysis of Meteorological, Oceanographic, and Climate Data

Andrew Dawson[1]

[1] Atmospheric, Oceanic & Planetary Physics, Department of Physics, University of Oxford, Oxford, UK
andrew.dawson@physics.ox.ac.uk

Behind each library

The *eofs* library provides a high-level Python interface for computing empirical orthogonal functions (EOFs) and related quantities, with a focus on correctness and ease of use. The library is implemented in a modular hierarchical fashion, allowing computations using plain arrays, or the inclusion of metadata. The software provides a convenient package for users wanting to perform EOF analysis in Python, and integrates with popular libraries from atmospheric and climate science. The software is available on Github.

Keywords: EOF analysis; Meteorology; Oceanography; Climate; Python

(1) Overview

Introduction

Data sets in meteorology, oceanography, and climate are typically very large, containing data covering large spatial areas, observed or modelled over long periods of time. Studying variability in these data sets can be challenging. For example, consider the task of computing a time-series of sea surface temperature on a latitude-longitude grid. First one must correctly weight the input data to account for spatial variability in the size of grid cells due to convergence of the meridians. The input data must then be reconfigured into a 2-dimensional form, and care taken to remove any missing values (e.g., values

Journal of
open research software

Dawson, A 2016 eofs: A Library for EOF Analysis of Meteorological, Oceanographic, and Climate Data. *Journal of Open Research Software,* 4: e14, DOI: http://dx.doi.org/10.5334/jors.122

SOFTWARE METAPAPER

eofs: A Library for EOF Analysis of Meteorological, Oceanographic, and Climate Data

Andrew Dawson[1]

[1] Atmospheric, Oceanic & Planetary Physics, Department of Physics, University of Oxford, Oxford, UK
andrew.dawson@physics.ox.ac.uk

The *eofs* library provides a high-level Python interface for computing empirical orthogonal functions (EOFs) and related quantities, with a focus on correctness and ease of use. The library is implemented in a modular hierarchical fashion, allowing computations using plain arrays, or the inclusion of metadata. The software provides a convenient package for users wanting to perform EOF analysis in Python, and integrates with popular libraries from atmospheric and climate science. The software is available on Github.

Keywords: EOF analysis; Meteorology; Oceanography; Climate; Python

(1) Overview

Introduction

Data sets in meteorology, oceanography, and climate are typically very large, containing data covering large spatial areas, observed or modelled over long periods of time. Studying variability in these data sets can be challenging. Consider for example a time-series of sea surface temperature on a latitude-longitude grid. First one must correctly weight the input data to account for spatial variability in the size of grid cells due to convergence of the meridians. The input data must then be reconfigured into a 2-dimensional form, and care taken to remove any missing values (e.g., values

WHY

Transparency — Climate science needs this!

Is there a crisis in science? — **Reproducibility**

Code-sharing — Python is a growing community

Big data is the future — **Innovation**

Open-science — Science deserves to be shared