Coupling another model to TreeMig

WHY COUPLE MODELS? 2

HOW DOES THE COUPLING WORK? 3

- Definitions and Abbreviations 3
- Coupling concept 3

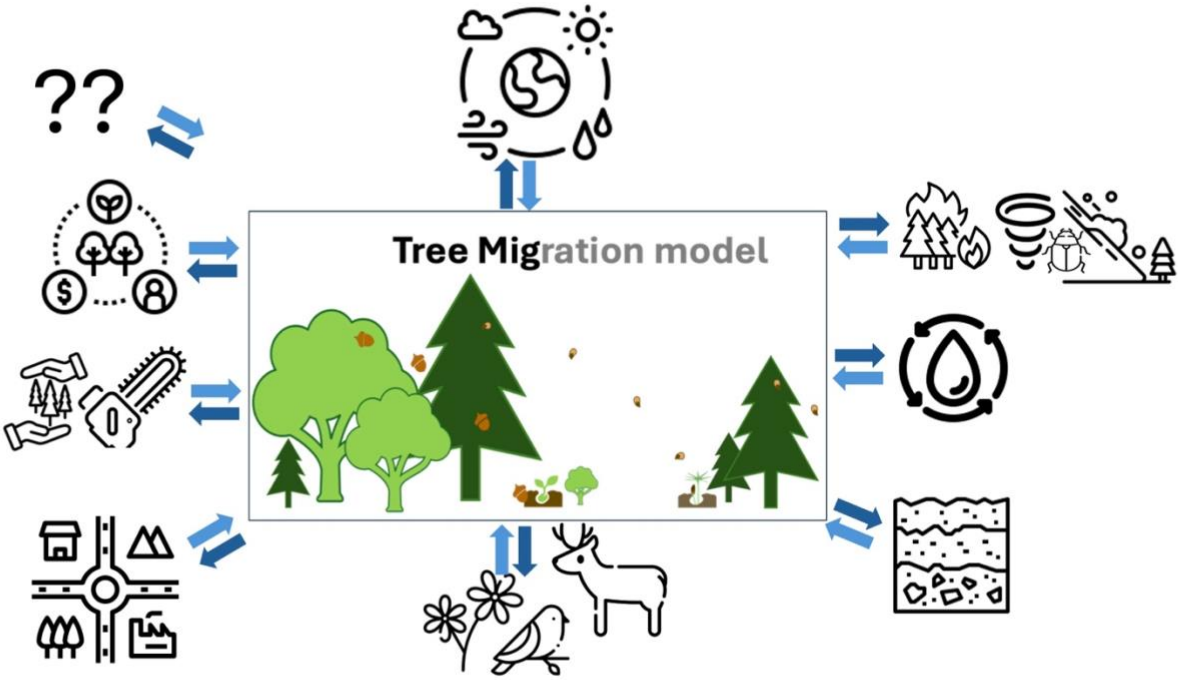
HOW TO DO THE COUPLING 6

SCRIPTS 8

EXAMPLE MODELS FOR COUPLING 9

- Example model DoNothing 9
- Example model ChangeTMStateAtTip1 10
- Example model ChangeBioclimRandomlyAfterTip1 11
- Example model ChangeBioclimDepOnTMAfterTip1 13
- Example model VegClimChange 15
- Example model TreeHerbivoreDynamics 19

REFERENCES 22



Why couple models?

TreeMig (treemig.wsl.ch, (Lischke *et al.* 2006) is a forest landscape model used in various applications, in different regions and times. With the TreeMig R-package (treemig.wsl.ch, (Lischke *et al.*)) it can now be used easily, either by a graphical user interface simplifying all the data preparation settings for TreeMig, or by R-scripts that can be exported from the GUI and then further modified to the user's needs.

The spatial forest dynamics, such as modelled in TreeMig, however does not act in isolation, but is coupled to other processes. It is influenced by external drivers such as climate, disturbances and also strongly impacted by human actions. And the output variables of TreeMig – tree species biomass, structure and composition in space and time – influence other external processes and features such as habitats for other organisms, the local hydrology, the protection function against disturbances, or the timber that can be harvested.

If these coupling is one-way, i.e. forest dynamics does not influence the drivers and is not influenced by the external processes, the coupling can be relatively easily implemented, by providing the input data before the TreeMig simulation or by running after the TreeMig simulation separate models depending on the TreeMig output.

However, if the coupling is two-way, i.e. forest dynamics does influence the drivers or is influenced by the external variables, it is more complex.

In former studies, such as coupling TreeMig with an avalanche release model (Zurbriggen *et al.* 2014) or a hydrological model (Speich *et al.* 2020), this involved modifying the TreeMig Fortran code. This is complex and error prone.

TreeMig offers already the possibility of a coupling to another model without changing the Fortran code. This can be done by iteratively running TreeMig and the other model, reading in the current TreeMig state state and output and changing the TreeMig state according to the external model, such as for rockfall (Moos *et al.* 2021; Moos and Lischke 2022) or river floodplain vegetation dynamics (Reindl 2025), or hypothetical forest management (cf. TreeMig tutorial or case study applications (Lischke *et al.*)). But this iterative approach and in particular handling the TreeMig state file is tricky and error prone, and therefore requires in most cases assistance by the TreeMig developers.

To overcome this hurdle, we have developed a package which simplifies such a coupling for the user. All the user has to provide is the R-script for running TreeMig, the model to be coupled to TreeMig – or an R-wrapper around it – the settings and inputs for it in a standardized but flexible way, and some settings for the coupling.

How does the coupling work?

Definitions and Abbreviations

- TreeMig: TM, TM-state, TM-input, TM-output
- Other model: OM, OM-state, OM-input
- Coupling time points: CTP, can be any, not necessarily equidistant
- Sub simulation period (coupling period): SSP, between CTP_i and CTP_{i+1}
    - CTP_i: at beginning of SSP, in code ti
    - CTP_{i+1}: at beginning of SSP, in code tip1

Coupling function(s): CF

Coupling concept

The entire simulation time is split into sub simulation periods SSPs. TreeMig and the other model run independently and one after the other in each SSP. At each coupling time point CTP, they exchange information by modifying TM's and OM's initial values and input for the following period.

The coupling function supports passing the information between the models in a suitable format.

The user defines the CTPs and which information is passed, and provides the other model and its settings, parameters, and driver data. The coupled simulation itself is done in the function `CoupleOtherModelToTreeMig`, which iterates over the coupling time points, runs TM and the OM for this SSP and exchanges the information at the next CTP.

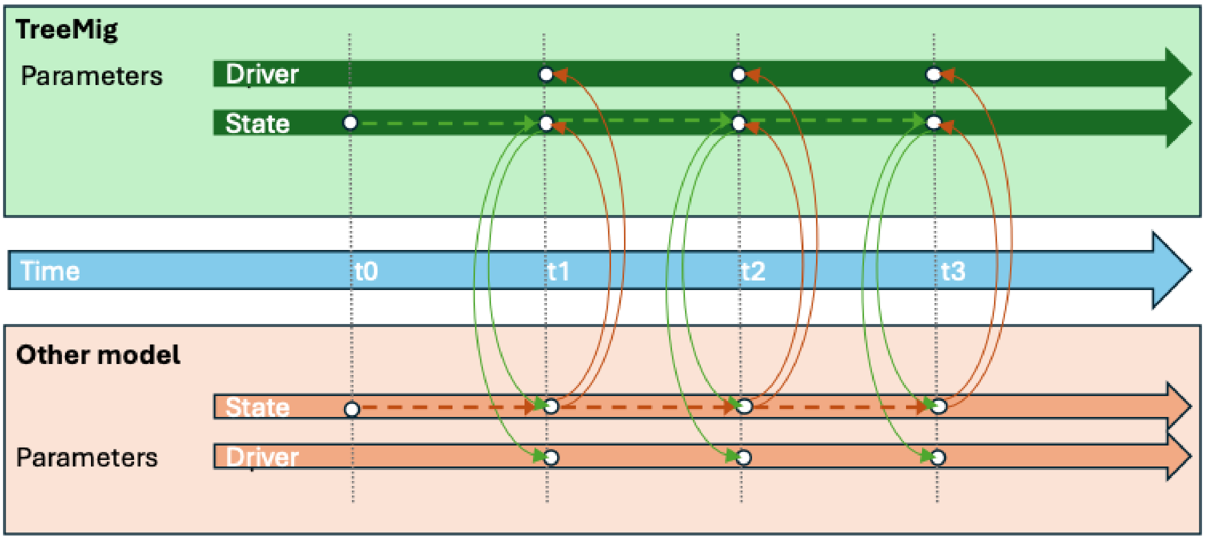


Figure 1: General coupling concept. At each coupling time point CTP (t1,t3, t3), information is transferred between TreeMig and the other model, and state and drivers of the models can be changed accordingly.

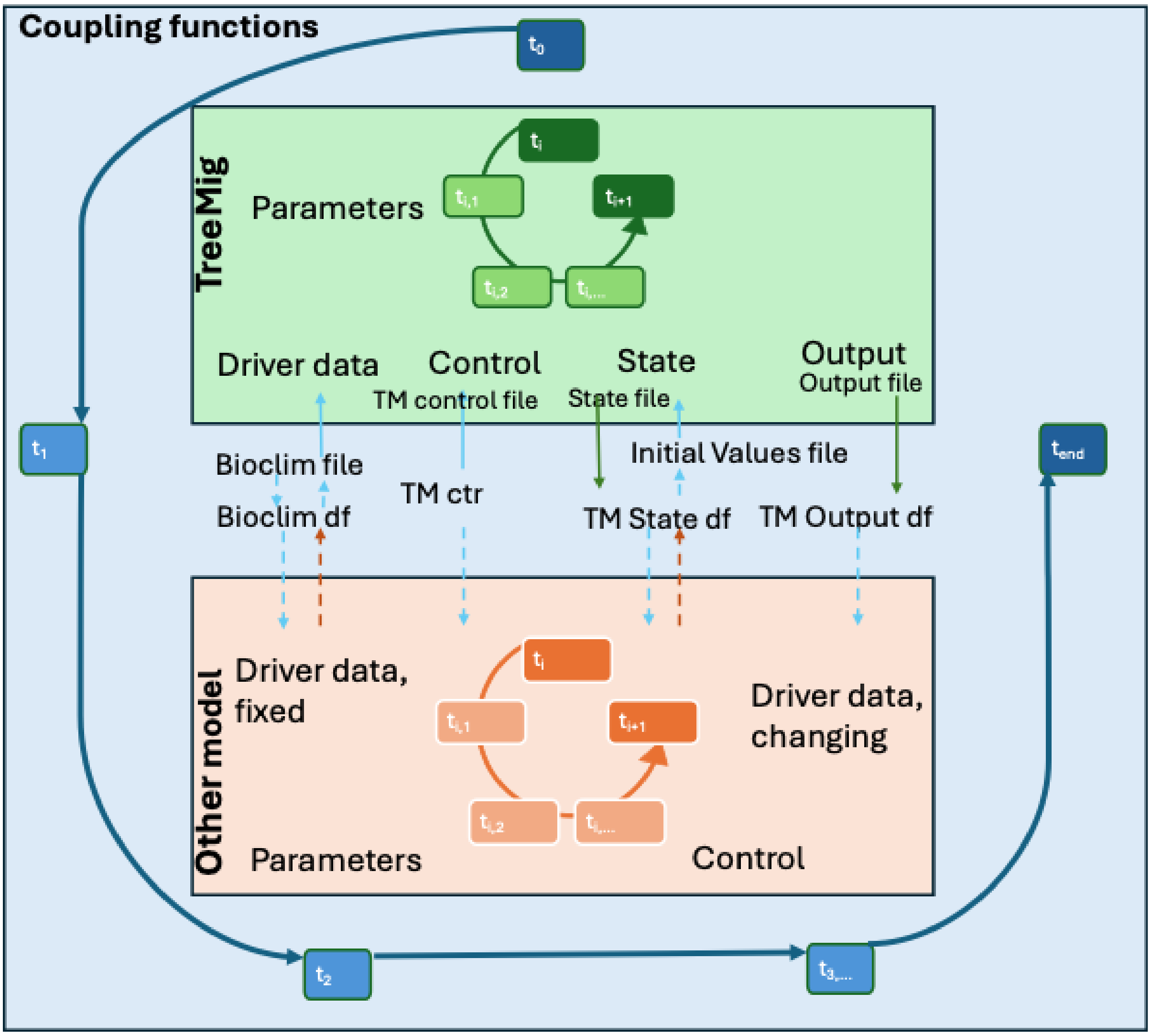


Figure 2: Coupling function(s). Potential information passing at one CTP. Solid arrows: mandatory, dashed arrows optional, to be chosen by user.

TM can be influenced by changing:

1. the TM-state
    a. only at CTP
        i. so this is basically a disturbance to TreeMig
    b. is converted to/from statefile
2. the TM-input (bioclimate)
    a. at coupling time point (disturbance like)
    b. or in sub-simulation period (pertaining change)
    c. is converted to/from bioclimate file
3. or the parameters. We ignore this last one.

OM can be affected by TM in various ways:

1. OM-state
    a. acting only at CTPs

    b. or a dummy (“fake”) TreeMig variable starting from a value simulated and passed by TreeMig acting in the entire SSP
2. OM-Input
    a. Either only in CTP (“disturbance”)
    b. or in the entire SSP

Structure/Flow

1. Coupling function provides time-loop over the different CTPs.
2. Interface
    a. via R dataframes
        i. For state, output and drivers
    b. Calls TreeMig
    c. Calls OM: OM has to meet the requirements of the coupling function. Therefore, the other model has to be formulated in or called from a user provided function with the input and output structure of the function OtherMod. This allows also to call binaries of models in other programming languages by the system statement.
    d. From TreeMig to interface:
        i. state-, output-, and bioclim-files are transformed into dataframes
    e. From OM to Interface: within the OM function the state and bioclim dataframes must be provided. The interface then transforms them into the state- and bioclimfiles for TreeMig.
    f. From interface to TreeMig,
        i. state-, and bioclim-dataframes are transformed into state-, and bioclim-files in TreeMig format
    g. From interface to OM
        i. TM state-, and bioclim-files transformed to dataframes and provided to OM
        ii. General information about the TM simulation, such as simulationID, simulation area and spatial resolution
        iii. Fixed data for the OM: parameters (if not hardwired in OM), input data that are not changed by the simulation
        iv. Variable data for the OM: state of the OM at CTP, changing input data
3. TM
    a. gets
        i. modified state-file (or rather InitialValues.txt)
        ii. modified bioclim file
    b. Runs with them
    c. Outputs
        i. statefile
        ii. output files
4. Other model
    a. Gets (if wished by user)
        i. TM state_dataframe
        ii. TM output dataframes for wished variables
        iii. bioclim dataframe

b. Runs
    i. Calculates/modifies
        1. Own dynamics
        2. Own initial state
        3. Own drivers (if appropriate)

c. Modifies TM state-dataframe

d. Modifies BC- dataframe

Storage of information during coupling iteration

Both, TM and OM need external and internal information to run. This are the control settings, the model parameters, the driver data, that can be the same over the entire simulation or be changed by the models, the internal state of the models at the last CTP used for initializing the models at the new CTP, and also the output data, that should be given out not only for the single SSPs but for the entire simulation period.

For TreeMig, this is handled by the coupling function itself. The results are put in TM_result_df_list_wholeIteration, a list of data frames, one for each output variable.

For OM, there are three lists to deal with that.

- OM_fixed_data is for values that do not change over the simulation. These are typically simulation settings and model parameters. Also, non-changing driver data can be passed by this list. These driver data can be transient and variable over the simulation period but stay the same.
- OM_var_data is for values that change during the simulation. These can be the states of a fully dynamic OM at the CTPs or changed driver data.
- It is up to the user to define these lists and pass their initial content when calling the coupling function.
- OM_var_data_wholeIteration is the overall output of OM, it is up to the user to define via the placeholder function Glue_OM_Results together, how to do append the SSP results of OM together. Either by a user provided function, or by the functions ReplaceOldOMResults, MakeOMList, or MakeOMDataframe.

How to do the coupling

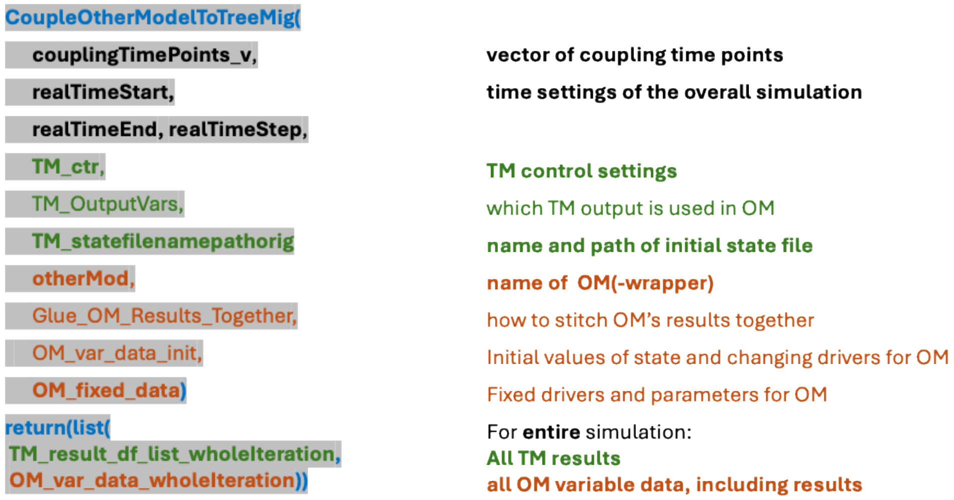
Preparation

At the start of coupling, the states of TreeMig (i.e. the state file) and, if applicable, of the OM must be known. This can be achieved by running spin-up simulations before the coupling, for TreeMig most easily with a TM-R script generated by the TreeMig framework GUI. The bioclimate, settings and state file of TreeMig at the end of this spinup must be saved and backed up. The settings for the coupling need to be defined.

The smallest possible interval between CTPs is one year. This means that it only makes sense to couple models to TreeMig that interact with TreeMig at most once a year. Large CTPs speed up the simulation, as the time-consuming reading and writing of state and output files to the CTPs occurs less frequently, but they can cause discretisation errors.

Run the coupling simulation

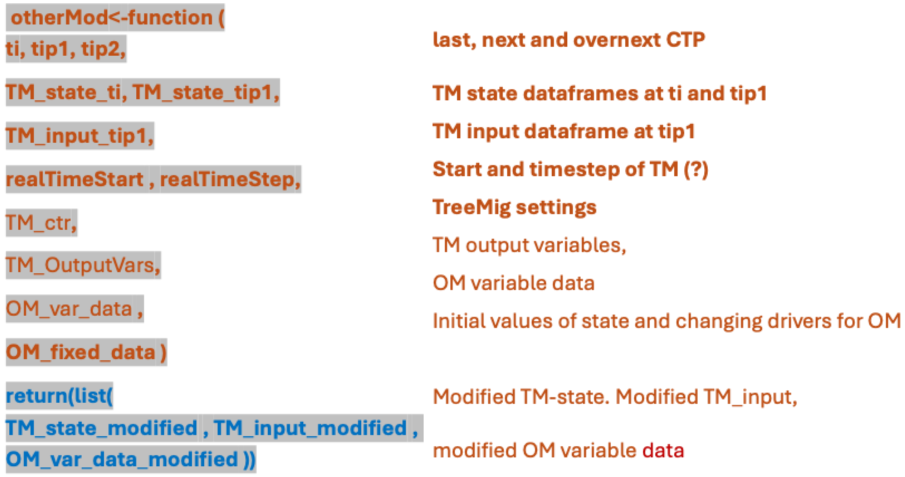
TreeMig and AM are then simulated in a coupled manner using the CoupleOtherModelToTreeMig function, with the following information passed:



CoupleOtherModelToTreeMig(	
couplingTimePoints_v,	**vector of coupling time points**
realTimeStart,	**time settings of the overall simulation**
realTimeEnd, realTimeStep,	
TM_ctr,	**TM control settings**
TM_OutputVars,	which TM output is used in OM
TM_statefilenamepathorig	**name and path of initial state file**
otherMod,	**name of OM(-wrapper)**
Glue_OM_Results_Together,	how to stitch OM's results together
OM_var_data_init,	Initial values of state and changing drivers for OM
OM_fixed_data)	Fixed drivers and parameters for OM
return(list(**TM_result_df_list_wholeIteration,** **OM_var_data_wholeIteration))**	For **entire** simulation: **All TM results** **all OM variable data, including results**

Requirements for the other model

The other model must be written in R, i.e. it will often be a wrapper around the original model. It has the following interface:



otherMod<-function (**ti, tip1, tip2,**	**last, next and overnext CTP**
TM_state_ti, TM_state_tip1,	**TM state dataframes at ti and tip1**
TM_input_tip1,	**TM input dataframe at tip1**
realTimeStart , realTimeStep,	**Start and timestep of TM (?)**
TM_ctr,	**TreeMig settings**
TM_OutputVars,	TM output variables,
OM_var_data ,	OM variable data
OM_fixed_data)	Initial values of state and changing drivers for OM
return(list(**TM_state_modified , TM_input_modified ,** **OM_var_data_modified))**	Modified TM-state. Modified TM_input, modified OM variable data

Within the other model, at a given CTP the TreeMig state data frame and/or the bioclimate are modified according to the rules of OM and based on the TreeMig state and output at the previous CTP.

Scripts

- CoupleOtherModelToTreeMig.R: coupling function and various helper functions. Contains
    - CoupleOtherModelToTreeMig
    - otherMod
    - WriteTMOutputVars
    - ReadTMOutput
    - Glue_TM_Results_Together
    - Glue_OM_Results_Together
    - MakeOMList
    - MakeOMDataFrame
    - ReplaceOldOMResults
    - ReadStateFile2Dataframe
    - WriteDataFrame2Statefile
    - TailorTMSteteFie2TMInputfFile
    - GetTM_OutputAndTransformCoords
    - GetTM_ OutputAndTransformCoordsAndYears
- TestGeneralCoupling.R : calling coupling function with different OMs from VariousOtherModels.
- VariousOtherModels.R: collection of OMs, see below

Example models for coupling

In `VariousOtherModels.R`

Can be tested separately, or with `TestCouplingOM_and_TM.R`.

Example model DoNothing

What is demonstrated

Just to test whether errors arise from the coupling itself

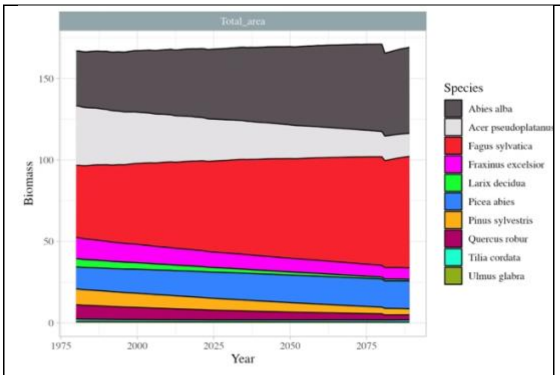
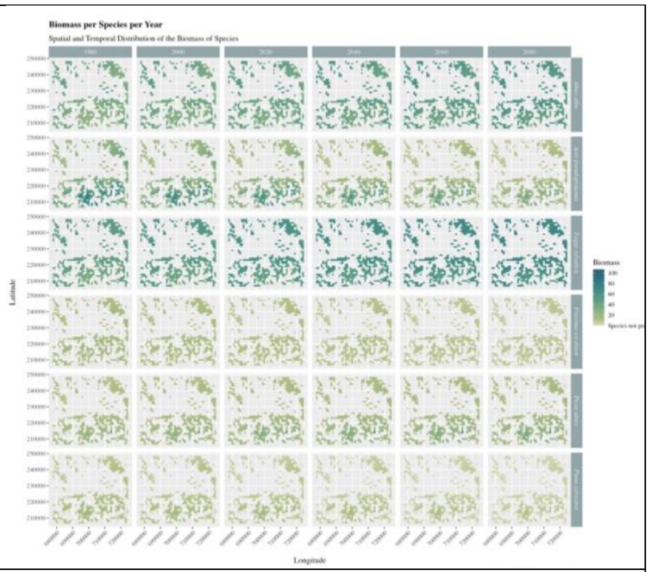
Information passing

- TM -> CF: TM state file, bioclim file
- CF -> TM: TM state file (initial values file), bioclim file

Model

OM does not do anything and neither changes anything in TreeMig. It just returns the state and input transfered from TreeMig. As input only TM_state_tip1 and TM_input_tip1 are needed.

Results

	
TreeMig Biomass (t/ha) in entire coupling simulation time. Only the most dominant species are shown. Left: average over entire simulation area, right: spatial distribution at selected time points.	

Example model **ChangeTMStateAtTip1**

What is demonstrated

One way coupling OM -> TM state, without dynamics but with a certain feedback from TM.

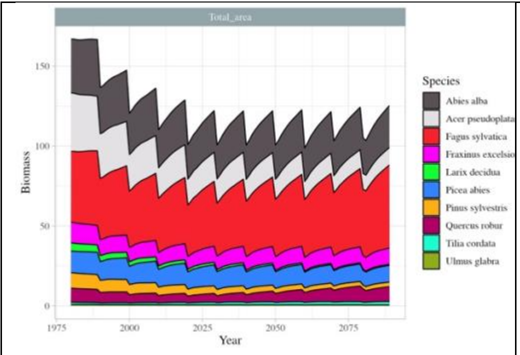
Information passing

- TM -> CF: TM state file
- OM -> CF: TM state dataframe
- CF -> TM: TM state file (initial values file)

Model

In randomly chosen cells with a yearly probability, multiplied with tip1-ti, a given fraction of trees above a certain height is killed. So, the dynamics of TreeMig feeds back via the height of the trees, but does not affect the disturbances.

E.g. random disturbances, windthrow

	
TreeMig Biomass (t/ha) in entire coupling simulation time. Only the most dominant species are shown. Left: average over entire simulation area, right: spatial distribution at selected time points.	

Results

Species biomasses decrease at every CTP, overall reduction to a new pseudoequilibrium.

Example model ChangeBioclimRandomlyAfterTip1

What is demonstrated

One way coupling: other model changes randomly input of TM (bioclim), without dynamics or feedback from TM

Information passing

- TM -> CF: bioclim file
- CF -> OM: bioclim data frame
- OM -> CF: bioclim data frame
- CF -> TM: changed bioclim file

Model

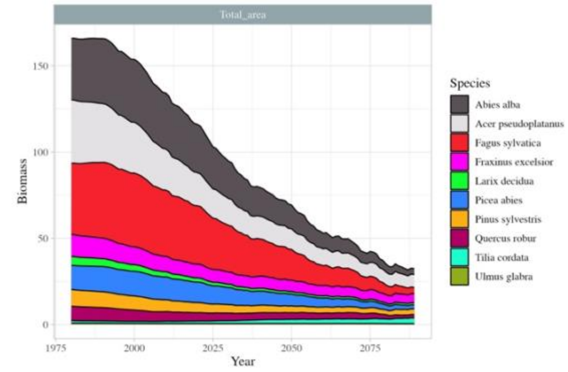
At every tip1, the stockability of random cells is set to zero, that means the cells are omitted in bioclim file. Alternatively, the bioclim values in these random cells are set to unfavourable values for the follow-up SSP.

This is done by

- either erasing these cells in the bioclim (TM_input_tip1); nonstockable ="dropcells"
- or by setting the DD values in these cells to -999;   nonstockable ="setCellsToMin999"

As input in OM only ti, tip1, OM_fixed_data for parameter, and TM_input_tip1 are needed.

Results

	
TreeMig Biomass (t/ha) in entire coupling simulation time. Only the most dominant species are shown. Left: average over entire simulation area, right: spatial distribution at selected time points.	

In TreeMig, Biomass decreases in cells that are affected by the dis   in one SSP, in the next SSP these cells become inhabitable again. But the species then start from biomass zero and have to be colonized from other cells, which results in a low biomass. This is similar to very strong and frequent disturbances. The mean biomass is always calculated over all available (non-dropped) cells.

Example model ChangeBioclimDepOnTMAfterTip1

What is demonstrated

One way coupling OM -> input of TM, without dynamics, but with feedback from TM.

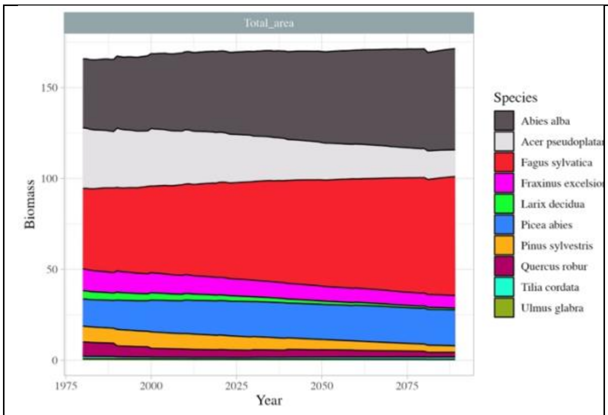
Information passing

- TM -> CF: TM output file, bioclim file
- CF -> OM: bioclim data frame, output (biomass) data frame
- OM -> CF: changed bioclim data frame
- CF -> TM: changed bioclim file

Model

Cells with low biomass that neighbour empty cells are omitted from BC for the next SSPs, i.e. the rest of the simulation. This mimics a sprawl of non forested areas, e.g. by intensified deforestation along roads or urban sprawl. The biomass is read in from the TM output for biomass.

Results

	
TreeMig Biomass (t/ha) in entire coupling simulation time. Only the most dominant species are shown. Left: average over entire simulation area, only stockable cells, right: spatial distribution at selected time points.	

Here the dropped cells remain dropped for the rest of the simulation. Since the biomass is averaged only over the forested (non-dropped) cells, it remains high.

Example model VegClimChange

Very simple climate change model, focusing on the effect of Albedo and carbon sequestration on the temperature.

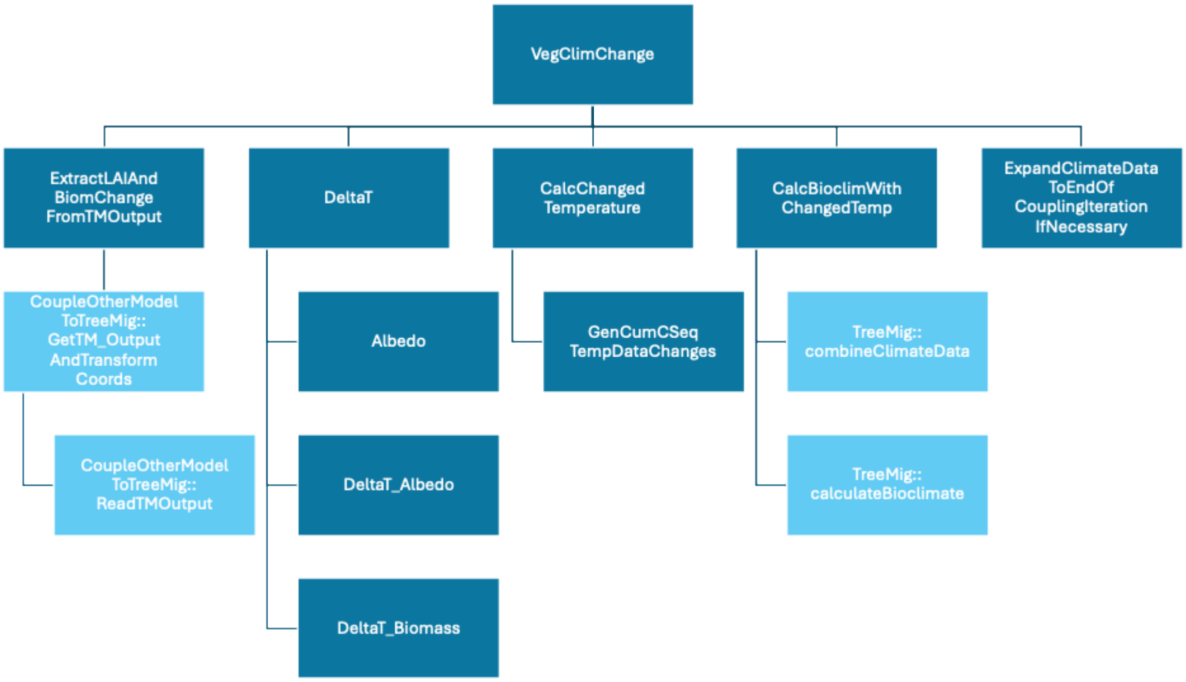
What is demonstrated

Two way coupling with aggregated dynamics of the other model .

Information passing

- TM-> CF: Tree LAI and biomass by TreeMig output files, read in and transformed with GetTM_OutputAndTransformCoords
- CF -> OM: OM_var_data: current temperature (and precip) data, OM_fixed_data: parameters
- OM -> CF: Modified bioclimate data frame
- CF -> TM: Modified bioclimate file

Function structure



Model

General

Temperature T at time t depends on general influences on the previous temperature $T_{std}(t)$, on albedo with the function $\Delta T_{albedo}(LAI(t))$, and on carbon sequestration $\Delta T_{seq}(\Delta Biomass(t))$. DeltaT

$$T = T_{std} + \Delta T_{albedo} + \Delta T_{seq}$$

We assume, that in a given climate data set the effect of the vegetation change is not included, i.e. that it only includes T_{std}. T_{std} is further assumed to be result of a vegetation in equilibrium, i.e. without net carbon sequestration and with an albedo of the value $albedo_{mean}$.

Carbon sequestration effect:

The carbon sequestration effect on temperature of one year *t* is assumed to be proportional to the biomass change per ha and to reduce temperature. DeltaT_Biomass

$$\widetilde{\Delta T_{seq}}\big(\Delta Biomass(t)\big) = -k_{biom}\Delta Biomass(t).$$

$$\Delta Biomass(t) = \frac{Biomass(t_{i+1})-Biomass(t_i)}{t_{i+1}-t_i}$$ ExtractLAIAndBiomChangeFromTMOutput

It is averaged over the period (ti,tip1), and the simulation area.

$$\overline{\Delta T_{seq}}(t) = mean(\widetilde{\Delta T_{seq}}\big(\Delta Biomass(t)\big)$$

Sequestered carbon is removed from the atmosphere, released carbon is added to the atmosphere. This has two consequences:

1. Carbon in (or not in) the atmosphere affects the temperature also in future, and each years carbon sequestration (or release) changes this atmospheric carbon. Therefore, we accumulate the carbon change effect
   $$\Delta T_{seq,local}(t) = \int_{t0}^{t} \overline{\Delta T_{seq}}(x)dx$$
2. Via the atmospheric processes, carbon affects the temperature globally. Therefore, it matters how big the forest area is, i.e. which area is represented by our simulation. Thus, we use an area scale factor area scale that describes how many ha forests the simulation is assumed to be representative for.
   a. $\Delta T_{seq}(t) = \Delta T_{seq,local}(t) * area\ scale$

Parameters

- kbiom:
  - 1 Gigaton (10^9 tons) carbon sequestered corresponds roughly to 0.00048°C (4.8 10^{-4}) temp change (IPCC, SR15) https://www.ipcc.ch/sr15/, https://www.climate.gov/news-features/understanding-climate/carbon-dioxide-removal-noaa-state-science-factsheet#:~:text=Carbon%20dioxide%20removal%20(CDR)%20refers,underlying%20cause%20of%20climate%20change.
  - https://www.carbonbrief.org/guest-post-refining-the-remaining-1-5c-carbon-budget/
  - Carbon gain or loss of 1 t/ha := 4.8 10^{-4} * 10 $^{-9}$ °C= 4.8 * 10 $^{-13}$ °C
- area scale :
  - earth forests $4*10^{12}$ ha https://www.fao.org/interactive/forest-resources-assessment/2020/en/
  - we assume e.g. 1/100 of these forests to be represented by the simulation

Albedo effect:

The albedo effect is given by

$$\widetilde{\Delta T_{albedo}}\big(albedo(t)\big) = -k_{albedo} * (albedo(t) - albedo_{mean})$$ DeltaT_Albedo

$$albedo(t) = \max\,(0, albedo_0 - k_{LAI}\ LAI(t))$$ Albedo

The albedo decreases linearily with LAI from the maximum value $albedo_0$ with no forest. Temperature decreases linearily with increasing albedo.

Albedo affects temperature locally . Here we average it only over the simulation area and coupling period, but do not accumulate it.

$$\Delta T_{albedo}(t) = mean(\ \widetilde{\Delta T}_{albedo}\left(albedo(x)\right)\ )$$

Parameters

- albedo$_{mean}$ is unclear. It maybe refers to the albedo of grass. And maybe to the albedo of a constant standard land cover which entered into the standard climate data.
- k_{albedo}:
    - dT/dAlbedo = 0.09°C/0.1=0.9°C/1
    - https://www.sciencedirect.com/science/article/abs/pii/S0038092X21000475

Total temperature effect:

Then we get the new temperature by just adding

$$T(t) = T_{orig}(t) + \Delta T_{albedo}(t) + \Delta T_{seq}(t).$$ CalcChangedTemperature

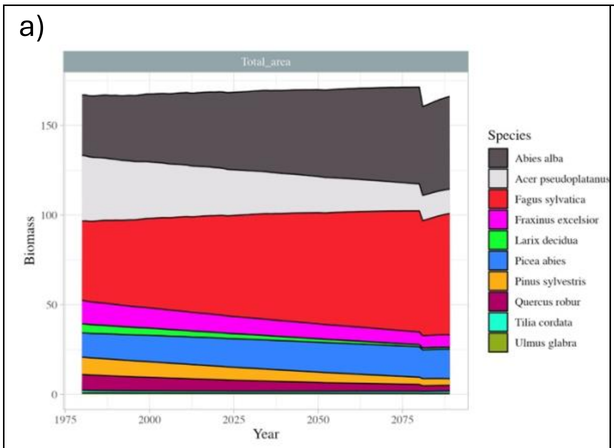
This new temperature is then applied to the next coupling period.

Bioclimate:

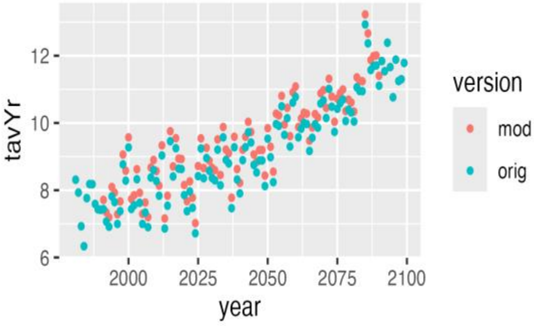
With this new temperature the bioclimate is calculated new for the future.

Results

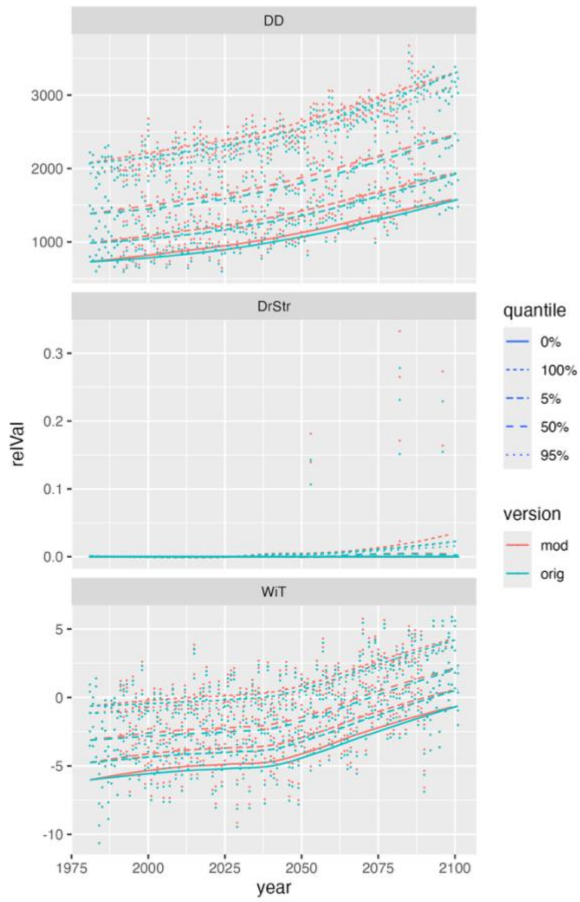
a)



b)



c)



a: TreeMig Biomass (t/ha) affected by feedback in entire coupling simulation time, only in stockable cells

b: original (turquoise) and modified by feedback (red) temperature input, averaged over simulation area

c: original and modified by feedback bioclimatic variables as dots for each cell and year, and as quantile lines over the years.

We see a slight increase in biomass, similar to the DoNothing simulation, but with a smaller drop at the end of the century. This goes along with slightly increased temperatures due to reduced albedo.

Example model TreeHerbivoreDynamics

Simple antagonist dynamics model. A herbivore feeds on the trees of selected species. Their biomasses are reduced within the coupling interval, but without own dynamics. The biomass simulated by TreeMig within the coupling interval is then corrected according to the reduction by the herbivore.

What is demonstrated

Two way coupling with actual dynamics of the other model within each coupling interval. This dynamics even depends on the tree biomass, which is simulated as “fake” within the other model.

Information passing

- TM-> OM
    - Tree biomass by TreeMig output files, read in and transformed with GetTM_OutputAndTransformCoordsAndYears
    - Original state data frame TM_state_tip1 (via coupling)
- OM -> TreeMig
    - Modified statefile via modified state data frame TM_state_tip1_mod.
- Coupling <-> OM
    - OM_var_data: herbivore biomass
- Coupling -> OM
    - OM_fixed_data: parameters

Model

Herbivores die and feed on the trees with a saturation function. The state variables are units of biomass of the herbivores and of the trees. Tree biomass during the coupling interval is only eaten. The dynamics run per grid-cell. No dispersal of herbivores.

At beginning of coupling intervals

The biomass $B_T(ti)$ is calculated by reading in the biomass output file of the previous TreeMig simulation and summing over all food species. The biomass $B_H(ti)$ of the herbivores is the last one of the preceding coupling interval. Also $B_T(tip1)$is determined from TreeMig (and called now $B_{T,TM}(tip1)$).

Dynamics within coupling intervals

Between the coupling time points, the herbivores die with mortality μ and feed with a feeding rate f, which depends on their own biomass and on the total tree biomass of food species. The food species are selected by the user, and can also be all_species.

$$B_H(t+1) = B_H(t) - \mu\, B_H(t) + B_H(t)\, eff\ f(\, B_H(t),\, B_T(t))$$

The feeding function per biomass (“capita”) is a saturation function with a maximum per biomass (“capita”) rate f_{max} and depends on the available tree biomass $B_T(t)$ per biomass unit of the herbivore $B_H(t)$. The half saturation constant K_m determines the the slope of the saturation function.

$$f\big(\, B_H(t),\, B_T(t)\big) = f_{max} \frac{B_T(t)/\, B_H(t)}{K_m + B_T(t)/\, B_H(t)}$$

The uptaken tree biomass $B_T(t)$ is converted into herbivore biomass with a certain efficiency eff.

The tree biomass between the coupling time points decreases accordingly, without any other dynamics.

$$B_t(t+1) = B_T(t) - \ B_H(t)\ \ f(\,B_H(t),\,B_T(t))$$

The dynamics is run with a yearly time step, but allowing for a finer temporal discretization (nsmallsteps within one 1-year step) within each time step. μ and f_{max} are scaled with 1/nsmallsteps.

At end of coupling intervals

At the end of the coupling interval, at tip1, the change (decrease) of tree biomass since ti due to the feeding is determined.

$$\Delta B_T(tip1) = B_T(tip1) - B_T(ti)$$

This (negative) feeding related change is now added to the TM calculated biomass $B_{T,TM}(tip1)$ at tip1.

The ratio between this value and the TM calculated biomass gives the fraction of remaining tree biomass.

$$FractB_T(tip1) = \frac{B_{T,TM}(tip1) + \Delta B_T(tip1)}{B_{T,TM}(tip1)}$$

This fraction is now multiplied to the population densities $N_{sp,h}$ in all height classes *h* of the food species *sp* in the state dataframe, yielding the modified state $N_{mod,sp,h}(tip1)$.

$$N_{mod,sp,h}(tip1) = \ FractB_T(tip1)\,N_{sp,h}(tip1)$$

The non-food species remain at their values.

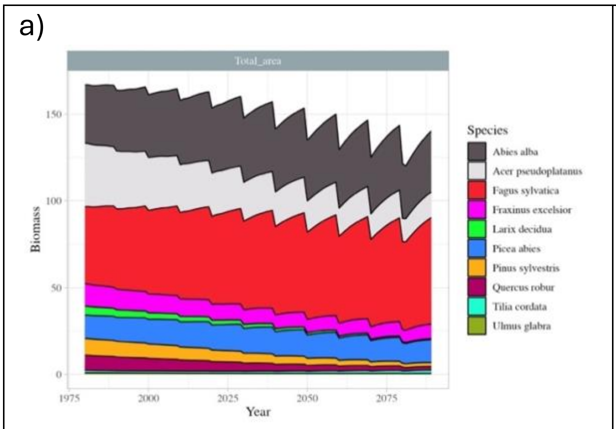
$N_{mod,sp,h}(tip1)$ and $B_H(tip1)$ are then the initialisation for the TreeMig and Herbivore dynamics model in the subsequent coupling interval.

Parameters

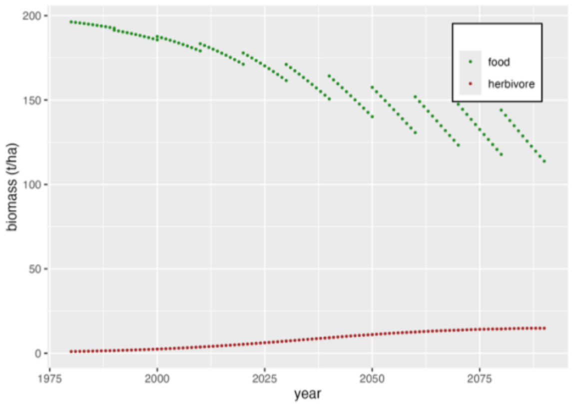
- f_{max}: maximum feeding rate [kg/year] per unit biomass
- K_m : half saturation constant of feeding function
- eff: efficiency to convert tree biomass into herbivore biomass
- μ: mortality of herbivore

Results

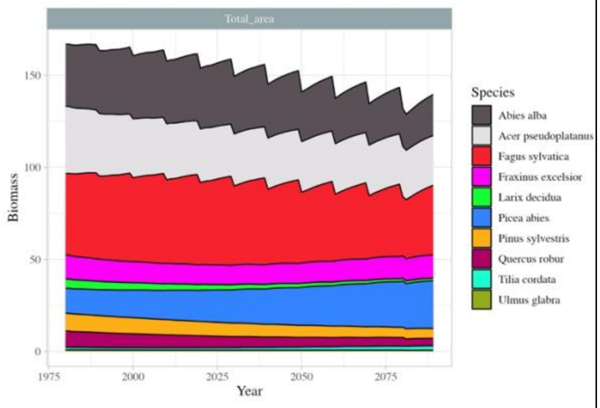
a)



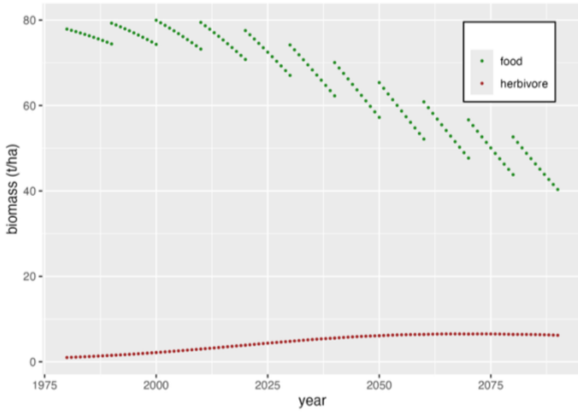
b)



c)



d)



TreeMig Biomass (t/ha) affected by herbivore eating in entire coupling simulation time a) herbirore eating all species c) herbivore only eating *Abies alba* and *Fagus sylvatica*

Biomass of herbivore and food trees in the herbivore model b) herbivore eating all species d) herbivore only eating *Abies alba* and *Fagus sylvatica*

We see a general decrease of the tree biomasss with a sawtooth pattern. At each coupling time point the tree biomass simulated by TreeMig (left a) and c)) is decreased suddenly by the total feeding over the last SSP, but it recovers partly during the next SSP. In the herbivore model (right), the herbivores increase. The available food starts from a level that is increased by the biomass growth of TreeMig, and then it decreases nearly linearly.

The sawtooth pattern thus results from the mutual adjustment of the initial values of tree biomass in each SSP by the respective other model and illustrates the discretisation error.

References

Lischke, H., Zimmermann, N. E., Bolliger, J., Rickebusch, S. and Loffler, T. J. 2006. TreeMig: A forest-landscape model for simulating spatio-temporal patterns from stand to landscape scale. - Ecological Modelling 199: 409-420. doi: 10.1016/j.ecolmodel.2005.11.046.

Lischke, H., Böhm, D., Schmatz, D. R., Psomas, A., Löffler, T., Joss, A. and Scherrer, D. 2026. TreeMig framework: an R-package with GUI to simulate spatio-temporal forest dynamics. - Ecography in press.

Moos, C., Guisan, A., Randin, C. F. and Lischke, H. 2021. Climate Change Impacts the Protective Effect of Forests: A Case Study in Switzerland. - Frontiers in Forests and Global Change 4: doi: 10.3389/ffgc.2021.682923.

Moos, C. and Lischke, H. 2022. Modelling the effect of rockfall on forest development in a dynamic forest model. - Ecosphere 13: e3909. doi: https://doi.org/10.1002/ecs2.3909.

Reindl, M. F. 2025. Modelling riparian vegetation at the Vjosa floodplains : assessing species competition and the impacts of disturbance. BOKU university pp. vii, 114 Seiten, Illustrationen, Diagramme. https://permalink.obvsg.at/bok/AC17768474

Speich, M., Zappa, M. and Lischke, H. 2020. FORests and HYdrology under Climate Change in Switzerland v1.0: a spatially distributed model combining hydrology and forest dynamics. - Geoscientific Model Development 13: 537-564. doi: doi.org/10.5194/gmd-13-537-2020.

Zurbriggen, N., Nabel, J. E. M. S., Teich, M., Bebi, P. and Lischke, H. 2014. Explicit avalanche-forest feedback simulations improve the performance of a coupled avalanche-forest model. - Ecological Complexity 17: 56-66. doi: 10.1016/j.ecocom.2013.09.002.