



Project Title	Biodiversity Digital Twin for Advanced Modelling, Simulation and Prediction Capabilities
Project Acronym	BioDT
Project Number	101057437
Type of Project	RIA - Research and Innovation Action
Topics	HORIZON-INFRA-2021-TECH-01-01
Starting Date of Project	1 June 2022
Ending Date of Project	31 May 2025
Duration of the Project	36 months
Website	www.biodt.eu

## D6.2 - Report on Model Upscaling Approaches

<b>Work Package</b>	WP6   Simulation, Modelling and Data Analytics
<b>Task</b>	T6.2   Upscaling and Implementing the DT
<b>Lead Authors</b>	Franziska Taubert (UFZ), Otso Ovaskainen (JYU), Tuomas Rossi (CSC)
<b>Contributors</b>	Bekir Afsar (JYU), Chris Andrews (UKCEH), Thomas Banitz (UFZ), Jan Dick (UKCEH), Ahmed El-Gabbas (UFZ), Tobias Frøslev (GBIF), Desalegn Chala Gelete (UiO), Jürgen Groeneveld (UFZ), Kate Ingenloff (GBIF), Simon Rolph (UKCEH)
<b>Peer Reviewers</b>	Tomáš Martinovič (IT4I@VSB), Christos Arvanitidis (LifeWatch ERIC)
<b>Version</b>	V1.0
<b>Due Date</b>	31/05/2024
<b>Submission Date</b>	30/05/2024

### Dissemination Level

<input checked="" type="checkbox"/>	PU: Public
<input type="checkbox"/>	SEN: Sensitive – limited under the conditions of the Grant Agreement
<input type="checkbox"/>	EU-RES. Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)
<input type="checkbox"/>	EU-CON. Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)
<input type="checkbox"/>	EU-SEC. Classified Information: SECRET UE (Commission Decision 2005/444/EC)



Funded by  
the European Union

## Version History

Revision	Date	Editors	Comments
0.1	30/03/2024	All WP6 members	Table of Contents, first draft of sections
0.2	08/05/2024	Christos Arvanitidis (LifeWatch ERIC), Tomáš Martinovič (IT4I@VSB), Franziska Taubert (UFZ)	Reviewer comments
0.3	16/05/2024	All WP6 members	Revision of draft
0.4	17/05/2024	Franziska Taubert (UFZ), Tuomas Rossi (CSC)	Final edits and clean-up
0.5	22/05/2024	Tuomas Rossi (CSC), Ahmed El-Gabbas (UFZ)	Minor text updates
0.6	27/05/2024	Franziska Taubert (UFZ)	Cleaned-up document
1.0	30/05/2024	Tuomas Rossi (CSC), Anna-Liisa Allas (CSC)	Submission-ready version incorporating BioDT Project Management Office (PMO) minor editorial changes

## Glossary of Terms

Item	Description
API	Application Programming Interface
ASF	African Swine Fever
CNN	Convolutional Neural Network
CORINE	Coordination of Information on the Environment
CWR	Crop Wild Relative
DAP	Data Access Protocol
DDDAS	Dynamic Data Driven Application Systems
DEIMS-SDR	Dynamic Ecological Information Management System - Site and Dataset Registry
EASIN	European Alien Species Information Network
ECS	Entity Component System
eLTER	Integrated European Long-Term Ecosystem, critical zone and socio-ecological Research
GUI	Graphical User Interface
HMSC	Hierarchical Modelling of Species Communities
IAS	Invasive Alien Species
IASDT	Invasive Alien Species Digital Twin

OPeNDAP	Open-source Project for a Network Data Access Protocol
PD	Phylogenetic Diversity
pDT	Prototype Digital Twin
UC	Use Case

## Keywords

Biodiversity, Digital Twin, Modelling, Simulation, Prediction, High-Performance Computing

## Disclaimer

The BioDT project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101057437 (<https://doi.org/10.3030/101057437>). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

## Executive Summary

Digital twins have been well developed in engineering but their full potential in other domains remains poorly understood. The Biodiversity Digital Twin project (BioDT) aims to test the potential of the Digital Twin concept in ecology, especially in conservation biology, where no operational Digital Twins yet exist. To explore the potential of digital twins in general, BioDT is based on a multitude of use cases (UCs) that differ in terms of their scope and complexity. Some of the UCs are based on long-term research programs that have developed relevant models and data, in which case the main challenge of the UC is to integrate the already developed components within the Digital Twin. Other UCs are based on more novel ideas and do not yet have operational models or even well-defined data sources, and hence these need to be developed simultaneously while building the Digital Twin. The prototype Digital Twins developed from the different UCs (short: pDT) are implemented through reproducible and well-documented pipelines that describe the interplay between data, models, and users, as well as the continuously updating nature of each pDT. The envisaged pDTs are expected to vary in number and phases of development cycles – a feature which we anticipated by the selection of UCs ranging in scope and complexity – to guide and inspire future teams aiming to develop pDTs for their specific topic in biodiversity research.

The development of each pDT comes with its own challenges like computational expensive runtime or uncertainty in data and respective model parameter. The developed pDT pipelines – from data streaming, over processing, model simulation to visualization – and their testing for pilot studies has already been comprehensively described in the previous report D6.1 *Pipeline release and validation*. This report D6.2 *Report on model upscaling approaches* focuses now on approaches that have been implemented or conceptualized with the aim to overcome challenges in running the developed pDT pipeline, for example, for larger scales, datasets, model simulations or user interactions than previously tested for the pilot study. The different approaches implemented to deal with upscaling issues for each pipeline of the pDTs range from implementations on the pan-European pre-exascale supercomputer LUMI (either with CPU or GPU), parallelization of pipeline or model runs as well as multi-scale model parameterization approaches using transfer functions. Each pDT, unique in its design and storyline, faces unique challenges. The solutions found and implemented within each pDT, however, are often based on general approaches, by which we close the report with a conclusion section on the *lessons learnt*.

## Table of Contents

1. Overview of prototype Digital Twins.....	6
2. Description of upscaling approaches for each pDT .....	7
2.1 Species response to environmental change.....	7
2.1.1 Biodiversity dynamics .....	7
2.1.2 Ecosystem services .....	13
2.2 Genetically detected biodiversity.....	15
2.2.1 Crop wild relatives and genetic resources for food security .....	15
2.2.2 DNA detected biodiversity, poorly known habitats .....	16
2.3 Dynamics and threats from and for species of policy concern .....	17
2.3.1 Invasive species .....	17
2.4 Species interactions with each other and with humans .....	19
2.4.1 Disease outbreaks.....	19
2.4.2 Pollinators.....	20
3. Conclusions and lessons learnt.....	22
3.1 Conclusions.....	22
3.2 Lessons learnt on model upscaling.....	22
3.3 Future work .....	23

# 1. Overview of prototype Digital Twins

The ten use cases (UCs) cover a wide range of taxa, systems, and applications. This diversity reflects that biodiversity, conservation and ecological research are, by nature, multidimensional and they encompass complex socio-ecological systems. The UCs differ in terms of their scope, complexity, and levels of scale, extent, granularity of processes (from agents to large scale processes) as well as context. Some of the UCs build on long-term research programs that have existing models and data, therefore the main challenge of these UCs is to consolidate the existing components as a digital twin. In contrast, other UCs are pioneering DT approaches for novel research themes and prior to the BioDT project commencing may not have had operational models or well-defined data sources, and hence these are being developed as 'from the ground up' as DTs.

The UCs are summarised here:

- Two UCs, *Grassland Biodiversity Dynamics*, and *Forest/Bird Biodiversity Dynamics*, focus on species-rich communities and how they respond to environmental change and management practices.
- The UC *Real time bird monitoring with citizen science data* tests the feasibility of involving a large number of citizens (>100,000) in close-to-real-time biodiversity monitoring.
- The UC *Cultural Ecosystem Services* focuses on cultural services provided by biodiversity and how people use the associated resources. Three UCs are studying genetically detected biodiversity: (i) *Crop Wild Relatives and genetic resources for food security*, (ii) *DNA Detected Biodiversity (poorly known habitats, phylogenetic diversity)* and (iii) *Genetically Detected Biodiversity in Cryptic Habitats (prioritisation of sampling effort)*. The first UC attempts, in the context of food security, to improve the genetic pool of domesticated crops by exploring genetic resources from crop wild relatives. The latter two UCs try to improve the global cover of DNA-based biodiversity monitoring and develop DTs to identify priority areas for sampling, in particular on cryptic or poorly studied habitats.
- The UC *Invasive Alien Species* aims at predicting the conditions and extent of the future spread of invasive alien plant species, which are of political concerns as alien species are a major threat to biodiversity and hence ecosystem functions and services and the associated societal goods and benefits.
- The UC *Disease Outbreaks* seeks to improve the management of the spread of wildlife diseases. As an example, the spread of African swine fever is modelled, which is a threat to domestic livestock and therefore of major social, economic and political concern.
- The UC *Pollinators* focuses on honey bees, because they provide the ecosystem service of pollination. In addition, they are exposed, like pollinators in general, to multiple stressors, in particular modern agricultural practices (monocultures, pesticide applications) and climate change, with a high risk of depletion of their populations.

The overall task of BioDT is to develop and deploy *prototype Digital Twins* (pDTs). This requires close collaboration of researchers whose main expertise is in technical platforms, data and UCs, workflows and FAIR principles, simulation, modelling and data analysis, and the integration with research infrastructures. The pDTs corresponding to each UC are implemented through reproducible and well-documented pipelines that describe the interplay between data, models, and users, as well as the continuously updating nature of each pDT. A comprehensive description of the developed and implemented pipelines for each pDT as well as their testing for a pilot study case has been published in the Deliverable Report D6.1 on *Pipeline release and validation* and will be found here: <https://biодt.eu/documents-publications>.

## 2. Description of upscaling approaches for each pDT

### 2.1 Species response to environmental change

#### 2.1.1 Biodiversity dynamics

##### 2.1.1.1 Grassland biodiversity dynamics

###### 2.1.1.1.1 Brief summary of the developed pipeline from D6.1

The pDT of grassland biodiversity dynamics will allow end-users (e.g. researchers, farmers, regulatory decision-makers) to select a specific grassland site, monitor its current state, and project its future state under different management and climate scenarios.

The pDT pipeline includes retrieving and processing required data, running simulations with the model GRASSMIND (Taubert et al., 2020), exploring the simulation output by users, and comparing simulation output with observation data. Particularly, the pDT allows users to select a specific grassland site in Europe by either providing a location (spatial coordinates) or a DEIMS.ID from which the (representative) location can be derived (for sites listed in the eLTER DEIMS-SDR, Wohner et al., 2019). Thereupon, weather, soil and management data for that location and a (user-specified or default) time period get retrieved from different public sources (e.g. Muñoz Sabater 2019; Simons, Koster, and Droogers 2020; Poggio et al., 2021) and prepared as input data to run simulations. The simulated grassland vegetation dynamics serve to compute and visualize various metrics of biodiversity and productivity as time series, which the users can analyse. Users can also request new simulation scenarios, especially short- or long-term future scenarios to explore different (combinations of) management options, climate change or climate extremes. This may lead to running tens of thousands of GRASSMIND simulations for many different sites and scenarios, which will highly benefit from the parallel processing capabilities in LUMI-C. If available, the simulated vegetation dynamics can be compared to observations (e.g. from eLTER grassland sites), or users may communicate the need for more data.

###### 2.1.1.1.2 Implemented approaches for upscaling

Three upscaling approaches have been implemented: (1) implementation of the pipeline on LUMI-C, (2) parallelization of GRASSMIND model runs, and (3) development of a generic site-transferable GRASSMIND parameterization.

- (1) The developed pipeline scripts for pre- and post-processing input and output data, as well as the GRASSMIND model itself run on LUMI-C. Therefore, we generally developed Python scripts for smooth deployment in different environments. The scripts are stored on the BioDT repository on GitHub (<https://github.com/BioDT>), they get continuously improved and extended, and they will be available as open source.
- (2) We have compiled the GRASSMIND code natively on LUMI-C and developed scripts to execute independent GRASSMIND simulations in parallel by using GNU Parallel tool (<https://www.gnu.org/software/parallel>).
- (3) A concept to derive a generic site-transferable GRASSMIND parameterization has been developed. Such a generic parameterization allows to apply the pipeline to any grassland site selected by the pDT user without the need for recalibrating the GRASSMIND model, which is computationally intensive and often simply impossible due to lacking observation data for the selected site. Such a generic parameterization means that certain standard assumptions need to be made (especially on the representative trait values of plant functional types), which may not always fully match the local conditions at a specific site (with a particular site-specific composition of species belonging to these plant functional types).

A promising approach to overcome the disadvantages described under (3) are transfer functions (cf. Samaniego, Kumar, and Attinger 2010; Rödig et al., 2017), i.e. model parameters as a function of site-specific variables (like weather and soil conditions, or plant species community composition). Development of transfer functions will be initiated as part of the next task 6.3 (*“Testing the predictive capacity of the DT”*). Adding such transfer functions to the generic parameterization would account for the differences that certainly exist among European grassland sites, particularly in plant species composition and plant trait adaptation to local conditions (e.g. differently evolved specific leaf area for Mediterranean versus UK sites due to different precipitation levels). In this way, GRASSMIND would remain transferable to new sites without recalibration, but model prediction uncertainties could most likely be reduced.

#### 2.1.1.1.3 Challenges

Challenges mostly have arisen in terms of limited data availability. Lacking data need to be replaced with default assumptions, which in turn increase the uncertainty in model predictions. Lacking data include, for example, Europe-wide information on grassland management (i.e. mowing frequency and dates, fertilization amount and dates, irrigation amount and dates, grazing dates, periods, number and type of grazing animals). Such information is partly available for specific years and regions or countries (e.g. Lange et al., 2022b; Schwieder et al., 2022), but usually does not cover management practices further in the past (and also not grassland establishment).

Analyzing and communicating the effects of uncertainties in management data requires a sensitivity analysis comparing various management options, which largely increases the necessary number of scenarios and simulations. For example, for Germany the explicit dates of grassland mowing are available from 2017 onwards and updated annually (Schwieder et al., 2024b; 2024a), but information on fertilization, irrigation and grazing is much more limited (Lange et al., 2022a). For fertilization, we can create a default scenario based on German legislation and farmers' common practices (Vogt et al., 2019). However, farmers' actual management might deviate, e.g. untypical fertilization dates or less than the maximum allowed amounts of fertilizer (if soils are rich in humus or past yields had been high). Each potential deviation should ideally be tested in an own simulation scenario for uncertainty analysis. Such analyses will be part of the next task 6.3 (*“Testing the predictive capacity of the DT”*).

A challenge for deriving the generic GRASSMIND parameterization (cf. 2.1.1.1.2) is the need for many model simulation runs. Moreover, the parameterization process cannot be fully automatized. Especially during development, it requires substantial analysis and interpretation of calibration results and expert testing of calibration settings (e.g. optimization algorithm and specifications, several choices for the definition of the objective function to assess the model fit to observation data). Finally, any results and insights generated during calibration need to be re-evaluated when the model itself gets modified and improved.

Another data limitation arises for the transfer functions to enhance the generic model parameterization. The development of such transfer functions requires a substantial amount of long-term observations with a good coverage of the pDT application area of European grassland sites. With our BioDT call to data holders of all eLTER grassland sites we were able to collect a good data basis for deriving the generic parameterization and transfer functions. However, data gaps remain and for areas or specific grassland types not covered the pDT may later show a lower predictive performance.

While the individual GRASSMIND simulations are fast to execute (seconds), a performance bottleneck of the pDT is the large number of individual simulations expected to be run. This bottleneck is efficiently solved by the developed parallelization scripts outside the model code. Another performance bottleneck is the long queuing time when retrieving location-specific weather data from the Copernicus ERA5-Land data set (Muñoz Sabater 2019). This bottleneck can only partly be solved by parallel requests due to the strict Copernicus Climate Data Store API limits of per-user requests.



#### 2.1.1.1.4 Analysis of relative performance improvements

To estimate the expected performance improvements, the runtime for preparing input files (without downloading raw input data) and simulating 160 instances of GRASSMIND (10 year simulation period, 1m<sup>2</sup> area) is 8 minutes on a Windows laptop without parallelization, 2 minutes with parallelization (10 cores), 25 seconds on a Windows-based HPC system Model Server Grid with parallelization (56 cores), and 5 seconds on a single LUMI-C node (128 cores).

Evidently, such 160 simulations can be calculated on a laptop, but the benefit of LUMI supercomputer comes when processing tens of thousands of simulations. Based on the obtained runtime estimates, we expect about 20-fold time-to-solution improvement by using a single LUMI-C node in comparison to a 10-core Windows laptop. Runtime could be further reduced by using multiple LUMI-C nodes and the optimal run configuration will be determined by the size of the production calculations. These performance improvements become crucial when unfolding the potential of the pDT to run simulations for many grassland sites, many climate and management scenarios, many stochastic replicates to approximate mean grassland dynamics, and particularly during model (re)calibration which requires very large numbers of simulation runs.

#### 2.1.1.1.5 Model source code release

The GRASSMIND model code is still improving and will be available as open-source in a GitLab repository. The developed parallelization scripts as well as pre- and post-processing scripts will be made publicly available under the BioDT GitHub (<https://github.com/BioDT>). At the moment, access is invitation only but can already be shared upon request.

#### 2.1.1.1.6 References

Lange, Maximilian, Hannes Feilhauer, Ingolf Kühn, and Daniel Doktor. 2022a. Land-Use Intensity Quantification and Management Classifications in Grasslands of Germany 2017/2018. Mendeley Data. <https://doi.org/10.17632/m9rrv26dvf.1>

———. 2022b. Mapping Land-Use Intensity of Grasslands in Germany with Machine Learning and Sentinel-2 Time Series. *Remote Sensing of Environment* 277 (August): 112888. <https://doi.org/10.1016/j.rse.2022.112888>

Muñoz Sabater, J. 2019. ERA5-Land Hourly Data from 1950 to Present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS). <https://doi.org/10.24381/cds.e2161bac>

Poggio, Laura, Luis M. de Sousa, Niels H. Batjes, Gerard B. M. Heuvelink, Bas Kempen, Eloi Ribeiro, and David Rossiter. 2021. SoilGrids 2.0: Producing Soil Information for the Globe with Quantified Spatial Uncertainty. *SOIL* 7 (1): 217–40. <https://doi.org/10.5194/soil-7-217-2021>

Rödig, Edna, Matthias Cuntz, Jens Heinke, Anja Rammig, and Andreas Huth. 2017. Spatial Heterogeneity of Biomass and Forest Structure of the Amazon Rain Forest: Linking Remote Sensing, Forest Modelling and Field Inventory. *Global Ecology and Biogeography* 26 (11): 1292–1302. <https://doi.org/10.1111/geb.12639>

Samaniego, L., R. Kumar, and S. Attinger. 2010. Multiscale Parameter Regionalization of a Grid-Based Hydrologic Model at the Mesoscale. *Water Resources Research* 46. <https://doi.org/10.1029/2008wr007327>

Schwieder, Marcel, Felix Lobert, Gideon Okpoti Tetteh, and Stefan Erasmi. 2024a. Grassland Mowing Events across Germany Detected from Combined Sentinel-2 and Landsat Time Series for the Year 2022. Zenodo. <https://doi.org/10.5281/zenodo.10610283>

———. 2024b. Grassland Mowing Events across Germany Detected from Combined Sentinel-2 and Landsat Time Series for the Years 2017 - 2021. Zenodo. <https://doi.org/10.5281/zenodo.10609590>

Schwieder, Marcel, Maximilian Wesemeyer, David Frantz, Kira Pfoch, Stefan Erasmi, Jürgen Pickert, Claas Nendel, and Patrick Hostert. 2022. Mapping Grassland Mowing Events across Germany Based on Combined Sentinel-2 and Landsat 8 Time Series. *Remote Sensing of Environment* 269 (February): 112795. <https://doi.org/10.1016/j.rse.2021.112795>

<https://www.biodt.eu/>

Simons, G., R. Koster, and P. Droogers. 2020. HiHydroSoil v2.0 - A High Resolution Soil Map of Global Hydraulic Properties. 134. FutureWater. Wageningen, The Netherlands.

Taubert, Franziska, Jessica Hetzer, Julia S. Schmid, and Andreas Huth. 2020. The Role of Species Traits for Grassland Productivity. *Ecosphere* 11 (7): e03205. <https://doi.org/10.1002/ecs2.3205>

Vogt, Juliane, Valentin Klaus, Steffen Both, Cornelia Fürstenau, Sonja Gockel, Martin Gossner, Johannes Heinze, et al. 2019. Eleven Years' Data of Grassland Management in Germany. *Biodiversity Data Journal* 7 (September): e36387. <https://doi.org/10.3897/BDJ.7.e36387>

Wohner, Christoph, Johannes Peterseil, Dimitris Poursanidis, Tomáš Kliment, Mike Wilson, Michael Mirtl, and Nektarios Chrysoulakis. 2019. DEIMS-SDR – A Web Portal to Document Research Sites and Their Associated Data. *Ecological Informatics* 51 (May): 15–24. <https://doi.org/10.1016/j.ecoinf.2019.01.005>

### 2.1.1.2 Forest/bird biodiversity dynamics

#### 2.1.1.2.1 Brief summary of the developed pipeline from D6.1

The forest biodiversity dynamics pDT aims to explore how various forest management strategies and climate change scenarios impact forest ecosystems and biodiversity. The main goal is to identify the most suitable treatment option that enhances biodiversity through conservation or adaptability for a particular forest across different climate scenarios. Through the use of the LANDIS-II forest simulator (Scheller et al., 2007) and HMSC biodiversity models (Ovaskainen et al., 2017; Ovaskainen and Abrego, 2020), the pDT predicts future environmental conditions and biodiversity responses under different management options and climate scenarios.

The developed pipeline integrates forest simulation, biodiversity modelling, and decision support to optimize forest management strategies in Finnish forests. Key components of the pipeline include specialized scripts for data preparation and interpretation. These scripts enable the preparation of input files for LANDIS-II, the translation of LANDIS-II outputs for HMSC, and the interpretation of models' outputs to evaluate the objective functions in the multiobjective optimization. The pipeline's pilot study focuses on the Uusimaa region, evaluating objective functions such as sustainable timber production, carbon storage, deadwood volume, and habitat suitability for birds. By addressing computational challenges and engaging stakeholders in the decision-making process, the pipeline aims to provide a robust framework for informed forest management decisions that balance ecological, social and economic objectives.

#### 2.1.1.2.2 Implemented approaches for upscaling

The LANDIS-II model is developed using C# / .NET, which makes its deployment on HPC environment non-trivial. A clean solution has been obtained by containerizing the model and required libraries. Successful test executions have been performed on the CPU partition of the LUMI supercomputer by using Singularity / Apptainer. Parallelization of LANDIS-II model has not been developed yet, but the plan is to parallelize LANDIS-II simulations by subdividing input maps into smaller areas, processed in parallel, and then integrating outputs for smaller areas to generate full outputs. This parallelization strategy enhances efficiency by distributing computational workload across multiple cores, thereby accelerating model runtime.

The computationally expensive parts of HMSC R code have been GPU-accelerated by using TensorFlow in collaboration with other on-going projects, yielding significant speed up on HPC systems (Rahman et al., 2024). This acceleration enables more efficient processing of large datasets, reducing the overall computation time and facilitating the scalability of the model to cover larger spatial extents, such as regional or country-level analyses.

#### 2.1.1.2.3 Challenges

The primary challenge in model computation time necessitates a strategic approach to address the computational intensity of running LANDIS-II and HMSC models across the entire study area. To mitigate this

challenge, the initial pilot study focuses on the Uusimaa region, allowing for comprehensive analysis of management regimes and climate scenarios while conserving computational resources. By simulating all necessary combinations and pre-computing optimal solutions before interacting with stakeholders, the pilot study aims to save time and resources during the decision-making processes. Once completed, the study will scale up to cover the entirety of Finland by leveraging the high-performance computing capabilities of LUMI. Additionally, ongoing efforts to translate HMSC into TensorFlow/Python, with anticipated GPU acceleration, hold promise for significantly reducing overall model runtime compared to CPU-only versions, enhancing computational efficiency and scalability.

#### 2.1.1.2.4 Analysis of relative performance improvements

Efforts are underway to enhance the performance of the pipeline through ongoing work on various fronts. On a local laptop, running a scenario for the Uusimaa region on LANDIS-II with a resolution of 250 x 250m takes approximately 70 minutes. However, the preparation of finer resolution input files (100x100m) is underway, which may further increase runtime. Detailed performance analysis of the whole pipeline will be performed once the planned parallelization scheme for LANDIS-II simulations has been implemented. Regarding HMSC calculations, the ongoing translation of HMSC into TensorFlow/Python with GPU acceleration promises substantial reductions in overall model runtime, potentially by up to three orders of magnitude compared to CPU-only implementations (Rahman et al., 2024). This transition holds significant potential for enhancing the performance of the pipeline.

#### 2.1.1.2.5 Model source code release

All the models and codes are available under open source licences. The LANDIS-II forest model and used extensions are available at <https://www.landis-ii.org> and <https://github.com/LANDIS-II-Foundation>. The HMSC-HPC code is currently available at <https://github.com/aniskhan25/hmsc-hpc>. The configured LANDIS-II and HMSC models for Finnish forests are shared among the collaborators within this pDT and can be accessible upon request as instructed in this GitHub repository: [https://github.com/BeAfsar/forest\\_birds\\_pDT](https://github.com/BeAfsar/forest_birds_pDT).

The developed execution scripts as well as pre- and postprocessing scripts will be available at the BioDT GitHub organization <https://github.com/BioDT>.

#### 2.1.1.2.6 References

Ovaskainen, O., Tikhonov, G., Norberg, A., Guillaume Blanchet, F., Duan, L., Dunson, D., ... & Abrego, N. (2017). How to make more out of community data? A conceptual framework and its implementation as models and software. *Ecology Letters*, 20(5), 561-576. <https://doi.org/10.1111/ele.12757>

Ovaskainen, O., & Abrego, N. (2020). *Joint species distribution modelling: with applications in R*. Cambridge University Press.

Rahman, A. U., Tikhonov, G., Oksanen, J., Rossi, T., & Ovaskainen, O. (2024). Accelerating joint species distribution modeling with Hmsc-HPC: A 1000x faster GPU deployment. *bioRxiv*, <https://doi.org/10.1101/2024.02.13.580046>



Scheller, R. M., Domingo, J. B., Sturtevant, B. R., Williams, J. S., Rudy, A., Gustafson, E. J., & Mladenoff, D. J. (2007). Design, development, and application of LANDIS-II, a spatial landscape simulation model with flexible temporal and spatial resolution. *Ecological Modelling*, 201(3-4), 409-419. <https://doi.org/10.1016/j.ecolmodel.2006.10.009>

### 2.1.1.3 Real-time bird monitoring with citizen science data

#### 2.1.1.3.1 Brief summary of the developed pipeline from D6.1

This pDT aims to investigate if and how citizen science can be employed to real-time bird monitoring, in a way that produces robust data also for scientific analyses. The pDT aims to develop a www-portal that shows

<https://www.biodt.eu/>

 @BiodiversityDT |  company/biodt

data and predictions with minimal delay compared to the real-world system. A pilot study is being implemented within Finland. If the pilot study is successful, the approach could easily be scaled up for the whole Europe. The choice of this pDT is done because the number of bird enthusiasts is huge.

As the bird classification model, we use a convolutional neural network (CNN) (Lauha et al., 2022). For model input, we use  $128 \times 129$  matrices, which correspond to spectrograms of 3 sec audio clips. These inputs are images, where each pixel value corresponds to the intensity of a certain frequency bin at a certain time point. The model was trained using labeled data from four bird audio repositories (Xeno-canto (<https://xeno-canto.org>); Macaulay (<https://www.macaulaylibrary.org>); Bird Sound Global (<https://bsg.laji.fi>); (Lehikoinen et al., 2022) and applying several kinds of data augmentations (Lauha et al., 2022).

As the biodiversity model, we use HMSC (Ovaskainen and Abrego, 2020), which belongs to the class of joint species distribution models (Warton et al., 2015). HMSC models describe species-environmental relationships using a generalized linear model and are able to borrow information among species based on their trait and phylogenetic similarity. Fitting HMSC models to large data sets and making spatially extensive predictions is computationally intensive. While the R-version of HMSC (Tikhonov et al., 2020) is partially HPC-compatible, it does not utilize optimal resources e.g., GPU computation, compromising its scalability to large data.

#### 2.1.1.3.2 Implemented approaches for upscaling

To overcome the main computational bottleneck, we have translated HMSC into TensorFlow/Python (Rahman et al., 2024), which version we have also implemented in LUMI-G.

#### 2.1.1.3.3 Challenges

While the GPU-accelerated version of HMSC is up-and-running in LUMI and can perform parameter estimation 1,000 times faster than the R-implementation of HMSC, this version has been so far implemented only for model fitting, not for making predictions. For the real-time bird monitoring prototype, making predictions is the computationally most intensive part, especially given that we aim to conduct that for the whole Finland at 1 ha resolution at an hourly basis. Implementing a GPU-accelerated version of HMSC prediction can be done using partially the same components as those used for parameter estimation, but still requires a substantial amount of recoding.

#### 2.1.1.3.4 Analysis of relative performance improvements

We have made extensive comparisons between the GPU-accelerated version of HMSC and the R-version of HMSC in publication (Rahman et al., 2024). The results show that the relative computational gain is largest in computationally intensive problems (many species, many sampling locations, models including phylogenetic or spatial components), where HMSC-HPC is typically 1,000 times faster than the R-implementation. As an exception, spatial models that are based on the NNGP approximation do not benefit from the current implementation and hence are not suitable for this prototype.

#### 2.1.1.3.5 Model source code release

HMSC is currently implemented as an R-package (Tikhonov et al., 2020) that is available in CRAN (<https://cran.r-project.org/web/packages/Hmsc/index.html>) and GitHub (<https://github.com/hmsc-r/HMSC>).

The GPU-accelerated version HMSC-HPC is available at GitHub (<https://github.com/aiskhan25/hmsc-hpc>).

#### 2.1.1.3.6 References

Lauha, P., Somervuo, P., Lehikoinen, P., Geres, L., Richter, T., Seibold, S. and Ovaskainen, O. (2022). Domain-specific neural networks improve automated bird sound recognition already with small amount of local data. *Methods in Ecology and Evolution* 13, 2799-2810. <https://doi.org/10.1111/2041-210X.14003>

Lehikoinen, P., Rannisto, M., Camargo, U., Aintila, A., Lauha, P., Piirainen, E., Somervuo, P., & Ovaskainen, O. (2022). Data from: Crowdsourcing training material for automated bird sound classification - a pilot study. *Zenodo Repository*. <https://doi.org/10.5281/zenodo.7030863>

Ovaskainen, O., & Abrego, N. (2020). Joint species distribution modelling: with applications in R. Cambridge University Press.

Tikhonov, G., Opedal, Ø. H., Abrego, N., Lehikoinen, A., de Jonge, M. M., Oksanen, J., & Ovaskainen, O. (2020). Joint species distribution modelling with the R-package HMSC. *Methods in Ecology and Evolution*, 11(3), 442-447. <https://doi.org/10.1111/2041-210X.13345>

Rahman, A. U., Tikhonov, G., Oksanen, J., Rossi, T. and Ovaskainen, O. Accelerating joint species distribution modelling with Hmsc-HPC: A 1000x faster GPU deployment. *BioRxiv* 2024.02.13.580046

Warton, D. I., Blanchet, F. G., O'Hara, R. B., Ovaskainen, O., Taskinen, S., Walker, S. C., & Hui, F. K. (2015). So many variables: joint modeling in community ecology. *Trends in Ecology & Evolution*, 30(12), 766-779. <https://doi.org/10.1016/j.tree.2015.09.007>

## 2.1.2 Ecosystem services

### 2.1.2.1 Cultural ecosystem services

#### 2.1.2.1.1 Brief summary of the developed pipeline from D6.1

This prototype digital twin focuses on developing a tool to support the understanding and management of cultural ecosystem services, which are non-material benefits people derive from ecosystems (Dick et al., 2022; Zulian et al., 2018). This includes recreation, tourism, intellectual development, and aesthetic experiences. The digital twin tracks changes in usage of services and resources, aiming to minimize trade-offs between recreational use and impacts on biodiversity.

The Cultural Ecosystem Services prototype digital twin consists of two main components: a recreation potential model and a species distribution model. The recreation potential model assesses the suitability of areas for recreational activities based on various factors like land cover, natural features, and access. The species distribution model quantifies biodiversity using data from sources like GBIF<sup>1</sup> and eLTER<sup>2</sup>, incorporating environmental factors like climate and vegetation.

Data sources include detailed information on land cover, natural features, and species occurrences. The recreation potential model uses data such as land cover maps, land use maps and proximity to infrastructure, while the species distribution model utilizes environmental variables like climate and vegetation cover.

The models are implemented using R programming language, with the recreation potential model employing the raster package for geospatial computations. The biodiversity component utilizes the *flexsdm* package for species distribution modelling.

#### 2.1.2.1.2 Implemented approaches for upscaling

The pipeline implementation on LUMI involves utilising Singularity containers to containerize the biodiversity model within a consistent environment. Users can pull the container from a designated repository, ensuring reproducibility and portability. Once the container is available, users execute the model by running a SLURM bash script, which specifies the necessary parameters and dependencies. This approach enables efficient utilization of LUMI's resources and facilitates the execution of multiple model runs in parallel, contributing to the scalability and robustness of the analysis pipeline.

---

<sup>1</sup> <https://www.gbif.org/>

<sup>2</sup> <https://elter-ri.eu/>

<https://www.biodt.eu/>



Neither the biodiversity nor recreational potential components of the pDT are suitable for GPU acceleration due to several reasons. Firstly, the models primarily rely on CPU-intensive tasks such as data pre-processing, statistical modelling, and raster processing, which may not benefit significantly from GPU acceleration. GPU acceleration is typically advantageous for tasks with highly parallelizable computations, such as matrix operations or deep learning algorithms, which are not extensively utilized in this model. Furthermore, the specific libraries and packages used in the model, such as *flexsdm* and *terra*, may not have GPU-accelerated versions available.

The recreational potential model is currently being upscaled from regional to national (covering the geographic extent of Scotland). This presents several challenges, including a reduction in model fidelity due to the use of national-scale datasets and the associated loss of local knowledge, as well as requiring greater computational resource.

No work was carried out on the use of transfer functions. Transfer functions could provide a valuable method for identifying specific parameters that influence local ecological dynamics, allowing for the scaling up of biodiversity models from local to regional or national levels. By analysing these functions, the pDT could pinpoint key environmental factors or anthropogenic influences that drive biodiversity patterns. This approach could enable a more nuanced understanding of how local ecological processes translate to broader spatial scales, facilitating more accurate and comprehensive conservation planning and management strategies at regional or national levels.

#### 2.1.2.1.3 Challenges

Unlike models that are agent-based or easily discretized into parallelizable tasks, this model uses statistical models that are difficult to distribute across multiple computer nodes efficiently, beyond running one job per species. However, the relatively short runtime of the models at present do not present an urgent need to make the pipeline faster on HPC. Ultimately, the models selected for the pDT faced challenges in upscaling because these models were not well suited for HPC.

#### 2.1.2.1.4 Analysis of relative performance improvements

The models have not been re-run since the pilot study. In the pilot study, biodiversity models were run for a target 100 species. Each successful model run for each species took between 5 and 20 minutes to complete, depending on the volume of data available for each species, with more data resulting in longer runtime

#### 2.1.2.1.5 Model source code release

The developed model code and execution scripts are published under the BioDT GitHub organization <https://github.com/BioDT/uc-ces>

#### 2.1.2.1.6 References

Dick J, Andrews C, Orenstein D, Teff-Seker Y, Zulian G. (2022). A mixed-methods approach to analyse recreational values and implications for management of protected areas: A case study of Cairngorms National Park, UK. *Ecosystem Services* 56. <https://doi.org/10.1016/j.ecoser.2022.101460>

Zulian G, Stange E, Woods H, Carvalho L, Dick J, Andrews C, Baró F, Vizcaino P, Barton D, Nowel M, Rusch G, Autunes P, Fernandes J, Ferraz D, Ferreira dos Santos R, Aszalós R, Arany I, Czúcz B, Priess J, Hoyer C, Bürger-Patricio G, Lapola D, Mederly P, Halabuk A, Bezak P, Kopperoinen L, Viinikka A. (2018). Practical application of spatial ecosystem service models to aid decision support. *Ecosystem Services* 29: 465-480. <https://doi.org/10.1016/j.ecoser.2017.11.005>

## 2.2 Genetically detected biodiversity

### 2.2.1 *Crop wild relatives and genetic resources for food security*

#### 2.2.1.1 Crop wild relatives and genetic resources for food security

##### 2.2.1.1.1 Brief summary of the developed pipeline from D6.1

The main objective of the crop wild relatives (CWR) pDT is to facilitate the identification and utilization of novel genetic resources from CWR through automating data flow, automated modelling runs, uncertainty analysis, and timely alerts on potential genetic resources of interest for plant breeders, policymakers, and conservation scientists. Our objective includes the creation of habitat suitability maps for all CWR with sufficient occurrence data. The pDT is designed to be adaptable across different crop species and traits, empowering users to address key research questions in pre-breeding, such as identifying geographic areas where populations of CWR harboring beneficial genetic traits for enhancing crop resilience to environmental stress are potentially growing. Additionally, the pDT facilitates the assessment of gaps in existing collection efforts, aiding in the strategic planning of future genetic resource collections.

To provide the best experience of interaction with the pDT for the end-users, the pDT will have a web interface based on the R Shiny (<https://rstudio.github.io/shiny/>) application. The interface will feature dropdown menus for:

- crops and their corresponding wild relatives,
- habitat suitability maps, and
- abiotic stress ranges among others.

This will allow users to effectively map the optimal overlap between environmental stress factors and habitat suitability to identify geographic areas where populations resilient to stress can potentially thrive. Plant breeders can utilize the genetic resources of these populations to develop crops with high resilience to climate change driven environmental stress.

##### 2.2.1.1.2 Implemented approaches for upscaling

For deployment of the model on LUMI, the R environment has been containerized with Docker and the container image can be pulled and executed on the CPU partition of the LUMI supercomputer through Apptainer / Singularity. Initial tests have been run on LUMI-C with this setup by using parallelization within the R code. The current implementation is not scalable across multiple nodes, and additional parallelization outside model code is planned to be implemented.

##### 2.2.1.1.3 Challenges

The CWR pDT aims to run tens of thousands of CWR species using different algorithms and model replications. As the different model runs are independent, they can be efficiently executed in parallel, resolving this computational challenge. The primary challenge lies in the fact that while some stress factors are completely unavailable, others are only available with either coarse spatial and thematic resolutions or reduced accuracy.

##### 2.2.1.1.4 Analysis of relative performance improvements

The large parallel computing capacity of LUMI-C is expected to be highly suitable for achieving the aimed large-scale model processing once the improved parallelization has been implemented. In case of smaller workloads, the containerised solution is directly executable also on cloud environments. The pDT aims to regularly run models for species with updated data. Running models for several thousand species is not easily achieved on laptop or desktop computers. The main advantage of using LUMI is to run these models in parallel to obtain output in a reasonable period of time.

#### 2.2.1.1.5 Model source code release

The developed model code and execution scripts are published under the BioDT GitHub organization <https://github.com/BioDT>.

### 2.2.2 DNA detected biodiversity, poorly known habitats

#### 2.2.2.1 Genetically detected biodiversity in cryptic habitats – prioritisation of sampling effort

##### 2.2.2.1.1 Brief summary of the developed pipeline from D6.1

No code has been produced so far for the model itself. The envisioned pDT model will leverage existing data to optimize the placement of future biodiversity samples based on user-defined constraints such as geographic, landscape, and taxonomic factors, along with prioritization parameters like community heterogeneity. It will use inputs like OTU tables to calculate metrics for ranking and prioritizing potential sampling areas. The user interface allows customization of prioritization criteria and visualizes prioritized areas on an interactive map, facilitating strategic decisions for biodiversity sampling and addressing knowledge gaps within certain constraints.

A web-based tool (<https://edna-tool.gbif-uat.org/>) has been developed and is being tested for facilitating uptake of more data (eDNA metabarcoding data) relevant for the model in GBIF. This tool is now being tested and used by early adopters.

##### 2.2.2.1.2 Implemented approaches for upscaling

None for the pDT model itself.

The supporting tool for formatting/publishing data already scales to realistic dataset sizes.

##### 2.2.2.1.3 Challenges

Certain computational challenges with growing data-files are expected if we use the matrix format (species by sites). BIOM files may solve this, or alternative approaches.

##### 2.2.2.1.4 Analysis of relative performance improvements

None have been performed.

##### 2.2.2.1.5 Model source code release

No code has been produced for the pDT model. The code for the associated data formatting tool is fully openly available in GitHub (<https://github.com/gbif/edna-tool-ui> and <https://github.com/gbif/edna-tool-backend>).

#### 2.2.2.2 DNA detected biodiversity, poorly known habitats - phylogenetic diversity

##### 2.2.2.2.1 Brief summary of the developed pipeline from D6.1

Phylogenetic diversity (PD) measures the evolutionary history embodied in a set of taxa, highlighting the importance of conserving biodiversity as a storehouse of potential future benefits for humanity. It offers a more evolutionary perspective on biodiversity compared to traditional taxonomic richness, which counts species within a genus or area. The PhyloNext pipeline utilizes GBIF data and the Open Tree of Life to compute PD metrics, aiming to facilitate conservation efforts and policy making by identifying areas of high evolutionary diversity. However, current limitations include long computation times and lack of interactive visualization capabilities. The envisioned pDT model would consist of this pipeline, documented and enhanced with modern computing infrastructures, and is seen as a potentially valuable tool for exploring and visualizing conservation strategies, with future improvements poised to increase its usability and



effectiveness in conservation planning. The development would be based on improving an existing prototype GUI.

#### 2.2.2.2.2 Implemented approaches for upscaling

Nothing implemented yet as of May 2024. PhyloNext, leveraging containerization technologies like Docker, was successfully installed on the pre-exascale supercomputer LUMI, with initial tests proving successful. This pipeline is highly adaptable to both HPC and cloud environments, offering flexibility in resource allocation per task via SLURM or dedicating a fixed node for optimized operations. It supports S3-compatible object storage for improved data access speed and includes a resource usage profiler for optimizing resource demands. Additionally, the integrated Biodiverse program uses optimizations like caching to speed up complex calculations, further enhancing the pipeline's efficiency.

#### 2.2.2.2.3 Challenges

Some of the calculations in the Biodiverse program may be bottlenecks and difficult to scale up.

#### 2.2.2.2.4 Analysis of relative performance improvements

None as of May 2024.

#### 2.2.2.2.5 Model source code release

The original PhyloNext code and pipeline is hosted at GitHub (Mikryukov et al., <https://phylonext.github.io>). No additional code has been developed so far. Source code for the preliminary GUI is openly available in two GitHub repositories (backend: <https://github.com/gbif/phylonext-ws>; frontend: <https://github.com/gbif/phylonext-ui>).

#### 2.2.2.2.6 References

Mikryukov V, Abarenkov K, Laffan S, Robertson T, McTavish EJ, Stjernegaard Jeppesen T, Waller J, Blissett M, Kõljalg U, Miller JT. PhyloNext: a pipeline for phylogenetic diversity analysis of GBIF-mediated data. URL: <https://phylonext.github.io/>

## 2.3 Dynamics and threats from and for species of policy concern

### 2.3.1 Invasive species

#### 2.3.1.1 Invasive Alien Plant Species Dynamics

##### 2.3.1.1.1 Brief summary of the developed pipeline from D6.1

All model input data are currently ready to be used in the models. This includes species occurrence records (GBIF<sup>3</sup>, EASIN<sup>4</sup>, and eLTER<sup>5</sup>), selected climate variables (CHELSEA<sup>6</sup>), habitat information (derived from Corine land cover<sup>7</sup>), and other predictors (e.g., road and railway intensity as a proxy for site accessibility and invasive alien species (IAS) dispersal routes). Spatially explicit joint Species Distribution Models (jSDMs) are fitted at the habitat level (i.e., a separate model per habitat type) using the HMSC-R package (Tikhonov et al., 2020a, 2024). The data pipelines follow the Dynamic Data Driven Application Systems paradigm (DDDAS), including design concepts like feedback loops and state management (Darema et al., 2023). The feedback loop components check for “new” data, and the state management components manage things like versioning

---

<sup>3</sup> <https://www.gbif.org/>

<sup>4</sup> <https://easin.jrc.ec.europa.eu/>

<sup>5</sup> <https://elter-ri.eu/>

<sup>6</sup> <https://chelsa-climate.org/>

<sup>7</sup> <https://land.copernicus.eu/en/products/corine-land-cover>  
<https://www.biodt.eu/>

and the state of data within the pDT. jSDMs will run in LUMI using singularity containers as the feedback loops sense new data or periodically at six-month intervals.

#### 2.3.1.1.2 Implemented approaches for upscaling

The data workflows are implemented as Directed Acyclic Graphs (DAGs) inside LUMI'S SLURM task scheduling systems using the PyDoit<sup>8</sup> library. The workflow DAG is organised in this pDT with a hierarchy of Tasks and Actions. Tasks are individual nodes in the DAGs and can run in parallel. Each Task can have multiple Actions, which run sequentially. In this pDT, there is a separate Task for each data source that is submitted to the LUMI SLURM system.

Spatially explicit jSDMs are memory intensive and require high processing time, particularly for models run at large spatial scales like Europe. Therefore, initial models on a subset of species and a small study area were tested locally first (on a Windows server PC) before the implementation of the full data on LUMI. Based on initial models, we found it would be difficult to calibrate models with a full Gaussian process (GP), even for a small study area. We decided to only work with Gaussian predictive process (GPP) models (Tikhonov et al., 2020b).

Our next step is to upscale these models on LUMI. Timely, a recent extension to HMSC has become available. The new extension (HMSC-HPC; Ur Rahman et al., 2024) provides a GPU-compatible Python implementation of the HMSC models using the TensorFlow library. The HMSC-HPC can accelerate the model fitting time up to 1000 times (Ur Rahman et al., 2024). The HMSC-HPC extension seems very promising to reduce the running time of our models on LUMI, which is limited to 2-3 days maximum wall time per SLURM job.

#### 2.3.1.1.3 Challenges

Although LUMI provides ample computational resources necessary for running a complex digital twin workflow, a possible limitation could rather come from the management of the computational project of the pDT on LUMI. Projects on LUMI have by default a 1-year shelf life, and then the code and data need to be moved to a renewed project. This would present a challenge to achieving true automation for the pDT, as a manual intervention is required to do the migration. Migrations could be tedious in some cases, and might even become a source for bugs in the workflow code if the underlying LUMI environment or assigned computational resources change from project to project.

#### 2.3.1.1.4 Analysis of relative performance improvements

We fitted initial models for species associated with forest habitat type in Germany (90 species, 3581 sampling units (i.e., locations), 9 covariates, 1 spatial random variable (GPP)). HMSC-R models for these data took ca. 40 hours per chain on parallel (4 MCMC chains); while using HMSC-HPC on LUMI-CPU took c.a. 100 minutes per chain on average. LUMI-GPU for the same data took only ca. 20 minutes per chain; i.e., 120-time speed up on average as compared to HMSC-R for this sample data. We see the use of the HMSC-HPC on LUMI-GPU as very promising to speed up our model fitting when upscaling to the full species list at the full European spatial extent.

#### 2.3.1.1.5 Model source code release

The code for the HMSC R package is available at <https://github.com/hmsc-r/HMSC>. The HMSC-HPC extension is available at <https://github.com/aniskhan25/hmsc-hpc>. Source code of the IAS-pDT workflow (including scripts for the models) are currently available in a private Git repository for the development phase and will become publicly available at the BioDT GitHub organization (<https://github.com/BioDT>).

---

<sup>8</sup> <https://pydoit.org/>  
<https://www.biodt.eu/>

#### 2.3.1.1.6 References

Darema, F., Blasch, E. P., Ravela, S., & Aved, A. J. (2023). The Dynamic Data Driven Applications Systems (DDAS) Paradigm and Emerging Directions. *Handbook of Dynamic Data Driven Applications Systems: Volume 2*, 1-51.

Tikhonov G., Opedal OH., Abrego N., Lehtikoinen A., de Jonge MMJ., Oksanen J., Ovaskainen O. (2020a). Joint species distribution modelling with the r-package Hmsc. *Methods Ecol Evol* 11 (3): 442-447. <https://doi.org/10.1111/2041-210X.13345>

Tikhonov G., Duan L., Abrego N., Newell G., White M., Dunson D., Ovaskainen, O. (2020a). Computationally efficient joint species distribution modeling of big spatial data. *Ecology* 10(2): e02929. <https://doi.org/10.1002/ecy.2929>

Tikhonov G., Ovaskainen O., Oksanen J., de Jonge M., Opedal O., Dallas T. (2024) Hmsc: Hierarchical Model of Species Communities. R package version 3.0-14, <https://www.helsinki.fi/en/researchgroups/statistical-ecology/software/hmsc>.

Ur Rahman A., Tikhonov G., Oksanen J., Rossi T., Ovaskainen O. (2024) Accelerating joint species distribution modeling with Hmsc-HPC: A 1000x faster GPU deployment. *bioRxiv* 2024.02.13.580046; doi: <https://doi.org/10.1101/2024.02.13.580046>

## 2.4 Species interactions with each other and with humans

### 2.4.1 Disease outbreaks

#### 2.4.1.1 Wild boar–domestic pig wildlife disease modeling and pathogen spillover use case

##### 2.4.1.1.1 Brief summary of the developed pipeline from D6.1

The disease outbreaks pDT aims to inform data-driven responses to manage spread of wildlife diseases, specifically African swine fever (ASF) in European wild boar populations. The implemented model is a stochastic, spatially- and temporally- explicit wild boar–ASF mechanistic model on a structured landscape (Lange and Thulke, 2017). The core code is written in Rust (<https://www.rust-lang.org/>) as a modularized entity component system (ECS) such that all model sub-modules have the same interface and structure. A Python (<https://www.python.org/>) wrapper allows for custom configuration of the sub-modules. The initial pDT runs the model using a default configuration that will be made available with project documentation.

##### 2.4.1.1.2 Implemented approaches for upscaling

The process of upscaling to LUMI has not yet begun. The team is in the initial stages of model migration to LUMI.

##### 2.4.1.1.3 Challenges

For full model functionality several model development goals require implementation (programming) in the local environment, the most notable of those goals including (1) automation of ingestion and processing of user-provided barrier data (shapefiles) and (2) standardization of pDT model outputs. That latter goal requiring consultation with the full pDT and external collaborators to reach an agreement on the full suite of outputs (static and dynamic) to be shared with pDT users. Once migrated to LUMI, additional coding will be necessary as testing multiple virus barrier scenarios will require parallelization, a feature not currently integrated into the code. Consequently, additional coding will be necessary post-migration to LUMI.

##### 2.4.1.1.4 Analysis of relative performance improvements

The disease outbreaks model is not computationally intensive and performs well in a local environment utilising an Intel i7 vPro 8th generation HP laptop running Windows. The core of the simulation, written in Rust as ECS, ensures uniformity in interface and structure across all sub-modules. Python wrappers facilitate <https://www.biodt.eu/>

easy customization of these sub-modules, enhancing flexibility in configuring simulation parameters. The code's design lends itself well to future migration to the LUMI environment. Noteworthy runtime statistics reveal an overall runtime of 8.07 seconds, with a per-tick runtime of 15.51 milliseconds across 520 ticks, underscoring the efficiency and potential scalability of the model. As model migration to LUMI has not yet happened, the model has yet to be tested in the LUMI environment and no comparison of performance can be made at this time.

#### 2.4.1.1.5 Model source code release

Model components are modularised and well-documented (Lange and Thulke, 2017). Full model code is currently available by invitation only on the Helmholtz AAI, although full documentation is openly available for the Rust modules ([https://ecoepi.eu/ecoepi/supl/ODD-swifco-rs/rust/swifco\\_rs/index.html](https://ecoepi.eu/ecoepi/supl/ODD-swifco-rs/rust/swifco_rs/index.html)) as well as the Python wrappers (<https://ecoepi.eu/ecoepi/supl/ODD-swifco-rs/index.html>). Model components implemented for the pDT are currently available by invitation only at the BioDT GitHub organization (<https://github.com/BioDT>).

#### References

Lange M., Thulke H.H., 2017. Elucidating transmission parameters of African swine fever through wild boar carcasses by combining spatio-temporal notification data and agent-based modelling. *SERR* 31:379-391. doi:10.1007/s00477-016-1358-8

## 2.4.2 Pollinators

### 2.4.2.1 Honey bee dynamics in agricultural landscapes

#### 2.4.2.1.1 Brief summary of the developed pipeline from D6.1

The Honeybee prototype Digital Twin will allow to run the simulation model BEEHAVE (Becher et al., 2014) for any point in Germany using landcover data (Preidl et al., 2020) and weather data from the Deutscher Wetterdienst (German Meteorological Service). The model computes the number of adult honey bees, the stored honey, number of flights and the number parasitic mites on a daily resolution. If the model is applied several thousand times, maps for bee vitality and honey production on the national scale can be produced. Thus, upscaling means in this project extending single local application to the national scale. Future work will expand this workflow to other countries towards the European scale.

#### 2.4.2.1.2 Implemented approaches for upscaling

The simulation experiments are specified and executed by R scripts using the *nlr*x package (Salecker et al., 2019). The software required for executing the model (NetLogo, Java, R with *nlr*x and other packages) have been bundled in a Docker container image that can be pulled and executed on the CPU partition of the LUMI supercomputer through Apptainer / Singularity and on a cloud through Docker. The execution of the containerized BEEHAVE model has been parallelized on LUMI over individual inputs by using HyperQueue task scheduler (<https://github.com/It4innovations/hyperqueue>).

#### 2.4.2.1.3 Challenges

The main computational challenge of this use case is the large amount of input data, parameters, and/or scenarios requiring processing with the BEEHAVE model. As such individual BEEHAVE simulations are independent, they can be executed in parallel, which is efficiently achieved with the implemented parallelization approach.

#### 2.4.2.1.4 Analysis of relative performance improvements

As an exploratory study, we calculated a prediction of the number of surviving bees and honey storage for a three-year period using a regular grid spanning around 3,500 locations in Germany, based on the surrounding land cover types and weather data. By utilizing the developed parallelization scheme, this calculation took about an hour on eight LUMI-C nodes. As a rough estimate, the same calculation would have taken more

than a week on a laptop. While the run configuration on LUMI requires still optimization for maximum efficiency, it is clear that the capability to execute the pDT in parallel over hundreds or thousands of cores and to leverage the large computing capacity of LUMI-C is highly advantageous.

#### 2.4.2.1.5 Model source code release

The model code, developed parallelization scripts as well as pre- and post-processing scripts are published under the BioDT GitHub organization <https://github.com/BioDT>.

#### 2.4.2.1.6 References

Becher, M.A., Grimm, V., Thorbek, P., Horn, J., Kennedy, P.J., Osborne, J.L., (2014). BEEHAVE: a systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure. *Journal of Applied Ecology* 51(2) 470-482.

Preidl, S., Lange, M., & Doktor, D. (2020). Introducing APiC for regionalised land cover mapping on the national scale using Sentinel-2A imagery. *Remote Sensing of Environment*, 240, 111673.

Salecker, J., Sciaini, M., Meyer, K. M., & Wiegand, K. (2019). The nlrx r package: A next-generation framework for reproducible NetLogo model analyses. *Methods in Ecology and Evolution*, 10(11), 1854-1863.

## 3. Conclusions and lessons learnt

### 3.1 Conclusions

The current development status of the pDTs has been largely determined by how well developed and established the used models were already at the start of the project. Hence, also the development of pipeline scripts and upscaling of models is most advanced for pDTs where models existed (e.g. pDTs focusing on biodiversity dynamics or pollinators) while other pDTs have conceptualised their pipelines with delayed implementation of corresponding scripts and upscaling approaches. This fundamental difference in the development of DTs addressing various aspects of biodiversity was anticipated from the beginning of the project, serving as a tool to outline a wide variety of challenges that may arise in the process as well as possible solutions to overcome them.

The development nature of the project and the uniqueness of pDTs have resulted in that the pDTs have worked rather independently from each other with limited shared generic building blocks. This is partly because there were no complete example pDT workflows before the work started. The developed codes and workflows could be generalized and harmonized among pDTs to obtain synergies in maintenance as well as base for future developments.

To summarize the found technical solutions, it is beneficial to utilize containerization for model deployment and task parallelization for model upscaling where possible and sufficient for the pDT. GPU-acceleration has significant potential for improving model performance, but it typically requires considerable amount of work and reimplementation of the model, and hence it is limited to applications with largest gains. The rationale behind these conclusions is elaborated in the following.

### 3.2 Lessons learnt on model upscaling

Before upscaling, the models and scripts need to be deployed on the HPC environment like the pre-exascale EuroHPC LUMI supercomputer, which is the main HPC platform for BioDT. For many use cases, the deployment has been efficiently done by containerizing the models. This has been especially useful for models that rely on programming frameworks that are uncommon in HPC environments as well as for maintaining R or Python environments comprising of numerous packages. Containers are also performance-wise practical to reduce the Lustre file system load caused otherwise by numerous small file accesses in R or Python environments. As further added benefit, containers provide versioning and reproducible runtime environment, making them transferrable to different HPC systems and project spaces. Containers can be built using multiple frameworks and the chosen approach has been Docker as it is directly compatible with execution on cloud environment for the use cases that do not need extensive computing resources. This approach is also compatible with HPC execution as the Docker containers can be seamlessly converted to Singularity / Aptainer that is the container execution framework on LUMI.

Model upscaling approaches depend on the case to case, but as a general approach, handling parallelization outside the model code has been found efficient for use cases requiring processing of a large number of independent input data, parameters, and/or scenarios. This approach has a significant benefit that extensive rewriting of scientific models is not needed, but the scientist can keep using their existing codes and programming languages they are most familiar with, which ensures long-term sustainability of the HPC usage. The approach is also modular as the task scheduler (such as HyperQueue or GNU Parallel) can be changed as needed without changes to the model code. However, this approach is limited to upscaling with CPU resources only unless the model code supports GPU execution.

GPU-acceleration of existing model codes is a considerable reimplementation effort but can result in significant gains in model runtime. This has been done in collaboration with other projects for the HMSC model used by multiple use cases as well as by larger scientific community.

Multi-scale model parameterizations are a useful approach when transferring pDT pipelines (and esp. models) of local sites to other sites or to larger regions without having detailed knowledge on how model parameters might change across regions. In such cases, transfer functions can be developed to estimate unknown model parameters from environmental conditions and to overcome computational efforts to calibrate local-scale models at each site stand-alone (which is often impossible esp. for sites with unavailable observation data). However, the approach still requires methodological research beforehand in order to identify the transfer functions. Therefore, the sites with available data still have to be calibrated at least once in order to correlate the so far unknown model parameters to environmental data like precipitation or soil properties. If transfer functions are known, a pDT pipeline and model can be easily transferred across sites and regions without the need of computationally intensive model calibration. However, such approaches are only useful for mechanistic, complex local-scale models and will rarely be used in the pDTs developed here.

### 3.3 Future work

For the pDTs with fully developed performant pipeline, the next steps will focus on model validation and improving the predictive performance of models, on further testing and refining the scripts, on making them more robust and permanent as well as on developing user interfaces and required functionalities. Model validation will be performed pDT-wise as storylines, pipelines and models differ significantly from each other, while experiences and tools in applying methodologies (e.g. global sensitivity analysis) will be shared among the pDTs. Especially for designing user interactions, an important step for all pDTs is the involvement of end users by letting them try and use the pDTs for their own questions and data. Two workshops dealing with three pDTs (Grassland Biodiversity Dynamics, Crop Wild Relatives, Honeybee vitality) have already been held in November 2023 and January 2024.