Received: 1 October 2021 | Accepted: 24 August 2022

DOI: 10.1111/2041-210X.14034

Methods in Ecology and Evolution

RESEARCH ARTICLE

Realising the Promise of Large Data and Complex Models

Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation

Ceres Barros[1] | Yong Luo[1,2,3] | Alex M. Chubaty[4] | Ian M. S. Eddy[2] | Tatiane Micheletti[1] | Céline Boisvenue[1,2] | David W. Andison[5] | Steven G. Cumming[6] | Eliot J. B. McIntire[1,2]

[1]Faculty of Forestry, University of British Columbia, Vancouver, British Columbia, Canada; [2]Canadian Forest Service (Pacific Forestry Centre), Natural Resources Canada, Victoria, British Columbia, Canada; [3]Forest Analysis and Inventory Branch, BC Ministry of Forests, Lands, Natural Resource Operations and Rural Development, Victoria, British Columbia, Canada; [4]FOR-CAST Research & Analytics, Calgary, Alberta, Canada; [5]Bandaloop Landscape-Ecosystem Services Ltd., Nelson, British Columbia, Canada and [6]Faculté de Foresterie, de Géographie et de Géomatique, Département des Sciences du Bois et de la Forêt, Pavillon Abitibi-Price, Université Laval, Québec, Canada

Correspondence
Ceres Barros
Email: ceres.barros@ubc.ca

Funding information
Alberta Innovates; Environment and Climate Change Canada; fRI Research; Mitacs Accelerate Award, Grant/Award Number: IT10289; Mitacs Elevate Fellowship, Grant/Award Number: IT12391; Forest Resources Improvement Association

Handling Editor: Rachel McCrea

Abstract

1. Modelling is widely used in ecology and its utility continues to increase as scientists, managers and policy-makers face pressure to effectively manage ecosystems and meet conservation goals with limited resources. As the urgency to forecast ecosystem responses to global change grows, so do the number and complexity of predictive ecological models and the value of iterative prediction, both of which demand validation and cross-model comparisons. This challenges ecologists to provide predictive models that are reusable, interoperable, transparent and able to accommodate updates to both data and algorithms.
2. We propose a practical solution to this challenge based on the PERFICT principles (frequent Predictions and Evaluations of Reusable, Freely accessible, Interoperable models, built within Continuous workflows that are routinely Tested), using a modular and integrated framework. We present its general implementation across seven common components of ecological model applications—(i) the modelling toolkit; (ii) data acquisition and treatment; (iii) model parameterisation and calibration; (iv) obtaining predictions; (v) model validation; (vi) analysing and presenting model outputs; and (vii) testing model code—and apply it to two approaches used to predict species distributions: (1) a static statistical model, and (2) a complex spatiotemporally dynamic model.
3. Adopting a *continuous* workflow enabled us to *reuse* our models in new study areas, update predictions with new data, and re-parameterise with different *interoperable* modules using *freely accessible* data sources, all with minimal user

input. This allowed *repeating predictions* and automatically *evaluating* their quality, while centralised inputs, parameters and outputs, facilitated ensemble forecasting and tracking uncertainty. Importantly, the integrated model validation promotes a continuous evaluation of the quality of more- or less-parsimonious models, which is valuable in predictive ecological modelling.

4. By linking all stages of an ecological modelling exercise, it is possible to overcome common challenges faced by ecological modellers, such as changing study areas, choosing between different modelling approaches, and evaluating the appropriateness of the model. This ultimately creates a more equitable and robust playing field for both modellers and end users (e.g. managers), and contributes to position predictive ecology as a central contributor to global change forecasting.

KEYWORDS
dynamic and nondynamic ecological modelling, forest stand dynamics, predictive ecology, *R*, reproducible and continuous workflows, species distribution models, tutorial

1 | INTRODUCTION

Predictive models have been used by ecologists for decades and their utility is expanding to meet the increasing need for ecological forecasts that can inform ecosystem management and conservation (Bodner et al., 2021; Dietze, 2017). As a result, many applied ecological models have been developed using different approaches—for example statistical (Taylor & White, 2020), dynamic simulation (Seidl et al., 2011)—and scales—for example population ranges (Stewart et al., 2020), regional vegetation dynamics (Barros et al., 2017), biome productivity models (Boisvenue & Running, 2010), among others. Despite these efforts, there is increasing demand for greater model complexity at finer spatial resolutions (DeAngelis & Yurek, 2017), for cross-model comparisons, ensemble forecasts and estimates of forecasting uncertainty (Bodner et al., 2020; Shifley et al., 2017). Addressing these demands is in part challenged by disconnection between the steps involved in applied ecological modelling and by gaps between ecologists and software developers.

Most applied ecological modelling projects have in common several steps—acquiring and processing data, parameter estimation, executing the models, evaluating model performance, analysing outputs and reporting—that can be difficult to integrate seamlessly into one workflow. For example, data processing and evaluating model performance are usually performed independently from the parameterisation phase. Consequently, transferring a model to a different context (e.g. new study area), or altering a model's initial conditions (Shifley et al., 2017) can be time-consuming and challenging, particularly to others than the original developers (Borregaard & Hart, 2016). Additionally, in the realm of complex dynamic ecological models, few ecologists or teams have both the required ecological expertise and the necessary programming skills and/or support to develop new models or improve existing ones (Vedder et al., 2021). This results in (i) model implementations that are difficult for most ecologists to understand, change or improve (Thiele & Grimm, 2015); (ii) a lack of contribution to modelling efforts from the wider community of ecologists; and (iii) the need for specialised software engineers, which can limit continuous model development and maintenance to well-funded teams (Vedder et al., 2021).

These challenges have limited the reproducibility and transparency (sensu Powers & Hampton, 2019) of modelling workflows and, therefore, their reusability and durability. This makes **iterative forecasting** (sensu Dietze, 2017; see bold terms in Glossary, Table 1) cross-model comparisons, and evaluations of model predictions harder (Schuwirth et al., 2019), distancing predictive ecology from ensemble forecasting of ecosystem responses to global changes—an approach used in climate forecasting (IPCC, 2022). Many have called for alternative ways of addressing ecological modelling problems through open, flexible and reproducible frameworks (Briscoe et al., 2019; Ellison, 2010; Powers & Hampton, 2019), which have been used in the context of nondynamic ecological models (Taylor & White, 2020; White et al., 2019). Yet, bringing these principles together in a **continuous workflow** and across modelling paradigms (i.e. **dynamic** and **static models**) will be key to overcoming barriers to transparency and reusability of predictive ecological models (Fer et al., 2021; Vedder et al., 2021). As a solution, McIntire et al. (2022) proposed the **PERFICT** principles—making frequent Predictions and Evaluations of Reusable, Freely accessible, Interoperable models, built within Continuous workflows that are routinely Tested—to integrate all steps of a modelling exercise, including data preparation, geospatial processing, estimating model parameters, calibration, prediction, **validation**, visualisation and analysis of simulation results, and model **code testing**.

Here, we present two complete, real-world implementations of the PERFICT approach, focused on predicting species distributions: a single-species nondynamic (i.e. static) empirical model and a complex, multi-species, dynamic simulation model—both applied to boreal tree species in Canada. We identify some of the concrete benefits that emerged from our workflow, namely higher model transferability

TABLE 1 Glossary of key terms.

Term	Description
Caching	The ability to store the results of an operation in cache memory, to avoid re-running algorithms whose inputs have not changed
Code testing	Testing the code of a model for potential errors that can lead to breakage, or abnormal results
Continuous workflow	A workflow that does not need the user's direct intervention during its execution
Dynamic vs. static (nondynamic) ecological models	Here, we distinguish two ecological modelling paradigms: nondynamic (static) ecological models and dynamic ecological models. Dynamic models involve time dependency, for example this year's predictions are next year's inputs. An example of static modelling is to predict species distributions from a statistical model relating observed occurrences and environmental conditions. An example of dynamic modelling is to predict species distributions from a spatially explicit age-based population model
FAIR data	Findable, Accessible, Interoperable and Reusable data (Wilkinson et al., 2016) that enhance both model transferability and reproducibility
Iterative forecasting	The act of updating a model's forecasts several times and as new data become available
Interoperable model/workflow	The ability to remove, replace or add model or workflow sub-components, because these can operate together in different combinations
LandR and LandR Biomass	LandR is a family of *SpaDES* modules geared towards simulating forest dynamics, with several parameterisation modules focused on Canadian boreal and montane forests, simulating fire disturbances and climate change impacts on forest and fire dynamics (among others currently being developed). Not to be confused with the LandR *R* package, which groups *R* functions used across LandR modules. LandR Biomass is a group of these modules focused on forest dynamics and their parameterisation
Model validation	Testing whether a model's outputs are ecologically accurate, by comparing them with out-of-sample data
Modular workflow/model	A workflow or model whose sub-components can be easily changed and turned on or off (e.g. changing a parameterisation approach or the input data), a necessary condition for interoperability
Nimble workflow/model (nimbleness)	A workflow or model that can be easily transferred into other contexts, such as a different geographical area or changing the parameterisation, prediction and validation approaches and algorithms
PERFICT	Seven principles that should improve the reusability, transparency and nimbleness of ecological models, while bridging gaps between data, models and decisions: frequent Predictions and Evaluations of Reusable, Freely accessible, Interoperable models, built within Continuous workflows that are routinely Tested (McIntire et al., 2022)
SpaDES	A group of *R* packages (*R* meta-package) that provides a toolkit for developing ecological models. Although geared towards spatiotemporally explicit modelling, it can accommodate any type of model as long as it can be written in *R* or any language that *R* can call (e.g. *Python*, *Java*, *C++*)
SpaDES event	A section of a module usually contained in an *R* function that executes a relatively self-contained process (e.g. tree cohort growth, or fitting a species distribution model)
SpaDES model/workflow	Usually a group of *SpaDES* modules that work together as a "model" to generate predictions. *SpaDES* models can be static or dynamic depending on the type of processes their modules include
SpaDES module	Usually a self-contained piece of a *SpaDES* model or workflow that encapsulates a step, process or mechanism in the model. For instance, parameter estimation for a simulation module can be contained in a single 'data module' (see Box 1). Here, we distinguish 'data' and 'calibration' modules, which are focused on preparing data and inputs for another module; 'prediction' and 'simulation' modules, which run static or dynamic predictions, respectively; and 'validation' modules, which validate predictions against out-of-sample data

(changing study area and parameterisation approach) and more efficient model validation, model selection and ensemble forecasting.

2 | MATERIALS AND METHODS

2.1 | Workflow overview

In the PERFICT approach, ecological model development integrates software dependency management, data acquisition and treatment, model parameterisation, calibration and validation, and the analysis and presentation of results within a continuous workflow that facilitates routine testing and wider scrutiny (Box 1). McIntire et al. (2022) also emphasise the importance of **interoperability** (via modularity and standardisation) to increase model longevity and accelerate scientific progress. PERFICT workflow development is thus **modular** and implies the use of a commonly understood programming language and platform (McIntire et al., 2022)—hosted accessibly and linked to all associated data. We present the implementation of these principles using two ecological forecasting examples in Canadian boreal forests, and through seven common components of ecological modelling applications: (i) the modelling toolkit; (ii) data acquisition and treatment; (iii) model parameterisation and calibration; (iv) obtaining predictions; (v) model validation; (vi) analysing and presenting model outputs; and (vii) testing model code.

2.2 | Modelling toolkit

We use the ***SpaDES*** toolkit (Spatial Discrete Event Simulation; Chubaty & McIntire, 2019; https://SpaDES.predictiveecology.org/) and *R* (R Core Team, 2022) to implement the PERFICT principles. *R* is widely used in ecology for data manipulation, analysis, simulation and presentation (Hesselbarth et al., 2021). It interfaces with other languages (e.g. *C++*, *Python*, *Java*), and provides analytical (e.g. statistical, geospatial and graphical) and development tools (e.g. benchmarking, caching, package management) that support integrated and reusable workflows. *SpaDES* facilitates building continuous, interoperable and reusable models via a standardised structure composed of **modules**, each being an *R* script that implements distinct and semi-autonomous algorithms through **event** scheduling (Box 1). It empowers ecological modellers with tools that establish explicit links with data, with package versioning and **caching** tools, and with automatic simulation setup, replication and saving—all of which enhance model transferability and **nimbleness** (Box 1).

2.3 | Input data and parameterisation

Model data and parameters can be provided by a range of methods from simple (e.g. expert-supplied parameter tables) to sophisticated (e.g. calibration by extensive bootstrapping or simulation approaches). The PERFICT approach is to integrate all such steps within the modelling workflow. The *SpaDES* toolkit promotes this in two ways. First, 'data/calibration modules' script and document all data-processing (default data sources, data downloading and loading into *R*, formatting and cleaning, quality control, etc.) and calibration steps (e.g. frequentist, Cobos et al., 2019, or Bayesian parameter estimation, Speich et al., 2021) that create inputs to the 'prediction/simulation modules' (Box 1). Second, default input objects and parameter values are provided to ensure all modules can run correctly with minimal user intervention (Box 1). Input objects may come from external data (e.g. online repositories specified in the metadata and accessed automatically), be self-generated, or directly supplied by the user. Default parameter values are specified as part of the module metadata, but may also be estimated by the module or supplied by the user. When caching (Box 1) and **FAIR** data (Findable, Accessible, Interoperable and Reusable data; Wilkinson et al., 2016) are used, a *SpaDES* workflow can update and re-parameterise itself automatically when inputs change (e.g. when datasets are updated or the study area changes) or internal algorithms change (e.g. changing the calibration approach).

2.4 | Obtaining model predictions

The principle of interoperability can be extended to model prediction. This is not in itself novel. Species-distribution-modelling tools exist to fit, compare and combine different statistical algorithms (e.g. DISMO *R* package; Hijmans et al., 2021). Interoperability of complex dynamic algorithms should also facilitate cross-model comparisons (Fer et al., 2021) and selecting relevant factors and processes to predict the system's response (Topping et al., 2015). We implement predictive interoperability as either switchable components within a single prediction module (see Example 1 below), or as alternate modules implementing different mechanisms or hypotheses (e.g. swapping of fire models in Micheletti et al., 2021).

To ensure reproducible predictions, all steps leading up to model parameterisation and execution need to be themselves transparent and reproducible (two 'PERFICT' requirements). We integrate model setup, execution and post hoc analysis, in *R* control scripts. Scripting protocols ensure that all required packages are installed and loaded (Box 1), establish working directories, define modules used and, optionally, supply parameters and inputs to override module defaults (see Examples, below).

2.5 | Validation

Model validation, that is, assessing model output quality, is usually achieved by comparing outputs with observed data (van Vliet et al., 2016). An integrated and automated model validation facilitates re-assessing model quality when upstream data and algorithms change ('evaluate frequently' in PERFICT), and running cross-model comparisons if the validation approach is transversal to different prediction algorithms. In our workflow, we integrate generic validation approaches as an event within a module or as a separate module, so that they are executed automatically after predictions and can be used with different prediction algorithms.

2.6 | Analysis and presentation of results

We leverage *SpaDES* features that create complete simulation objects (*simLists*; Box 1) to enhance reproducibility, and to facilitate post hoc analyses and the inspection of parameters and outputs. For instance, we use *simLists* to consult model parameter values and the fit of statistical models used for calibration *after* the simulation and without needing to save objects to disk or to search through module code. We also embed visual outputs within our modules, which *SpaDES* can automatically plot and save.

2.7 | Testing the model code

Software testing—assessing the model's technical soundness (Osherove, 2013)—differs from model validation—assessing its ecological accuracy. Within the former, we distinguish assertions, unit tests and integration tests. Assertions and unit tests evaluate individual code components (typically the result of a function call) or check the structure of a particular object. Integration tests evaluate whether several model components or processes are integrated correctly. Assertions

BOX 1 Using *SpaDES* to achieve a PERFICT modelling workflow

Ecological modelling has often been spread across several independent environments (e.g. proprietary GIS software for spatial processing, statistical software for data analyses) and used manual intervention (e.g. data acquisition), which can lead to brittle workflows (Vedder et al., 2021). Our implementation integrates all these steps into a continuous and transparent workflow written in *R*, with version control (*git*), and automated testing (via GitHub Actions and CRAN; Figure B1a). Each step benefits from the same abstraction: each *expects* a set of inputs and *creates* a set of outputs identified by the modeller, and is wrapped into a stand-alone piece called a 'module'. Modules are chained into a single workflow or 'model', and reused in new configurations, provided that a module's expected inputs match another's outputs or externally supplied objects. We use the SpaDES *R* meta-package to enable this structure. The *SpaDES* toolkit standardises simulation workflow/model building and execution via *SpaDES*-modules (*R* scripts of known structure) that are integrated and executed automatically by *SpaDES* functions via an *R* control script (Figure B1b; see 6. below). Each module contains a metadata

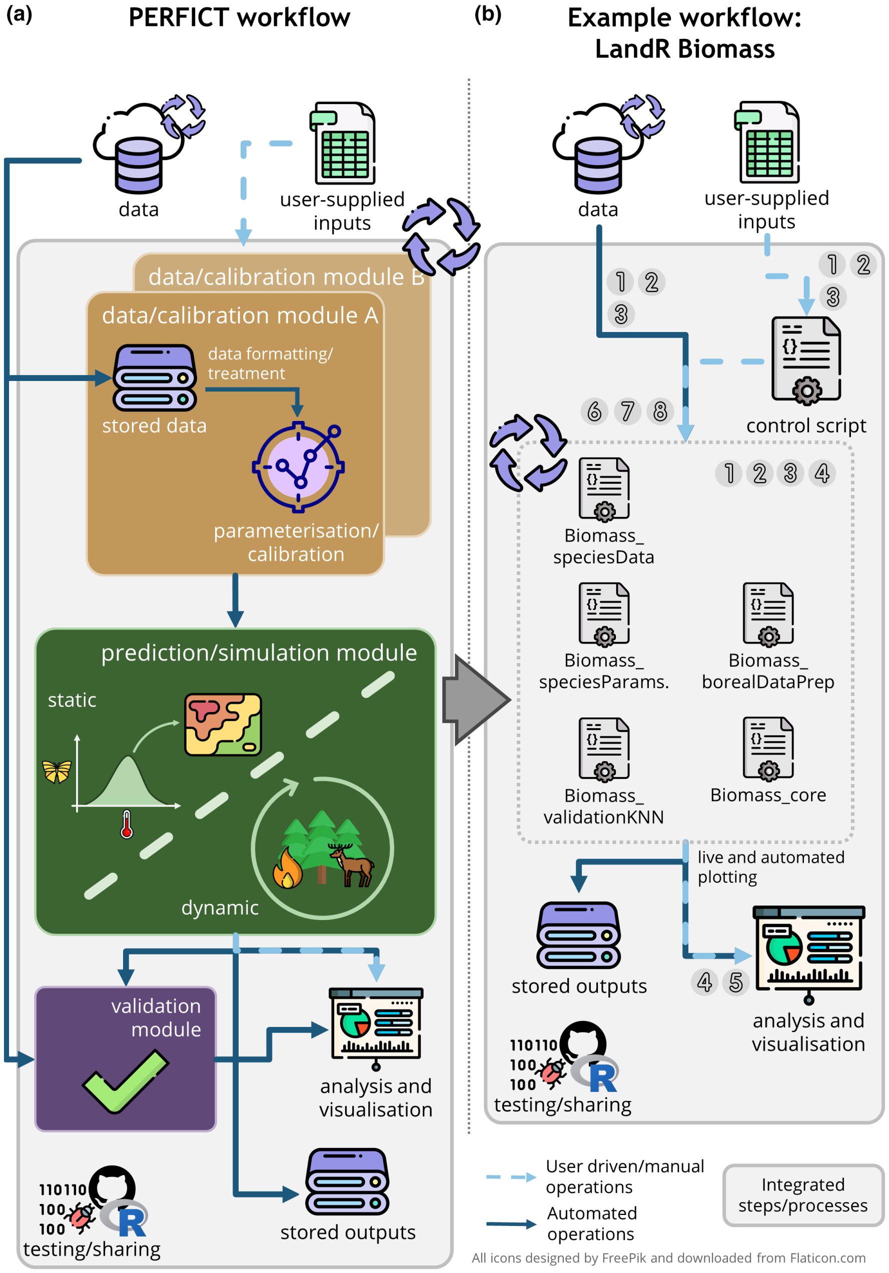
(a)
PERFICT workflow
(b)
Example workflow: LandR Biomass
data
user-supplied inputs
data/calibration module B
data/calibration module A
data formatting/ treatment
stored data
parameterisation/ calibration
prediction/simulation module
static
dynamic
validation module
analysis and visualisation
stored outputs
110110
100
100
testing/sharing
data
user-supplied inputs
1 2 3
1 2 3
control script
6 7 8
1 2 3 4
Biomass_ speciesData
Biomass_ speciesParams.
Biomass_ borealDataPrep
Biomass_ validationKNN
Biomass_core
live and automated plotting
stored outputs
4 5
analysis and visualisation
testing/sharing
User driven/manual operations
Automated operations
Integrated steps/processes
All icons designed by FreePik and downloaded from Flaticon.com

FIGURE B1 PERFICT workflow scheme (a) and its application with LandR Biomass (b). Numbers refer to tools enumerated in Box 1. LandR Biomass modules are briefly described in Table 2.

BOX 1 Continued

section, an event scheduler function and one or more events (algorithms that execute a given process, for example, spatial data processing or computing growth increments in a population model) that alter data objects shared with other modules (as in Fall & Fall, 2001). Module metadata specify all package dependencies, expected data objects ('input objects' in *SpaDES*) and parameter values ('parameters'), and created outputs. *SpaDES* analyses each module's metadata to deduce their dependency relationships in a model setup, and schedules and executes events across modules in the correct order (unless cycles exist, which require user-specified ordering). This bottom-up approach where modules describe their own dependencies, differs from top-down approaches where, for example, a project *Makefile* describes dependencies and sequencing (*Make*, GNU Project, 2020; *Make*-inspired *R* packages like TARGETS, Landau, 2021). This allows decentralised module development and execution, which facilitates running arbitrary combinations of modules, since module interdependencies and sequencing are re-deduced for each combination. Input objects can also be explicitly linked to online data by URLs in the metadata, enabling modules to download data automatically ("stored data" in Figure B1a) and satisfy their expected inputs when these are not supplied by an upstream module or user. Special 'save' events allow saving any module output, at any frequency, without changing module code. Embedded caching mechanisms allow saving the results of time-consuming steps (e.g. statistical modelling, geoprocessing) and avoid re-calculation unless inputs or algorithms change. When they do, these tasks are updated automatically (purple rotating arrows, Figure B1). Hence, when default data and parameters are general enough (geographically and ecologically), modules can re-estimate parameters in new locations, allowing complex models to be executed with little user intervention. Finally, *SpaDES* supports cross-platform portability by managing package dependencies via the REQUIRE package (McIntire, 2022; see 8. below).

Here, we distinguish three types of *SpaDES* modules: 'data/calibration modules', which prepare model inputs and parameters; 'prediction/simulation modules', which generate predictions using either static or dynamic mechanisms; and 'validation modules', which evaluate predictions against independent data. Modules can cross these boundaries or be structured in other ways—for example, a prediction module may perform calibration if fitting the predictive model prior to generating predictions (Example 1 in main text). Since the content of a module's events and metadata is arbitrary, *SpaDES* can accommodate any inter- and intra-module design and any data, parameterisation, validation and predictive model that can be written or interfaced in *R*. For example, a calibration module may use frequentist (e.g. Cobos et al., 2019) or Bayesian approaches (e.g. Speich et al., 2021), or may interface with a *python*-based model (using, e.g. the RETICULATE package; Ushey et al., 2022) that calibrates parameters.

Below, we enumerate some of the most useful tools we use when implementing our workflows, and Figure B1b is annotated with numbers indicating where they are most applied. Superscript letters denote *R* packages: 'rep' for REPRODUCIBLE (McIntire & Chubaty, 2021a), 'S' for SPADES. *CORE*, *'Se' for* SPADES. EXPERIMENT (McIntire & Chubaty, 2021b) and 'Req' for REQUIRE.

Data-related tools

1. *prepInputs*$^{\text{rep}}$—a function that sources/downloads and imports a dataset: it can also automate common geo-processing steps (e.g. cropping, masking and re-projection).
2. *Cache*$^{\text{rep}}$—a wrapper for other function calls that caches function results to secondary storage, with automatic retrieval if inputs and algorithms have not changed. Especially useful for long-running code (e.g. data preparation, parameter estimation).
3. *suppliedElsewhere*$^{\text{S}}$—detects whether an object is being supplied by the user or another module. This allows developers to specify module defaults, while maintaining flexibility to accept inputs from upstream sources. Supports interoperability.

Model object

4. *simList*$^{\text{S/Se}}$—an object class that contains a complete and executable model environment, including all modules with their parameters, dependencies, input and output data. All simulation objects are accessible from the *simList*, facilitating postsimulation inspection of any model outputs.

Modelling-related tools

5. *moduleDiagram*$^{\text{S}}$, *objectDiagram*$^{\text{S}}$—diagram and plot module dependencies, and the emergent data flows within and between modules.
6. *simInit*$^{\text{S}}$, *SpaDES*$^{\text{S}}$, *experiment2*$^{\text{Se}}$—initialise and run the simulation by executing all module events, using user-supplied parameters and inputs, if applicable. *experiment2* runs a simulation experiment (via multiple calls to *spades*), specified as sets of initial parameter values, model inputs, modules and levels of replication. It also organises outputs in a file system hierarchy following the experimental design, which facilitates using outputs for post hoc analyses (e.g. calculating summary statistics or combining results into an ensemble).
7. *restartSpaDES*$^{\text{S}}$—resumes an interrupted simulation. Invaluable during model development and debugging. It recreates the *simList* from current module code and state, so that execution can resume at the interrupted event. Enables developers to make and test code changes without re-running the entire workflow.

Package management tools

8. *Require*$^{\text{Req}}$—Checks and installs module package dependencies specified in metadata, handling version-specific installation and loading (unlike a standard *library*() call) from both CRAN and GitHub. It can also initialise full workflow libraries to a known state.

are simpler and faster to implement and are embedded in the model code, instead of running externally like unit tests (Sarma et al., 2016), and so can function as integration tests by detecting errors introduced by upstream processes. We follow the PERFICT principle of "routine testing", by embedding assertions that test both integration and data integrity across modules. Because this adds overhead, our modules can skip assertions for production runs. We also test individual functions and algorithms with unit tests and simpler assertions executed automatically on CRAN (e.g. *SpaDES* packages) or GitHub Actions Continuous Integration (CI), when workflows and modules have their own public GitHub repositories. Publicly hosting workflows and modules on GitHub also allows other users to test them under different configurations, report bugs and directly contribute to improvements.

2.8 | Examples

We present two "real-world" implementations and designs of our workflow for species distribution modelling. Example 1 uses a single-species statistical model and was kept deliberately simple to work as tutorial for spatial ecological modelling with *SpaDES*. Example 2 presents a complex dynamic landscape model used to simulate forest dynamics. Example 2 is subdivided into two sub-examples that demonstrate how PERFICT and *SpaDES* enhance the transferability (Example 2.1) and validation (Example 2.2) of complex ecological models. We use relatively small study areas in boreal forests in Canada, but the approach is scalable to larger areas (see Micheletti et al., 2021). See Table 2 for a brief description of all modules and

TABLE 2 Modules used in examples 1 and 2, their repository URLs and online manuals where applicable

Type of module	Module name	Description and URL	Manual section
Data	*climateData*	Prepares climate input layers. https://github.com/CeresBarros/SpaDES4Dummies/tree/master/modules/climateData	
	speciesAbundanceData	Prepares species percent cover input layers. https://github.com/CeresBarros/SpaDES4Dummies/tree/master/modules/speciesAbundanceData	
	Biomass_speciesData	Prepares species input layers from multiple data sources. https://github.com/PredictiveEcology/Biomass_speciesData	https://landr-manual.predictiveecology.org/landr-biomass_speciesdata-module.html
	Biomass_borealDataPrep	Prepares multiple inputs and parameters used by *Biomass_core*; customised for Western Canadian boreal and montane forests. https://github.com/PredictiveEcology/Biomass_borealDataPrep	https://landr-manual.predictiveecology.org/landr-biomass_borealdataprep-module.html
Calibration	*Biomass_speciesParameters*	Estimates invariant species traits (growth and mortality-related life-history traits) and adjusts spatially varying species traits (*maxB* and *maxANPP*) used by *Biomass_core* from forest growth plot data. https://github.com/PredictiveEcology/Biomass_speciesParameters	https://landr-manual.predictiveecology.org/landr-biomass_speciesparameters-module.html
Prediction (static)	*projectSpeciesDist*	Fits species distribution models using baseline climate and species layers prepared by *climateData and speciesAbundanceData* modules, and projects species distributions using projected climate layers. https://github.com/CeresBarros/SpaDES4Dummies/tree/master/modules/projectSpeciesDist	
Simulation	*Biomass_core*	Simulates tree species growth, mortality, ageing, and dispersal. Updates biomass following other modules' events, and produces summary figures and tables. https://github.com/PredictiveEcology/Biomass_core	https://landr-manual.predictiveecology.org/landr-biomass_core-module.html
Validation	*Biomass_validationKNN*	Obtains and prepares observed and simulated data for validation of succession vegetation dynamics between two time points (currently 2001 and 2011). https://github.com/PredictiveEcology/Biomass_validationKNN	https://landr-manual.predictiveecology.org/landr-biomass_validationknn-module.html

their online repositories. We ran and tested all code using *R* v4.2.0 on a Windows 10 OS.

2.8.1 | Example 1—Static forecasts of white spruce distribution shifts

We demonstrate how to wrap a static species distribution model (SDM) in a continuous workflow linking raw data to predictions and their evaluation. The example is part of the SpaDES4Dummies guide (https://ceresbarros.github.io/SpaDES4Dummies/), offering an introduction to module creation. The SDM predicts presences and absences of white spruce, *Picea glauca* (Moench) Voss, across a randomly-selected study area (ca. 912,360 ha) in Alberta, Canada, as a function of four bioclimatic variables (annual mean temperature, temperature seasonality, annual precipitation and precipitation seasonality), using one of two prediction algorithms: MaxEnt (Phillips et al., 2006) or a generalised linear model (GLM) fitted with a logit link function. Two data modules, *speciesAbundanceData* and *climateData*, download and prepare FAIR data to fit the SDMs, and cache these operations. *speciesData* obtains white spruce percent cover data for 2001 from Beaudoin et al. (2017) (converted to presence/absence), and *climateData* obtains bioclimatic variables under baseline and future conditions from WorldClim (Fick & Hijmans, 2017), but other data can be used (e.g. different climate scenarios; Barros, Chubaty, & Micheletti, 2022). The prediction module, *projectSpeciesDist*, then takes the data modules' output objects to (i) fit the chosen statistical model (via a module parameter), (ii) automatically validate it, (iii) project the species distributions under baseline and future climate conditions, and (iv) prepare visual outputs (iii and iv are repeated through several climate periods). In complex models, these steps can be encapsulated into separate modules (see Example 2), but here this would add unnecessary complexity to a tutorial. We invite users to build upon and change existing modules by, for example, breaking the prediction module into several modules.

All three modules included lists of necessary *R* packages (which *SpaDES* installs automatically), fully documented parameters and inputs, and assertions that checked for data inconsistencies and missing input objects (when defaults are not available). The workflow is encapsulated in a control script that ensures reproducible *R* package installation and model setup. We ran the entire workflow twice to demonstrate timing differences before and after caching spatial data operations. Each time, we fit two SDMs (MaxEnt and GLM) relating white spruce presences/absences (from 2001) and average climate values from 1970–2000, and generate forecasts for four future climate periods (2021 to 2100 in 20-year increments). The full code is available in the tutorial webpage (https://ceresbarros.github.io/SpaDES4Dummies/part2.html; Barros, Chubaty, & Micheletti, 2022).

2.8.2 | Example 2—Predicting forest changes with LandR Biomass

The benefits of a PERFICT approach and *SpaDES* are most evident in complex dynamic ecological simulation models, which often have rich input data requirements, important parameterisation efforts and long computation times (Micheletti et al., 2021). This example shows how we enhanced the transferability and validation of **LandR Biomass**, a forest landscape model. LandR Biomass predicts forest biomass changes in a spatiotemporally explicit manner, by simulating the population dynamics, dispersal, interactions and responses to disturbances of cohorts of different tree species, using a hybrid modelling approach. It was originally based on LANDIS-II Biomass Succession Extension v3.2.1 (LBSE; Scheller & Mladenoff, 2004; Scheller & Miranda, 2015), from which it has since diverged. Re-implementing LBSE in *R/SpaDES* revealed an opportunity for algorithm alterations for both computation efficiency and ecological realism. We kept the same ecological succession dynamics, but improved performance at larger spatial scales and calculations of growth and germination across tree cohorts. We also integrated simulation, data-driven- and automated parameterisation, and model validation in a modular workflow (see *Biomass_core* manual, Appendix 1; Barros, Chubaty, & McIntire, 2022b). Here, we present the modules involved in parameterising, running and validating forest succession dynamics without disturbances.

LandR Biomass's core simulation module (*Biomass_core*) runs forest dynamics on a pixel basis within a user-defined study area (hereafter, 'the landscape'). For this, it requires starting values of tree cohort biomass and age across the landscape, and species- (i.e. traits) and pixel-level parameters that influence cohort growth, mortality, dispersal and germination success, and responses to disturbances (not used here; see *Biomass_core* manual, Appendix 1). In our integrated workflow, two data modules (*Biomass_speciesData* and *Biomass_borealDataPrep*) obtain spatial and trait data necessary to prepare and/or estimate species traits and pixel-level parameters, and initial tree cohort biomass and age across the landscape. By default, all data comes from FAIR sources, including LANDIS-II trait tables, published literature and spatial layers of stand biomass, age and species cover used for trait estimation and initial landscape conditions (Table 2.1 Appendix 2; see *Biomass_speciesData* and *Biomass_borealDataPrep* manuals, Appendix 1). Another module (*Biomass_speciesParameters*) then (re-)calibrates species growth traits (in this case, previously estimated by *Biomass_borealDataPrep*). For each species, it compares observed growth curves built from an independent species biomass dataset with theoretical growth curves from *Biomass_core* runs using different trait combinations (Table 2.1), using maximum likelihood. From the best matching theoretical curve, it extracts the growth trait values to be used in the simulation (see *Biomass_speciesParameters* manual, Appendix 1). *Biomass_core* also produces visual outputs automatically, and we used internal *SpaDES* saving mechanisms to output pixel-level tables of tree cohort biomass and age per year. Finally, the validation module (*Biomass_validationKNN*) compares simulation outputs to a second snapshot of species biomass conditions across the same landscape (Table 2.1). It prepares both the observed and simulated data, before comparing them using visual outputs, and two validation metrics calculated on several landscape- and pixel-level properties: mean absolute deviations (MAD) from observed values and the sum of the negative log-likelihoods of simulated values (SNLL; *Biomass_validationKNN* manual, Appendix 1). We chose to calculate MAD and SNLL instead of mean squared error and R^2 metrics, because these tend to support increasing complexity and overfitted models (Gimenez-Nadal et al., 2019). Also, estimating the number of parameters (k) involved in complex simulation models can be challenging and prevent calculating the Akaike information criterion (AIC; $\mathrm{AIC} = 2\,k + 2\mathit{SNLL}$) and AIC-related metrics. In these cases, we advise interpreting differences in SNLL within certain boundaries for k.

As in Example 1, default parameter values, input objects and data sources are documented in each module's metadata, but can be overridden by the user (e.g. by providing a traits table with different values). By default, the data, calibration and validation modules automatically download and process data from FAIR sources that provide meaningful data and parameters for Western Canadian boreal forests (Table 2.1). This allows automated parameterisation and validation with little user input. All *R* packages and their minimum versions were also declared in each module's metadata section. *R* functions shared between modules were grouped in the LandR *R* package (McIntire et al., 2021), to increase module interoperability and easily propagate changes to functions used across modules. A control script encapsulated the simulation workflow, including setup and production of additional publication figures. A comprehensive description of these LandR Biomass modules is available in the online manual (Barros, Chubaty, & McIntire, 2022b). The full workflow and source code is hosted at https://github.com/CeresBarros/LandRBiomass_publication (Barros, Chubaty, & McIntire, 2022a) with data sources and parameter values listed in Appendix 2.

Our practical examples of LandR Biomass show its application and validation in different study areas (Example 2.1) and with alternative parameterisations (Example 2.2). In Example 2.1, we ran simulations for two equal-size sets of study areas in nonmanaged forests in northwest Saskatchewan, Canada: the 'base case' (set A) and 'study area change' (set B) simulations (Figure 3.1 Appendix 3). Each set consisted of a larger area used for parameterisation (ca. 185,000 ha) and a smaller area used for simulating forest dynamics, encompassed within the first (ca. 46,400 ha). We recall that each study area has its own set of starting conditions and parameter values. In Example 2.2, we ran one additional simulation on set A of study areas without the (re-)calibration module (*Biomass_speciesParameters*), thus changing parameter values with respect to the base case. We called this simulation 'alternative parameterisation' and compared it with the base case simulation. We ran all three simulation setups in 250 × 250 m pixels, for 30 years starting from 2001 vegetation conditions, and replicated 10 times (using *experiment2*; Box 1) to capture stochasticity from demographic and dispersal processes, the only sources of variation in each setup. Outputs of each setup were validated using species biomass data from 2011 (at the 10th year of simulation).

Since we aim to demonstrate the usefulness of our framework, we focus our results on its technical advantages. We use Example 2.2 to show how integrated validation can provide immediate evaluation of distinct parameterisation approaches, and how the *SpaDES* toolkit facilitates building an ensemble forecast of species biomasses, calculated as the average yearly total biomass of each species across parameterisation approaches and replicates.

TABLE 3 Caching speeds up simulations. Values indicate how much faster second-time runs of the example workflows were with respect to a first-time run, which started without any predownloaded or preformatted data ('base case' simulation in Example 2). All simulations were run using Intel Xeon W-2245 CPU @ 3.90GHz processors on Windows 10 OS, with an average download speed of 100 Mbit/s

		Times faster than first run	
Example 1	*climateData*[a]	30	
	speciesAbundanceData[a]	2	
	projectSpeciesDist[b]	2	
Example 2		Study area change run (Example 2.1)	Alternative param. run (Example 2.2)
	Biomass_speciesData[a]	3	348
	Biomass_borealDataPrep[a]	1	1
	Biomass_speciesParameters[c]	6	
	Biomass_core[b]	1[e]	1[e]
	Biomass_validationKNN[d]	1	1

[a] Data module.

[b] Prediction/simulation module.

[c] Calibration module.

[d] Validation module.

[e] Average across repetitions.

3 | RESULTS

3.1 | Model testing and transparency

Hosting our modelling frameworks in public GitHub repositories contributed to their transparent and collaborative development and accelerated code improvement. For instance, in Example 1, package installation errors were quickly detected by a collaborator when attempting to run the workflow on a different machine. Similarly, the LandR Biomass modules presented in Example 2 have seven active code developers, each detecting potential issues and contributing enhancements via GitHub, while simultaneously reusing these modules across multiple projects (e.g. Micheletti et al., 2021). Embedded assertions provided comprehensive test coverage and ensured workflow integrity in both examples. Unit tests, which tested particular components of the *Biomass_core* simulation module (Example 2), covered 36.3% of the module code, as estimated by the COVR *R* package (Hester, 2020)—this excludes any embedded assertions and package tests.

3.2 | Speed and transferability improvements of the PERFICT workflow

Integrating all modelling steps under the same modular workflow and using caching mechanisms significantly reduced data-to-results time costs (both examples) and improved workflow transferability to new geographical contexts (Example 2.1) and parameterisation approaches (Example 2.2). Given the small size of the study areas used in both examples, most of the time taken in a first-time simulation run (i.e. a simulation that required obtaining and processing all data and parameters) was spent downloading and preparing input data. This was reflected in the faster computation times of second runs of data modules across both examples (Table 3). Caching operations and using publicly available national data also allowed LandR Biomass to detect whether the input study areas had changed and to act accordingly, by avoiding re-downloading national datasets, but re-performing all geoprocessing operations and re-parameterising for the new location (Example 2.1), or by detecting that all the inputs had been already prepared in the first run (Example 2.2). This resulted in over 300 times faster model parameterisation (Table 3).

3.3 | Frequent predictions and integrated model validation with a PERFICT workflow

Explicitly linking data, model parameterisation and prediction also facilitated rerunning the prediction and validation steps, when input data or parameterisation approaches were changed. For instance, in Example 1, adding new forecasts under a more optimistic climate scenario was as simple as supplying new climate data URLs (Figure 1). In Example 2.1, changing study areas involved switching two polygon objects. In Example 2.2, to change the parameterisation approach, we turned the calibration module "off". In all cases, minimal code changes were needed to obtain a new set of predictions and all

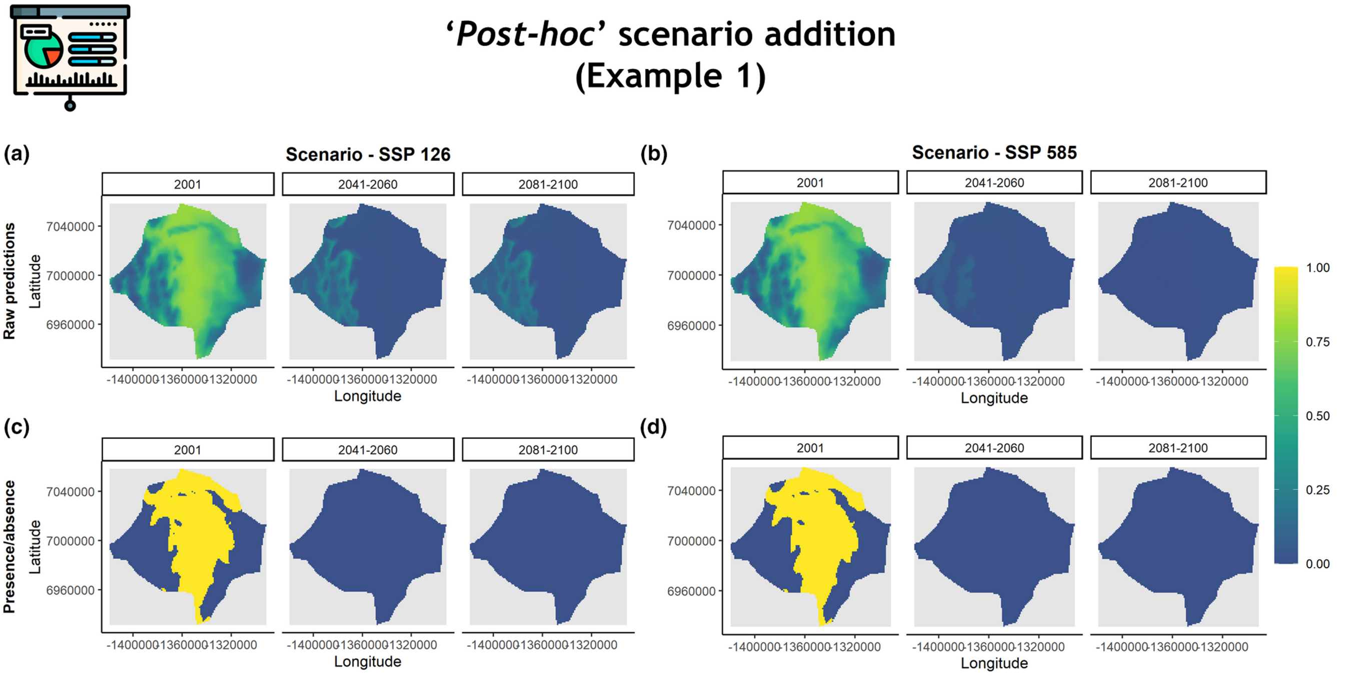

FIGURE 1 Direct links with data facilitate adding new scenarios 'post hoc'. In Example 1, by default white spruce distributions are forecasted using the climate projections under the shared socioeconomic pathway (SSP) 585 scenario from the IPCC 6th coupled model Intercomparison Project (b, d). We added a more optimistic climate scenario SSP 126 (a, c), with minimal intervention (Barros, Chubaty, & Micheletti, 2022). The figure shows MaxEnt forecasts of presence probabilities (a, b) and presences/absences (c, d), for baseline conditions (2001) and two forecast periods (2041–2060 and 2081–2100).

necessary model re-fitting and re-parameterisation were done on-the-fly. In Example 2.2, this enabled rapidly comparing model outputs with different parameters (Figure 2) and facilitated building an ensemble prediction (Figure 3).

Similarly, integrating model validation steps within the prediction module (Example 1) or in a validation module (Example 2), guaranteed that the models were re-evaluated whenever predictions changed. Furthermore, the centralisation of both the fitted statistical models and validation outputs in simulation objects allowed easy inspection of the performance of alternative models. For example, in Example 1 we saw that the GLM-based SDM performed worse than MaxEnt at predicting white spruce occurrences, as suggested by the lower AUC value of the GLM (Table 3.1 Appendix 3). In Example 2.2, total species biomass dynamics were visually similar between the base case and alternative parameterisation simulations (Figure 2a,b), as were MAD values of relative species abundance, of the number of pixels where a species was present ('species presences'), and of the number of pixels where a species had highest biomass ('species dominance'; Figure 2c,d). Yet, SNLL values suggested different model fit between the two simulations. Assuming equal number of parameters in the two simulations, SNLL values indicated that calibrating species traits using more data (base case) provided better predictions of total species biomass at the landscape- and pixel-levels (lower SNLL values) than using default parameters from LANDIS-II (alternative parameterisation), but poorer predictions of species dominance across the landscape (higher SNLL values; Table 3.1). Allowing for a "worst case" rough estimate of $k = 10$ per estimated species trait ($2 * k * 4$ traits = penalty of 80 for the base case), yielded the same results, with the exception that the base case approach becomes worse at predicting species presences. We note that it is difficult to accurately know the value that k should have for "expert-derived" parameters, and assume no penalty for the alternative parameterisation. Hence, under both assumptions SNLL values indicated that the base case

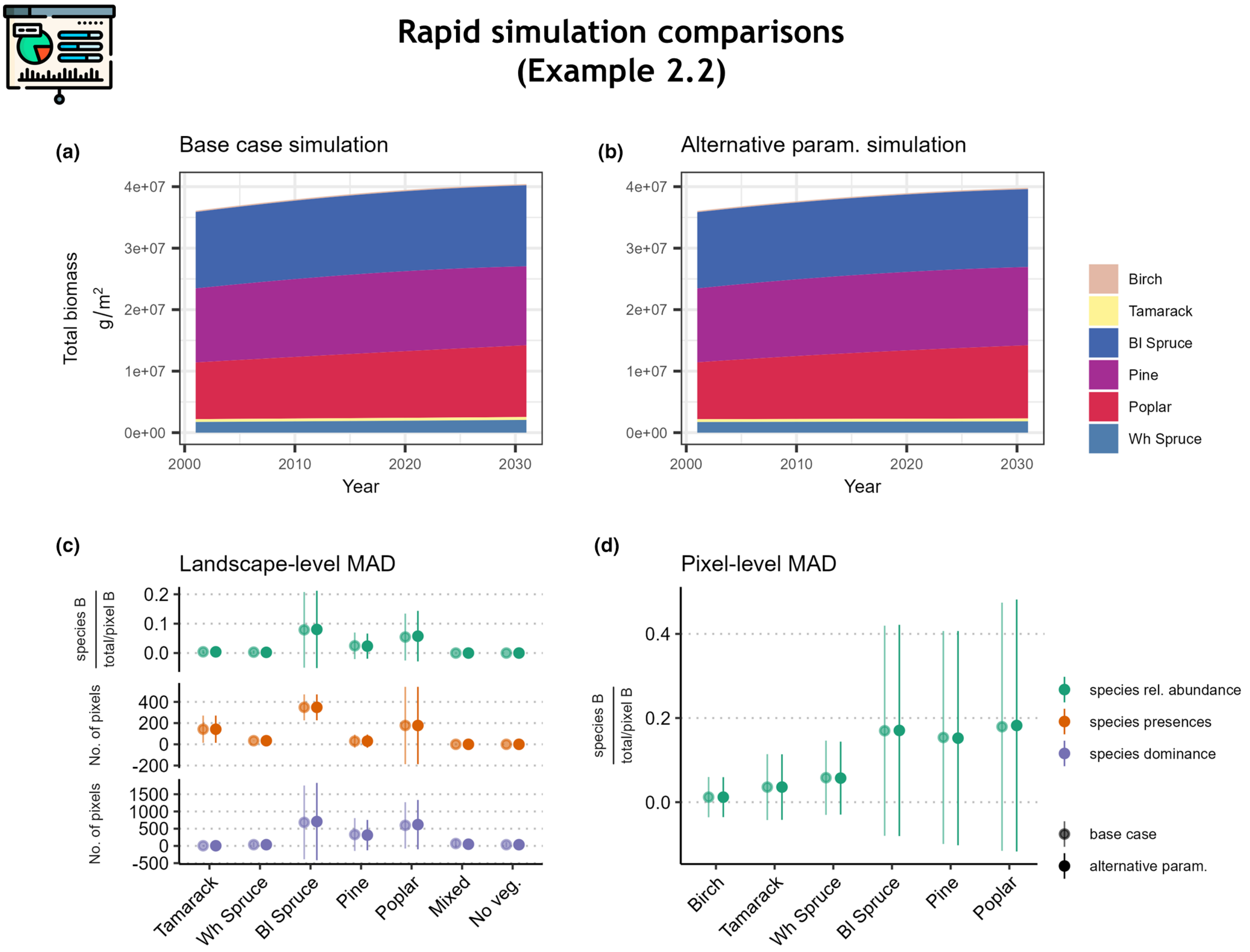

FIGURE 2 Built-in simulation visual outputs facilitate comparing different simulations. We show a few visual outputs automatically generated by LandR Biomass modules used in Example 2.2. Panels (a and b) show simulated temporal trends in total species biomass across the landscape, for one simulation replicate; panels (c and d) show mean absolute deviations (MAD) of different simulated properties from observed data. Landscape-level MAD were calculated on properties measured across the landscape, while pixel-level MAD are calculated per pixel—see Appendix 1 for details.

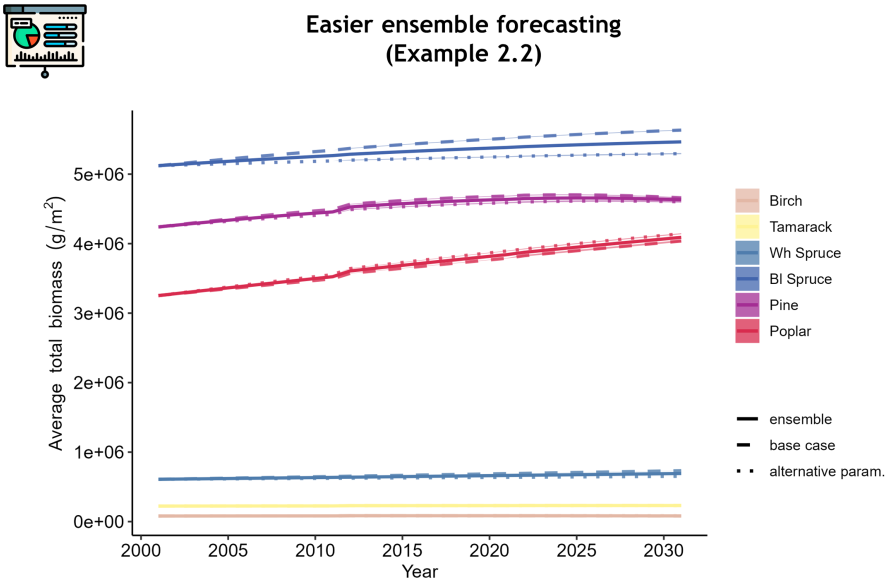

FIGURE 3 *simList* objects facilitate building ensemble forecasts, by centralising simulation outputs. We show total landscape-level species biomass generated by LandR Biomass using two parameterisation approaches (base case and alternative parameterisation), and an ensemble projection obtained by averaging values across simulations (and their repetitions). Lines show averages across replicates (replicates + approaches for the ensemble); shaded areas show variation (standard deviation) across replicates for each parameterisation approach, which was small.

provided better predictions of fine scale species properties (i.e. biomass partition) and worse predictions of coarse scale properties (i.e., dominance and presence/absence) and we can conclude that our results are robust to unknown *k*.

4 | DISCUSSION

We share a solution to two problems that can hamper the progress of predictive ecology: the disconnection among modelling steps and the gap between ecologists and modelling software. Existing practices like version control (*git*), scripting workflows in open software, and iterative approaches to forecasting are important steps towards achieving these goals, but are insufficient to guarantee the transferability, reusability and nimbleness necessary to accelerate cross-model comparisons, integrate validation and facilitate ensemble forecasts across ecological models (Fer et al., 2021). Following the PERFICT principles and using *R*, the *SpaDES* toolkit and *git* provided a modelling standard and tools that made this possible, by combining all modelling steps in a single uninterrupted modelling workflow, developed collaboratively and in a widely understood language.

Establishing these direct links among data, models, and results also offers a solution for ecological modelling in light of recent calls for transparency and increased attention towards tampering and manipulating scientific results (Hopf et al., 2019). In applied contexts, rapid and constant flows among data, prediction and interpretable outputs are often necessary to meet stakeholder expectations (Bodner et al., 2021). Reliably achieving this requires transparency, reproducibility, and nimble workflows that enable the timely ingestion of new data and scientific advancements, while supporting uncertainty assessments. Our integrated and modular workflow allowed running LandR Biomass in a new study area (Example 2.1), using alternative parameterisation approaches and showing ensemble model results (Example 2.2) effortlessly. Furthermore, the built-in presentation and analysis of results, and open-source framework facilitated inspecting and interpreting model outputs by nondevelopers, increasing model reproducibility and transparency.

Automating the parameterisation and validation of predictive ecological models may also be important to support their uptake by ecology practitioners from nonmodelling backgrounds, to respond more quickly to stakeholder demands, and to address challenges associated with cross-model validation and tracking uncertainty of complex models (Fer et al., 2021). In our case, we linked our models with nation-wide data that allowed automatically estimating parameters and validating outputs across Canada. This, together with exporting statistical parameterisation models, facilitated easily assessing parameter and overall model uncertainty (e.g. Table 2.7 Appendix 2). For instance, in Example 1, fitting and validating two accessible (via the *simList*) types of species distribution models, facilitated cross-model comparisons and inspecting model coefficients. In Example 2, automated visual outputs and validation allowed the immediate assessment of the effects of new parameter values on species biomass trends (Figure 2) and model fit (Table 3.1), while standardised simulation structures (*simLists*) facilitated inspecting model parameterisation and adding a post hoc ensemble forecast (Figure 3). As in other models, this automation does not preclude careful inspection of estimated parameters and their effect on predictions (and potentially adapting module code) when transferring the models to different contexts. Yet, using an integrated, modular and open model like LandR Biomass makes these changes easier to accomplish. Just as ecologists use advanced statistical methods shared by experienced statisticians and programmers without necessarily knowing their mathematical formulations, we believe that complex simulation models developed by ecological modellers to calibrate themselves within reasonable boundaries (potentially enforced with code assertions and thorough documentation) can be used analogously. Ultimately, our approach facilitates identifying potential limitations related to input data, parameterisation approach, or modelled processes even in a complex landscape model, and by both developers and nondevelopers—for example if LandR

Biomass fit decreases with different user-supplied data, a first step to improving predictions could be to inspect and potentially change parameter-estimation models, which can be done without changing the module code (see e.g. *Biomass_borealDataPrep* manual in Appendix 1).

Furthermore, the integration and automation of validation steps within a complex simulation model workflow is particularly useful in an iterative forecasting context (Fer et al., 2021). Models do not become "validated" and ubiquitously applicable after passing one or more arbitrary tests (Augusiak et al., 2014). Rather they gain a level of endorsement that can be compared with other models, or using other parameterisations and data, when confronted with the same tests. And so, validation should be frequently reassessed as models develop and new models emerge (Augusiak et al., 2014; McIntire et al., 2022). This can be challenging due to data limitations. For instance, the effects of projected climate change on ecosystems cannot be validated today, as changes have not yet occurred, and long-term forest measurements necessary to validate forest dynamics are seldom available at landscape scales. By integrating validation within the model framework, this step can be automated and repeated as new data becomes available (Fer et al., 2021). This means that model accuracy can be assessed and reported seamlessly following simulations, and becomes itself a transparent and reusable procedure that can updated if model inputs change. In our examples, we added built-in model validation as part of the model-fitting step (Example 1) or as a separate module (Example 2), using FAIR validation datasets and generic validation approaches as much as possible. In LandR Biomass, other datasets and approaches could be used for validation (e.g. Landsat data, Matasci et al., 2018); however, our goal was to develop a module that could be used anywhere in Canada and provide simple and easily interpretable validation metrics. Now that the infrastructure is in place, expanding the module to use different data and validation approaches should be straightforward and lead to further model improvements. Finally, the integrated validation provided a direct evaluation of different modelling assumptions (both examples) and parsimony (Example 2.2), which are important for both inference and prediction. In Example 1, this meant a direct assessment of the accuracy of the statistical algorithm, which revealed better performance of MaxEnt over GLM for predicting range shifts. In Example 2.2, our assessment of the accuracy of two parameterisation approaches of different complexity (Figure 2, Table 3.1) suggested that the more complex model with calibrated growth- and mortality-related parameters (base case) predicted total species biomass better, but not species dominance. We expected that calibrating model parameters with higher quality data would improve model performance across all validated properties. The immediate feedback showed that the substantial extra effort involved in the calibration improved forecasts at finer scales (i.e. pixel-based species biomass partitioning), but not for coarser patterns (i.e. species dominance across the landscape). An automated calculation of likelihood metrics in an ecological simulation context can thus contribute to an objective and continuous evaluation of parsimony (Topping et al., 2015), by revealing when increasing model complexity is worthwhile.

The initial time we invested to adopt a PERFICT workflow was largely compensated by the flexibility provided; it led to higher scrutiny, increased exploration, and hopefully to improved modelling and science. The data integration across all stages of the workflow also reduced development costs among collaborators when updates were needed (i.e. only sub-components were updated), and reduced implementation and development times for other projects since modules could be reused (e.g. Micheletti et al., 2021). Caching accelerated model development (the developer could avoid repeating slow operations, while keeping the workflow intact at all stages) and promoted iterative forecasting (our workflows only re-run the components that require updating). Additionally, collaborative development enabled faster identification and resolution of issues. We aim to further alleviate development costs, particularly in LandR Biomass, with increased module test coverage and more comprehensive routine integration testing. We note, however, that access to continuous integrated testing of large complex models is a bottleneck to many teams, especially when funding resources cannot cover the cost of access to computer resources and their maintenance (Vedder et al., 2021). We also intend to periodically assess workflow accessibility to nondevelopers (i.e. end users) and improve where needed. We have successfully used workflows similar to Example 1 to teach *SpaDES* workshops (see Part 1 of the SpaDES4SDummies guide; Barros, Chubaty, & Micheletti, 2022) to practitioners in ecology, and LandR Biomass modules are being adopted by new users. To facilitate learning, we are streamlining module manual creation across *SpaDES* modules, using LandR Biomass as a first application (see Appendix 1 and Barros, Chubaty, & McIntire, 2022b), and provide project and workflow templates (https://github.com/PredictiveEcology/SpaDES.project).

The re-implementation of LBSE in a PERFICT workflow brought other advantages. As suggested by Thiele and Grimm (2015), porting existing models into a more transparent and usable format (e.g. more widely understood language) benefits from increased scrutiny by the scientific community, improving a model's scientific quality and longevity. Porting LBSE into *R*/*SpaDES* revealed algorithm-documentation inconsistencies and allowed improvements to model mechanics. It also allowed tightly coupling data and simulation workflows with statistical, visualisation and geoprocessing tools. Although *SpaDES* modules can interface with models in other languages (thereby avoiding complete model rewrites), re-writing LBSE in *R* allowed all co-authors to understand the implementation, selectively modify algorithms to improve performance and ecological realism, link LandR Biomass to other models (e.g. different fire models; Micheletti et al., 2021), use it in different ecological and geographical contexts (Barros et al., 2020; Micheletti et al., 2021) and across operating systems.

Here, we demonstrated a practical solution to two pervasive problems in ecological simulation modelling that can encumber the use of large datasets and complex models, hinder scientific progress, decrease model transparency, and reduce the durability and reusability of ecological simulation models. By demonstrating how ecologists can develop applied ecological models using transparent and reusable integrated workflows and harnessing community contributions, we hope to contribute to moving predictive ecology forward in the field of global change research and forecasting.

AUTHOR CONTRIBUTIONS

Ceres Barros, Yong Luo and Eliot McIntire designed the study; Yong Luo, Eliot McIntire, Alex Chubaty, Ceres Barros, Ian Eddy and Steven Cumming re-implemented the LBSE model in *R* and developed additional modelling code; Yong Luo assessed model accuracy (with respect to LANDIS-II) and performance; Ceres Barros performed example simulations and analysed outputs; Alex Chubaty, Tatiane Micheletti, Ian Eddy and Eliot McIntire independently tested *R* code; Ceres Barros, Yong Luo, Eliot McIntire, Céline Boisvenue and Ian Eddy worked on ensuring the ecological representativeness of LandR Biomass; David Andison provided funding support and participated in genesis of *SpaDES*; Ceres Barros led the elaboration of the LandR Biomass Manual and led manuscript writing, with equal writing contribution from all authors, who gave final approval for publication.

ACKNOWLEDGEMENTS

We thank Geneviève Degré-Timmons for contributing to improving this manuscript with her modelling novice perspective, as well as anonymous reviewers and Editors. We acknowledge funding from a Mitacs Accelerate Award no. IT10289 and Mitacs Elevate Fellowship no. IT12391, in partnership with fRI Research, both awarded to Dr Ceres Barros, funding from Alberta Innovates and the Forest Resources Improvement Association of Alberta, and funding from the Environment and Climate Change Canada, Canadian Wildlife Service. We also acknowledge the World Climate Research Programme, which coordinated and promoted CMIP6, the climate modelling groups for producing and making available their model output, the Earth System Grid Federation (ESGF) for archiving the data and providing access, and WorldClim for downscaling and sharing climate projections and preparing bioclimatic variables.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1111/2041-210X.14034.

DATA AVAILABILITY STATEMENT

All code is available on open GitHub repositories and archived in Zenodo (Example 1: Barros, Chubaty, & Micheletti, 2022; Example 2: Barros, Chubaty, & McIntire, 2022a). All data used here is open access and can be obtained by running the integrated workflow code scripts (for Example 1, Part2_SDMs.R script in Barros, Chubaty, & Micheletti, 2022; for Example 2, global.R script in Barros, Chubaty, & McIntire, 2022a). Note that all code was tested on a Windows 10 OS, using R 4.2.2.

ORCID

Ceres Barros https://orcid.org/0000-0003-4036-977X
Yong Luo https://orcid.org/0000-0002-3748-9773
Alex M. Chubaty https://orcid.org/0000-0001-7146-8135
Ian M. S. Eddy https://orcid.org/0000-0001-7397-2116
Tatiane Micheletti https://orcid.org/0000-0003-4838-8342
Céline Boisvenue https://orcid.org/0000-0002-6031-7961
David W. Andison https://orcid.org/0000-0001-7135-7160
Steven G. Cumming https://orcid.org/0000-0002-7862-2913
Eliot J. B. McIntire https://orcid.org/0000-0002-6914-8316

REFERENCES

Augusiak, J., Van den Brink, P. J., & Grimm, V. (2014). Merging validation and evaluation of ecological models to 'evaludation': A review of terminology and a practical approach. *Ecological Modelling*, *280*, 117–128. https://doi.org/10.1016/j.ecolmodel.2013.11.009

Barros, C., Chubaty, A., & McIntire, E. (2022a). CeresBarros/LandRBiomass_publication: LandR biomass workflow v1.0.1 (v1.0.1). *Zenodo*. https://doi.org/10.5281/ZENODO.7262431

Barros, C., Chubaty, A., & McIntire, E. (2022b). PredictiveEcology/LandR-Manual: V1.0.3 (v1.0.3). *Zenodo*. https://doi.org/10.5281/ZENODO.7293682

Barros, C., Chubaty, A., & Micheletti, T. (2022). CeresBarros/SpaDES4Dummies: SpaDES 4 Dummies guide v1.2.0 (v1.2.0). *Zenodo*. https://doi.org/10.5281/ZENODO.7154710

Barros, C., Guéguen, M., Douzet, R., Carboni, M., Boulangeat, I., Zimmermann, N. E., Münkemüller, T., & Thuiller, W. (2017). Extreme climate events counteract the effects of climate and land-use changes in alpine tree lines. *Journal of Applied Ecology*, *54*(1), 39–50. https://doi.org/10.1111/1365-2664.12742

Barros, C., McIntire, E. J. B., & Andison, D. W. (2020). *Spatio-temporal dynamic modelling of mixed-severity fire regimes in the SW foothills of Alberta*. fRI Research. https://www.landscapesinmotion.ca/resources/2020/10/20/report-spatio-temporal-dynamic-modelling

Beaudoin, A., Bernier, P. Y., Villemaire, P., Guindon, L., & Guo, X. J. (2017). *Species composition, forest properties and land cover types across Canada's forests at 250m resolution for 2001 and 2011 [TIFF]*. Natural Resources Canada. https://doi.org/10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990

Bodner, K., Fortin, M.-J., & Molnár, P. K. (2020). Making predictive modelling ART: Accurate, reliable, and transparent. *Ecosphere*, *11*(6), e03160. https://doi.org/10.1002/ecs2.3160

Bodner, K., Rauen Firkowski, C., Bennett, J. R., Brookson, C., Dietze, M., Green, S., Hughes, J., Kerr, J., Kunegel-Lion, M., Leroux, S. J., McIntire, E., Molnár, P. K., Simpkins, C., Tekwa, E., Watts, A., & Fortin, M.-J. (2021). Bridging the divide between ecological forecasts and environmental decision making. *Ecosphere*, *12*(12), e03869. https://doi.org/10.1002/ecs2.3869

Boisvenue, C., & Running, S. W. (2010). Simulations show decreasing carbon stocks and potential for carbon emissions in Rocky Mountain forests over the next century. *Ecological Applications*, *20*(5), 1302–1319. https://doi.org/10.1890/09-0504.1

Borregaard, M. K., & Hart, E. M. (2016). Towards a more reproducible ecology. *Ecography*, *39*(4), 349–353. https://doi.org/10.1111/ecog.02493

Briscoe, N. J., Elith, J., Salguero-Gómez, R., Lahoz-Monfort, J. J., Camac, J. S., Giljohann, K. M., Holden, M. H., Hradsky, B. A., Kearney, M. R., McMahon, S. M., Phillips, B. L., Regan, T. J., Rhodes, J. R., Vesk, P. A., Wintle, B. A., Yen, J. D. L., & Guillera-Arroita, G. (2019). Forecasting species range dynamics with process-explicit models: Matching methods to applications. *Ecology Letters*, *22*(11), 1940–1956. https://doi.org/10.1111/ele.13348

Chubaty, A. M., & McIntire, E. J. B. (2019). *SpaDES: Develop and run spatially explicit discrete event simulation models* (v2.0.4). https://CRAN.R-project.org/package=SpaDES

Cobos, M. E., Peterson, A. T., Barve, N., & Osorio-Olvera, L. (2019). Kuenm: An R package for detailed development of ecological niche models using Maxent. *PeerJ*, *7*, e6281. https://doi.org/10.7717/peerj.6281

DeAngelis, D. L., & Yurek, S. (2017). Spatially explicit modeling in ecology: A review. *Ecosystems*, *20*(2), 284–300. https://doi.org/10.1007/s10021-016-0066-z

Dietze, M. C. (2017). *Ecological forecasting*. https://press.princeton.edu/books/hardcover/9780691160573/ecological-forecasting

Ellison, A. M. (2010). Repeatability and transparency in ecological research. *Ecology*, *91*(9), 2536–2539. https://doi.org/10.1890/09-0032.1

Fall, A., & Fall, J. (2001). A domain-specific language for models of landscape dynamics. *Ecological Modelling*, *18*, 1–18.

Fer, I., Gardella, A. K., Shiklomanov, A. N., Campbell, E. E., Cowdery, E. M., De Kauwe, M. G., Desai, A., Duveneck, M. J., Fisher, J. B., Haynes, K. D., Hoffman, F. M., Johnston, M. R., Kooper, R., LeBauer, D. S., Mantooth, J., Parton, W. J., Poulter, B., Quaife, T., Raiho, A., … Dietze, M. C. (2021). Beyond ecosystem modeling: A roadmap to community cyberinfrastructure for ecological data-model integration. *Global Change Biology*, *27*(1), 13–26. https://doi.org/10.1111/gcb.15409

Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: New 1-km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, *37*(12), 4302–4315. https://doi.org/10.1002/joc.5086

Gimenez-Nadal, J. I., Molina, J. A., & Velilla, J. (2019). Modelling commuting time in the US: Bootstrapping techniques to avoid overfitting. *Papers in Regional Science*, *98*(4), 1667–1684. https://doi.org/10.1111/pirs.12424

GNU Project. (2020). *GNU Make Manual [GNU OS]*. Free Software Foundation. https://www.gnu.org/software/make/manual/

Hesselbarth, M. H. K., Nowosad, J., Signer, J., & Graham, L. J. (2021). Open-source tools in R for landscape ecology. *Current Landscape Ecology Reports*, *6*, 97–111. https://doi.org/10.1007/s40823-021-00067-y

Hester, J. (2020). *covr*: *Test coverage for packages* (3.5.1). https://CRAN.R-project.org/package=covr

Hijmans, R. J., Phillips, S., Leathwick, J., & Elith, J. (2021). *dismo*: *Species distribution modeling* (1.3-5). https://CRAN.R-project.org/package=dismo

Hopf, H., Krief, A., Mehta, G., & Matlin, S. A. (2019). Fake science and the knowledge crisis: Ignorance can be fatal. *Royal Society Open Science*, *6*(5), 190161. https://doi.org/10.1098/rsos.190161

IPCC. (2022). *Climate change 2022: Impacts, adaptation, and vulnerability*. Cambridge University Press. https://report.ipcc.ch/ar6wg2/pdf/IPCC_AR6_WGII_FinalDraft_FullReport.pdf

Landau, W. (2021). The targets R package: A dynamic make-like function-oriented pipeline toolkit for reproducibility and high-performance computing. *Journal of Open Source Software*, *6*(57), 2959. https://doi.org/10.21105/joss.02959

Matasci, G., Hermosilla, T., Wulder, M. A., White, J. C., Coops, N. C., Hobart, G. W., Bolton, D. K., Tompalski, P., & Bater, C. W. (2018). Three decades of forest structural dynamics over Canada's forested ecosystems using Landsat time-series and lidar plots. *Remote Sensing of Environment*, *216*, 697–714. https://doi.org/10.1016/j.rse.2018.07.024

McIntire, E. J. B. (2022). *Require*: *Installing and loading R packages for reproducible workflows* (0.0.13.9003).

McIntire, E. J. B., Chubaty, A., Cumming, S., Andison, D., Barros, C., Boisvenue, C., Hache, S., Luo, Y., Micheletti, T., & Stewart, F. (2022). PERFICT: A re-imagined foundation for predictive ecology. *Ecology Letters*, *25*, 1345–1351. https://doi.org/10.1111/ele.13994

McIntire, E. J. B., & Chubaty, A. M. (2021a). *Reproducible*: *A set of tools that enhance reproducibility beyond package management* (v1.2.8). https://reproducible.predictiveecology.org, https://github.com/PredictiveEcology/reproducible

McIntire, E. J. B., & Chubaty, A. M. (2021b). *SpaDES.experiment*: *Simulation experiments within the SpaDES ecosystem* (v0.0.2.9002). https://github.com/PredictiveEcology/SpaDES.experiment

McIntire, E. J. B., Chubaty, A. M., Barros, C., Eddy, I. M. S., Luo, Y., & Micheletti, T. (2021). *LandR*: *Landscape ecosystem modelling in R* (1.0.5.9006).

Micheletti, T., Stewart, F. E. C., Cumming, S. G., Haché, S., Stralberg, D., Tremblay, J. A., Barros, C., Eddy, I. M. S., Chubaty, A. M., Leblond, M., Pankratz, R. F., Mahon, C. L., Van Wilgenburg, S. L., Bayne, E. M., Schmiegelow, F. K. A., & McIntire, E. J. B. (2021). Assessing pathways of climate change effects in SpaDES: An application to boreal landbirds of Northwest Territories Canada. *Frontiers in Ecology and Evolution*, *9*. https://doi.org/10.3389/fevo.2021.679673

Osherove, R. (2013). *The art of unit testing: With examples in C*. Simon and Schuster.

Phillips, S. J., Anderson, R. P., & Schapire, R. E. (2006). Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, *190*(3), 231–259. https://doi.org/10.1016/j.ecolmodel.2005.03.026

Powers, S. M., & Hampton, S. E. (2019). Open science, reproducibility, and transparency in ecology. *Ecological Applications*, *29*(1), e01822. https://doi.org/10.1002/eap.1822

R Core Team. (2022). *R: A language and environment for statistical computing (4.2.0)*. R Foundation for Statistical Computing. https://www.R-project.org/

Sarma, G. P., Jacobs, T. W., Watts, M. D., Ghayoomie, S. V., Larson, S. D., & Gerkin, R. C. (2016). Unit testing, model validation, and biological simulation. *F1000Research*, *5*, 1946. https://doi.org/10.12688/f1000research.9315.1

Scheller, R. M., & Miranda, B. R. (2015). *LANDIS-II biomass succession v3.2 extension – User Guide* (3.2).

Scheller, R. M., & Mladenoff, D. J. (2004). A forest growth and biomass module for a landscape simulation model, LANDIS: Design, validation, and application. *Ecological Modelling*, *180*(1), 211–229. https://doi.org/10.1016/j.ecolmodel.2004.01.022

Schuwirth, N., Borgwardt, F., Domisch, S., Friedrichs, M., Kattwinkel, M., Kneis, D., Kuemmerlen, M., Langhans, S. D., Martínez-López, J., & Vermeiren, P. (2019). How to make ecological models useful for environmental management. *Ecological Modelling*, *411*, 108784. https://doi.org/10.1016/j.ecolmodel.2019.108784

Seidl, R., Fernandes, P. M., Fonseca, T. F., Gillet, F., Jönsson, A. M., Merganičová, K., Netherer, S., Arpaci, A., Bontemps, J.-D., Bugmann, H., González-Olabarria, J. R., Lasch, P., Meredieu, C., Moreira, F., Schelhaas, M.-J., & Mohren, F. (2011). Modelling natural disturbances in forest ecosystems: A review. *Ecological Modelling*, *222*(4), 903–924. https://doi.org/10.1016/j.ecolmodel.2010.09.040

Shifley, S. R., He, H. S., Lischke, H., Wang, W. J., Jin, W., Gustafson, E. J., Thompson, J. R., Thompson, F. R., Dijak, W. D., & Yang, J. (2017). The past and future of modeling forest dynamics: From growth and yield curves to forest landscape models. *Landscape Ecology*, *32*(7), 1307–1325. https://doi.org/10.1007/s10980-017-0540-9

Speich, M., Dormann, C. F., & Hartig, F. (2021). Sequential Monte-Carlo algorithms for Bayesian model calibration – A review and method comparison. *Ecological Modelling*, *455*, 109608. https://doi.org/10.1016/j.ecolmodel.2021.109608

Stewart, F. E. C., Nowak, J. J., Micheletti, T., McIntire, E. J. B., Schmiegelow, F. K. A., & Cumming, S. G. (2020). Boreal Caribou can coexist with natural but not industrial disturbances. *The Journal of Wildlife Management*, *84*(8), 1435–1444. https://doi.org/10.1002/jwmg.21937

Taylor, S. D., & White, E. P. (2020). Automated data-intensive forecasting of plant phenology throughout the United States. *Ecological Applications*, *30*(1), e02025. https://doi.org/10.1002/eap.2025

Thiele, J. C., & Grimm, V. (2015). Replicating and breaking models: Good for you and good for ecology. *Oikos*, *124*(6), 691–696. https://doi.org/10.1111/oik.02170

Topping, C. J., Alrøe, H. F., Farrell, K. N., & Grimm, V. (2015). Per Aspera ad Astra: Through complex population modeling to predictive theory. *The American Naturalist*, *186*(5), 669–674. https://doi.org/10.1086/683181

Ushey, K., Allaire, J., & Tang, Y. (2022). *Reticulate*: *Interface to "python"* [manual].

van Vliet, J., Bregt, A. K., Brown, D. G., van Delden, H., Heckbert, S., & Verburg, P. H. (2016). A review of current calibration and validation practices in land-change modeling. *Environmental Modelling & Software*, *82*, 174–182. https://doi.org/10.1016/j.envsoft.2016.04.017

Vedder, D., Ankenbrand, M., & Sarmento Cabral, J. (2021). Dealing with software complexity in individual-based models. *Methods in Ecology and Evolution*, *12*(12), 2324–2333. https://doi.org/10.1111/2041-210X.13716

White, E. P., Yenni, G. M., Taylor, S. D., Christensen, E. M., Bledsoe, E. K., Simonis, J. L., & Ernest, S. K. M. (2019). Developing an automated iterative near-term forecasting system for an ecological study. *Methods in Ecology and Evolution*, *10*(3), 332–344. https://doi.org/10.1111/2041-210X.13104

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J. W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., … Mons, B. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*, *3*(1). https://doi.org/10.1038/sdata.2016.18

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Barros, C., Luo, Y., Chubaty, A. M., Eddy, I. M. S., Micheletti, T., Boisvenue, C., Andison, D. W., Cumming, S. G., & McIntire, E. J. B. (2023). Empowering ecological modellers with a PERFICT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation. *Methods in Ecology and Evolution*, *14*, 173–188. https://doi.org/10.1111/2041-210X.14034