HANAMI

HPC ALLIANCE FOR APPLICATIONS AND SUPERCOMPUTING INNOVATION: THE EUROPE – JAPAN COLLABORATION

Funded by the European Union

EuroHPC Joint Undertaking

This project received funding from the European High Performance Computing Joint Undertaking (EuroHPC JU) under the European Union's Horizon Europe framework program for research and innovation and Grant Agreement No.101136269. Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or EuroHPC Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

DELIVERABLE D4.1

Development and deployment of the HANAMI Earth System Model workflow and diagnostics package

Project Title	Hpc AlliaNce for Applications and supercoMputing Innovation: the Europe - Japan collaboration
Project Ref	EuroHPC International Cooperation (HORIZON-EUROHPC-JU-2022-INCO-04)
Project Acronym	HANAMI
Project Number	101139786
Type of Action	HORIZON JU Research and Innovation Actions
Topic	HORIZON JU Research and Innovation Actions
Starting Date of Project	2024-03-01
Ending Date of Project	2028-02-28
Duration of the Project	36 months
Website	http://hanami-project.com/

Work Package	4
Task	4.1
Lead Authors	Kai Keller (BSC), Marta Alerany (BSC), Leo Arriola (BSC)
Contributors	Mario Acosta (BSC)
Peer Reviewers	Sam Hatfield (ECMWF), Joachim Bierkamp (DKRZ)
Version	1.1
Due Date	31/08/2025
Submission Date	01/09/2025

Dissemination Level

x	PU: Public
	SEN: Sensitive – limited under the conditions of the Grant Agreement
	EU-RES. Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)
	EU-CON. Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)
	EU-SEC. Classified Information: SECRET UE (Commission Decision 2005/444/EC)

Disclaimer

Version History

Revision	Date	Editors	Comments
V0.9	15/08/2025	Marta Alerany, Leo Arriola, Kai Keller, Mario Acosta	First draft of the deliverable before the internal review
V1.0	28/08/2025	Marta Alerany, Leo Arriola, Kai Keller, Mario Acosta, Sam Hatfield, Joachim Bierkamp	Final Draft of the deliverable after including the comments from the interanl reviewers, before submission to the CMO
V1.1	01/09/2025	France Boillod-Cerneux	Finalize the template format for the deliverable

Executive Summary

The deliverable discusses the current status of the HANAMI Earth System Model workflow implementation and diagnostic package. In addition to the description of technical progress, we focus on the emergence of new collaborations within the HANAMI framework. Our monthly meetings and individual travels and stays, funded by HANAMI, helped establish strong and sustainable collaborations with our Japanese partners, particularly with the members of the high-resolution modeling teams at JAMSTEC, NIES, and Riken-RCCS.

Table of Contents

1. Introduction 7

2. The HANAMI ESM workflow 9
2.1 The Autosubmit workflow manager 9
2.2 The HANAMI workflow implementation 10
2.3 Cluster access and job submission 12
2.3.1 Fugaku 12
2.3.2 Miyabi 13
2.3.3 Marenostrum 5 13
2.4 Status of the model deployment 13
2.5 Workflow features and ongoing work 14
3. The HANAMI diagnostics package 14
3.1 Visualization diagnostics 15
3.2 Replicability 16
3.3 Scientific skill metrics 19
3.4 Structure of pyhanami 21

Table of Figures

Figure 1 - Technical setup used to set up the Autosubmit workflow manager. 10

Figure 2 - Structure of the jobs that compose the HANAMI workflow and their dependencies, including the platforms where these jobs run. 10
Figure 3 - Time series plot generated with pyhanami. It includes the evolution of the air surface temperature from 1990 to 2014 for two different simulation datasets of LENS2. 15
Figure 4 - Spatial plots generated with pyhanami. They depict the absolute difference (left) and effect size together with significant differences (right) for values of air surface temperature from 1990 to 2014 between two different simulation datasets of LENS2. 16
Figure 5 - Interface of the HANAMI Shiny app displaying bias plots for the meridional wind output by double- and single-precision versions of IFS-NEMO. 16
Figure 6- Matrix plot displaying the outcome of the replicability test applied to compare a double-precision vs a single-precision version of IFS-NEMO. 18
Figure 7 - Results of the climatological bias (left) and spatial correlation (right) evaluations of tropical cyclones comparing NICAM16-7S data (last row) to observational datasets for the period 1990-2010..21

1. Introduction

In project 2 / task 4.1 of Work Package (WP) 4, we aim to develop an automatic workflow for testing three aspects of Earth System Models (ESM):

- Replicability,
- Projection skill, and
- Computational performance

Most model developers have their own kit of diagnostics and tests that are applied to gather information on those three aspects. The knowledge of how to obtain sensible results from specific tests often lies at individual sites and is rarely shared. However, there are robust methodologies available and widely accepted by the community. We aim to combine those methodologies inside a single workflow with a generic interface applicable to any weather and climate model and accessible on any cluster.

The workflow will be implemented using the Autosubmit workflow manager. Autosubmit is designed to meet climate research necessities. Unlike other workflow solutions in the domain, it integrates the capabilities of an experiment manager, workflow orchestrator and monitor in a self-contained application. The experiment manager allows for defining and configuring experiments, supported by a hierarchical database that ensures reproducibility and traceability. The orchestrator is designed to run complex workflows in research and operational mode by managing their dependencies and interfacing with local and remote hosts. These multi-scale workflows can involve from a few to thousands of steps and from one to multiple platforms. Autosubmit facilitates easy and fast integration and relocation to new platforms.

The work presented in this deliverable would not have been possible without the profound collaboration with our Japanese partners. Within the first 18 months of HANAMI, we built strong bonds, especially with Dr. Hisashi Yashiro from the National Institute for Environmental Studies (NIES), Dr. Chihiro Kodama from the Japan Agency for Marine-Earth Science and Technology (JAMSTEC), and Dr. Hirofumi Tomita, the Computational Climate Science Research Team leader at the RIKEN Advanced Institute for Computational Science. Leo Arriola from the Barcelona Supercomputing Center (BSC) stayed 5 weeks at NIES, working together with Dr. Hisashi Yashiro on porting NICOCO to the HANAMI workflow. NICOCO is a coupled atmosphere and ocean ESM. The model name, NICOCO, is a combination from the Nonhydrostatic ICosahedral Atmospheric Model (NICAM) and the ocean model COCO (former acronym of CCSR (Center for Climate System Research) Ocean Component Model). Marta Alerany from the BSC stayed 6 weeks at JAMSTEC working closely together with researchers at JAMSTEC identifying scientific performance metrics and diagnostics for high-resolution climate models. Based on these collaborations, we have successfully applied to a JHPCN project call, together with our Japanese partners, and have been awarded a 1-year project including Miyabi-C and Miyabi-G computing resources. Further, we applied together to the Fugaku general access call, where we hope to perform large-scale ensemble experiments to support the developments of the HANAMI workflow, if awarded.

The remainder of the deliverable reports about the status of the implementation and a brief explanation of the developments. In Section X we explain the foundation of the HANAMI workflow and its features. In Section X, we describe the replicability test that we have developed and report about the scientific performance skills and diagnostics developed within HANAMI.

2. The HANAMI ESM workflow

The HANAMI workflow is an Autosubmit-based workflow that allows running climate simulations of ESMs on HPC platforms. Autosubmit-based workflows provide high degrees of customization, as the Autosubmit workflow manager combines experiment management capabilities with typical HPC climate workflow management functionalities.

At its current stage, the workflow fully supports the compilation and running of the Japanese NICOCO coupled model (NICAM and COCO Coupled) on Marenostrum5 and Fugaku. The workflow can compile and run climate simulations on both platforms. In addition, the workflow also supports running the dummy Lorenz96 model, which was used as an initial proof of concept. Furthermore, building on prior work to port EC-Earth4 to Fugaku, initial implementations of the workflow also supported EC-Earth4 and its compilation. However, due to rapid development of EC-Earth4, the efforts on maintaining the compatibility with Fujitsu compilers on Fugaku was stopped and NICOCO was prioritized.

2.1 The Autosubmit workflow manager

Autosubmit is a Python-based open source application that can conveniently be installed as a pip package. Moreover, it can be easily containerized and used in cloud environments, or installed inside Conda environments. This flexibility allows Autosubmit to be readily used in most HPC environments.

It is recommended to deploy Autosubmit on a virtual machine (VM) with passwordless access to the HPCs. Users can then connect via ssh to the Autosubmit VM and submit workflows with unique IDs, while Autosubmit orchestrates the resulting workflow jobs on the targeted HPC by communicating with the batch schedulers. This setup is represented in Figure 1. In HANAMI, we use the Autosubmit VM hosted at the Barcelona Supercomputing Center. The advantage of this arrangement is a centralized Autosubmit experiment (workflow) database. All experiments from any user are assigned a unique ID, which allows one to copy experiments and reproduce their results. This is represented by the colored matrix in the VM, which represents the different experiments or instances of the workflows.

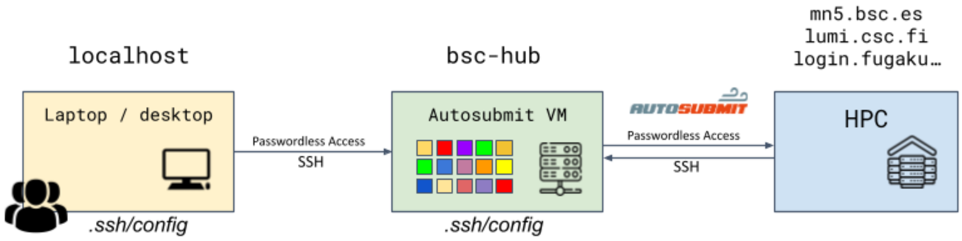

Figure 1 - Technical setup used to set up the Autosubmit workflow manager.

In Autosubmit, workflows can be conceived as directed graphs of elements of a set of interdependent jobs which run once or multiple times on different platforms. For each job, a script must be provided, that Autosubmit will invoke to process the job's task. This can include the submission of the climate simulation, the compilation of the climate model, initial condition perturbations, etc. In addition to the script, the job definition must also include on what platform it should run; locally, on an HPC, on a VM, or any platform that is required.

All the basic workflow configuration (i.e, jobs, platforms, etc.) are managed by YAML files, which Autosubmit parses to generate the final run-scripts. It then uses job states to internally monitor and operate the jobs, whether for re-submitting a failed job or for running the subsequent children jobs. In addition to these basic workflow definitions, Autosubmit also includes basic parameters which are more climate focused. For instance, the start and end dates of the simulation jobs, the number of simulation chunks (time divisions of long-term climate simulations), and the number of members.

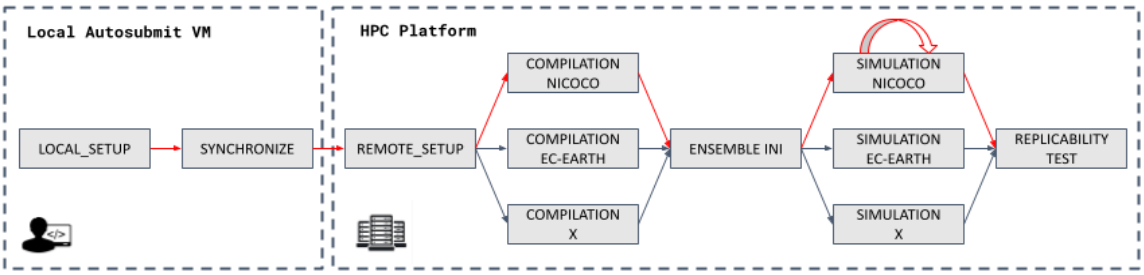

Figure 2 - Structure of the jobs that compose the HANAMI workflow and their dependencies, including the platforms where these jobs run.

2.2 The HANAMI workflow implementation

The design principles that have been taken into consideration during the HANAMI workflow development have been portability, flexibility, and the capacity to "plug in" with ease additional climate models. In order to achieve this, the workflow provides a generic interface which allows orchestrating common necessary jobs for all models. These include steps like synchronizing the model and workflow sources between the different platforms, and performing analyses on standardized output datasets. Model-specific jobs are embedded into the workflow, while the dependency flow remains model-agnostic (see Figure 2). In Table 1, we provide a brief description of each job in the current version of the HANAMI workflow.

HANAMI Workflow jobs		
Job name	Job description	Platform
LOCAL_SETUP	Allows cloning the model sources. Compresses the whole workflow codebase.	Autosubmit VM
SYNCHRONIZE	Uses rsync to synchronize the workflow and model sources from the VM to the remote platform.	Autosubmit VM
REMOTE_SETUP	Uncompresses the workflow and model sources. Creates the necessary folder structure and also allows cloning model sources if necessary.	HPC Platform
COMPILATION_X	Compiles the different models and allows controlling multiple parameters.	HPC Platform
ENSEMBLE_INI	Will include IC perturbations for running ensembles. Work in progress.	HPC Platform
SIMULATION_X	Runs a chunk of the model simulation and modifies model namelists accordingly.	HPC Platform
REPLICABILITY_TEST	Will include running the pyhanami replicability tests. Work in progress.	TDB

Table 1 - Description of the jobs that compose the HANAMI workflow.

Expanding on the “plug-in” concept, the model and platform-dependent jobs are isolated as much as possible. Each model has its own compilation and simulation job, plus additional support libraries. This separation allows the workflow to have a generic interface; one only needs to provide a few common paths and configuration files in order to integrate a new model or platform into Autosubmit. In order to achieve this generic interface, the workflow provides a set of YAML configuration files which allow for users to create the desired workflow structure. These files provide the generic jobs with the context for each combination of platform, model, mode that is desired.

Currently, the configuration for the workflow is subdivided by each model. Users can define any YAML fields which they see fit and might require for operating the model. Although this file structure is flexible, at the moment, for each model there are files for each platform-simulation combination. The number of nodes, processes, job time limits, compilation modes and more can be found there. In addition to these files, there are more generic and general configuration files which include paths and platform definitions which are required for the workflow.

All these different configuration files represent different modes and contexts for different types of runs. They specify what combination of options are required for a specific run. The way that the workflow and Autosubmit know which option to run is controlled through the main.yml and minimal.yml files, which are not in the workflow repository but rather are directly created in the Autosubmit VM. The loading of the configuration is hierarchical, meaning that some files have more priority than others, and are loaded before. In this sense, the main.yml and minimal.yml files have higher priority than configuration files that are in the workflow repository, and therefore act as "switches" which load the different configuration files for the different modes of running.

```
# Main configuration file for climate simulation workflow
# This file controls the execution environment, compilation settings, and experiment parameters

RUN:
  MODEL: "lorenz96"     # Options: "nicoco" (NICAM-COCO coupled model), "lorenz96" (Lorenz-96 model)
  PLATFORM: "fugaku"    # Options: "fugaku" (RIKEN), "marenostrum5" (BSC)
  SIMULATION: "test"    # Type of simulation to perform. Currently only "test" mode is supported

COMPILATION:
  METHOD: "precompiled" # Options: "precompiled", "source"
  ROOT_PATH: "/path/to/binaries"  # Required when METHOD is "precompiled"
  SOURCE_CONFIG:        # Only used when METHOD is "source"
    CLONE_ON: "local"   # Options: "local", "remote"
    REPO: "https://github.com/example/lorenz96"  # Git repository URL
    BRANCH: "main"      # Git branch to checkout
    COMMIT: "abc123def456"  # Git commit hash for exact version control

EXPERIMENT:
  DATELIST: 19500101    # Start date of simulation in YYYYMMDD format
  MEMBERS: fc0          # Ensemble member identifier
  CHUNKSIZEUNIT: day    # Time unit for chunking the simulation: "day", "month", "year"
  CHUNKSIZE: 1          # Size of each chunk in CHUNKSIZEUNIT
  NUMCHUNKS: 2          # Number of chunks to process: Total simulation length = CHUNKSIZE *
NUMCHUNKS
  CALENDAR: standard    # Calendar type for date calculations
```

Listing 1 - Example of a main.yml file for the HANAMI workflow.

In Listing 1 various "switch" parameters are shown. As described, these represent the user interface, meaning that the user only has to modify these high-level parameters to then load all the underlying configuration files.

2.3 Cluster access and job submission

Porting workflows to different HPC platforms often involves significant adaptation. These adaptations are mainly due to variations in both hardware architectures and software environments, including the type of job scheduler employed and system-specific authentication mechanisms.

2.3.1 Fugaku

Fugaku is a Fujitsu supercomputer built on the ARM architecture, specifically the Fujitsu A64FX microprocessor, which features 48 cores per node and uses Scalable Vector Extensions (SVE). It consists of 158,976 nodes connected by Tofu interconnect. Giving that it is a Fujitsu machine, it requires the use of Fujitsu compilers for code compilation. Unlike many European systems, it does not use the Slurm scheduler; instead, it employs a custom scheduler that is operated through commands such as pjsub and pjstat. To support Fugaku, additional developments were necessary within Autosubmit to enable proper communication with this non-standard scheduler.

2.3.2 Miyabi

Miyabi is an Intel/Nvidia-based machine, similar in architecture to MareNostrum5. Miyabi-G (accelerator-node group) consists of 1,120 nodes of NVIDIA GH200 Grace Hopper superchips, and Miyabi-C (general-purpose CPU-node group) consists of 190 nodes of two Intel Xeon CPU Max 9480. It uses InfiniBand NDR to interconnect the different nodes. However, it presents significant differences and challenges regrading its authentication mechanism: it uses two-factor authentication (2FA) based on one-time passwords (OTP) for SSH logins. This requirement complicates the automation process, as Autosubmit typically assumes non-interactive, passwordless access to the target platform.

While some features within Autosubmit have been designed to partially support such cases, their functionality remains limited. One possible workaround under consideration involves installing Autosubmit directly on Miyabi's login nodes, thereby circumventing the 2FA process. However, this approach introduces new operational considerations and is still under evaluation.

Additionally, Miyabi uses a proprietary job scheduler based on commands such as qsub, similar to Fugaku's departure from Slurm. As a result, further enhancements to Autosubmit may be required to fully support job submission and monitoring on this system.

2.3.3 Marenostrum 5

Marenostrum5 is also an Intel/Nvidia-based machine. The Marenostrum 4 GPP (general purpose partition, cpu-based nodes) includes 6,408 nodes based on Intel Sapphire Rapids (4th generation Intel Xeon Scalable processors) plus an additional 72 nodes featuring Intel Sapphire Rapids but with HBM (High Bandwith Memory). The Marenostrum 5 ACC (Accelerated Partition) is composed of 1,120 nodes based on the combination of Intel Xeon Sapphire Rapids processors and NVIDIA Hopper GPUs, offering a total (CPUs+GPUs) of 680,960 compute units. In contrast to Miyabi, it uses Slurm as the job scheduler. As it is our own in-house computer, Autosubmit runs natively on MareNostrum5.

2.4 Status of the model deployment

The models that have been worked on and ported to the workflow during the development of the HANAMI have been: NICOCO (NICAM & COCO), EC-EARTH4, and the dummy model Lorenz96.

Currently, NICOCO can be compiled and executed on both Marenostrum5 and Fugaku. Due to the model's requirements, it relies on personal Spack instances to load the software environment. While functional, this approach introduces some issues and inconsistencies. The plan is to move towards a container-based setup in order to provide a more robust and reproducible environment. Regarding performance, NICOCO generally seems to run better on Fugaku, although no profiling or scaling tests have been carried out yet.

Within the scope of HANAMI, the EC-EARTH4 model was also compiled on Fugaku. This required several changes to the compile wrappers and parts of the source code to make it compatible with the Fujitsu compilers. Although compilation was successful, the model was not fully run or tested, so this remains an open task for future work. With regard to MareNostrum5, the model has already been ported and run in the scope of other projects.

2.5 Workflow features and ongoing work

As mentioned in the other sections, the workflow currently supports compiling the NICOCO model and in order to achieve this, an adapted version of NICOCO is used. This version of NICOCO includes the compilation wrapper scripts that were created in order to compile and run on MareNostrum5. Other changes, mostly regarding the Fugaku environment, which was outdated, were added. In addition to the compilation, on both Marenostrum5 and on Fugaku, the workflow can also run the model for various resolutions. It must be mentioned that no performance profiling has been done on NICOCO, therefore the resource configuration being used is likely not the most optimum.

In order to run the NICOCO model, two scripts were developed which allow patching the different namelists. For NICAM, which uses an F90 namelist format, a Python script using the f90nml package was implemented. For COCO, which uses a DSL (domain-specific language) an additional Python script was created. The development of this namelist patcher was more arduous as COCO doesn't follow strict rules in its namelists. Using these two scripts, for each simulation job the workflow modifies the dates, restart paths and more namelist parameters that are required to run a simulation chunk.

Additionally, the workflow also allows obtaining CPMIP metrics, like simulated years per day (SYPD), actual simulated years per day (ASYPD), and others. This is calculated directly via Autosubmit, which constantly pulls information from the different runs and then calculates a set of metrics. Once these metrics are calculated, they can be retrieved through its API or visualized through the Autosubmit GUI.

The work being exposed here is still under development, and multiple features are planned or still ongoing. Here is a brief description of the various open lines of work:

- Adding multi-member support. Current work is focused on adding an automated initial condition perturbation method in the workflow for NICOCO.
- Porting the HANAMI workflow to additional platforms, including LUMI and Miyabi. We are currently porting Autosubmit to Miyabi.
- Improving environment management on different platforms by using containers and abandoning personal Spack instances.

3. The HANAMI diagnostics package

The HANAMI diagnostics package pyhanami is a pip-installable Python package envisioned as a tool to evaluate the replicability and scientific skill of ESMs. Even though the functionalities that it provides will also be included in the generic workflow explained above, its goal is to allow researchers to easily take advantage of these without the need to install and adapt the workflow to their computing environments.

The package includes several visualization diagnostics, the replicability test we developed within the HANAMI project, and a set of scalar scientific skill metrics that we are currently working on and that aim to evaluate how well a given ESM reproduces specific climate phenomena. The following subsections provide a detailed explanation of each of these capabilities.

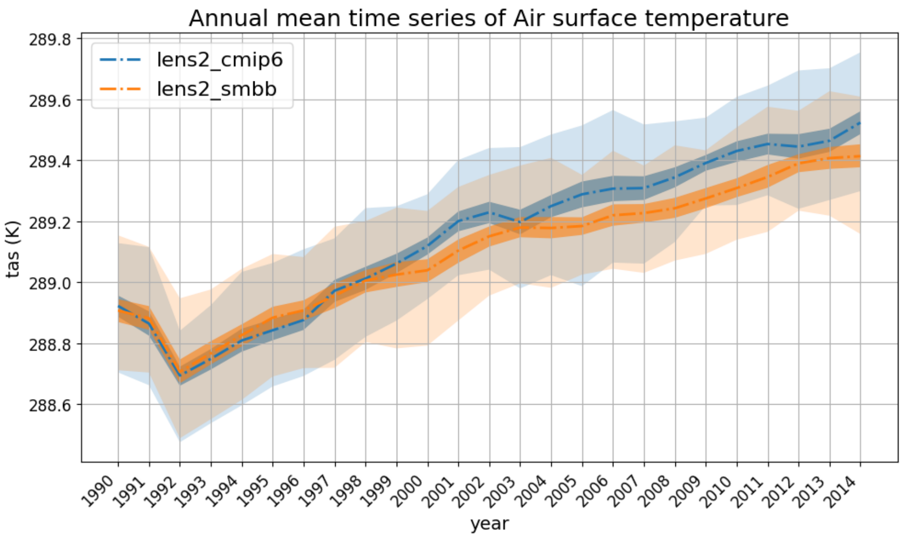

Figure 3 - Time series plot generated with pyhanami. It includes the evolution of the air surface temperature from 1990 to 2014 for two different simulation datasets of LENS2.

3.1 Visualization diagnostics

pyhanami provides a suite of visualization diagnostics designed to facilitate a first interpretation and comparison of ESMs' simulation outputs. Its features include the generation of time series plots for any number of simulation datasets, clearly displaying mean values along with percentile bands (2.5%, 5%, 95%, and 97.5%) for a certain time period and a selected climate variable. These plots can optionally include observational data for a reference, allowing users to quickly assess how accurately the simulations align with real-world measurements.

Additionally, the package incorporates tools to create spatially colored global maps that display the absolute mean difference between two simulation ensembles for a selected variable, helping to identify regional discrepancies. This is further enabled through effect size maps, which compute Cohen’s d between the ensembles. Areas where differences are statistically significant based on a t-test are highlighted in these plots, offering insights into

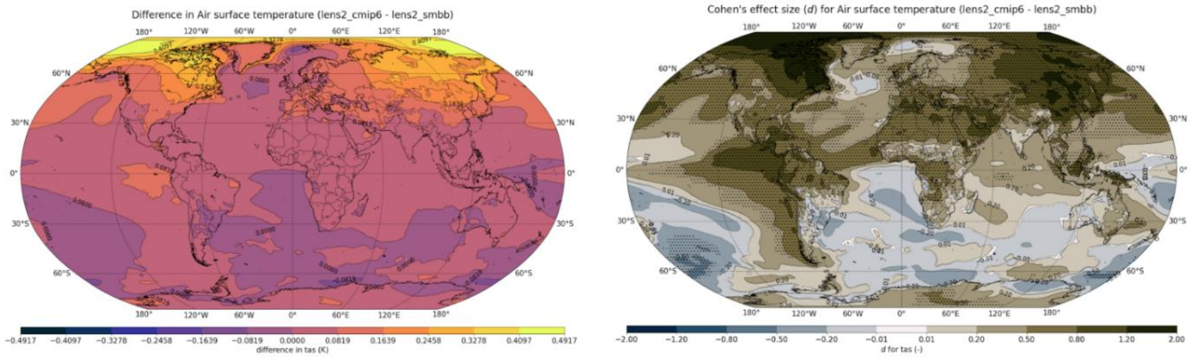

Figure 4 - Spatial plots generated with pyhanami. They depict the absolute difference (left) and effect size together with significant differences (right) for values of air surface temperature from 1990 to 2014 between two different simulation datasets of LENS2.

the spatial distribution of the observed changes. Overall, these visualization tools enable a deeper understanding of model variability across both temporal and spatial scales, serving as a first straightforward analysis before moving onto more rigorous scalar diagnostics.

Figures 3 and 4 show examples of the described plots generated with pyhanami using simulations from the Community Earth System Model (CESM) v2 Large Ensemble Community Project (LENS2).

Finally, we have created a Shiny app (https://earth.bsc.es/shiny/hanami/) where we are compiling these spatial plots, along with bias plots (comparing simulations to observations), for the simulation datasets analyzed so far (see Figure 5). Currently, it includes plots comparing different forcings of LENS2 and different floating-point precisions of IFS-NEMO (a coupled atmosphere-ocean model being developed as part of the Destination Earth initiative). Our plan is to continue adding these plots as the package is applied to new datasets. The app allows users to easily toggle between different climate variables, periods, and comparisons, facilitating the examination of these datasets in a centralized and interactive way.

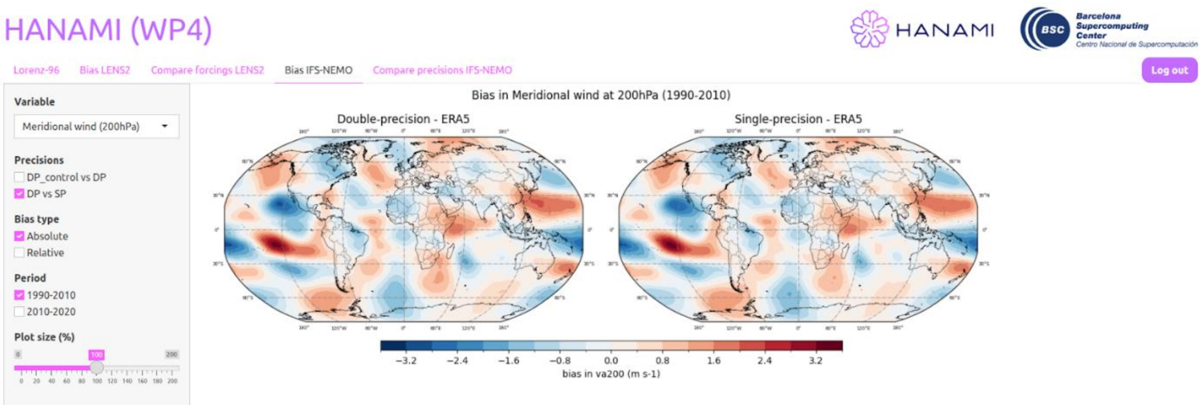

Figure 5 - Interface of the HANAMI Shiny app displaying bias plots for the meridional wind output by double- and single-precision versions of IFS-NEMO.

3.2 Replicability

We say that an ESM is replicable if performing the same experiment (with the same model and forcing) using different clusters, computing environments, or different compilers leads to identical results representing the same climate. However, bit-for-bit replicability is not feasible in practice due to the chaotic nature of this type of model. It has been shown that the nonlinearity of these models results in significantly different trajectories, even for something as trivial as a different set of compiler flags. Because of this, we ask and check for statistical indistinguishability instead, meaning that a model is replicable if there are no statistically significant differences between the ensembles generated by it using two different computing environments.

As part of HANAMI WP4, we have developed a test to verify replicability and compiled our results in a paper that is currently under review (Keller et al., 2025). Apart from introducing the structure of the test, the paper addresses the lack of a methodology for accurately assessing the performance of this type of test. We introduce a procedure to evaluate the statistical power and sensitivity of current replicability methods, and make improvements based on the results.

The replicability test that we developed and analyzed in the paper builds on the methodology presented by (F.Massonet et al., 2020) to check statistical indistinguishability between simulated ensembles. The structure of the test is the following: given two ensembles generated in two different computing environments, we assign a score to each ensemble member, resulting in two distributions of score which are then compared by combining several statistical tests, such as the Kolmogorov-Smirnov test, with the null hypothesis that both samples are drawn from the same underlying population.

When running an ESM, the surface of the Earth is represented as a grid composed of multiple cells. We use the climatologies (mean values over a span of time) of climate variables at each of these cells to compute the weights for the test. Some of the variables used are the temperature of the air in the atmosphere, the precipitation rate, the pressure at mean sea level, or the salinity of the sea surface. Directly averaging the values of these variables spatially would result in the loss of important information regarding regional differences among the ensemble members. Due to this, we first apply various metrics at the cell level, which contrast the value of one member to either the mean of the entire simulated ensemble or the mean of a common reference set of observations.

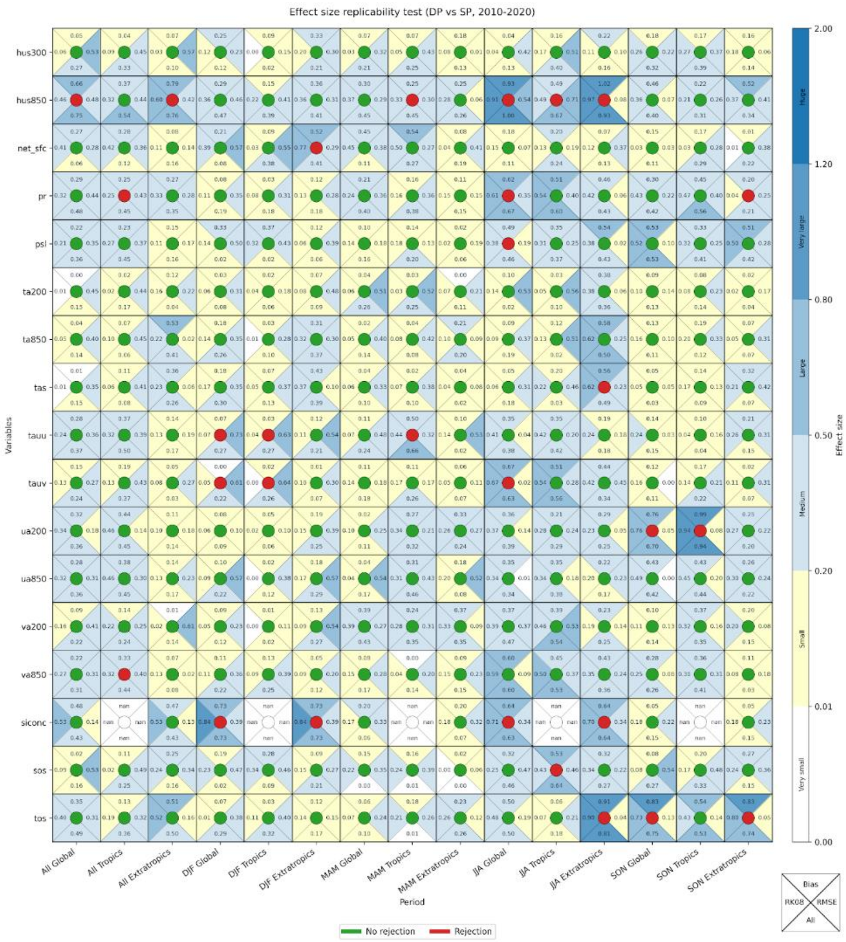

Figure 6- Matrix plot displaying the outcome of the replicability test applied to compare a double-precision vs a single-precision version of IFS-NEMO.

For our test, we combine the outcome of three different metrics and evaluate them with four distinct statistical tests, rejecting the null hypothesis whenever at least one of the tests results in a rejection. In the paper, we show that this methodology leads to a higher power and a lower false positive rate for the test. These two magnitudes represent the probability that the test will reject the null hypothesis correctly when two samples come from different populations and erroneously when the populations are the same, respectively. Our methodology is capable of resolving the differences in the data better, triggering for smaller effect sizes. We demonstrated this by applying the new version of the test to large (1000-member) ensembles generated with the simplified Lorenz96 model, and to a smaller (100-member) real simulated ensemble from LENS2.

The described replicability test is integrated into the pyhanami package. It allows users to test for replicability across a set of climate variables, distinguishing by region (global, tropics (below 30º) and extratropics (above 30º)) and season (all, DJF (December–February), MAM (March–May), JJA (June–August), and SON (September–November)). This separation facilitates the identification of more specific issues that may be obscured when averaging the simulation results globally or over the entire year. Figure 6 provides an example of how the outcome of the test is presented to the user. The plot includes the tested climate variables as rows and several combinations of regions and seasons as columns. Each square is divided in four triangles containing the effect size obtained with each of the four metrics used, and a circle in the middle displaying the results of the test (red for a rejection and green when the null hypothesis cannot be ruled out).

3.3 Scientific skill metrics

For ESMs to be reliable and serve as the scientific basis for public policy, they need to provide high confidence in future climate projections. A rigorous way to quantify this confidence involves evaluating their scientific model skill, which refers to the ability of the model to accurately represent and predict various aspects of the climate system. These include their capacity to forecast future climate changes or to capture complex patterns and relationships within the system.

Although there are already many metrics available to assess this feature of ESMs, most of them are spatial or visual metrics that are susceptible to subjective or biased interpretation, and usually require expert knowledge. This makes comparing different models and/or model versions challenging and time-consuming. We aim for a quantitative evaluation and, hence, we are compiling and implementing a robust and reliable set of scalar metrics that output a numeric score. This will allow researchers to easily contrast results across different models, platforms and locations.

We have been compiling pre-existing scalar metrics related to several phenomena, among others, El Niño-Southern Oscillation (ENSO), the Madden-Julian oscillation (MJO), or the Monsoon. We analyzed the requirements of each of these metrics in terms of variables, pre-processing of the simulated data, domain, and available observations. We found that some of these metrics were already implemented and openly available as part of the Program for Climate Model Diagnosis and Intercomparison (PCMDI) Metrics Package. However, assessment of some phenomena, such as tropical cyclones (TCs) or the Boreal Summer Intraseasonal Oscillation (BSISO), was missing. Even though several metrics or indices have been proposed for the latter, there is currently no standard implementation or package to help researchers check some crucial phenomena.

In order to identify phenomena that had been overlooked and help define new scalar metrics, especially those relevant for high-resolution climate models, Marta Alerany completed a 6-week stay at JAMSTEC in Yokohama (Japan) between May and June 2025. The stay took place in the Research Center for Environmental Modeling and Application (CEMA). It was conducted under the supervision of Chihiro Kodama, a researcher within the Cloud-Resolving Model Development and Application Group (CRM-DAG). This group focuses on high-resolution climate simulations to better understand several physical processes and their roles in severe weather events, such as typhoons, very relevant in that area. In particular, Chihiro has played a key role in the development of NICAM. Assessment of the simulation results by comparing with in-situ and satellite observations is key for this purpose. Due to this, the group is eager to obtain a standard set of numerical metrics that could ease and accelerate this goal.

Being at JAMSTEC in person facilitated many valuable discussions with researchers within the CRM-DAG group itself and beyond, thanks to Chihiro's extensive network. Among others, Marta had the chance to meet Hiroaki Tatebe, leader developer of the Model for

Interdisciplinary Research on Climate (MIROC) at JAMSTEC, one of Japan's flagship climate models, and with Daisuke Takasuka, assistant professor at Tohoku University. Hiroaki has developed a simple workflow to run MIROC and post-process its output data which computes global and zonal means of several climate variables in the atmosphere, ocean, and land. Meeting with him was useful to identify the most relevant variables that they usually check during the tuning stages of the model development, and to introduce him to Autosubmit. On the other hand, Daisuke's focus is on convection and precipitation characteristics. Specifically, his team has developed several metrics as part of a comparison of how well 3 km-resolution models reproduce precipitation (ICON, IFS, and NICAM, with IMERG as a reference). A fruitful discussion with him allowed us to obtain new complex metrics related to the randomness and clustering of convection, as well as more straightforward ones related to precipitation's development and cycle.

Within CRM-DAG, the group leader, Tomoe Nasuno, was very interested in phenomena related to subseasonal, seasonal, and interannual local extreme events. She suggested and provided references for many phenomena and indices that we had not considered before, some of them specific to the Asian Pacific climate and, then, promising for future collaboration and to encourage them to use our package. Marta also benefited greatly from the expertise of researcher Masuo Nakano, who is now focusing on TCs and whose input was crucial for implementing and validating metrics related to these during the stay. He has worked in the past on MJO and BSISO, and facilitated references for these phenomena that could help us extend and specialize the metrics that we had already collected related to MJO.

Apart from the new ideas and references gathered, a main result of the stay was the successful implementation of TC metrics and its application to NICAM data provided by Chihiro. This involved employing the TempestExtremes package to identify TC trajectories and features during the period 1990-2010 with a 1.25º-resolution version of the model (NICAM16-7S) and, afterwards, using the output to evaluate the TCs with the Cyclone Metrics Package (CyMeP). CyMeP computes several metrics involving the bias in the number of TCs, their intensity, and their duration, as well as their spatial and temporal correlation with respect to a reference dataset, which by default is taken as IBTrACS (the most complete global collection of TCs available). Figure 7 depicts an example of the comparison tables obtained, which include IBTrACS as OBS, several observational datasets (from ERAI to ERA5), and the NICAM simulations.

Reanalysis climatological bias (global)

5° x 5°	$\bar{b}_{clim,count}$	$\bar{b}_{clim,tcd}$	$\bar{b}_{clim,ace}$	$\bar{b}_{clim,pace}$	$\bar{b}_{clim,lmi}$
	#	days	10^4 kn^2	10^4 kn^2	° lat.
OBS	107.7	851.9	772.3	743.7	18.6
ERAI	-64.9	-588.6	-664.8	-580.5	5.0
20CRv3	-50.6	-478.4	-545.6	-415.3	1.1
MERRA	-57.4	-561.3	-656.0	-556.0	6.0
MERRA2	-16.4	-296.4	-502.6	-380.5	0.7
JRA	-36.5	-377.3	-543.9	-458.8	4.9
CFSR	-42.8	-438.7	-522.9	-500.7	4.2
ERA5	-26.9	-329.0	-499.7	-347.1	5.0
NICAM	16.6	-100.5	-337.6	-122.7	2.7

Low Bias — No Bias — High Bias

Reanalysis spatial correlation (global)

5° x 5°	$r_{xy,track}$	$r_{xy,gen}$	$r_{xy,u10}$	$r_{xy,slp}$	$r_{xy,ace}$	$r_{xy,pace}$
OBS	1.00	1.00	1.00	1.00	1.00	1.00
ERAI	0.86	0.71	0.40	0.21	0.83	0.91
20CRv3	0.93	0.80	0.80	0.85	0.93	0.97
MERRA	0.89	0.76	0.39	0.17	0.86	0.90
MERRA2	0.92	0.82	0.64	0.50	0.90	0.93
JRA	0.96	0.86	0.52	0.50	0.91	0.97
CFSR	0.94	0.81	0.66	0.31	0.94	0.96
ERA5	0.96	0.87	0.67	0.57	0.93	0.96
NICAM	0.84	0.76	0.59	0.51	0.82	0.86

Worse Performance — Better Performance

Figure 7 - Results of the climatological bias (left) and spatial correlation (right) evaluations of tropical cyclones comparing NICAM16-7S data (last row) to observational datasets for the period 1990-2010.

It is worth noting that the stay also served as a valuable opportunity to disseminate the work that we are doing within HANAMI WP4. Several members of the project attended and delivered presentations at the Japan Geoscience Union (JpGU) Meeting 2025 in Chiba (Japan). Particularly, Kai introduced the replicability test and some of the results published in the paper (Keller et al., 2025), while Marta focused on the generic automated workflow, the set of scientific skill metrics, and the self-contained pyhanami package. A similar presentation was given by Marta during a CEMA seminar open to all JAMSTEC members, which delved a bit deeper into the metrics identified and implemented during the stay. In all cases, the response was overwhelmingly positive, with many researchers interested in the automation aspect.

Currently, we are working on integrating the already implemented metrics into pyhanami and, in parallel, implementing and testing some of the newly suggested metrics and indices.

3.4 Structure of pyhanami

The pyhanami package is structured in a modular way that enables flexible and extensible workflows for analyzing multiple simulation ensembles. Several classes shape the core structure of the package. Starting with the SimulationData class, which loads and processes climate simulation ensembles, preparing them for further diagnostics and statistical analysis.

For visual analysis, the DataDiagnostics class provides functionality to generate the time series and spatial plots introduced in the previous sections. The class accepts SimulationData objects as inputs, and is designed to allow datasets to be added after initialization, enabling users to iteratively expand their comparisons as needed. Complementing the diagnostics tools, the ReplicabilityTest class implements the replicability test explained above. Like the diagnostics module, it allows dynamic management of datasets, making it suitable for iterative testing workflows.

The main missing features relate to data handling, both for simulated and observational data. As it is, the package accepts simulation ensembles passed either as a path to a .netcdf file or an xarray Dataset object. However, we are discussing the possibility of integrating support for a standard intake catalogue, which users could easily configure with all the simulated datasets they wish to study. The package would load the necessary data directly from this catalog.

Regarding observational data, the package currently requires a path to a directory containing observation files in a specific format. To simplify this process, we plan to provide several curated observational datasets bundled with the package, covering an extensive period up to the present and containing all variables required for the replicability test. In this way, the necessary datasets would be automatically downloaded upon installation of the package. Moreover, we are considering providing Python scripts to facilitate downloading missing variables from the most commonly used, publicly available observational products.

Overall, the codebase is organized to support future extensions, including integrating the described set of scientific skill metrics and automated report generation.

4. Conclusion

In this deliverable, we report the progress within task 4.1 of work package 4 of the HANAMI project. The aim of this task is to develop a generic workflow that enables running climate models in a reproducible fashion and to provide model diagnostics such as time series, spatial plots, and CPMIP computational performance metrics. While we first targeted porting EC-Earth4 to the workflow and deploying it on Fugaku and Marenostrum5, we decided to primarily focus on NICOCO, the NICAM atmospheric model coupled to the ocean model COCO. This decision is based on the rapid developments currently performed on EC-Earth to make it ready for the fast-track simulations of CMIP7. This made it very difficult to maintain the model within the workflow, keeping up with the developments. Nevertheless, we were able to deploy a version of EC-Earth4 on Fugaku, which will facilitate the later deployment of a more mature version. We made very good progress during the first 18 months of HANAMI providing a first stable version of the generic model workflow with ports of NICOCO and the Loren96 dummy model. The workflow is operated with the Autosubmit workflow manager using the instance deployed at the BSC. The workflow is capable of compiling NICOCO and performing climate simulations with it. Currently we support running on Fugaku and Marenostrum5.

We further report the progress of the HANAMI diagnostics package, a.k.a., pyhanami. Pyhanami is a pip-installable Python package that can be used to perform certain diagnostics on climate model outputs. Currently it includes a climate model replicability test, ensemble global mean time series, spatial bias and two-ensemble effect size plots, and a scientific skill metric assessing the resemblance of tropical cyclone duration, intensity, and spatio-temporal correlation with observations.

In addition to the technical progress, we also considerably improved the collaboration with our Japanese partners thanks to the funding for travels and stays provided by HANAMI. Leo Arriola was able to port NICOCO to the HANAMI workflow while staying 5 weeks at NIES, supervised by Dr. Hisashi Yashiro. Marta Alerany worked closely together with Dr. Chihiro Kodama and many others on scientific model performance metrics during her 6-week stay at JAMSTEC. In addition, we have been applying together with our Japanese partners to common projects. We successfully applied to a JHPCN call, granting us a 1-year project with computing resources on Miyabi, the new supercomputer of the University of Tokyo.

The next steps that we will address in the remaining 18 months of HANAMI are porting EC-Earth4 to the workflow and deploying it on Fugaku and Marensotrum5 in close collaboration to the EC-Earth consortium including the developments for CMIP7. We will further deploy NICOCO on Miyabi-C to perform a replicability test for NICOCO between the three clusters Miyabi-C, Fugaku, and Marenostrum5. We will also add ensemble support in the HANAMI worklfow and enable reporting the CPMIP computational performance metrics.