PLOS CLIMATE

RESEARCH ARTICLE

Matilda v1.0: An R package for probabilistic climate projections using a reduced complexity climate model

Joseph K. Brown*, Leeya Pressburger, Abigail Snyder, Kalyn Dorheim, Steven J. Smith, Claudia Tebaldi, Ben Bond-Lamberty

Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, MD, United States of America

* joseph.brown@pnnl.gov

OPEN ACCESS

Citation: Brown JK, Pressburger L, Snyder A, Dorheim K, Smith SJ, Tebaldi C, et al. (2024) Matilda v1.0: An R package for probabilistic climate projections using a reduced complexity climate model. PLOS Clim 3(5): e0000295. https://doi.org/10.1371/journal.pclm.0000295

Editor: Steven L. Forman, Baylor University, UNITED STATES

Received: September 8, 2023

Accepted: March 25, 2024

Published: May 2, 2024

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: https://doi.org/10.1371/journal.pclm.0000295

Data Availability Statement: All code to run analysis and recreate figures in this manuscript is available at https://github.com/JGCRI/Brown-etal_

Abstract

A primary advantage to using reduced complexity climate models (RCMs) has been their ability to quickly conduct probabilistic climate projections, a key component of uncertainty quantification in many impact studies and multisector systems. Providing frameworks for such analyses has been a target of several RCMs used in studies of the future co-evolution of the human and Earth systems. In this paper, we present Matilda, an open-science R software package that facilitates probabilistic climate projection analysis, implemented here using the Hector simple climate model in a seamless and easily applied framework. The primary goal of Matilda is to provide the user with a turn-key method to build parameter sets from literature-based prior distributions, run Hector iteratively to produce perturbed parameter ensembles (PPEs), weight ensembles for realism against observed historical climate data, and compute probabilistic projections for different climate variables. This workflow gives the user the ability to explore viable parameter space and propagate uncertainty to model ensembles with just a few lines of code. The package provides significant freedom to select different scoring criteria and algorithms to weight ensemble members, as well as the flexibility to implement custom criteria. Additionally, the architecture of the package simplifies the process of building and analyzing PPEs without requiring significant programming expertise, to accommodate diverse use cases. We present a case study that provides illustrative results of a probabilistic analysis of mean global surface temperature as an example of the software application.

1 Introduction

The human-Earth system is fundamentally integrated with impacts and feedbacks tightly interconnecting outcomes across human decisions and the broader environment. Human decisions regarding land use, water use, and energy consumption affect the broader Earth system, which can subsequently drive future human decisions [1,2]. Multisectoral models are those that include representations of energy, water, land, socioeconomic, and climate sectors in an

2024-PLOSClimate or https://doi.org/10.5281/zenodo.10695259.

Funding: This research was supported by the U.S. Department of Energy, Office of Science, as part of research in MultiSector Dynamics, Earth and Environmental System Modeling Program. The Pacific Northwest National Laboratory is operated for DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. The views and opinions expressed in this paper are those of the authors alone.

Competing interests: The authors have declared that no competing interests exist.

integrated framework. The Global Change Analysis Model (GCAM), and similar multisectoral models can be used to explore future scenarios with different water, energy, land use, and socioeconomic outcomes that interact with the Earth system. Representative Concentration Pathways (RCPs), for example, provide scenarios that reach varying magnitudes of radiative forcing by the end of the century based on changing GHG emissions and land use [1–3]. Shared Socio-economic Pathways (SSPs) provide scenarios driven by plausible changes in global developments including population and economic growth, fossil fuel dependency, and costs of environmental degradation [3,4]. The development of an SSP-RCP framework (hereafter, SSPs) combines the climate and societal futures of SSPs and RCPs [3,5]. These scenarios can be used in Earth system models (ESMs) to explore future climate outcomes given different possible emissions scenarios. However, the breadth at which ESMs can investigate the climate system comes at a significant computational expense.

Reduced complexity climate models (RCMs) play a significant role in quickly assessing how key climate variables may evolve in the future, and can do so in a probabilistic framework, made possible because of their simplified computational complexity [6–11]. By representing only the most critical Earth system processes with reduced resolution in temporal and spatial dimensions, RCMs are a useful alternative to more powerful but much slower ESMs in many cases [9,12]. The computational efficiency of RCMs makes them an ideal tool for constructing perturbed parameter ensemble (PPE) simulations, which are RCM ensembles built by running the model iteratively with different parameter sets [7,13,14]. This ability enables effective sampling of the parameter space, propagation of parameter uncertainty to RCM ensembles, and provides a framework for probabilistic projection quantification [7–9,15,16].

The capacity for RCMs to conduct probabilistic projections with PPEs is critical, but few RCMs have an easy-to-use and open-source workflow for this capability. In Phase 2 of the Reduced Complexity Model Intercomparison Project (RCMIP), Nicholls et al. [17] highlight the use of extant RCMs to perform probabilistic analyses to inform Earth system knowledge by creating PPEs. Among the RCMIP models investigated; MAGICC, FaIR, and Hector are some of the most relevant, with MAGICC and FaIR used in previous reports by the Intergovernmental Panel on Climate Change (IPCC) [12]. While both models are extensive and widely used for probabilistic projections [18–20], they have some drawbacks. For example, while FaIR demonstrates skillful outputs in RCMIP evaluations [17], it lacks a turnkey mechanism for computing probabilistic distributions of climate variables [7,14]. This places a significant programming responsibility on the user. Hector has similarly lacked a seamless method for probabilistic projections. MAGICC is also one of the best-performing RCMs in RCMIP and takes advantage of a rigorous statistical approach [10,21]. However, while it aims to shift to open-source, it is currently a closed-source model. To use MAGICC to the fullest capability of the model, users must contact model developers for access to the software package and probabilistic distribution.

We address some of these drawbacks in our development of the R package Matilda. Matilda is an open-science framework that provides a simplified method for conducting probabilistic climate projections without imposing a significant programming burden on the user. Our method generates parameter sets from Monte Carlo estimation of prior distributions from the literature. It then builds PPEs, weights them for realism against observed data, and computes probabilistic climate projections. While Matilda is flexible enough to operate with many RCMs, it was designed explicitly for seamless integration with Hector.

Hector is an open-source, object-oriented simple climate carbon-cycle model capable of emulating more complex ESMs and is executed in C++ [6,9,11]. It takes advantage of the computational benefits described above by operating on a global spatial scale and annual time step. The model functions by converting user-specified emissions to atmospheric

concentrations which are used to calculate radiative forcing [6,9]. Hector then uses total radiative forcing to derive global temperature change and other climate variables [9,11]. Despite its reduced complexity, Hector provides a good representation of CMIP6 outputs for major climate variables across SSP scenarios [22]. In addition to operating as a stand-alone carbon-climate model, Hector is also used as the default climate module for GCAM [6]. Hector can also be run as an R package and through a web-accessible interface (Hector UI), making it accommodating to a larger user base [22–24]. Hector's design and performance make it an excellent candidate for developing a user-friendly probabilistic climate projection tool in R. Pressburger et al. [13]show the benefits of applying such a framework to account for uncertainty of model parameters when assessing near- and long-term sources of atmospheric CO_2. With Matilda, users will be able to easily use Hector to conduct probabilistic climate projection analyses without the burden of complex coding requirements. Matilda thus provides seamless integration with the Hector reduced complexity climate model.

The objectives of this paper are twofold: 1) we introduce the Matilda R package that provides a simple framework for conducting probabilistic climate projection analyses using the Hector simple climate model and 2) we showcase the package functionality with a case study that provides illustrative results. We conclude with a list of future developments that can improve the long-term utility of Matilda.

2 Software description

Here we introduce the software functions and basic workflow of the package (Fig 1). We use applied examples to show package functionality by assessing climate change projections from the SSP 2–4.5 emissions scenario (i.e., middle of the road SSP with the year 2100 radiative

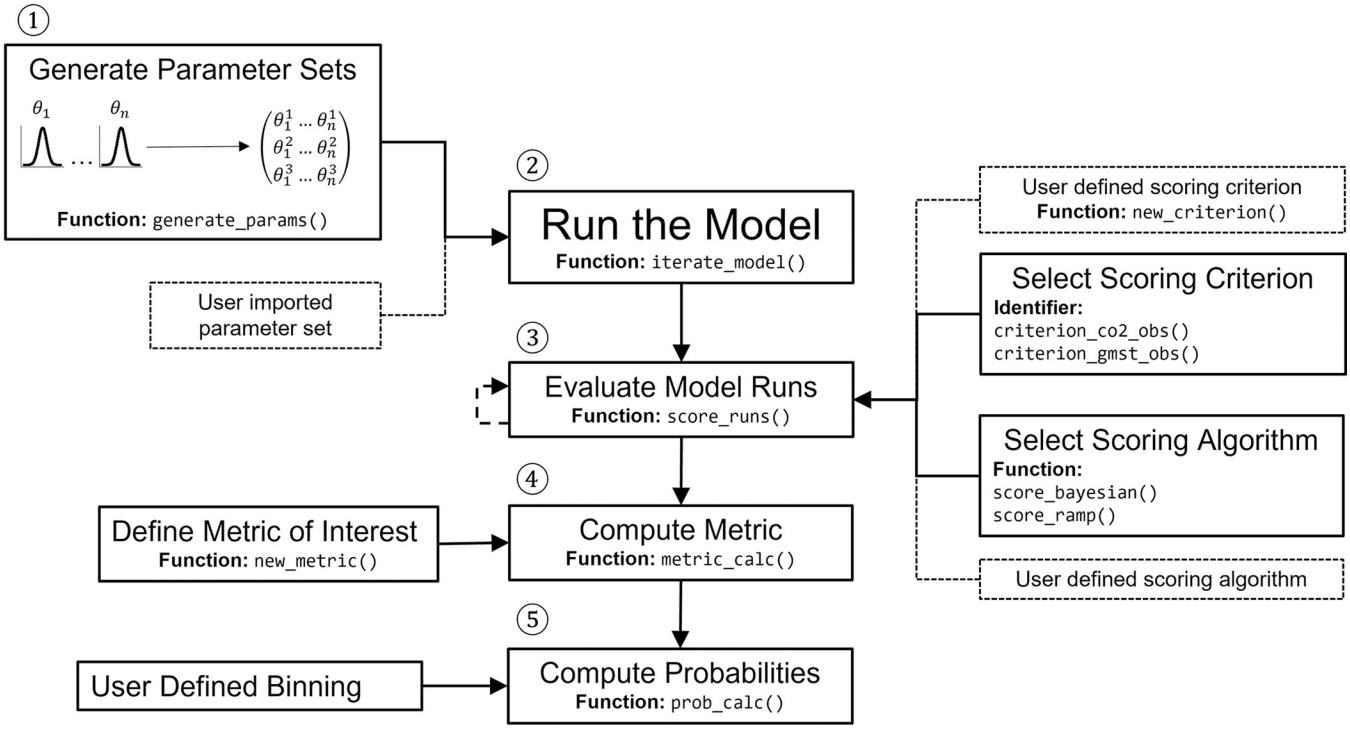

Fig 1. Matilda workflow. Diagram detailing the Matilda workflow to compute probabilistic projections. Dotted lines indicate opportunities for the user to define their own program specification. The dashed line in step 3 indicates the ability of the user to evaluate ensemble members repeatedly with different scoring criterion.

https://doi.org/10.1371/journal.pclm.0000295.g001

forcing level of 4.5 W/m^2), providing step-by-step code and explaining the significance of each function.

2.1 Installing the software

The Matilda software is available on GitHub. To install from our GitHub repository:

1)

```
library(remotes)

install_github("jgcri/matilda")
```

Once installed, the package is loaded as with any other R package:

2)

```
library(matilda)
```

Matilda functions are fully integrated with Hector's R interface, and therefore when Matilda is installed and loaded, the hector package (https://jgcri.github.io/hector/) is also loaded. Matilda requires the use of Hector V3.0 or newer [22].

2.2 Software documentation

Full descriptions of package functions can be accessed in the package's help documentation. Furthermore, detailed documentation and vignettes are available from our GitHub repository (github.com/jgcri/matilda).

2.3 Configuring a model core

An analysis in Matilda begins by setting up a Hector model instance, termed a "core". A Hector core is an object that contains information about model inputs, current state, and outputs for a specific Hector run. The information contained in an initiated core comes from an INI file holding metadata, emissions scenarios, and model parameters needed to run Hector.

We call newcore() (a hector function) to initiate a core containing information to conduct model runs using the SSP 2–4.5 emission scenario:

3)

```
ini_file <- system.file("input/hector_ssp245.ini", package
= "hector")

core_ssp245 <- newcore(ini_file, name = "SSP_245")
```

2.4 Parameter estimation and establishing parameter sets

The basis of running Hector in a probabilistic setup relies on establishing a set of parameter configurations that are used to run the model iteratively. Matilda uses parameter information gathered from the literature to inform prior distributions (Table 1). To build parameter sets, we draw parameter values from their prior distributions using Monte Carlo sampling. Each parameter is sampled from its marginal prior distribution independently of the rest of the parameters. The prior distribution for each parameter θ_i is defined using mean and standard deviation estimates as in:

$$\theta_i \sim N(\mu, \sigma) \quad (1)$$

where θ_i is a given Hector parameter and $N(\mu, \sigma)$ is the normal distribution of parameter θ_i using hyperparameters μ (mean) and σ (standard deviation). Some parameters have marginal distributions best represented using lognormal distribution, in such cases, (μ, σ) is substituted by $log(\mu, \sigma)$ (Table 1). Using informed prior marginal distributions from the literature as a starting point for building perturbed parameter sets enables the exploration of a range of possible parameter values from the multi-dimensional parameter space judged viable on the basis of existing knowledge [13,25]. Once each parameter set of draws is performed, the full parameter vector values are combined using a uniform multivariate distribution. This process assumes independence of parameters, meaning for example that any value from the univariate draws of parameter θ_1 is equally likely to be paired with any value from the univariate draws of parameter $\theta_{2...n}$. This parameter estimation process ultimately establishes parameter sets that account for parameter uncertainty and can be used to build an ensemble of model runs that will be evaluated against observational evidence for some salient output of those runs. In other words, we use prior information about individual parameters to build parameter sets, remaining agnostic about which sets will result in the most skilled model results until confronting Hector's output with observed data.

In Matilda, we build parameter sets by calling generate_params(). Parameter distributions are independent of the SSP scenario, however, to run this function the user must still provide an established Hector core. Additionally, the user must specify the number of parameter sets desired (draws). Using generate_params() will produce randomized draws each time it is run. Therefore, the user should either save the resulting data frame or use set.seed() if replication of parameter sets is critical to the analysis. In this example we use our previously established core

Table 1. Hector parameters used in Matilda. Hector parameters used to generate parameter sets in this work. The distributions are indicated as mean ± standard deviation. References from where distributions are derived are included.

Parameter	Description	Units	Distribution	Reference of uncertainty
α	Aerosol forcing scaling factor	Unitless	1.0 ± 0.23 (Normal)	Smith et al (2020) [26]
β	CO_2 fertilization factor	Unitless	0.55 ± 0.10 (Normal)	Jones et al (2013) [27]
ECS	Equilibrium Climate Sensitivity	˚C	3.0 ± 0.65 (Lognormal)	Sherwood et al (2020) [28]
K_{eff}	Ocean heat diffusivity	$cm^2\ s^{-1}$	1.16 ± 0.118 (Normal)	Vega-Westhoff et al (2019) [29]
NPP_0	Pre-industrial net primary productivity	Pg C yr^{-1}	56.2 ± 14.3 (Normal)	Ito (2011) [30]
Q_{10}	Temperature sensitivity of heterotrophic respiration	Unitless	2.2 ± 1.0 (Lognormal)	Davidson and Janssens (2006) [31]

https://doi.org/10.1371/journal.pclm.0000295.t001

to produce a set of 25 parameter configurations and display a subset of samples from the result:

4)

```
param_sets <- generate_params(core = core_ssp245, draws = 25)
print(param_sets)
##       BETA       Q10_RH       NPP_FLUX0       AERO_SCALE       DIFFUSIVITY
ECS
##    1   0.5429609   1.7033771   50.47088   1.2697224   1.070107   2.410262
##    2   0.5234430   1.4867288   51.41584   0.6596398   1.209486   3.092423
##    3   0.4225671   1.3724196   75.76174   0.9010108   1.037287   2.335147
##    4   0.4857051   1.9163999   86.77007   0.7581231   1.243966   2.712076
```

Parameters can be easily added or omitted from the new parameter set data frame. For example, to run the model with a subset of parameters, undesired columns can be omitted from the data frame. This will result in a data frame that only contains parameter distributions that the user wishes to perturb. Similarly, the user can characterize additional parameter distributions and add them as new columns to the data frame, as long as the parameter is described in Hector.

Once established, the parameter sets are used as inputs for independent Hector model runs. Thus, each model run represents a multivariate parameter combination as follows:

$$m_i = (\theta_{1_i}, \theta_{2_i}, \theta_{3_i}, \ldots, \theta_{n_i}) \tag{2}$$

where m_i is an individual ensemble member and $\theta_{\{1-n\}_i}$, are parameters sampled to build an independent configuration. Using different parameter sets to run Hector allows us to build PPEs and determine how different parameter combinations from a presumably viable parameter space interact to affect climate variable projections. This method effectively propagates parameter uncertainty to model ensemble uncertainty, a process described as forward uncertainty propagation [32].

2.5 Forward uncertainty propagation and running the model

We run Hector for each of the parameter sets by calling iterate_model(), which runs the model for each parameter set and combines the results into a data frame object representing the new PPE. To run iterate_model(), the same core object is used as in previous steps and we also must supply the object where parameter sets are stored:

5)

```
results <- iterate_model(core = core_ssp245, params = param_sets)
print(results)
##       scenario       year       variable       value       units       run_number
##    1 Unnamed Hector core 1745 CO2_concentration 277.1500 ppmv CO2 1
##    2 Unnamed Hector core 1746 CO2_concentration 277.1886 ppmv CO2 1
##    3 Unnamed Hector core 1747 CO2_concentration 277.2234 ppmv CO2 1
##    4 Unnamed Hector core 1748 CO2_concentration 277.2558 ppmv CO2 1
```

The resulting data frame returns 25 separate runs, as indicated by the run_number column; in this case, the total number of rows is 55600 (25 runs × 4 output variables × 556 years). Each run includes values for the major climate variables of a Hector default output (CO_2 concentration, total radiative forcing, CO_2 forcing, and global mean air temperature) for the years 1745–2300 (the time range defined by the SSP INI file we chose above).

While a core object and a data frame of parameter sets are the only required arguments to run iterate_model(), additional arguments can be supplied to reduce the variables and year range returned for each run using save_vars and save_years, respectively. This reduces the size of the data stored in memory, which may be important when running the model to build large ensembles (e.g., 15,000 members as in [13]). Any output variable from Hector can be returned using save_vars() for any year range subset from 1745–2300. In the following example, we supply these arguments to return values only for CO_2 concentration and global mean surface temperature anomaly for the year range 1745–2100:

6)

```
results <- iterate_model(core = core_ssp245, params =
param_sets, save_vars = c("CO2_concentration", "gmst"),
save_years = 1745:2100)
```

The resulting data frame has only 17800 rows, a 68% savings over the full example above.

2.6 Model evaluation approach and scoring model runs

Evaluating ensemble members is important in climate model assessment because we do not know *a priori* which parameter sets yield realistic simulations. Evaluation procedures allow us to gain insight into the uncertainty of model outputs, thereby enhancing the fidelity of results. The concept of weighting ensemble members is intuitive; members that are more skilled (i.e., agree better with the historical record) should receive a higher weight than members that are less skilled (i.e., present larger deviations from the historical record). Ensemble members closely aligned with historical climate data will contribute more information to our probabilistic projections than members with outputs deviating significantly from the historical record. Thus, the uncertainty from varying the model parameters *a priori* is "filtered" through an evaluation of the parameters performance against observations. Model weights are analogous to the likelihood factor in Bayes Theorem (defined in Sec. 2.6.1) and could potentially be used to update prior parameter distributions to approximate the posterior parameter distributions. While this is a capability of Matilda, we do not discuss it at length in this paper. It is also important to note the limitations to this approach. While the weighting approaches presented here can be effective for updating marginal distributions of parameters independently, they may not fully capture the complex dependencies present in multidimensional distributions. This limitation becomes noticeable as the dimensionality of parameter space increases. In these cases, formal Bayesian approaches (e.g., Markov Chain Monte Carlo) are better suited for exploring a multidimensional parameter space.

Scoring PPE members in Matilda is conducted using score_runs() which requires (1) a results data frame, (2) a scoring criterion, and (3) a scoring function/algorithm. The results data frame typically comes from calling iterate_model(), as above.

Scoring criteria define information used to compare ensemble members against observational data. A scoring criterion can be built by the user by calling new_criterion() and simply requires

the climate variable to be used in the comparison, the years of comparison, and observed data values for the years specified. For example, a new criterion can be created based on global mean surface temperature from a dataset containing observed warming values from 1990–2023:

7)

```
temp_data <- read.csv("example_temperature_data.csv")
head(temp_data)
##       year       anomaly_C
##    1   1990    0.3605625
##    2   1991    0.3388965
##    3   1992    0.1248968
##    4   1993    0.1656585
##    5   1994    0.2335498
##    6   1995    0.3768662
user_temp_criterion <- new_criterion(var = "gmst", years =
temp_data$year, obs_values = temp_data$anomaly_C)
print(user_temp_criterion)
## Criterion for screening Hector: gmst 1990 to 2023
```

This defines a custom criterion: a time series of 34 (1990–2023) values that will be compared against Hector's "gmst" (global mean surface temperature anomaly) output variable.

The Matilda package has internally available scoring criteria for easy use, including criterion_co2_obs() and criterion_gmst_obs(). Data contained in criterion_co2_obs() is pulled from the Mauna Loa record of observed annual mean atmospheric CO_2 concentration [33], while criterion_gmst_obs() uses observed annual mean global surface temperature anomaly data retrieved from the HadCRUT5 data set [34].

Scoring functions in Matilda apply different mathematical algorithms to compute model weights based on the results and scoring criterion. We provide multiple mechanisms to weight model outputs against observations, and users can define their own custom functions as well. There are currently two internally available scoring functions called score_bayesian() and score_ramp(), that differ in functionality and computational complexity.

2.6.1 Scoring function: Score_bayesian(). Bayesian probability theory provides a rigorous framework combining prior information, observational data, and model simulations for parameter estimation, model evaluation, and uncertainty quantification [35,36]. The Bayesian weighting scheme presented here was influenced by Bayesian model averaging (BMA) methodology. In BMA, multiple model structures, each with specific parameterizations, are used to make projections [35,37–40]. In the BMA method each model is *a priori* assumed equiprobable but once observations are introduced, model probabilities change. Models that are more consistent with observations are considered more likely, while the likelihood of models that are inconsistent with observations is reduced [41]. During this process no model is completely excluded from contributing information, but models are down weighted if they do not accurately represent observations. In this way, score_bayesian() in Matilda behaves like BMA by

computing weights for PPE members based on consistency with historical observations. Unlike BMA however, Matilda focuses solely on perturbing parameters of a single model (Hector) to quantify uncertainty and thus does not capture uncertainty associated with model structure. In this section, we detail the Bayesian weighting scheme performed in Matilda.

We approximate the formalism of Bayesian inference to analyze PPE members and assign weights according to the probability of ensemble member m_i conditional upon observed data $\mathbf{Y}$. This is described as the posterior probability and, consistent with Bayes' theorem, is proportional to the product of a chosen likelihood function of m_i given observed data and prior information of m_i (prior probability) [37]. We can express this in equation form as:

$$P(m_i|\mathbf{Y}) \propto L(m_i|\mathbf{Y}) \times P(m_i) \tag{3}$$

where $P(m_i|\mathbf{Y})$ is the posterior probability of m_i conditional upon observed data $\mathbf{Y}$, $L(m_i|\mathbf{Y})$ is the chosen likelihood function, and $P(m_i)$ is the prior information of ensemble member m_i.

As demonstrated in Eq 3, posterior probabilities are dependent on prior knowledge about ensemble member m_i and a likelihood function that quantifies the agreement between ensemble member m_i and observed data $\mathbf{Y}$. Here, we use a normal likelihood distribution function based on root-mean-square error (RMSE) which is commonly used as a statistical evaluation of model performance in climate research and is optimal under the assumption that errors are normally distributed [42,43]. For a given time series of observed data $_{(t)}$ (i.e., scoring criteria) and a corresponding time series of each ensemble member $m_{(t)}$, RMSE is a quantification of the averaged difference between the observed and modeled data and is calculated using the following formula:

$$\mathrm{RMSE}_i = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathbf{Y}_{(t)} - m_{i(t)}}{\sigma_{\mathbf{Y}(t)}}\right)^2} \tag{4}$$

here, RMSE_i is an independent RMSE value representing how well PPE member m_i agrees with observations where $\sigma_{\mathbf{Y}(t)}$ represents the standard deviation of the error associated with the observations $\mathbf{Y}$ at time t and N represents the total number of data points in the time series. We further use these RMSE_i values in our likelihood function. Assuming a normal distribution, our proportional likelihood can be calculated as:

$$L(m_i|\mathbf{Y}) = e^{-\frac{1}{2}\left(\frac{\mathrm{RMSE}_i}{\sigma_{\mathrm{RMSE}}}\right)^2} \tag{5}$$

In this equation, a decay relationship exists between RMSE_i and $(m_i|\mathbf{Y})$, indicating a gradual decrease in $(m_i|\mathbf{Y})$ as RMSE_i increases. In other words, the likelihood of an ensemble member decreases as the disagreement with observations increases. The value of σ_{RMSE} in Eq 5 plays a role in establishing the sensitivity of the likelihood to increased RMSE values, controlling the rate of likelihood decay. The relationship is explained in detail below and can be visualized in Fig 2.

As described in Sec. 2.4, we enforce an equally distributed prior (m_i) across all ensemble members because although we use prior information about individual parameters of Hector to build parameter sets, we do not know which sets will result in the most skilled model results before considering observed data. Taking all this information together, weights are estimated in score_runs() as normalized probabilities of each ensemble member, taking into account both agreement with observed data and prior beliefs about ensemble members using the following formula:

$$\omega_i = \frac{L(m_i|\mathbf{Y}) \times P(m_i)}{\sum_{i=1}^{N}(L(m_i|\mathbf{Y}) \times P(m_i))} \tag{6}$$

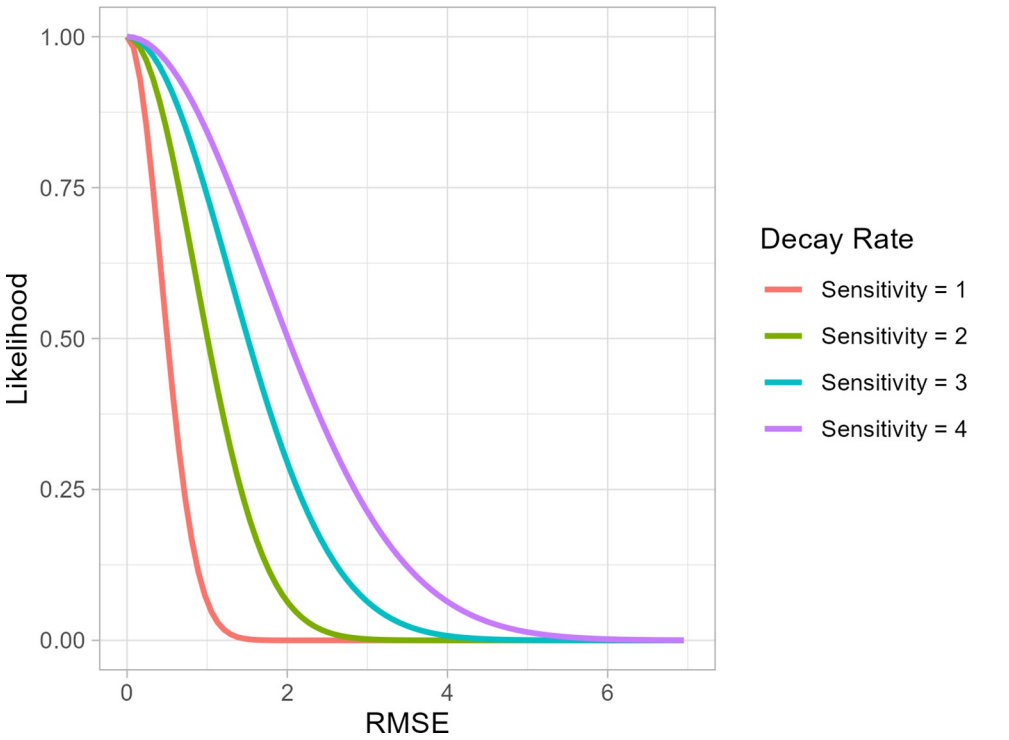

Fig 2. Decay rates from varying sensitivity values. Root-mean-square error (RMSE) plotted against likelihood, conditional upon observed data. Different colors indicate decay rates for different sensitivity values in score_bayesian (). Setting higher values to sensitivity decreases the deviation penalty applied to ensemble members.

https://doi.org/10.1371/journal.pclm.0000295.g002

where ω_i is a weight assigned to each ensemble member.

We provide an example of using score_bayesian() as the scoring function when calling score_runs(). For this example, we use the result produced in code block 5, and analyze the agreement between ensemble members and observed data with the criterion_co2_obs() scoring criterion:

8)

```
scored_hector_runs <- score_runs(results, criterion_co2_obs
(), score_bayesian, sigma = 29.4, sensitivity = 2)

print(scored_hector_runs)

##       weights       run_number
##    1    0.093578592    1
##    2    0.005328194    2
##    3    0.138975159    3
##    4    0.008768552    4
##    5    0.040364027    5
```

The resulting data frame returns 25 weights assigned for each Hector run (indicated by run_number). Weights for each ensemble member will be nonzero positive values that sum to 1. Weights closer to 1 represent ensemble members in strong agreement with observed data and weights closer to 0 correspond to members that are not in strong agreement with observed data. It is important to note that weights are asymptotic and thus no one ensemble member can score exactly 1 or reach a value of exactly 0 due to the normalization process.

The sigma parameter in code block 8 corresponds to $_{(t)}$ in Eq 7 and provides an option for the user to provide a time-varying penalty term in the RMSE_i calculation. This way, we address not only observational uncertainties but also changes in those uncertainties over time [44]. For example, uncertainties for global mean surface temperature data from ~1800s are several times higher in magnitude compared to temperature observations in more recent years. By providing an adjustable sigma parameter we account for such cases. By default, the sigma parameter assumes that the variability of residuals remains constant across all years (homoscedasticity) and uses the standard deviation of the observed data to penalize errors between Hector's output and observations uniformly across all years. The user can indicate an alternative (constant) sigma value (as shown in code block 8). Alternatively, a vector of sigma values can be provided which will apply a time-varying penalty to the error term in the RMSE_i calculation.

The sensitivity parameter in code block 8 is a multiplier applied to σ_{RMSE} that sets the decay rate determining how quickly likelihood values decrease as RMSE values increase. Because sensitivity represents the unit of acceptable deviation, a lower sensitivity value leads to a faster decay rate, meaning that the likelihood decreases more rapidly with increasing RMSE values. Conversely, a higher sensitivity reduces the decay rate, and ensemble member likelihood decreases more slowly with respect to increasing RMSE values. This results in more weight being assigned to ensemble members that have a lower likelihood. In Fig 2, we show how setting different sensitivity values in score_bayesian() leads to different decay rates, meaning that weights are distributed differently depending on the magnitude of the sensitivity parameter. In effect, this determines how severely an ensemble member is penalized as it departs from a criterion.

This parameter thus gives users the ability to govern the sensitivity of the likelihood decay as RMSE values increase. By adjusting the sensitivity value, the user has control over the weight assigned to different ensemble members based on their RMSE values. Setting a lower sensitivity value will result in more weight placed on ensemble members with low RMSE values. In comparison, a higher sensitivity value will give relatively more weight to ensemble members with higher RMSE values. The default sensitivity value is set to 'sensitivity = 1'which quantifies one-unit standard deviation of the RMSE values, this results in a relatively gradual tapering of the likelihood as RMSE increases. We also provide the option in score_bayesian() to set a value to sensitivity directly which will change the acceptable deviation from observed data.

We note that users should adjust sensitivity in line with the context and evaluation purpose specific to their analytical goals. The case study at the end of this paper provides an example of assessing ensemble member weights for different acceptable deviation limits and in Sec. 2.8 we offer a more general discussion of the caveats to model weighting.

2.6.2 Scoring function: Score_ramp(). The score_ramp() function is a simpler and more transparent scoring algorithm that computes the absolute difference between ensemble members $m_i(t)$ and observed data $\mathbf{Y}(t)$ at each time step:

$$D(t) = |\mathbf{Y}(t) - m_i(t)| \tag{7}$$

Scores are then computed based on how far absolute differences (t) are from arbitrarily selected minimum (w_1) and maximum (w_2) divergence values. For example, $D(t) \leq w_1$

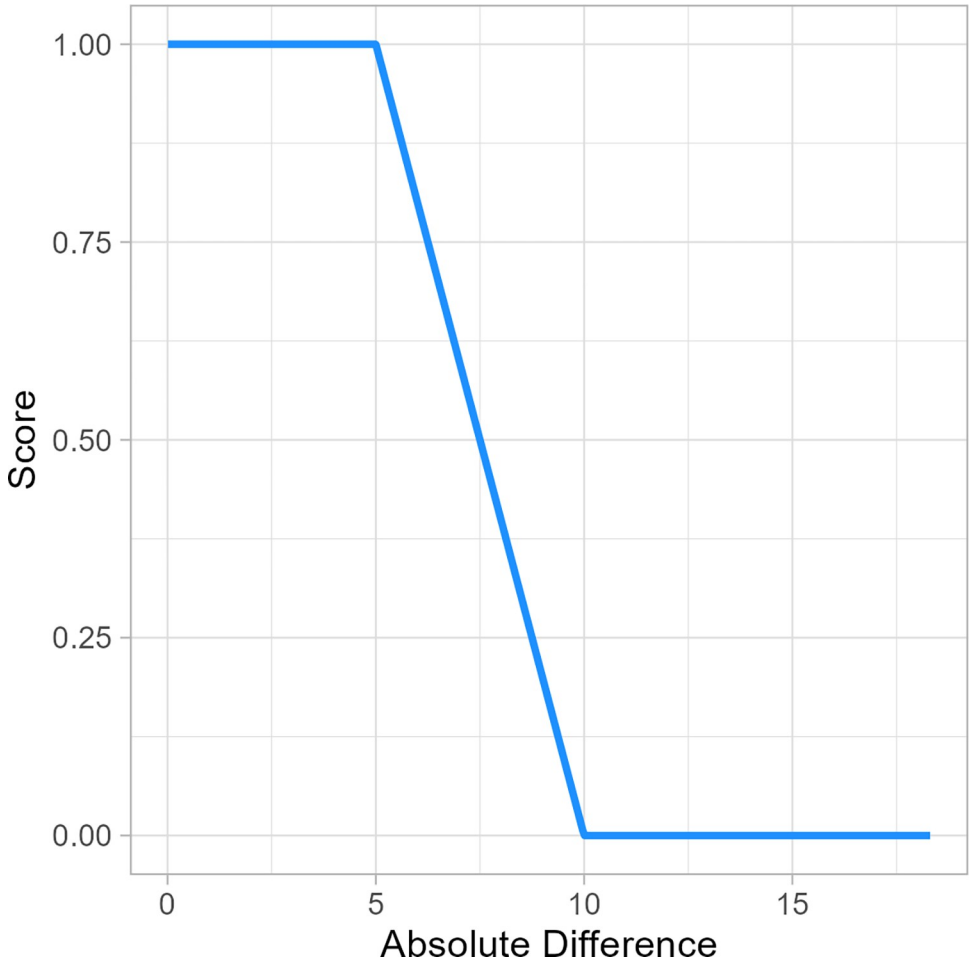

Fig 3. Decay rate for score_ramp(). Example of decay method for score_ramp() where $w_1 = 5$ and $w_2 = 10$. Ensemble members with an average deviation from observation < 5 will score 1 and ensemble members with an average deviation > 10 will score 0. Scores of ensemble members with average deviation between w_1 and w_2 will decrease from 1 linearly as average deviation approaches w_2.

https://doi.org/10.1371/journal.pclm.0000295.g003

indicates small differences between modeled and observed data at time t and will result in a score of 1, whereas $D(t) \geq w_2$ indicates significant divergence of modeled data from observed data and will result in a score of 0 (Fig 3). In cases where $D(t)$ falls between w_1 and w_2, scores are computed using a linear function that decreases as $D(t)$ values get closer to w_2 and further from w_1 (Fig 3).

We can express this linear decay with the following formula:

$$S(t) = \frac{(w_2 - D(t))}{(w_2 - w_1)} \tag{8}$$

where $S(t)$ is the score at time t. Once computed, scores are averaged across the entire time series, resulting in a single score for each ensemble member. Scores for ensemble members are normalized in score_runs() to assign a weight between 1–0 for skilled versus unskilled members, where more skilled ensemble members will be weighted closer to 1 while less skilled ensemble members will receive weights closer to 0. Similar to the normalization step above,

weights are estimated using the following formula:

$$\omega_i = \frac{S_{i(t)}}{\sum_{i=1}^{N}(S_{i(t)})} \tag{9}$$

where ω_i is a weight assigned to each ensemble member from the normalized mean score of each member $S_{i(t)}$.

Below we provide a code example using score_ramp() as the scoring function in a score_runs() call. As in code block 8, we use the results produced in code block 5 and assess the agreement between ensemble members and observed data with the criterion_co2_obs() scoring criterion:

9)

```
scores <- score_runs(result, criterion_co2_obs(), score_
ramp, w1 = 0, w2 = 10)

print(scores)

##       weights       run_number
##    1   1.044760e-02   1
##    2   7.575543e-10   2
##    3   6.271913e-02   3
##    4   8.717546e-21   4
##    5   6.774904e-13   5
```

Similar to scoring ensemble members with score_bayesian(), the resulting data frame returns 25 weights assigned to each ensemble member (indicated by run_number).

Ensemble members weighted according to our scoring algorithms can be used to visualize the uncertainty ranges that different parameter combinations and/or parameter sampling distributions generate after observational evidence is brought to bear on their results. In Fig 4, we show all ensemble members weighted using our two scoring algorithms. The ensemble shading visually demonstrates how score_bayesian(), with a default sigma and sensitivity, distributes weights more evenly across likely ensemble members, whereas score_ramp() assigns higher weights to ensemble members falling closer to the minimum divergence range of w_1 (when w_1 = 0 and w_2 = 10 ppm).

A time test on an analysis weighting 1000 ensemble members shows that the difference in computation time between the two scoring algorithms is negligible, with both functions computing weights for N = 1000 runs in a fraction of a second.

2.7 Defining and calculating metrics and probabilities

Once the ensemble members are scored, we can use them to compute informative metrics from model projections. Calculating metrics from the final weighted PPE requires (1) a results data frame and (2) a metric object, which must first be defined by the user.

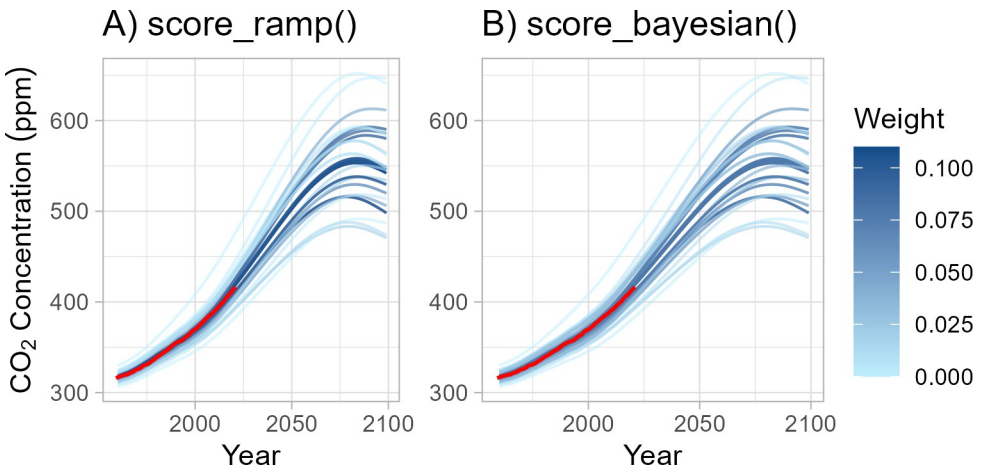

Fig 4. Weighted ensemble members using different scoring algorithms. Perturbed parameter ensemble (PPE) projections using 25 parameter sets plotted for atmospheric CO_2 concentration from 1960–2100 weighted using the A) score_ramp() and B) score_bayesian() algorithms. Ensemble member weights are indicated by color shading with the solid red line representing observed atmospheric CO_2 concentrations from 1959–2021.

https://doi.org/10.1371/journal.pclm.0000295.g004

Metric objects determine what data the user is most interested in extracting and summarizing from the results data frame. For example, a metric object can identify information needed to estimate global mean air temperature anomaly (global_tas) for the 20-year average used by the IPCC to represent long-term temperature change (2081–2100). We complete this by calling the function new_metric():

10)

```
metric_lt <- new_metric(var = "global_tas", years =
2081:2100, op = mean)

print(metric_lt)

## Probabilistic Hector Metric: mean global_tas 2081 to
2100
```

This defines a new custom metric object: obtain the mean global air temperature (global_tas) for the years 2081–2100. The argument op in code block 10 describes a statistical operation that can be performed on the model data to compute a descriptive statistic for each member of the ensemble. While we define a 20-year mean global_tas metric in this example, the user can also easily compute the median, min, max, standard deviation, etc. for each ensemble member. Additionally, users can specify a single year rather than quantifying statistics over a range of years (e.g., 2100 vs. 2081–2100).

Once this metric is defined, we call metric_calc() to compute metric values for each ensemble member using the results data frame:

11)

```
values_metric_lt <- metric_calc(results, metric_lt)

print(values_metric_lt)
```

```
##       run_number       metric_result
##    1    1                  2.315857
##    2    2                  3.083498
##    3    3                  2.232456
##    4    4                  2.474680
##    5    5                  1.975036
```

The resulting data frame returns 25 separate metric values (indicated by the metric_result column) representing the 2081–2100 mean warming of global air temperature.

When metrics of interest are calculated and weights are assigned to PPE members based on agreement with historical record, we have the necessary information to address questions such as, "*What is the probability that long-term mean warming will be between 2.0–4.0°C relative to pre-industrial reference*?"

We approach such a question by calling prob_calc(), a function that sums PPE weights as they are binned into metric ranges identified by the user. Running prob_calc() requires (1) a data frame where metric values can be identified, (2) bins defined by the user, and (3) a data frame where PPE weights can be identified. Here, we provide an example of prob_calc() usage:

12)

```
# Establishing metric ranges
temp_range <- c(1.5, 2.0, 2.5, 3.0, 3.5, 4.0, Inf)
# Producing probabilities
prob_calc(metrics = values_metric_lt$metric_result,
bins = temp_range,
scores = scored_hector_runs$weights)
bins          scores        probability
##    1    (1.5,2]    0.04765274    0.04765274
##    2    (2,2.5]    0.18914194    0.18914194
##    3    (2.5,3]    0.40841067    0.40841067
##    4    (3,3.5]    0.26876412    0.26876412
##    5    (3.5,4]    0.05342877    0.05342877
##    6    (4,Inf]    0.03260175    0.03260175
```

In this example we use PPE weights computed using the score_bayesian() algorithm. The prob_calc() result shows the total probability that long-term projections of mean warming will fall within each of the temperature ranges defined by our bins (1.5–2˚C, 2–2.5˚C, 2.5–3˚C,

3–3.5˚C, 3.5–4˚C, and >4˚C) for the SSP scenario represented in our core object (SSP 2–4.5). With the result above, for example, we can conclude that there is a ~90% probability that the long-term average global warming will be between 2.0–4.0˚C relative to pre-industrial reference.

2.8 Caveats to model weighting for probabilistic analysis

While we chose to provide the user with the flexibility to tailor our model weighting framework to fit the specifics of individual analyses, we encourage users to exercise caution and consider some caveats when using Matilda for probabilistic analyses. First, we provide several methods for weighting PPE members with different scoring criteria. It is important to note that decisions regarding the choice of scoring algorithm and criteria can be open-ended, and different approaches will yield different results. For example, likelihood weights computed using observed temperature alone will differ from those computed with observed CO_2 alone and weights computed using multiple lines of evidence (can be completed in Matilda using multi_criteria_weighting()) will potentially differ from those using either temperature or CO_2 individually. One approach to ensure credible results may include a sensitivity test of different weighting procedures. For example, we conducted a test in which we weighted a 1000-member ensemble using multiple lines of evidence (historical temperature and CO_2 concentration). In this test we show that altering the influence (weight) of the scoring criterion does not significantly change resulting warming probabilities. This would provide evidence that our results are not highly sensitive to changes in the influence of the scoring criterion (S1 Fig). Additionally, Pressburger et al. [13] provide an example of using multiple lines of evidence to apply filters to a PPE in an informative way.

Despite the similarities between the two scoring algorithms, the differences in approach and customization options can lead to variations in the behavior and performance of each method (Fig 4). It is therefore important to consider the specific goals of an analysis and characteristics of the data when selecting which scoring algorithm to use. The benefits of ramp scoring (score_ramp) is its simplicity, making it particularly useful for educational purposes or in preliminary analyses. As Pressburger et al. [13] show, this weighting method can be useful for methodologies aiming to cull ensemble members that fall outside of hard bounds. Bayesian scoring (score_bayesian) is a more complex approach that is conceptually closer to formal Bayesian calibration methods like BMA. The approximation of Bayesian methods allows for a more rigorous approach to model weighting. Unlike our ramp scoring approach, score_bayesian() has the ability to incorporate time-varying error, making it useful in cases where the uncertainty of observed data (scoring criterion) is expected to vary temporally.

Finally, it is important to note that score_bayesian() assumes independence of model-observation residuals. We acknowledge this assumption does not always hold true. Therefore, while our framework provides valuable results, there are limitations by not addressing more complex error structures, such as autocorrelation of residuals [44]. In such cases, more sophisticated modeling techniques may be necessary to improve accuracy of inferences from results [29,44].

3 Case study: Probabilistic temperature projections across four SSP scenarios

Here, we present a case study to demonstrate the core utility of Matilda. We note that this case study is meant only to show the utility of the package and is simply illustrative. It is not meant to be presented as a formal Bayesian probabilistic analysis of temperature change from SSP scenarios. In this case study we will use Hector with four SSP scenarios from CMIP6 (SSP1-

1.9, SSP1-2.6, SSP2-4.5, SSP3-7.0) [5] to compute mean temperature change over the long-term 20-year average presented in IPCC AR6 [45]. We interpret our results similar to the IPCC, using scaled likelihoods: very likely (90–100%), likely (66–100%), about as likely as not (33–66%), unlikely (0–33%), and very unlikely (0–10%) [45]. This case study examines the probabilities of long-term (2081–2100) global mean surface temperature change relative to a pre-industrial reference (1850–1900) in each of the four SSP scenarios.

After initiating cores for each of the four scenarios, we generate 1000 parameter sets using generate_params(). For consistency, these 1000 parameter sets remain the same for each scenario. We call iterate_model() to run Hector across all scenario cores using parameter sets to propagate parameter uncertainty to PPE members. When running the model iterations, we extract the global mean surface temperature (gmst) for the years 1960–2100. Running the model with 1000 parameter sets across four SSP scenarios takes ~100 minutes to run serially on a single processor. Our resulting ensemble members are weighted by calling score_runs(). We weighted ensemble members using observed mean annual global surface temperature as a scoring criterion (criterion_gmst_obs()) with the Bayesian scoring algorithm (score_bayesian). We assume the uncertainty of the observed data to be homoscedastic and therefore use the default sigma value.

Assessing different sensitivity values informs the acceptable RMSE ranges for ensemble members and weights them accordingly. For example, we show how weighting ensemble members using score_bayesian() with a sensitivity = 1 (default sensitivity) places a higher likelihood on ensemble members with lower RMSE values, while manually overriding the sensitivity to sensitivity = 2 decreases the penalty of ensemble members with relatively higher RMSE values (Fig 5).

In this example analysis, we maintain sensitivity at its default value and then visualize PPE members for each SSP scenario analyzed (Fig 6A). This data represents global mean surface temperature normalized to the reference period of 1850–1900 (pre-industrial as previously stated). Normalization of the output data is completed as a post-processing step. The data shown provide some evidence of the range of possible global mean surface temperature futures under each scenario. The most likely outcomes are those that most accurately reflect historical global mean surface temperature patterns (Fig 6A). To summarize the modeled data using our metric of interest (20-year mean of global surface temperature for the years 2081–2100), we first establish our metric definition using new_metric() and then call metric_calc() to compute metrics from our evaluated PPE. For each SSP scenario, we˚compute probabilistic projections of long-term mean warming using prob_calc() for 0.5˚C temperature bins for each SSP scenario (Fig 6B).

From this example, we can infer that when averaged over 2081–2100, the probability of remaining below 2.0˚C warming decreases steadily as we transition from the low emissions scenario (SSP1-1.9) to the higher emissions scenario (SSP3-7.0) (Fig 6B). We can compute approximate probabilities for different temperature ranges for the SSP scenarios by computing the weighted sum of probabilities in desired ranges. For example, in the low emission scenario SSP1-1.9, the probability that warming will remain below 2.0˚C in our long-term warming projection is ~96%. Alternatively, we can calculate precise probabilities by altering bin widths supplied to prob_calc() to ranges that can be compared with IPCC results [45]. For example, in IPCC AR6 the SSP1-1.9 scenario is *very likely* (90–100%) to be warmer by 1.0˚C-1.8˚C relative to pre-industrial reference, while the SSP3-7.0 scenario is *very likely* to be warmer by 2.8˚C-4.6˚C [45]. We find similar results here, where SSP1-1.9 is *very likely* (97%) to be warmer by 1.0˚C-2.0˚C and SSP3-7.0 is *very likely* (92%) to be warmer by 2.4˚C-4.8˚C relative to the pre-industrial reference. For scenarios SSP1-2.6 and SSP2-4.5, the corresponding *very likely* ranges in IPCC AR6 are 1.3–2.4˚C and 2.1–3.5˚C, respectively [45]. For these scenarios,

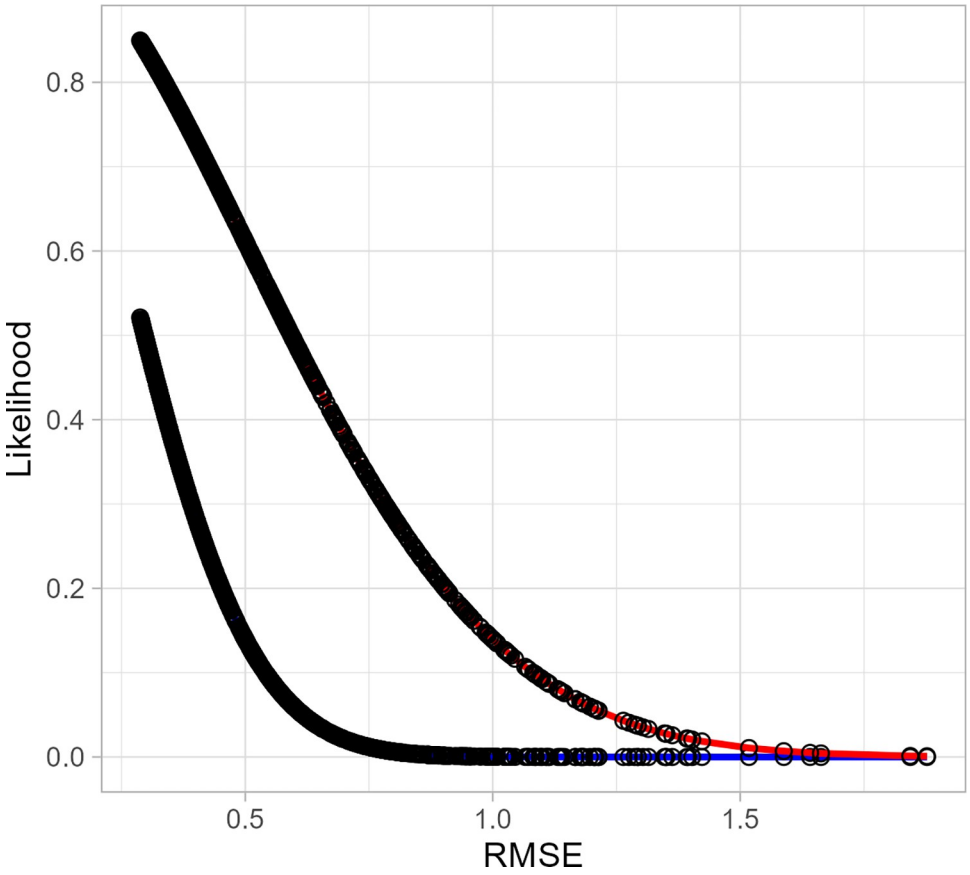

Fig 5. Likelihood of ensemble members given different sensitivity values. Likelihood of perturbed parameter ensemble (PPE) members for an example emissions scenario based on root mean square error (RMSE) using the score_bayesian() algorithm. Blue line shows the use of default sensitivity value: The algorithm penalizes ensemble members as RMSE values deviate from one unit of standard deviation. Red line shows the use of a customized sensitivity value: Setting sensitivity = 2 indicating two units of standard deviation and thus assigns weight to ensemble members falling within this acceptable deviation range. Black dots represent individual ensemble members.

https://doi.org/10.1371/journal.pclm.0000295.g005

we again find *very likely* temperature ranges similar to what IPCC AR6 indicates, where SSP1-2.6 is *very likely* (94%) to be warmer by 1.1˚C-2.6˚C and SSP2-4.5 is *very likely* (92%) to be warmer by 1.7˚C-3.8˚C relative to the pre-industrial reference.

The results from this example case study assume that models can be accurately evaluated using a single scoring criterion of historical data (such as observed global mean surface temperature). However, it is widely recognized that evaluating climate models often involves considering multiple lines of evidence. For example, comprehensive model evaluation should involve assessing performance across various observed climate variables (e.g., temperature, CO_2 concentrations, ocean heat content, etc. and the interactions that exist among those variables), as well as more complex models (i.e., ESMs). The form of the scoring function is also non-univocal in most applications and motivates the flexibility of Matilda in allowing different choices.

4 Conclusion

Using RCMs for probabilistic climate projections is critical for exploring uncertainty in the future integrated human-Earth system [13,17–19]. The use of RCMs presents a viable

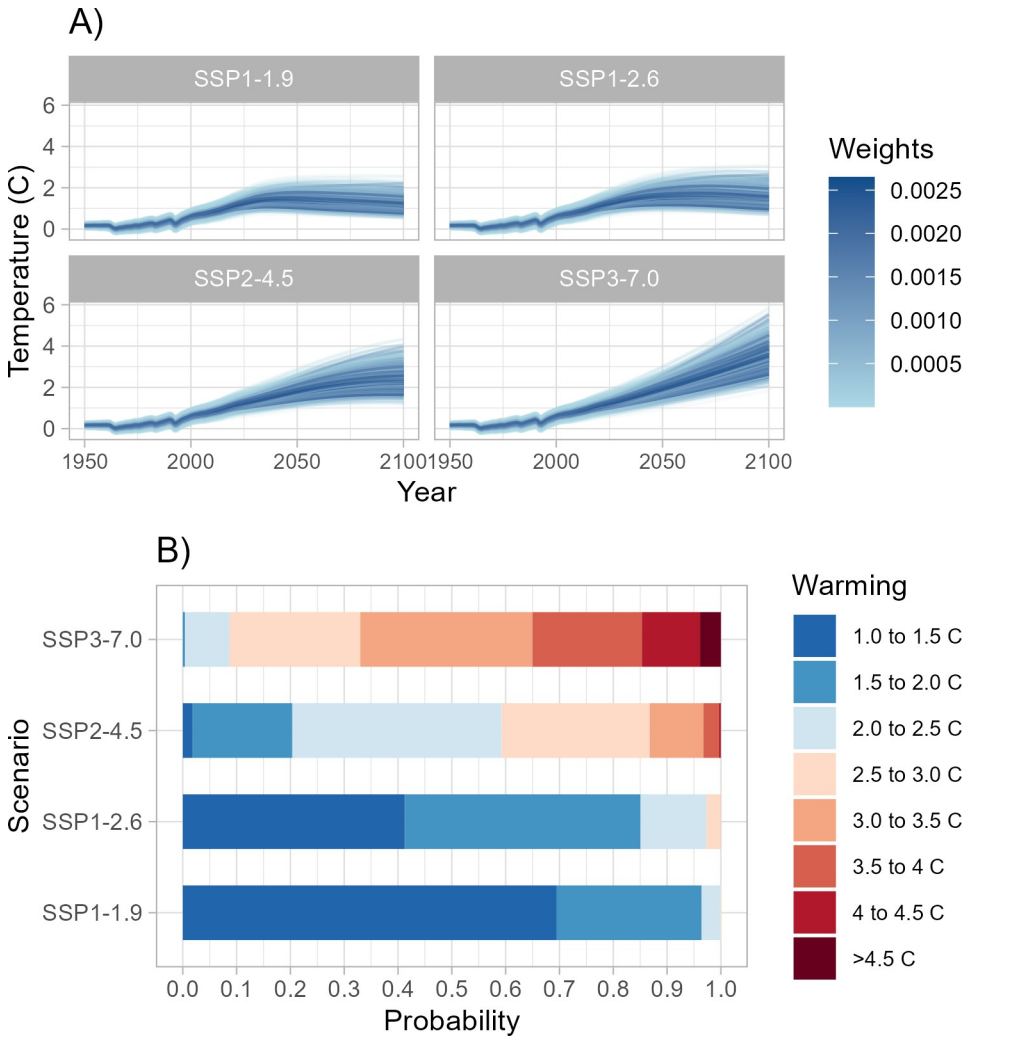

Fig 6. Global mean surface temperature projections and warming probabilities across four emissions scenarios. A) 1000-member perturbed parameter ensemble (PPE) projecting global mean surface temperature from 1950–2100 for each SSP scenario. Darker blue ensemble members represent those members that best reflect historical temperature observations. B) Stacked bars blocked by the probability of different temperature ranges for each SSP scenario. Lower emissions scenarios (SSP1-1.9 and SSP1-2.6) have a higher probability of temperature remaining below 2.0˚C than higher emissions scenarios (SSP2-4.5 and SSP3-7.0).

https://doi.org/10.1371/journal.pclm.0000295.g006

approach to tackle this challenge, as they possess the capability to simulate perturbed parameter ensembles (PPEs) rapidly and can emulate the behavior of more complex ESMs [17] for some key large-scale observable quantities. However, despite their proficiency, some challenges arise when employing several RCMs for probabilistic climate projection analysis. These challenges include closed-source designs and placing heavy programming responsibility on the user [7,10,14,21], which can make both analysis and interpretation difficult.

Matilda is an open-source, turn-key, flexible framework that provides tools to complete probabilistic climate projections using the Hector model. We show how this tool streamlines probabilistic projection analysis and makes such analytical approaches more accessible to the large community of R users, with seamless integration with Hector. By expanding the ways Hector can be used, Matilda can help address questions of climate uncertainty under different emissions scenarios and pursue other probabilistic analyses. We hope that Matilda can be

particularly valuable when coupled with GCAM and similar models to understand the propagation of uncertainty in the human-Earth system [6,19,20,46].

We aim to continue the development of Matilda in several ways. First, we aim to develop enhanced parameter sampling options to enable more robust sampling without relying heavily on *a priori* assumptions about the parametric form of prior distributions. Improving this process can be addressed by implementing more Bayesian approaches into parameter sampling (e.g., Markov Chain Monte Carlo sampling) [47]. While Matilda provides a practical method for approximating posterior distributions without explicitly performing MCMC sampling on each parameter, future versions of the package can be designed to automate this process. Implementing this approach will formalize the Bayesian updating process and make it possible to efficiently explore higher dimensional parameter spaces. Second, providing more methods for model evaluation will improve the robustness of an analysis. While our case study assumes that models can be accurately evaluated using a single scoring criterion like observed global mean surface temperature, it is widely recognized that climate model evaluation is often improved by considering multiple lines of evidence. Comprehensive model evaluation should involve assessing performance across various observed climate variables (e.g., temperature, CO_2 concentrations, ocean heat content, and the interactions among those variables) and/or existing ESMs. Finally, we intend to develop further Matilda's ability to be integrated with additional RCMs. This integration would provide a unified approach for conducting probabilistic projection analysis across different models. By doing so, we can effectively address questions that focus on clarifying uncertainties arising from structural differences among models within the RCM community.

Supporting information

S1 Fig. Sensitivity test using different scoring criterion weights of influence. Warming probabilities for a 1000-member perturbed parameter ensemble weighted using multiple lines of evidence. Lines of evidence were given different influence (weight) in the model weighting scheme to test sensitivity of different weighting procedures. Titles on each panel indicate the influence of each line of evidence. Stacked bars are blocked by the probability of different temperature ranges for each SSP scenario.
(TIF)

Author Contributions

Conceptualization: Joseph K. Brown, Ben Bond-Lamberty.

Data curation: Joseph K. Brown, Ben Bond-Lamberty.

Formal analysis: Joseph K. Brown.

Methodology: Joseph K. Brown, Leeya Pressburger, Abigail Snyder, Kalyn Dorheim, Steven J. Smith, Claudia Tebaldi, Ben Bond-Lamberty.

Project administration: Ben Bond-Lamberty.

Software: Joseph K. Brown, Leeya Pressburger, Abigail Snyder, Claudia Tebaldi, Ben Bond-Lamberty.

Supervision: Ben Bond-Lamberty.

Visualization: Joseph K. Brown.

Writing – original draft: Joseph K. Brown.

Writing – review & editing: Leeya Pressburger, Abigail Snyder, Kalyn Dorheim, Steven J. Smith, Claudia Tebaldi, Ben Bond-Lamberty.

References

1. Moss RH, Edmonds JA, Hibbard KA, Manning MR, Rose SK, van Vuuren DP, et al. The next generation of scenarios for climate change research and assessment. Nature. 2010; 463: 747–756. https://doi.org/10.1038/nature08823 PMID: 20148028
2. van Vuuren DP, Kok MTJ, Girod B, Lucas PL, de Vries B. Scenarios in Global Environmental Assessments: Key characteristics and lessons for future use. Glob Environ Change. 2012; 22: 884–895.
3. Advances Alizadeh O. and challenges in climate modeling. Clim Change. 2022; 170: 18. https://doi.org/10.1007/s10584-021-03298-4
4. O'Neill BC, Kriegler E, Ebi KL, Kemp-Benedict E, Riahi K, Rothman DS, et al. The roads ahead: Narratives for shared socioeconomic pathways describing world futures in the 21st century. Glob Environ Change. 2017; 42: 169–180.
5. O'Neill BC, Carter TR, Ebi K, Harrison PA, Kemp-Benedict E, Kok K, et al. Achievements and needs for the climate change scenario framework. Nat Clim Chang. 2020; 10: 1074–1084. https://doi.org/10.1038/s41558-020-00952-0 PMID: 33262808
6. Hartin CA, Patel P, Schwarber A, Link RP, Bond-Lamberty BP. A simple object-oriented and open-source model for scientific and policy analyses of the global climate system–Hector v1.0. Geoscientific Model Development. 2015; 8: 939–955. https://doi.org/10.5194/gmd-8-939-2015
7. Leach NJ, Jenkins S, Nicholls Z, Smith CJ, Lynch J, Cain M, et al. FaIRv2.0.0: a generalized impulse response model for climate uncertainty and future scenario exploration. Geosci Model Dev. 2021; 14: 3007–3036.
8. Nicholls ZRJ, Meinshausen M, Lewis J, Gieseke R, Dommenget D, Dorheim K, et al. Reduced Complexity Model Intercomparison Project Phase 1: introduction and evaluation of global-mean temperature response. Geosci Model Dev. 2020; 13: 5175–5190.
9. Dorheim K, Link R, Hartin C, Kravitz B, Snyder A. Calibrating simple climate models to individual earth system models: Lessons learned from calibrating Hector. Earth Space Sci. 2020; 7. https://doi.org/10.1029/2019ea000980
10. Meinshausen M, Raper SCB, Wigley TML. Emulating coupled atmosphere-ocean and carbon cycle models with a simpler model, MAGICC6 –Part 1: Model description and calibration. Atmos Chem Phys. 2011; 11: 1417–1456.
11. Woodard, Shiklomanov, Kravitz. A permafrost implementation in the simple carbon–climate model Hector v. 2.3 pf. Geosci Model Dev. 2021; 14: 4751–4767. https://doi.org/10.5194/gmd-14-4751-2021
12. Kikstra JS, Nicholls ZRJ, Smith CJ, Lewis J, Lamboll RD, Byers E, et al. The IPCC Sixth Assessment Report WGIII climate assessment of mitigation pathways: from emissions to global temperatures. Geosci Model Dev. 2022; 15: 9075–9109.
13. Pressburger L, Dorheim K, Keenan TF, McJeon H, Smith SJ, Bond-Lamberty B. Quantifying airborne fraction trends and the destination of anthropogenic CO2 by tracking carbon flows in a simple climate model. Environ Res Lett. 2023; 18: 054005. https://doi.org/10.1088/1748-9326/acca35
14. Smith CJ, Forster PM, Allen M, Leach N, Millar RJ, Passerello GA, et al. FAIR v1.3: a simple emissions-based impulse response and carbon cycle model. Geosci Model Dev. 2018; 11: 2273–2297.
15. Frank P. Propagation of Error and the Reliability of Global Air Temperature Projections. Front Earth Sci Chin. 2019; 7. https://doi.org/10.3389/feart.2019.00223
16. Hall J, Fu G, Lawry J. Imprecise probabilities of climate change: aggregation of fuzzy scenarios and model uncertainties. Clim Change. 2007; 81: 265–281.
17. Nicholls Z, Meinshausen M, Lewis J, Corradi MR, Dorheim K, Gasser T, et al. Reduced Complexity Model Intercomparison Project Phase 2: Synthesizing Earth System Knowledge for Probabilistic Climate Projections. Earths Future. 2021; 9: e2020EF001900. https://doi.org/10.1029/2020EF001900 PMID: 34222555
18. Rogelj J, Fransen T, den Elzen MGJ, Lamboll RD, Schumer C, Kuramochi T, et al. Credibility gap in net-zero climate targets leaves world at high risk. Science. 2023; 380: 1014–1016. https://doi.org/10.1126/science.adg6248 PMID: 37289874
19. Ou Y, Iyer G, Clarke L, Edmonds J, Fawcett AA, Hultman N, et al. Can updated climate pledges limit warming well below 2 C. Science. 2021; 374: 693–695.
20. Fawcett AA, Iyer GC, Clarke LE, Edmonds JA, Hultman NE, McJeon HC, et al. Can Paris pledges avert severe climate change? Science. 2015; 350: 1168–1169.

21. Meinshausen M, Nicholls ZRJ, Lewis J, Gidden MJ, Vogel E, Freund M, et al. The shared socio-economic pathway (SSP) greenhouse gas concentrations and their extensions to 2500. Geosci Model Dev. 2020; 13: 3571–3605.

22. Dorheim K, Gering S, Gieseke R, Hartin C, Pressburger L, N. A, et al. Hector V3.1.1: functionality and performance of a reduced-complexity climate model. EGUsphere [preprint]. https://doi.org/10.5194/egusphere-2023-1477

23. Dorheim K, Bond-Lamberty B, Hartin C, Link R, Nicholson M, Pralit P, et al. Hector a simple carbon-climate model. 2023. https://doi.org/10.5281/zenodo.7951070

24. Evanoff J, Vernon C, Waldhoff S, Snyder A, Hartin C. hectorui: A web-based interactive scenario builder and visualization application for the Hector climate model. J Open Source Softw. 2020; 5: 2782.

25. Jonko A, Urban NM, Nadiga B. Towards Bayesian hierarchical inference of equilibrium climate sensitivity from a combination of CMIP5 climate models and observational data. Clim Change. 2018; 149: 247–260.

26. Smith CJ, Kramer RJ, Myhre G, Alterskjær K, Collins W, Sima A, et al. Effective radiative forcing and adjustments in CMIP6 models. Atmos Chem Phys. 2020; 20: 9591–9618.

27. Jones C, Robertson E, Arora V, Friedlingstein P, Shevliakova E, Bopp L, et al. Twenty-first-century compatible CO2 emissions and airborne fraction simulated by CMIP5 earth system models under four representative concentration pathways. J Clim. 2013; 26: 4398–4413.

28. Sherwood SC, Webb MJ, Annan JD, Armour KC, Forster PM, Hargreaves JC, et al. An assessment of earth's climate sensitivity using multiple lines of evidence. Rev Geophys. 2020; 58. https://doi.org/10.1029/2019RG000678 PMID: 33015673

29. Vega-Westhoff B, Sriver RL, Hartin CA, Wong TE, Keller K. Impacts of observational constraints related to sea level on estimates of climate sensitivity. Earths Future. 2019; 7: 677–690.

30. Ito A. A historical meta-analysis of global terrestrial net primary productivity: are estimates converging? Glob Chang Biol. 2011; 17: 3161–3175.

31. Davidson EA, Janssens IA. Temperature sensitivity of soil carbon decomposition and feedbacks to climate change. Nature. 2006; 440: 165–173. https://doi.org/10.1038/nature04514 PMID: 16525463

32. Eriksson O, Jauhiainen A, Maad Sasane S, Kramer A, Nair AG, Sartorius C, et al. Uncertainty quantification, propagation and characterization by Bayesian analysis combined with global sensitivity analysis applied to dynamical intracellular pathway models. Bioinformatics. 2018; 35: 284–292.

33. Tans P, Keeling R. Mauna Loa CO2 annual mean data. NOAA/ESRL. 2015.

34. Morice CP, Kennedy JJ, Rayner NA, Winn JP, Hogan E, Killick RE, et al. An updated assessment of near-surface temperature change from 1850: The HadCRUT5 data set. J Geophys Res. 2021;126. https://doi.org/10.1029/2019jd032361

35. Massoud EC, Lee H, Gibson PB, Loikith P, Waliser DE. Bayesian Model Averaging of Climate Model Projections Constrained by Precipitation Observations over the Contiguous United States. J Hydrometeorol. 2020; 21: 2401–2418.

36. Vrugt JA, Massoud EC. Uncertainty quantification of complex system models: Bayesian analysis. Handbook of Hydrometeorological Ensemble Forecasting. Springer; 2019. pp. 563–636.

37. Culka M. Uncertainty analysis using Bayesian Model Averaging: a case study of input variables to energy models and inference to associated uncertainties of energy scenarios. Energy Sustain Soc. 2016; 6: 1–24.

38. Samadi S, Pourreza-Bilondi M, Wilson CAME, Hitchcock DB. Bayesian model averaging with fixed and flexible priors: Theory, concepts, and calibration experiments for rainfall-runoff modeling. J Adv Model Earth Syst. 2020; 12. https://doi.org/10.1029/2019ms001924

39. Khan F, Pilz J, Ali S. Evaluation of CMIP5 models and ensemble climate projections using a Bayesian approach: a case study of the Upper Indus Basin, Pakistan. Environ Ecol Stat. 2021; 28: 383–404.

40. Massoud EC, Lee HK, Terando A, Wehner M. Bayesian weighting of climate models based on climate sensitivity. Communications Earth & Environment. 2023; 4: 1–8.

41. Hinne M, Gronau QF, van den Bergh D, Wagenmakers E-J. A Conceptual Introduction to Bayesian Model Averaging. Advances in Methods and Practices in Psychological Science. 2020; 3: 200–215.

42. Hodson TO. Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. Geosci Model Dev. 2022; 15: 5481–5487.

43. Chai T, Draxler RR. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geosci Model Dev. 2014; 7: 1247–1250.

44. Ruckert KL, Guan Y, Bakker AMR, Forest CE, Keller K. The effects of time-varying observation errors on semi-empirical sea-level projections. Clim Change. 2017; 140: 349–360.

45. Intergovernmental Panel on Climate Change (IPCC). Future Global Climate: Scenario-based Projections and Near-term Information. Climate Change 2021 –The Physical Science Basis: Working Group I Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press; 2023. pp. 553–672.

46. Riahi K, Grübler A, Nakicenovic N. Scenarios of long-term socio-economic and environmental development under climate stabilization. Technol Forecast Soc Change. 2007; 74: 887–935.

47. Tsutsui J. Minimal CMIP Emulator (MCE v1. 2): a new simplified method for probabilistic climate projections. Geoscientific Model Development. 2022; 15: 951–970.