Funded by the European Union Destination Earth implemented by ECMWF esa EUMETSAT

Replicability Testing and Scientific Skill Quantification in Earth System Models with *pyhanami*

Marta Alerany[1], Kai Keller[1], Chihiro Kodama[2], Masuo Nakano[2], Tomoe Nasuno[2], Mario Acosta[1]

[1]Barcelona Supercomputing Center (BSC-CNS), Barcelona, Spain
[2]Japan Agency for Marine-Earth Science and Technology, Yokohama, Japan

Funded by the European Union Destination Earth implemented by ECMWF esa EUMETSAT

Replicability Testing and Scientific Skill Quantification in Earth System Models with *pyhanami*

Marta Alerany[1], Kai Keller[1], Chihiro Kodama[2], Masuo Nakano[2], Tomoe Nasuno[2], Mario Acosta[1]

[1]Barcelona Supercomputing Center (BSC-CNS), Barcelona, Spain
[2]Japan Agency for Marine-Earth Science and Technology, Yokohama, Japan

pyhanami

Funded by the European Union Destination Earth implemented by ECMWF esa EUMETSAT

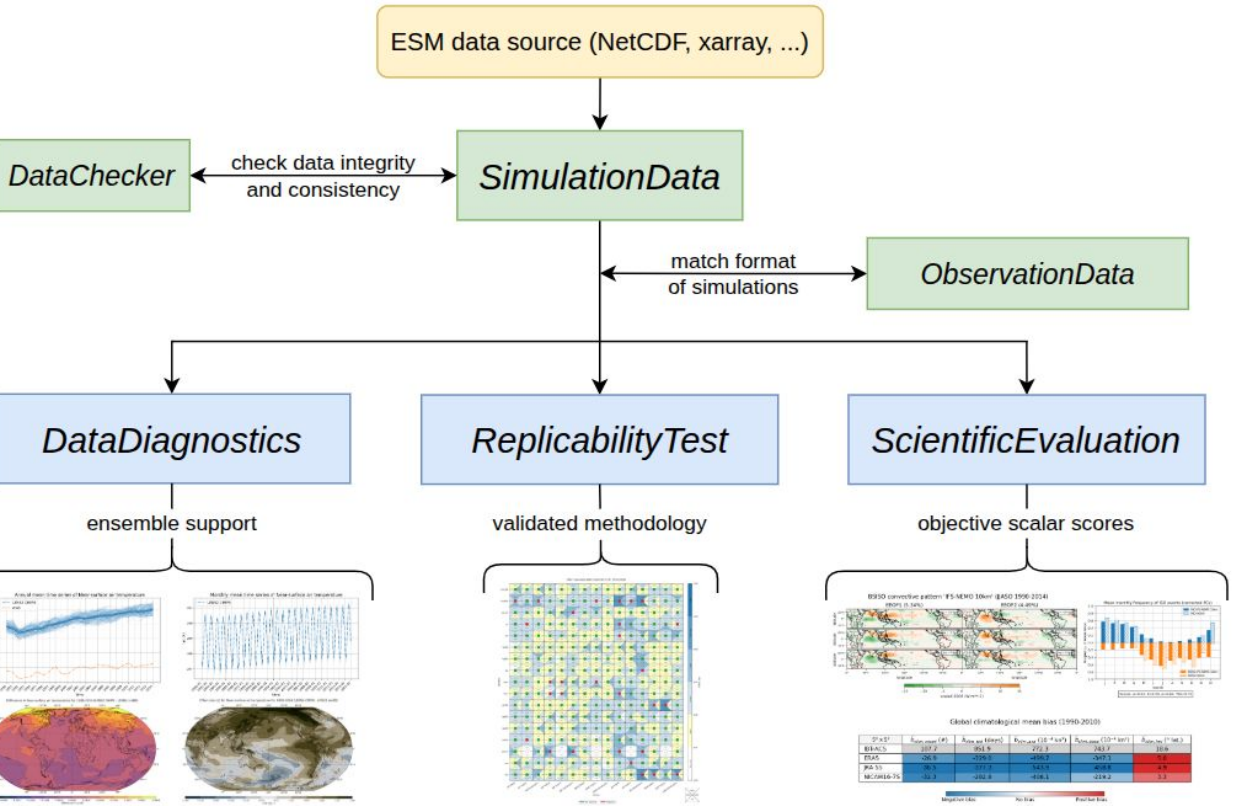
Replicability Testing and Scientific Skill Quantification in Earth System Models with *pyhanami*

Marta Alerany[1], Kai Keller[1], Chihiro Kodama[2], Masuo Nakano[2], Tomoe Nasuno[2], Mario Acosta[1]

[1]Barcelona Supercomputing Center (BSC-CNS), Barcelona, Spain
[2]Japan Agency for Marine-Earth Science and Technology, Yokohama, Japan

pyhanami

- Generic data access (NetCDF files / xarray datasets)
- Diagnostics plotting (intrinsic ensemble support)
- Replicability testing (validated methodology)
- Scientific evaluation (quantitative model validation and intercomparison)



Check our repository:

Funded by the European Union **Destination Earth** implemented by ECMWF esa EUMETSAT

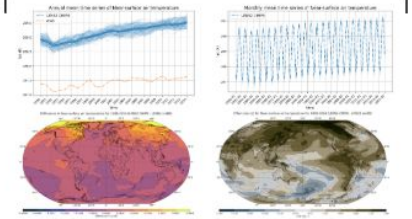
ESM data source (NetCDF, xarray, ...)

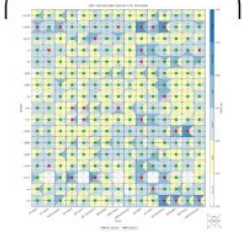
DataChecker ← check data integrity and consistency → *SimulationData*

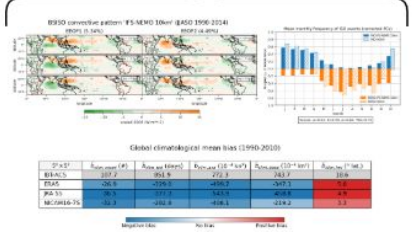
SimulationData ↔ match format of simulations ↔ *ObservationData*

SimulationData →

- *DataDiagnostics* → ensemble support
- *ReplicabilityTest* → validated methodology
- *ScientificEvaluation* → objective scalar scores







I'll be at PICO spot 1b.2!
marta.alerany@bsc.es

Check our repository:

Funded by the European Union Destination Earth implemented by ECMWF esa EUMETSAT

Replicability Testing and Scientific Skill Quantification in Earth System Models with *pyhanami*

Marta Alerany[1] (marta.alerany@bsc.es), Kai Keller[1], Chihiro Kodama[2], Masuo Nakano[2], Tomoe Nasuno[2], Mario Acosta[1]

[1]Barcelona Supercomputing Center (BSC-CNS), Barcelona, Spain
[2]Japan Agency for Marine-Earth Science and Technology, Yokohama, Japan

pyhanami

Check our repository:

Navigation panel

- Context
- Package design
- Visual diagnostics
- Replicability testing
- Scientific evaluation
- Ongoing and future features

EGU General Assembly

Sharing is encouraged

1. Context

Context

Destination Earth

Flagship initiative of the European Commission to develop a highly-accurate digital model of the Earth (a **digital twin** of the **Earth**) to:

- **Monitor and simulate** the Earth's system developments and human interventions
- **Anticipate** environmental disasters and resultant socio-economic crises
- Enable the **development and testing** of scenarios

Access data through its **Data Bridge**: unified API for all DestinE components + dedicated compute and data storage capabilities

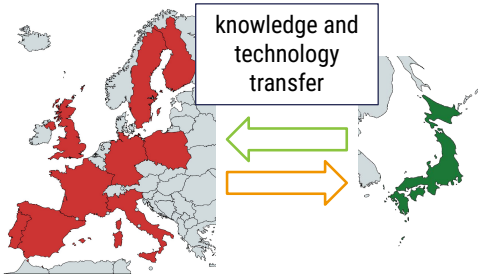
HANAMI → **Hpc AlliaNce for Applications and supercoMputing Innovation: the Europe-Japan collaboration**

Three Scientific Pillars:

- *WP4 - Climate and natural disaster*
- *WP5 - Biomedical science*
- *WP6 - Materials science*

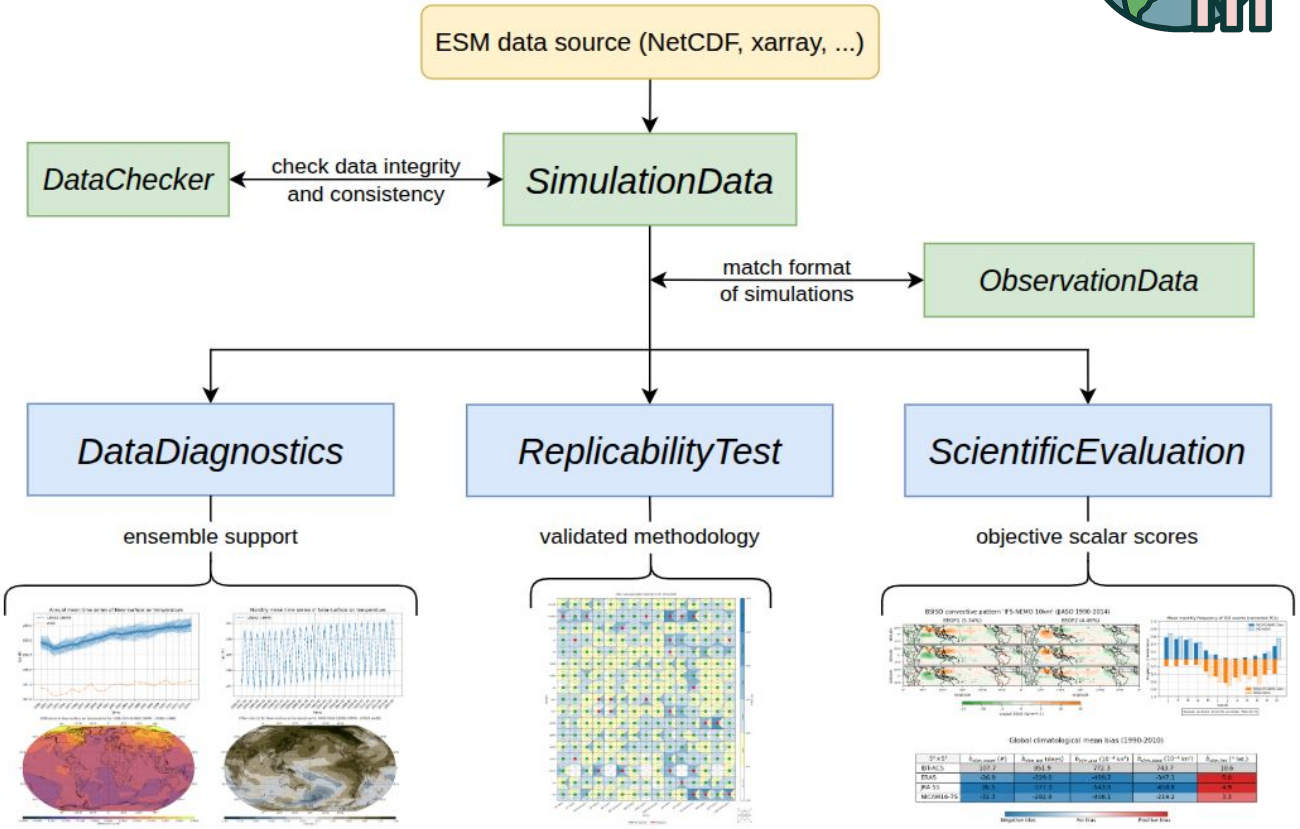
WP4 has three projects:

- *Earth-system-model performance assessment*
- *Earth-system-model benchmark suite (HPCW)*
- *Large-eddy cloud simulation model (UWLCM)*



2. Package design

Package design



Funded by the European Union
Implemented by ECMWF esa EUMETSAT

3. Visual diagnostics

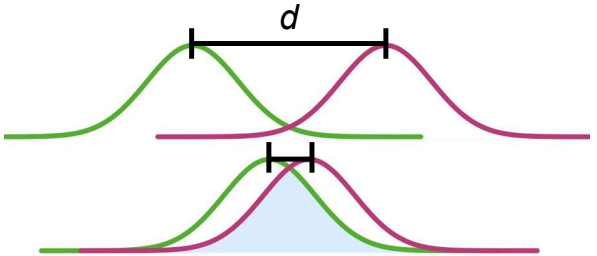
Visual diagnostics: types

Visual plots to quickly compare **ensembles of simulations**, options:

- **Time series** (ensemble mean/spread, individual members, observations)
- **Spatial plots:**
  - Bias (relative to observations)
  - Absolute difference (between ensemble means)
  - Effect size (*d*) + significant differences

$$d = \frac{\mu_1 - \mu_2}{\sigma} \quad \text{with} \quad \sigma = \sqrt{\frac{(n_1 - 1)\,\sigma_1^2 + (n_2 - 1)\,\sigma_2^2}{n_1 + n_2 - 2}}$$

$$= \sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}} \quad , \quad \text{for} \, n_1 = n_2$$



Visual diagnostics: implementation

To generate visual diagnostics of simulation datasets with **pyhanami** we need to initialize the *DataDiagnostics* class with the *SimulationData* objects that we want to analyze:

1. Load simulation data:

```
import pyhanami

# Load simulation datasets
sim_1 = pyhanami.SimulationData('source_sim_1', name='name_sim_1')
sim_2 = pyhanami.SimulationData('source_sim_2', name='name_sim_2')
```

2. Initialize diagnostics class:

```
# Initialize DataDiagnostics class with the SimulationData objects
diags = pyhanami.DataDiagnostics([sim_1, sim_2])
```

3. Create time series plots:

```
# Plot annual mean time series for two datasets together
diags.time_series_plot('var_name', ['name_sim_1', 'name_sim_2'], 'output_path')
```

click to see plot

```
# Plot annual mean time series for one dataset showing all ensemble members
diags.time_series_plot('var_name', 'name_sim_1', 'output_path', plot_ens=True)
```

click to see plot

```
# Plot monthly mean time series for one dataset
diags.time_series_plot('var_name', 'name_sim_1', 'output_path', time_freq='monthly')
```

click to see plot

4. Create spatial plots:

```
# Plot spatial bias for one dataset
diags.bias_plot('var_name', 'name_sim_1', 'output_path')
```

click to see plot

```
# Plot spatial absolute difference between two datasets
diags.abs_diff_plot('var_name', ['name_sim_1', 'name_sim_2'], 'output_path')
```

click to see plot

```
# Plot spatial effect size between two datasets centered at longitude 180º
diags.eff_size_plot('var_name', ['name_sim_1', 'name_sim_2'], 'output_path', clon=180)
```

click to see plot

Visual diagnostics: implementation

To generate
initialize th
analyze:

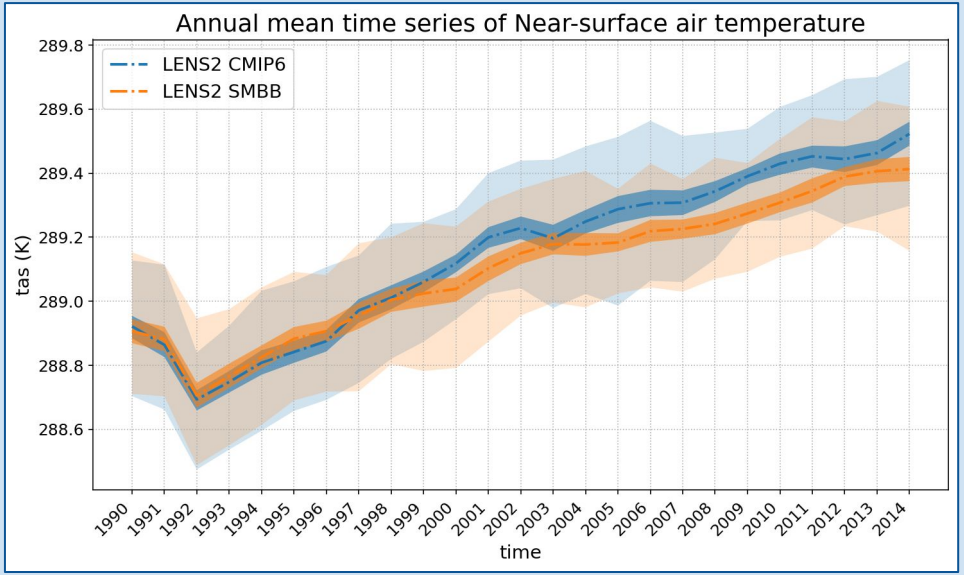
1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

3. Create tim

```
# Plot annual
diags.time_ser
click to see
# Plot annual
diags.time_ser
click to see
# Plot monthly
diags.time_ser
click to see
```

Use case: **time series** of CESM2-LENS2 simulation ensembles using different biomass burning protocols, the original CMIP6 one ('CMIP6') and a smoothed version ('SMBB') showing the ensembles mean and spread (50 members each, period 1990-2014):



minimize

```
ut_path')

ude 180º
ut_path', clon=180)
```

Visual diagnostics: implementation

To generate
initialize th
analyze:

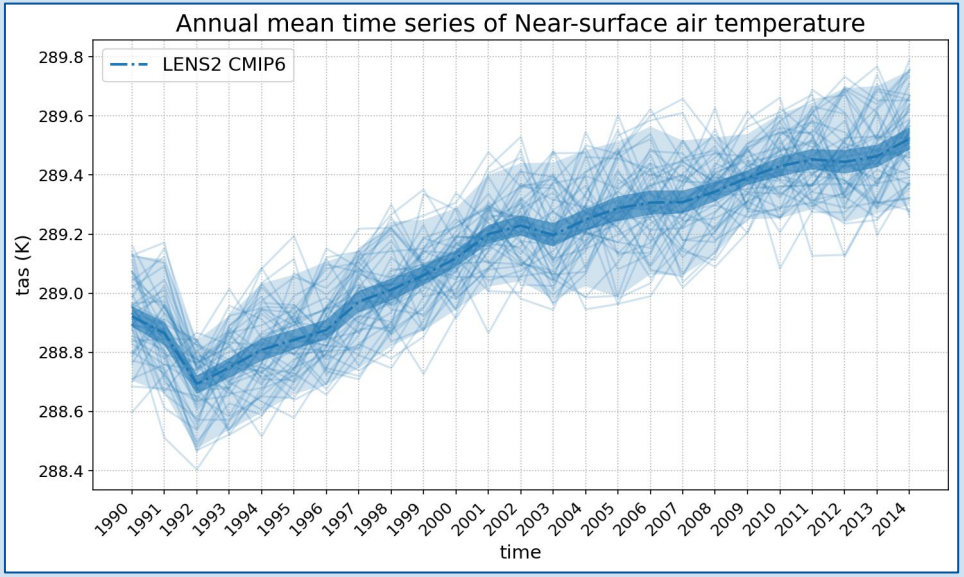
1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

3. Create tim

```
# Plot annual
diags.time_ser
click to see
# Plot annual
diags.time_ser
click to see
# Plot monthly
diags.time_ser
click to see
```

Use case: **time series** of a CESM2-LENS2 simulation ensemble showing ensemble mean, spread and individual members (50 members, period 1990-2014):



minimize

```
out_path')

ude 180º
out_path', clon=180)
```

Visual diagnostics: implementation

To generate
initialize th
analyze:

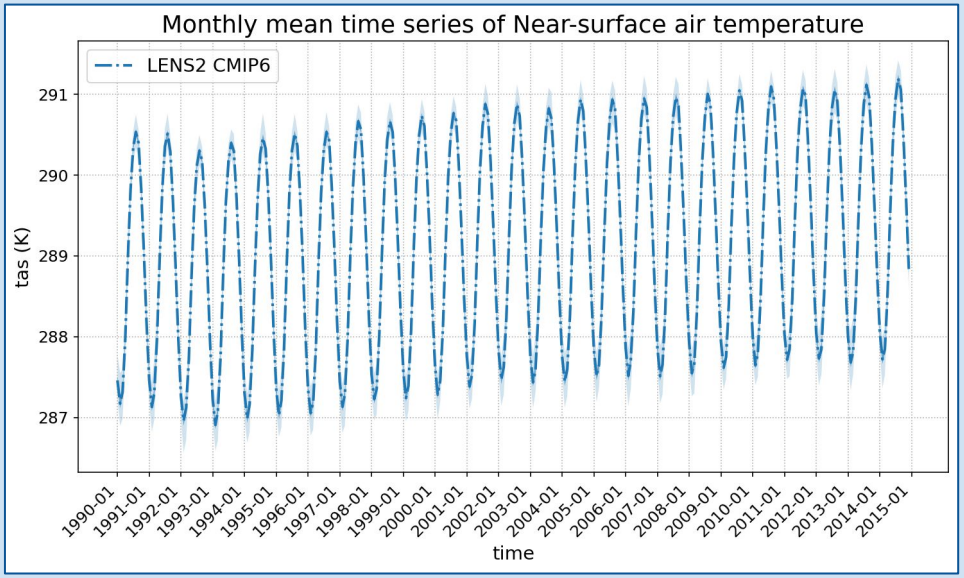
1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

3. Create tim

```
# Plot annual
diags.time_ser
click to see
# Plot annual
diags.time_ser
click to see
# Plot monthly
diags.time_ser
click to see
```

Use case: **monthly time series** of a CESM2-LENS2 simulation ensemble showing the ensemble mean and spread (50 members, period 1990-2014):



minimize

```
ut_path')

ude 180º
ut_path', clon=180)
```

Visual diagnostics: implementation

To generate
initialize th
analyze:

1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

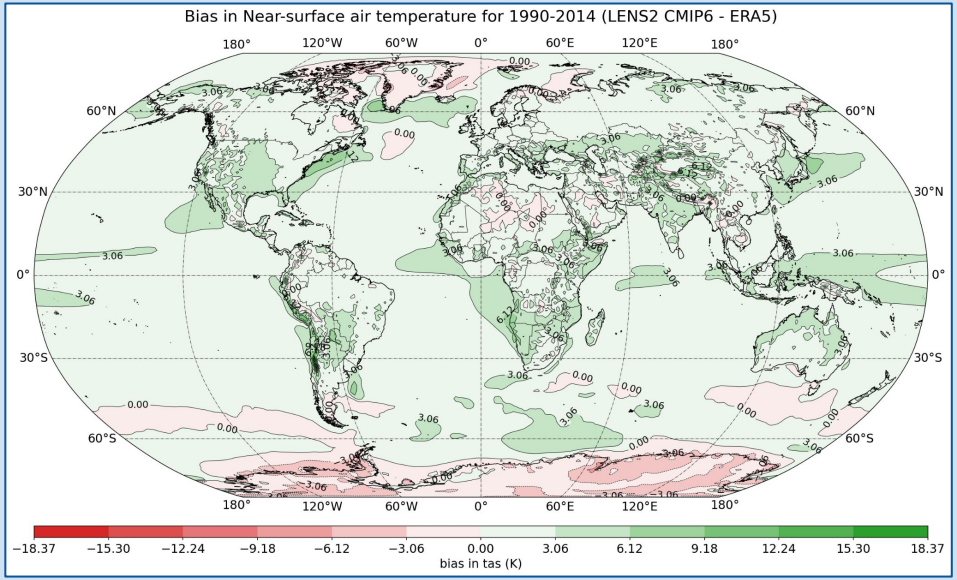
3. Create tim

```
# Plot annual
diags.time_ser
```
click to see

```
# Plot annual
diags.time_ser
```
click to see

```
# Plot monthly
diags.time_ser
```
click to see

Use case: **spatial bias** of a CESM2-LENS2 simulation ensemble relative to ERA5 (50 members, period 1990-2014):



minimize

```
ut_path')

ude 180º
ut_path', clon=180)
```

Visual diagnostics: implementation

To generate
initialize th
analyze:

1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

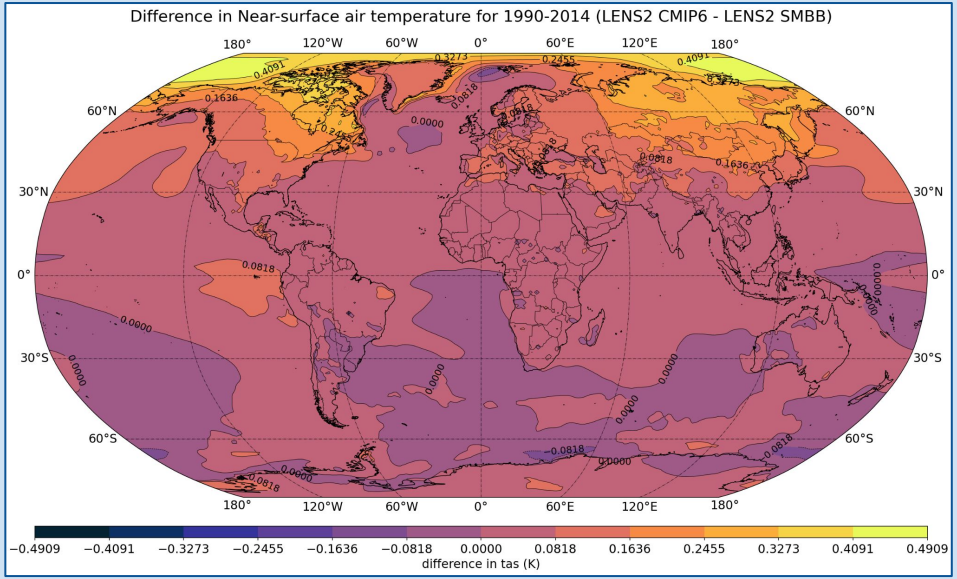
3. Create tim

```
# Plot annual
diags.time_ser
```
click to see

```
# Plot annual
diags.time_ser
```
click to see

```
# Plot monthly
diags.time_ser
```
click to see

Use case: **spatial absolute difference** between CESM2-LENS2 simulation ensembles using different biomass burning protocols, the original CMIP6 one ('CMIP6') and a smoothed version ('SMBB') (50 members each, period 1990-2014):



minimize

```
ut_path')

ude 180º
ut_path', clon=180)
```

Visual diagnostics: implementation

To generate
initialize th
analyze:

1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

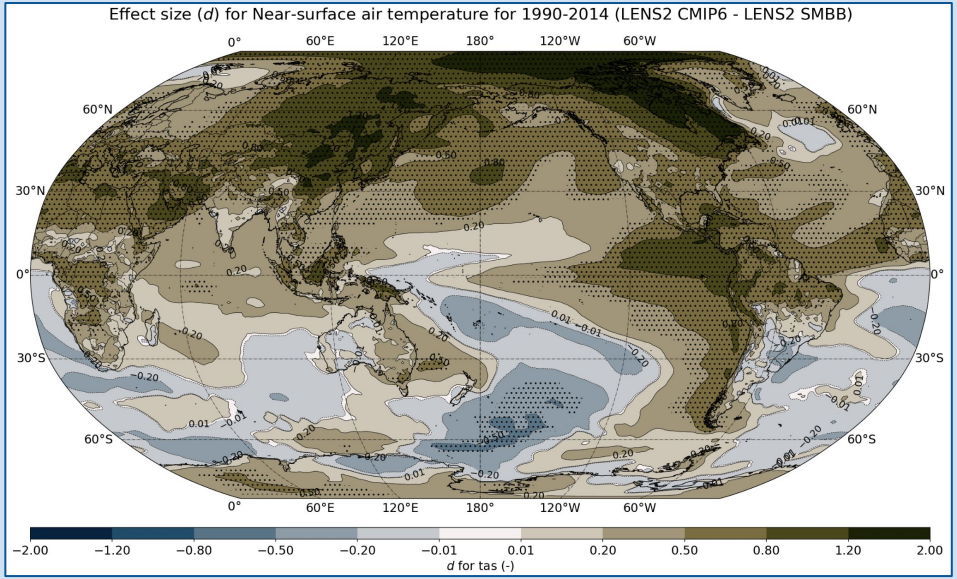
3. Create tim

```
# Plot annual
diags.time_ser
```
click to see

```
# Plot annual
diags.time_ser
```
click to see

```
# Plot monthly
diags.time_ser
```
click to see

Use case: **spatial effect size** between CESM2-LENS2 simulation ensembles using different biomass burning protocols, the original CMIP6 one ('CMIP6') and a smoothed version ('SMBB') centered at longitude 180º (50 members each, period 1990-2014):



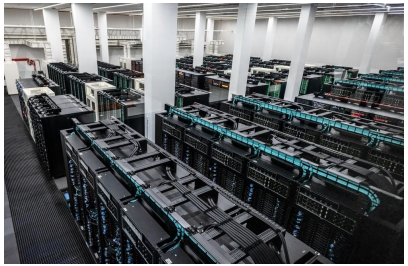
minimize

```
ut_path')

ude 180º
ut_path', clon=180)
```

4. Replicability testing

Replicability test: concept

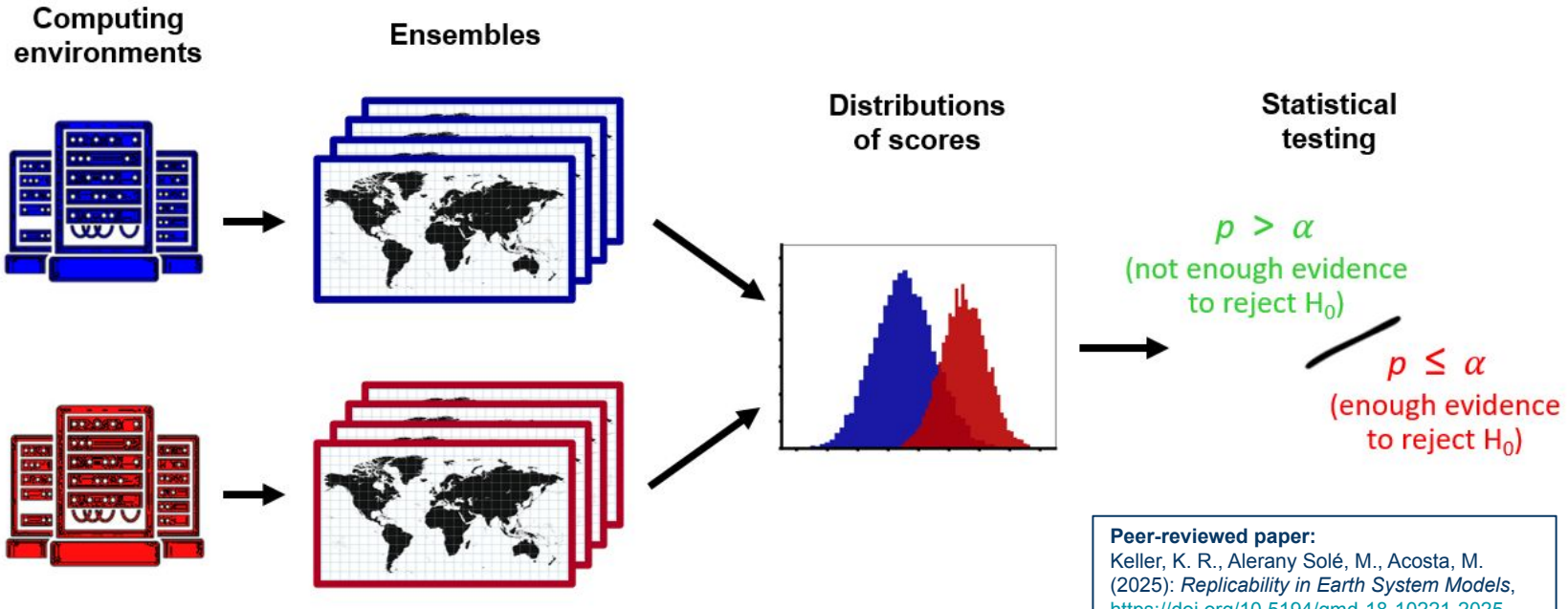
An Earth System Model is **replicable** if the **same experiment** run on two **different computing environments** yields the **same results:**

EC-EARTH +  ? = EC-EARTH + 

Replicability is essential to ensure that changes in the simulated climate respond only to scientific factors, not related to software/hardware differences; however, bit-to-bit replicability is not feasible in practice.

Replicability test: methodology

Validated methodology to detect **statistically significant differences** between two simulation ensembles:



Peer-reviewed paper:
Keller, K. R., Alerany Solé, M., Acosta, M. (2025): *Replicability in Earth System Models*, https://doi.org/10.5194/gmd-18-10221-2025

Note:
p = p-value, α = significance level

Replicability test: implementation

To perform a replicability test comparing to simulation datasets with **pyhanami** we need to initialize the *ReplicabilityTest* class with two *SimulationData* objects and use the *perform_rep_test method:*

1. Load simulation data:

```
import pyhanami

# Load simulation datasets
sim_1 = pyhanami.SimulationData('source_sim_1', name='name_sim_1')
sim_2 = pyhanami.SimulationData('source_sim_2', name='name_sim_2')
```

2. Initialize replicability class:

```
# Initialize ReplicabilityTest class with the SimulationData objects
tester = pyhanami.ReplicabilityTest([sim_1, sim_2])
```

3. Perform replicability test:

```
# Perform replicability test between both datasets
tester.perform_rep_test(['name_sim_1', 'name_sim_2'])
```

4. Create plot with results:

```
# Plot outcome of the test
tester.matrix_plot(['name_sim_1', 'name_sim_2'], 'output_path')
```

click to see plot

Replicability test: implementation

To perform a
need to init
the *perfor*

1. Load simu

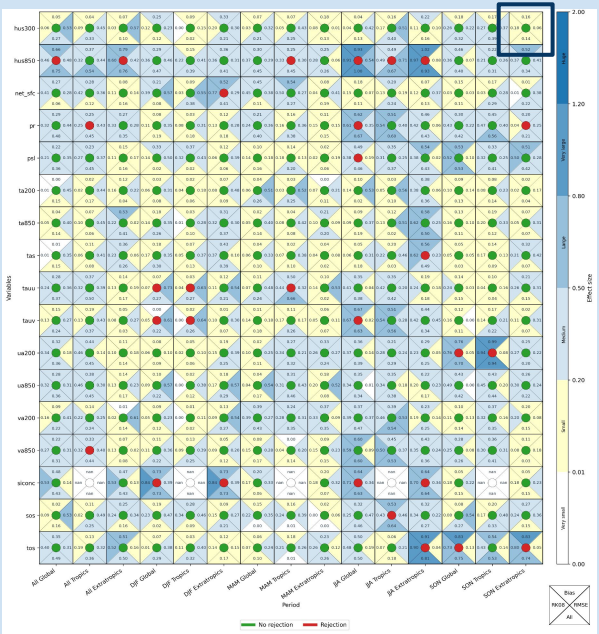
```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

3. Perform re

```
# Perform repl
tester.perform
```

Use case: **replicability test** between Double precision (DP) and Single precision (SP) versions of IFS-NEMO (15 members, 160 km + 100 km horizontal resolution (atmosphere + ocean), period 2010-2020):

click to enlarge



minimize

Replicability test: implementation

To perform a
need to init
the *perfor*

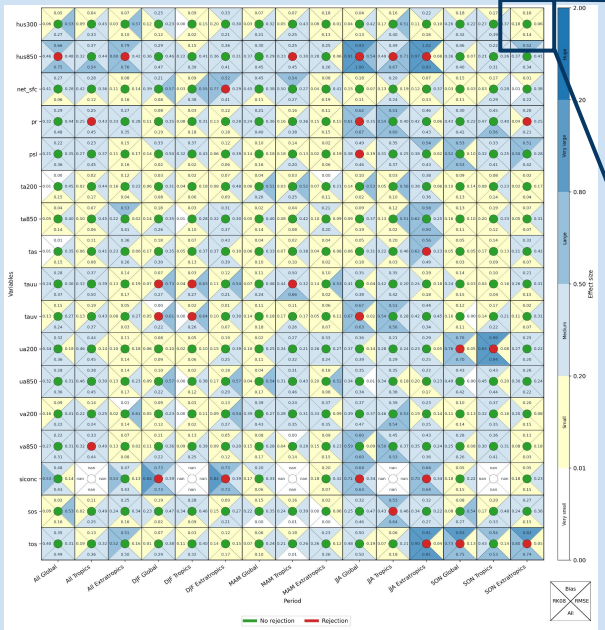
1. Load simu

```
import pyhanam

# Load simulat
sim_1 = pyhana
sim_2 = pyhana
```

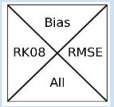
3. Perform re

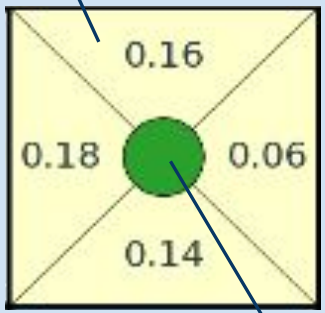
```
# Perform repl
tester.perform
```

Use case: **replicability test** between Double precision (DP) and Single precision (SP) versions of IFS-NEMO (15 members, 160 km + 100 km horizontal resolution (atmosphere + ocean), period 2010-2020):



effect size between the ensembles based on:





test outcome per variable and season/region

minimize

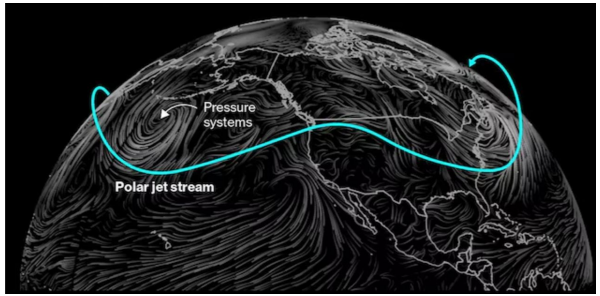
5. Scientific evaluation

Scientific evaluation: concept

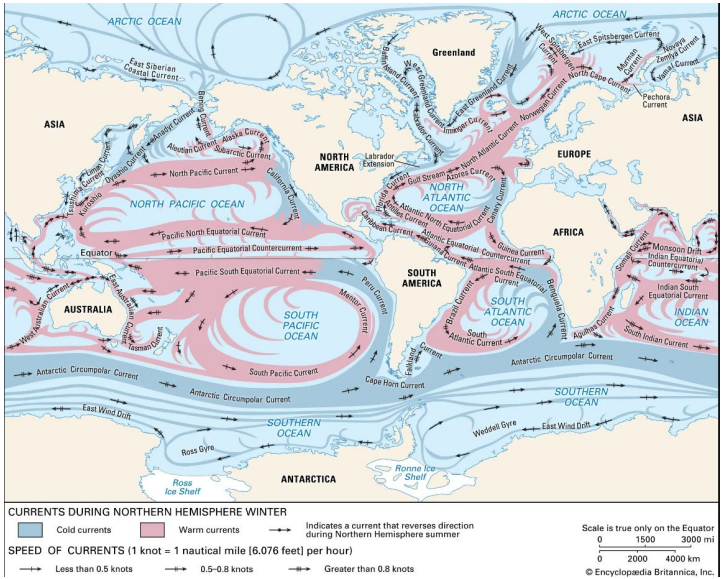
The package includes scientific skill evaluation of underrepresented climate phenomena.

By **scientific skill** we refer to a model's ability to accurately predict various aspects of the climate system:

- Simulate observed climate patterns
- Capture complex interactions and variability
- Forecast future climate changes



[1]



[2]

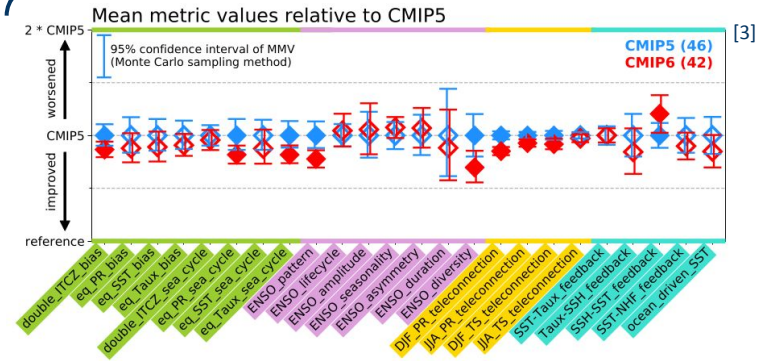
[1] L. Rosenthal & B.K. Sullivan (2024): *Severe flooding across globe linked to climate-fueled rain storms*
[2] Encyclopædia Britannica (Accessed: May 2025): *Major ocean current systems of the world*

Scientific evaluation: approach

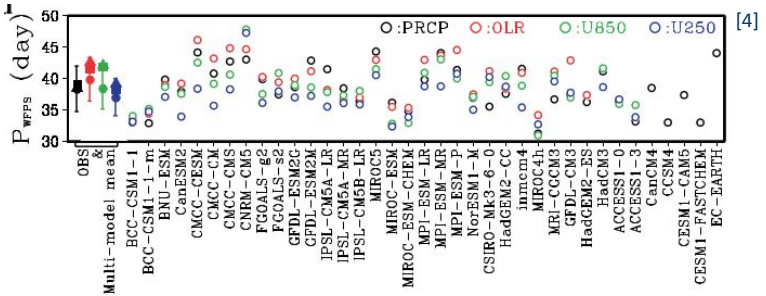
Commonly used: spatial/visual metrics → susceptible to subjective or biased interpretation

Our approach: combine standard visualizations with **scalar scores** → **quantitative** and quick evaluation

some have already been proposed



[3]



[4]

[3] Y.Y. Planton et al. (2021): *Evaluating Climate Models with the CLIVAR 2020 ENSO Metrics Package*
[4] M.-S. Ahn et al. (2017): *MJO simulation in CMIP5 climate models: MJO skill metrics and process-oriented diagnosis*

Scientific evaluation: phenomena

The package includes a growing suite of scalar scores for evaluating the following climate phenomena:

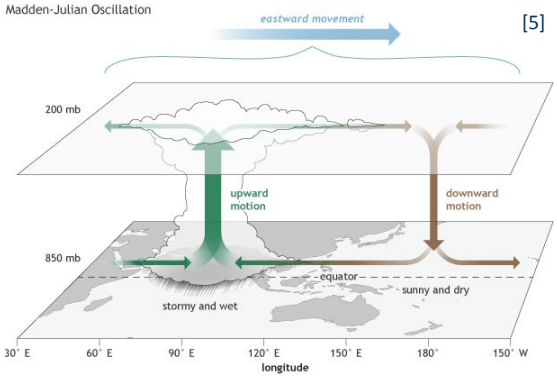
- **Tropical Intraseasonal Oscillation (ISO):**
  - **Madden-Julian Oscillation (MJO)**
  - **Boreal Summer ISO (BSISO)**

go to general ISO evaluation

go to specific MJO evaluation

- **Tropical Cyclones (TCs)**

go to TCs evaluation



[5]

[6]

[5] https://www.climate.gov/news-features/blogs/enso/what-mjo-and-why-do-we-care (Accessed: 28-04-2026)
[6] https://www.flickr.com/photos/gsfc/42828603210/ (Accessed: 28-04-2026)

Destination Earth
Funded by the European Union
Implemented by ECMWF esa EUMETSAT

Tropical Intraseasonal Oscillation (ISO)

Predominant phenomenon in the tropics throughout the year, characterized by large-scale convective anomalies that modulate tropical atmospheric circulation on 30-90 day timescales. We differentiate between two modes of ISO:

- **Madden-Julian Oscillation (MJO):** boreal winter, predominant eastward propagation along the equator
- **Boreal Summer ISO (BSISO):** boreal summer, not only eastward but also northward/northwestward propagation over the northern Indian Ocean and the western North Pacific

Evaluation workflow (applied to boreal summer and boreal winter separately [7]):

Extended Empirical Orthogonal Function (EEOF) analysis → **Bimodal ISO indices** (first two Principal Components (PCs)) → Mean monthly frequency of events (**ISO seasonality**) → **Scalar scores** comparing to observations

[7] K. Kikuchi (2020): *Extension of the bimodal intraseasonal oscillation index using JRA-55 reanalysis*

ISO: implementation

To evaluate the simulation of the ISO with **pyhanami** we need to initialize the *ScientificEvaluation* class with the *SimulationData* objects that we want to analyze and use the *compute_iso_scores* method:

1. Load simulation data:

```
import pyhanami

# Load simulation datasets
sim_1 = pyhanami.SimulationData('source_sim_1', name='name_sim_1')
```

2. Initialize evaluation class:

```
# Initialize ScientificEvaluation class with a SimulationData object
sciskill = pyhanami.ScientificEvaluation(sim_1)
```

3. Perform ISO analysis:

```
# Perform ISO analysis comparing to observations
iso_analysis = sciskill.compute_iso_scores('name_sim_1', obs=True)
```

4. Create plots with results:

```
# Plot EEFOs from observational data centered at longitude 180º
iso_analysis.eeof_plots('output_path', clon=180)
```

click to see plot

```
# Plot PCs (Bimodal ISO indices) from simulation data for specific years
iso_analysis.pc_plots('output_path', years=2000)
```

click to see plot

```
# Plot ISO seasonality for both simulation and observations
iso_analysis.freq_plot('output_path')
```

click to see plot

```
# Plot summary table with scalar scores
iso_analysis.scores_table('output_path')
```

click to see plot

go back to phenomena

ISO: implementation

To evaluate the simulation of the ISO with **pyhanami** we need to initialize the *Scientific* and

1. Lo

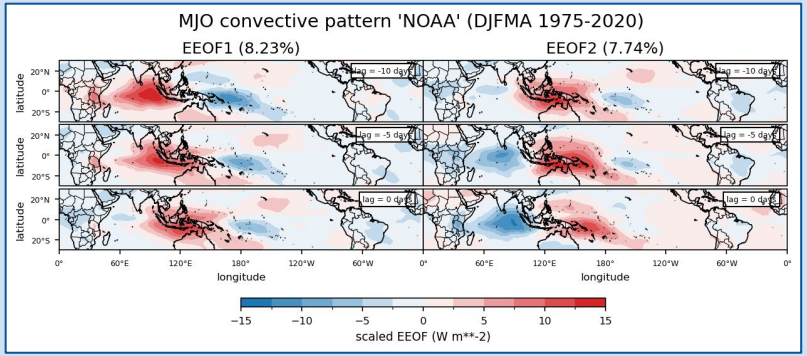
```
impo

# Lo
sim_
```

3. Pe

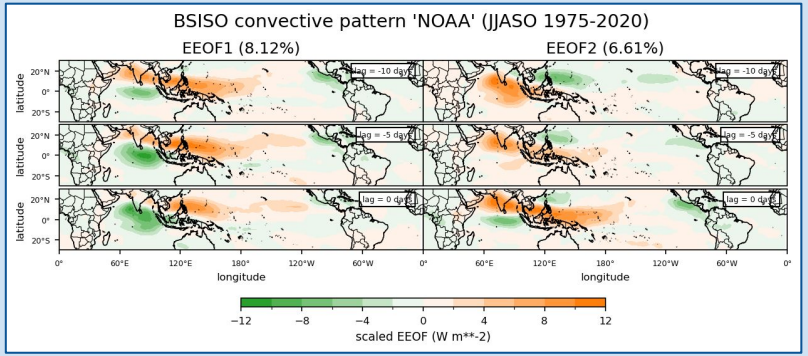
```
# Pe
iso_
```

Use case: **EEOFs** computed from NOAA observational data centered at longitude 180º (period 1975-2020):

Boreal winter



Boreal summer



minimize

```
iso_analysis.scores_table('output_path')
```

click to see plot

go back to phenomena

ISO: implementation

To evaluate
ScientificE
and use t

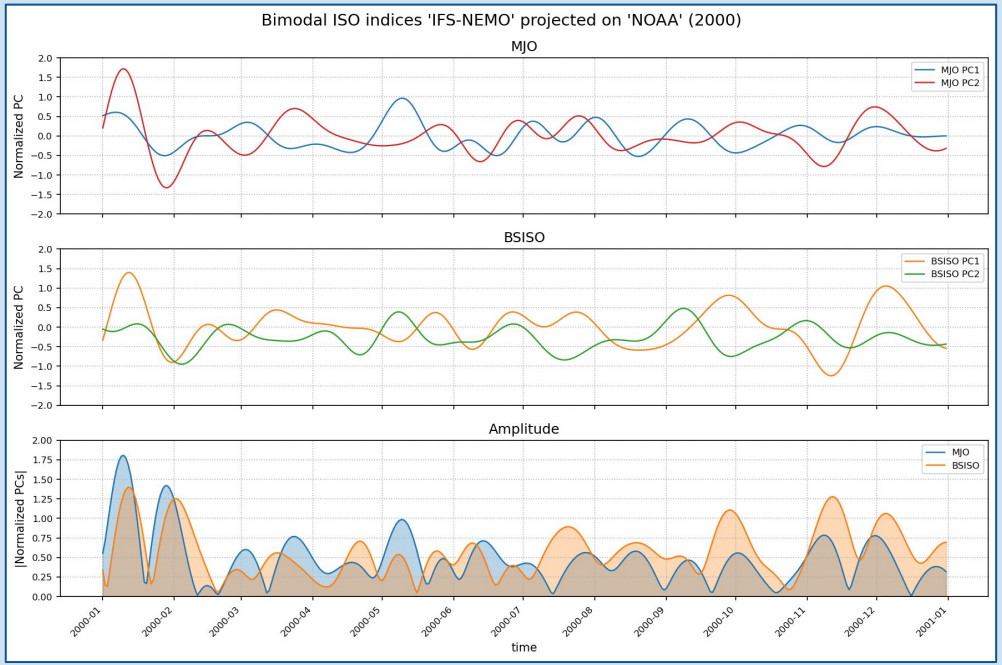
1. Load simu

```
import pyhanar

# Load simulat
sim_1 = pyhana
```

3. Perform IS

```
# Perform ISO
iso_analysis =
```

Use case: **PCs** for a single member IFS-NEMO simulation (10 km horizontal resolution, year 2000):



minimize

ct

ic years

phe

ISO: implementation

To evaluate
ScientificE
and use t

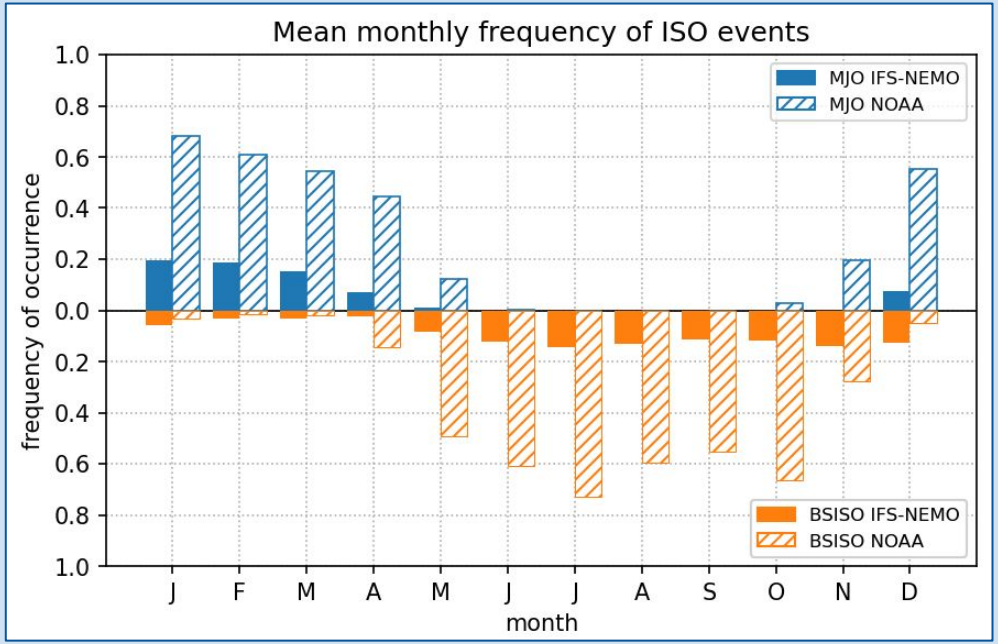
1. Load simu

```
import pyhana

# Load simula
sim_1 = pyhan
```

3. Perform I

```
# Perform ISO
iso_analysis
```

Use case: **ISO seasonality** for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):



minimize

ISO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhanar

# Load simulat
sim_1 = pyhana
```

3. Perform IS

```
# Perform ISO
iso_analysis =
```

Use case: **scalar scores** for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):

ISO scalar scores (1990-2014)

	α	R	σ	TSS
NOAA	1.00	1.00	1.00	1.00
IFS-NEMO	0.54	0.88	0.21	0.12

Worse performance — Better performance

Note:

α = PC's amplitude ratio,

σ = ISO seasonality standard deviations ratio,

R = ISO seasonality temporal correlation

TSS = ISO seasonality Taylor Skill Score

minimize

```
ct

ic years
```

Madden-Julian Oscillation (MJO)

Specific evaluation for MJO based on the Real-Time Multivariate MJO (RMM) indices [8] and wavenumber-frequency power spectrum analyses [9].

Evaluation workflows:

1. Combined Empirical Orthogonal Function (CEOF) analysis → **RMM indices** (first two Principal Components (PCs)) → Lead-lag correlation between RMM indices / MJO activity per phase (amplitude and active days) → **Scalar scores** comparing to observations

2. **Power spectrum** analysis → Focus on the MJO band → **Scalar scores** comparing to observations

[8] M.C. Wheeler et al. (2004): *An All-Season Real-Time Multivariate MJO Index: Development of an Index for Monitoring and Prediction*
[9] M. Wheeler et al. (1999): *Convectively Coupled Equatorial Waves: Analysis of Clouds and Temperature in the Wavenumber-Frequency Domain*

MJO: implementation

To evaluate the simulation of the MJO with **pyhanami** we need to initialize the *ScientificEvaluation* class with the *SimulationData* objects that we want to analyze and use the *compute_mjo_scores* method:

1. Load simulation data:

```
import pyhanami

# Load simulation datasets
sim_1 = pyhanami.SimulationData('source_sim_1', name='name_sim_1')
```

2. Initialize evaluation class:

```
# Initialize ScientificEvaluation class with a SimulationData object
sciskill = pyhanami.ScientificEvaluation(sim_1)
```

3. Perform MJO analysis:

```
# Perform MJO analysis comparing to observations
mjo_analysis = sciskill.compute_mjo_scores('name_sim_1')
```

4. Create plots with results related to CEOF analysis:

```
# Plot CEOFs
mjo_analysis.ceof_plots('output_path')
```
click to see plot

```
# Plot lead-lag correlation between RMM indices
mjo_analysis.lead_lag_corr_plot('output_path')
```
click to see plot

```
# Plot MJO activity per phase
mjo_analysis.activity_per_phase_plots('output_path')
```
click to see plot

```
# Plot summary tables with scalar scores
mjo_analysis.ceof_corr_table('output_path')
mjo_analysis.ceof_bias_table('output_path')
mjo_analysis.activity_per_phase_bias_tables('output_path')
```
click to see plot click to see plot

5. Create plots with results related to power analysis:

```
# Plot power spectrum
mjo_analysis.power_spectrum_plots('output_path')
```
click to see plot

```
# Plot summary table with scalar scores
mjo_analysis.power_bias_table('output_path')
```
click to see plot

go back to phenomena

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhanar

# Load simulat
sim_1 = pyhana
```

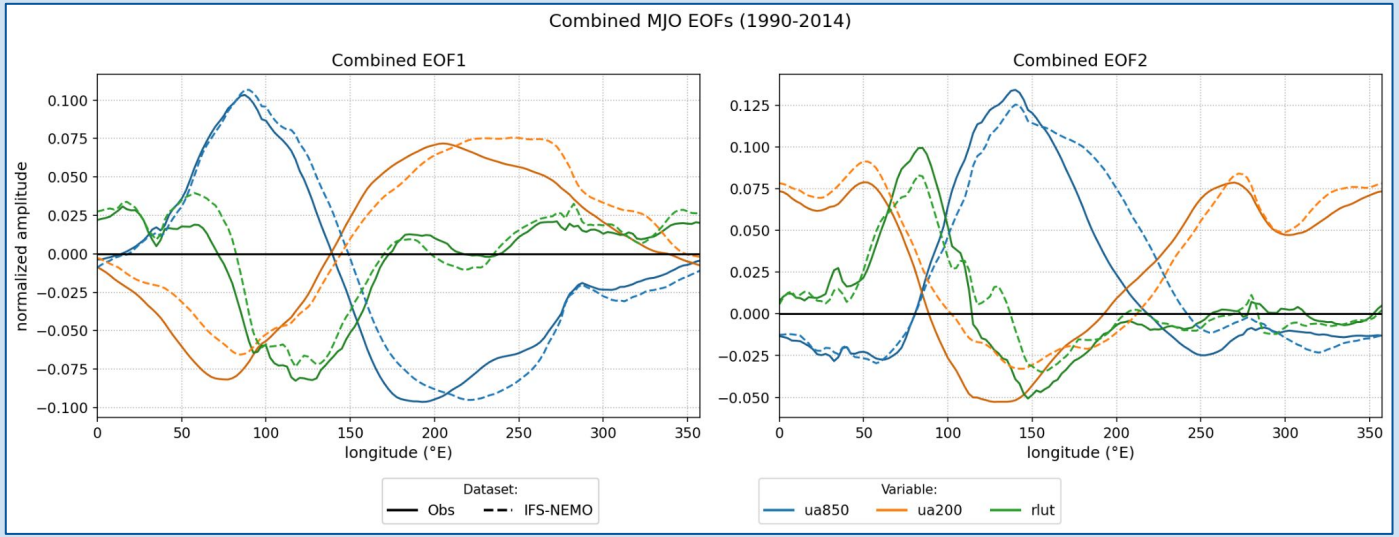
2. Initialize e

```
# Initialize S
sciskill = pyh
```

3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **CEOFs** for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):



minimize

sis:

sis:

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhanar

# Load simula
sim_1 = pyhana
```

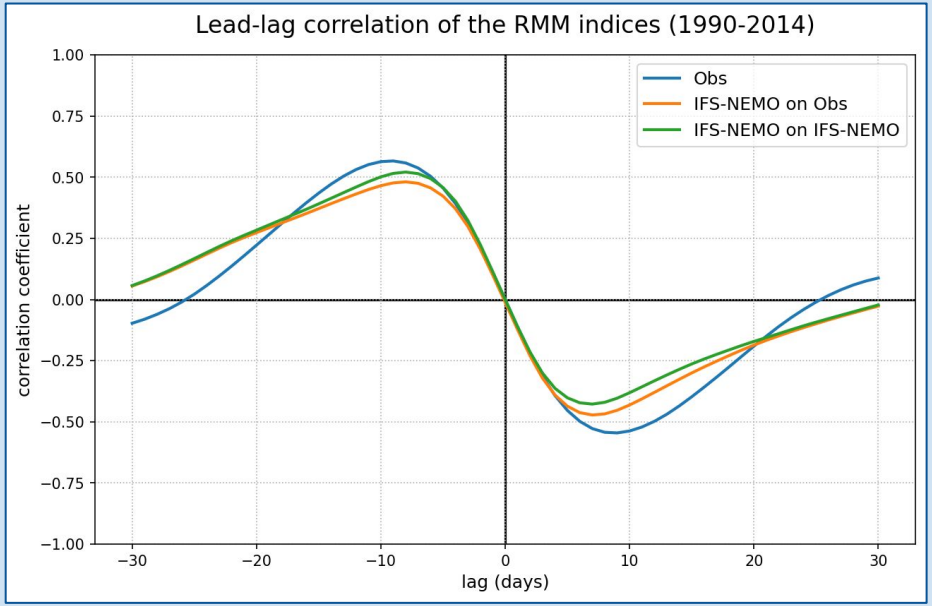
2. Initialize e

```
# Initialize S
sciskill = pyh
```

3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **lead-lag correlation** between RMM indices for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):



minimize

sis:

sis:

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhana

# Load simula
sim_1 = pyhana
```

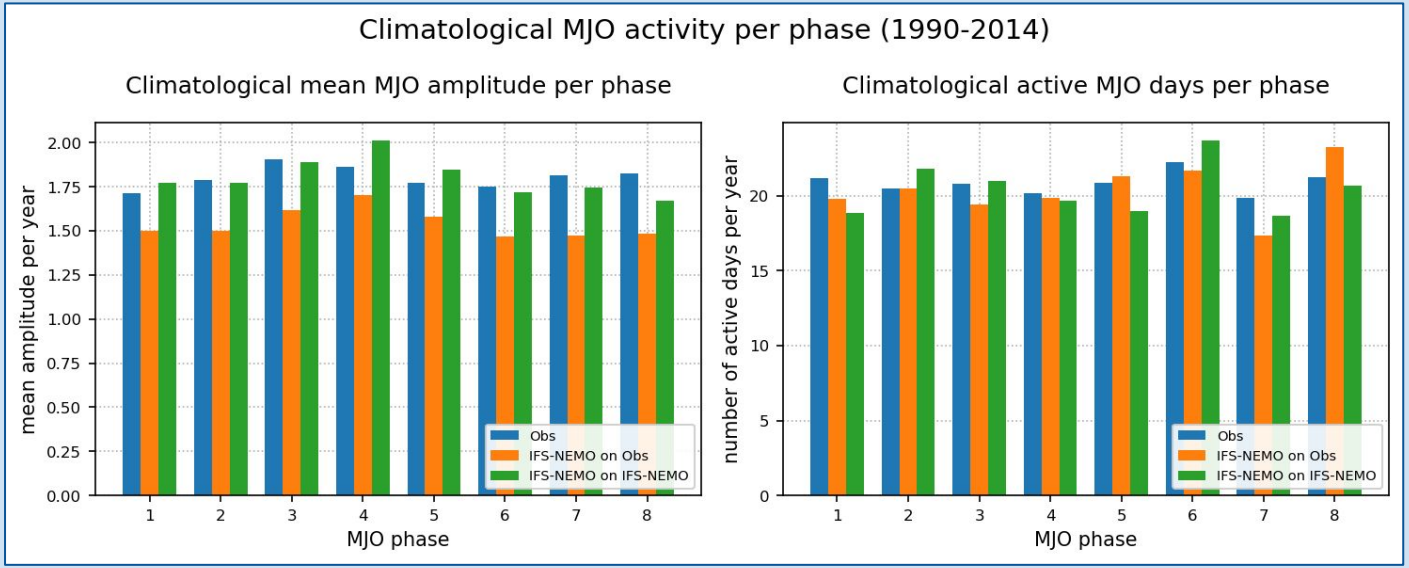
2. Initialize e

```
# Initialize S
sciskill = pyh
```

3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **MJO activity per phase** for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):



minimize

sis:

sis:

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhanar

# Load simula
sim_1 = pyhana
```

2. Initialize e

```
# Initialize S
sciskill = pyh
```

3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **scalar scores** related to the CEOF analysis for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):

Correlation between Combined EOFs (1990-2014)

2.5° x 2.5°	$r_{ua850,0}$	$r_{ua850,1}$	$r_{ua200,0}$	$r_{ua200,1}$	$r_{rlut,0}$	$r_{rlut,1}$
Obs	1.00	1.00	1.00	1.00	1.00	1.00
IFS-NEMO	0.97	0.93	0.96	0.95	0.96	0.94

Negative correlation (-1) — No correlation (0) — Positive correlation (1)

Bias derived from Combined EOF analysis (1990-2014)

2.5° x 2.5°	$b_{expl\ var,0}$ (%)	$b_{expl\ var,1}$ (%)	$b_{max\ lead\text{-}lag\ corr}$ (-)	$b_{MJO\ period}$ (days)
Obs	13.67	13.00	0.56	36.00
IFS-NEMO on Obs	0.00	0.00	-0.08	-6.00
IFS-NEMO on IFS-NEMO	-3.94	-2.67	-0.08	-6.00

Negative bias — No bias — Positive bias

minimize

sis:

sis:

click to see plot

METSAT

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhanar

# Load simulat
sim_1 = pyhana
```

2. Initialize e

```
# Initialize S
sciskill = pyh
```

3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **scalar scores** related to MJO activity for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):

Bias in climatological MJO activity per phase (1990-2014)

Bias in Climatological mean amplitude	$\overline{b}_{ph\,1}$	$\overline{b}_{ph\,2}$	$\overline{b}_{ph\,3}$	$\overline{b}_{ph\,4}$	$\overline{b}_{ph\,5}$	$\overline{b}_{ph\,6}$	$\overline{b}_{ph\,7}$	$\overline{b}_{ph\,8}$
Obs	1.71	1.79	1.90	1.86	1.77	1.75	1.82	1.83
IFS-NEMO on Obs	-0.21	-0.29	-0.29	-0.16	-0.19	-0.28	-0.34	-0.34
IFS-NEMO on IFS-NEMO	0.06	-0.02	-0.01	0.15	0.08	-0.03	-0.07	-0.15

Bias in Climatological active days	$\overline{b}_{ph\,1}$ (days)	$\overline{b}_{ph\,2}$ (days)	$\overline{b}_{ph\,3}$ (days)	$\overline{b}_{ph\,4}$ (days)	$\overline{b}_{ph\,5}$ (days)	$\overline{b}_{ph\,6}$ (days)	$\overline{b}_{ph\,7}$ (days)	$\overline{b}_{ph\,8}$ (days)
Obs	21	21	21	20	21	22	20	21
IFS-NEMO on Obs	-1	0	-1	-0	0	-1	-3	2
IFS-NEMO on IFS-NEMO	-2	1	0	-1	-2	1	-1	-1

Negative bias — No bias — Positive bias

minimize

sis:

sis:

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhana

# Load simula
sim_1 = pyhana
```

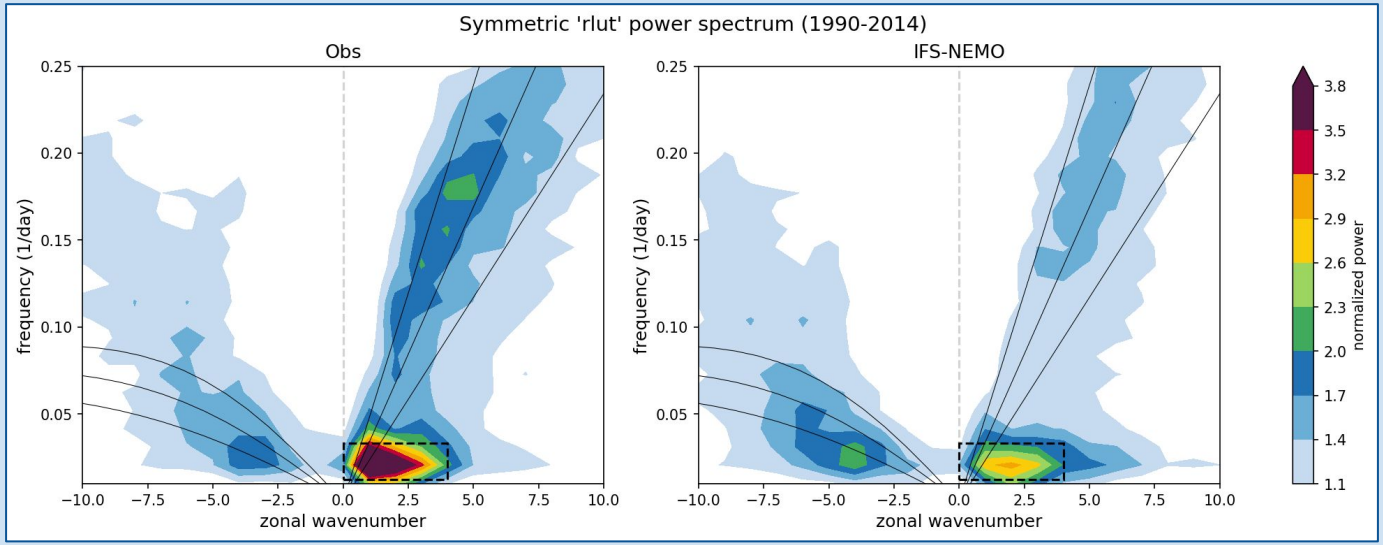
2. Initialize e

```
# Initialize S
sciskill = pyh
```

3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **power spectrum** of outgoing longwave radiation at the top of the atmosphere ('rlut') for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):



minimize

sis:

sis:

MJO: implementation

To evaluate
ScientificE
and use t

1. Load simu

```
import pyhanar

# Load simula
sim_1 = pyhana
```

2. Initialize e

```
# Initialize S
sciskill = pyh
```

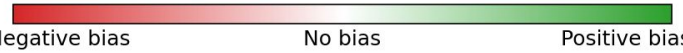
3. Perform M

```
# Perform MJO
mjo_analysis =
```

Use case: **scalar scores** related to the power analysis for a single member IFS-NEMO simulation (10 km horizontal resolution) compared to NOAA observational data (period 1990-2014):

Bias derived from power spectra (1990-2014)

2.5° x 2.5°	$b_{ua850}^{E/W}$	$b_{ua200}^{E/W}$	$b_{rlut}^{E/W}$	$b_{ua850}^{E/O}$	$b_{ua200}^{E/O}$	$b_{rlut}^{E/O}$	b_{ua850}^{period} (days)	b_{ua200}^{period} (days)	b_{rlut}^{period} (days)
Obs	1.93	1.92	1.80	1.00	1.00	1.00	41.90	41.67	41.39
IFS-NEMO	-0.34	-0.41	-0.44	-0.17	-0.22	-0.27	-0.58	-0.39	-0.33



minimize

sis:

sis:

Tropical Cyclones (TCs)

Warm-core, cyclonic storms associated with heavy precipitation and strong winds that begin over tropical oceans. They can vary in speed, size and intensity.

Evaluation workflow:

TCs trajectories → TCs metrics[10] → TCs **scalar scores**[10]

Metrics:

- Annual/monthly frequency (counts)
- Tropical Cyclone days (TCD)
- Accumulated cyclone energy (ACE)
- Pressure ACE (PACE)
- Latitude of lifetime-maximum intensity (LMI)

for each one →

Scalar scores:

- **Global climatological mean bias**
- **Global storm mean bias**
- **Global Spearman rank correlation**
- **Global Pearson correlation**

[10] C.M. Zarzycki et al. (2021): *Metrics for evaluating tropical cyclones in climate data*

TCs: implementation

To evaluate the simulation of the TCs with **pyhanami** we need to initialize the *ScientificEvaluation* class with the *SimulationData* objects that we want to analyze and use the *compute_tc_scores* method:

1. Load simulation data:

```
import pyhanami

# Load simulation datasets
sim_1 = pyhanami.SimulationData('source_sim_1', name='name_sim_1')
```

2. Initialize evaluation class:

```
# Initialize ScientificEvaluation class with a SimulationData object
sciskill = pyhanami.ScientificEvaluation(sim_1)
```

3. Perform TCs analysis:

```
# Perform TCs analysis comparing to observations
tc_analysis = sciskill.compute_tc_scores('name_sim_1')
```

4. Create plots with results:

```
# Plot summary tables with scalar scores
tc_analysis.clim_bias_table('output_path')
tc_analysis.storm_bias_table('output_path')
tc_analysis.temp_corr_table('output_path')
tc_analysis.spatial_corr_table('output_path')
```

click to see plot

go back to phenomena

Destination Earth
Funded by the European Union
Implemented by ECMWF esa EUMETSAT

TCs: implementation

Use case: **scalar scores** for a single member NICAM16-7S [8] simulation (56 km horizontal resolution) compared to reference observations and reanalyses (period 1990-2010):

Global climatological mean bias (1990-2010)

5° x 5°	$\overline{b}_{clim,\,count}$ (#)	$\overline{b}_{clim,\,tcd}$ (days)	$\overline{b}_{clim,\,ace}$ (10^{-4} kn²)	$\overline{b}_{clim,\,pace}$ (10^{-4} kn²)	$\overline{b}_{clim,\,lmi}$ (° lat.)
IBTrACS	107.7	851.9	772.3	743.7	18.6
ERA5	-26.9	-329.0	-499.7	-347.1	5.0
JRA-55	-36.5	-377.3	-543.9	-458.8	4.9
NICAM16-7S	-32.3	-282.8	-408.1	-219.2	3.3

Negative bias — No bias — Positive bias

Global seasonal correlation (1990-2010)

5° x 5°	$\rho_{s,\,count}$	$\rho_{s,\,tcd}$	$\rho_{s,\,ace}$	$\rho_{s,\,pace}$	$\rho_{s,\,lmi}$
IBTrACS	1.0	1.0	1.0	1.0	1.0
ERA5	0.9	0.8	0.9	0.9	0.8
JRA-55	0.9	0.9	0.9	0.9	0.9
NICAM16-7S	0.9	1.0	0.9	0.8	1.0

Negative correlation — No correlation — Positive correlation

minimize

[8] C. Kodama et al. (2021)

6. Ongoing and future features

Ongoing and future features

1. Additional climate phenomena (e.g., precipitation)
2. Intake data catalog (standardized data interface)
3. Integration into generic workflow:

Run simulations

EC-EARTH

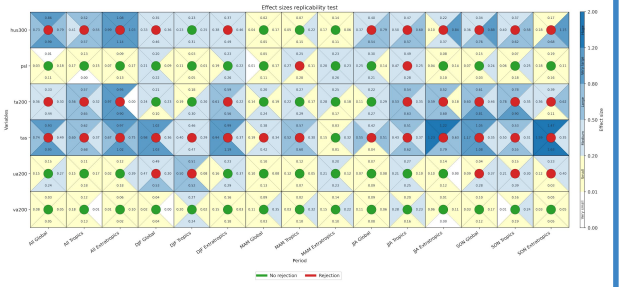
NICOCO

HANAMI workflow

+

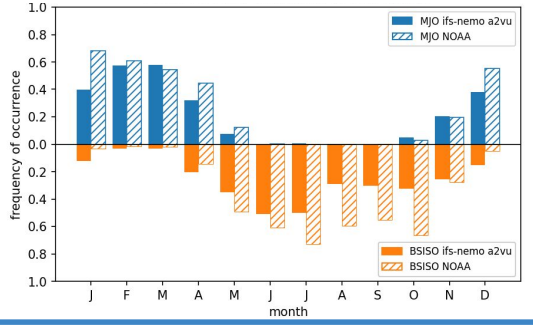
Apply diagnostics

Replicability test



+

Scientific evaluation



pyhanami Python package