Final Report

ODD amendment, code verification and example simulations for the landscape extension of the BEEHAVE$_{ecotox}$ model

Authors
Matthias Becher, Alexander Singer & Oliver Jakoby

RIFCON GmbH Report No.
R2260099-1

Completion Date
23.11.2023

Sponsor

BAYER AG
PRO BP R&D Crop Science
Building 6200, 0.16
40789 Monheim

Germany

Realisation

RIFCON GmbH
Goldbeckstraße 13
69493 Hirschberg

Germany

TABLE OF CONTENT

Introduction

This document provides information about the landscape extension for the BEEHAVE$_{ecotox}$ model (Preuss et al. 2022). In the first part, an ODD protocol (Overview, Design concepts and Details) by Grimm et al. (2006, 2010) describes the changes made to the code for the implementation of the extension. This is followed by a verification of the code and an example simulation. The model was implemented in NetLogo (Wilensky, 1999), version 5.3.1.

ODD amendment

The description of the model extension is following the ODD protocol (Overview, Design concepts and Details) by Grimm et al. (2006, 2010). The BEEHAVE$_{ecotox}$ landscape model is an updated version of BEEHAVE$_{ecotox}$ (Preuss et al. 2022). Hence, this amendment is restricted to the changes that have been made to the BEEHAVE$_{ecotox}$ model.

1. Purpose

The purpose of the landscape extension is to overcome the limitation of BEEHAVE$_{ecotox}$ to define available food sources and applied substances only via the graphic user interface. The new version now allows the user to also define more complex landscapes via input text files, similar to the ones used in the original BEEHAVE (Becher et al. 2014) version, but in a shorter format which simplifies input file creation and speeds up reading it in. Additionally, a number of potential plant protection products are defined via an input file of which one can be selected for application.

2. Entities, state variables and scales

In order to create multiple substances, a new Netlogo “breed” *substances* has been defined, to save the defining variables of a plant protection product. Information about the *substances* is either taken from the interface or read from a substance input file. This approach lays the foundation for a possible future expansion to use several substances simultaneously.

New Breeds

breed [substances substance]

Substances-own variables

These variables are similar to the respective (i.e. without the suffix “_Substance”) BEEHAVE$_{ecotox}$ variables defined via the GUI but can have unique values for each substance.

substanceName
etox_DT50_Substance
etox_Forager_Oral_LD50_Substance
etox_Forager_Oral_slope_Substance
etox_Forager_contact_LD50_Substance
etox_Forager_contact_slope_Substance
etox_Larvae_Oral_LD50_Substance

etox_Larvae_Oral_slope_Substance
etox_Forager_ImmediateMortality_Substance
etox_degradation_honey_Substance
etox_DT50_honey_Substance
etox_FF_Nursebees_Nectar_Substance
etox_FF_Nursebees_Pollen_Substance

New Global variables

A new *Input_File* format has been defined that provides the ecotoxicological information for each *flowerPatch*. In contrast to the original format, resources are not defined for each day of the year but via a single flowering period with maximal nectar and pollen availability. This reduces the size of the *Input_File* to one line per food source, which speeds up the setup process and simplifies file handling.
etoxFileFormat?: is true, if the new file format is used
SubstanceID: the ID (*who*) of the applied substance

Renaming of variables

For reasons of consistency, some *foragerSquadrons-own* variables have been renamed (to be similar to e.g. *ETOX_F_PPPNectarConc*):
ETOX_F_PPPNectarDose (previous name: *ETOX_PPPNectarDose*)
ETOX_F_PPPPollenDose (previous name: *ETOX_PPPPollenDose*)
ETOX_F_PPPWaterConc (previous name: *ETOX_PPPWaterConc*)

New flowerpatches-own

The new *Input_File* format requires the definition of the flowering period and the maximal resource availability during that time:

floweringStartDay: first day of year, when a *flowerPatch* offers resources
floweringStopDay: last day of year, when a *flowerPatch* offers resources
quantityMaxMyl: amount of nectar offered daily during the *floweringPeriod*
amountMaxPollen_g: amount of pollen offered daily during the *floweringPeriod*

Repeated applications are possible:
ETOX_NextAppDay_patch [day of year]: day, when the next application at the *flowerPatch* starts
ETOX_StopAppDay_patch [day of year]: day, when the next (or current) application stops
ETOX_ExposurePeriod_patch [days]: duration of the next (or current) application
ETOX_ApplicationList_patch: a list of all the start days of exposures
ETOX_ExposurePeriodsList_patch: a list of all the durations of exposures
ETOX_Nth_Application: counts the number of applications that have so far taken place in each year

New flower-Patches-own variables were introduced that generalize the GUI-parameters for the red and the green patch. This also allows the application of more than two food sources:
ETOX_PPPConcNectar_patch [µg/kg]: like *ETOX_PPPConcNectar_Red* (..._*Green*)
ETOX_PPPConcPollen_patch [µg/kg]: like *ETOX_PPPConcPollen_Red*
ETOX_PPPContact_patch [kg/ha]: like *ETOX_PPPContact_Red* (..._*Green*)

ETOX_WaterVolume_patch [ml]: like *ETOX_WaterVolume_Red* (..._*Green*)
ETOX_WaterConc_patch [µg/L]: like *ETOX_WaterConc_Red* (..._*Green*)
ETOX_RUD_patch [(ha*mg)/(kg*kg)]: like *ETOX_RUD* (i.e. now specific for each *flowerPatch*)

New procedures

New procedures have been implemented to read in a *Substance_File* and then create and parameterise a specific substance. *PlotSetupProc* simplifies the code when creating plots and *DefaultProc_ETOX* sets the parameters on the interface to their default value.

New Procedures BEEHAVE ETOX	Called by:
CreateSubstancesProc	*Setup*
CreateSubstancesSetParametersProc	*CreateSubstancesProc*
DefaultProc_ETOX	*DefaultProc*
PlotSetupProc	*Tplot_exposureandeffects_ETOX*
ReadSubstanceFileProc_ETOX	*CreateSubstancesProc*

CreateSubstancesProc: prepares the creation of a new *Substance* agent with data from the interface or an input file
CreateSubstancesSetParametersProc: here, the actual creation and parameterisation of the new agent takes place
DefaultProc_ETOX: sets the global variables defined on the interface to their default value
PlotSetupProc: simplifies the creation of plots in *Tplot_exposureandeffects_ETOX*
ReadSubstanceFileProc_ETOX: reads in the data for creating all *substances*, if they are defined via the *SUBSTANCE_FILE*.
The new procedures are described in detail in chapter "7. Submodels and alterations".

3. Process overview and scheduling

The changes made for this version of the $BEEHAVE_{ecotox}$ module mainly affect the setup processes, i.e. definitions of the landscape, substances, applications etc. This means that model results of the updated $BEEHAVE_{ecotox}$ version are the same (within the limits of stochastic variability) as the original $BEEHAVE_{ecotox}$ ones, given the same setup was implemented. (For a comparison of the results of both model versions see section "Code verification".
In order to set up landscapes with more than two food sources, the landscape has to be defined via an input file. While the original BEEHAVE *INPUT_FILE* defines resource availability in each patch for each day of the year, a new file format uses only a single line for each patch with a start and stop date for the flowering period (for details, see chapter 6). Additionally, details about pesticide applications can now also be defined via the *INPUT_FILE*. The updated scheduling of the BEEHAVE procedures is shown in Table 1.

Table 1: Scheduling of the BEEHAVE procedures. Procedures added after the publication of the original version (Becher et al. 2014) are shown in red with the version in parentheses. Procedures new to this updated BEEHAVE$_{ecotox}$ version are bold and highlighted.

Level 1	Level 2	Level 3	Level 4	Level 5	Level 6
Setup					
	ReadFileProc				
	ParameterizationProc				
	CreateFlowerPatchesProc				
		FlowerPatchesUpdateProc			
	Create_READ-IN_FlowerPatchesProc				
		FlowerPatchesUpdateProc			
	ReadBeeMappFileProc (BEEHAVE 2016)				
	CreateImagesProc				
	InitialConditionsProc (ETOX 2021)				
	CreateCohortsProc (ETOX 2021)				
	CreateSubstancesProc (ETOX update)				
		CreateSubstancesSetParametersProc (ETOX update)			
		ReadSubstanceFileProc_ETOX (ETOX update)			

StartProc					
	Go				
		DailyUpdateProc			
			TExposure_at_patch_ETOX (ETOX 2021)		
			BeeMappCorrectionProc (Beehave 2016)		
		SeasonProc_HoPoMo			
		WorkerEggsDevProc			
		DroneEggsDevProc			
		NewEggsProc			
		SwarmingProc			
		WorkerEggLayingProc			
		DroneEggLayingProc			
		WorkerLarvaeDevProc			
			MitesReleaseProc		
		DroneLarvaeDevProc			
			MitesReleaseProc		
		NewWorkerLarvaeProc			
		NewDroneLarvaeProc			
		WorkerPupaeDevProc			
			MitesReleaseProc		
			MitesReleaseProc		
		DronePupaeDevProc			
			MitesReleaseProc		
			MitesReleaseProc		
		NewWorkerPupaeProc			
		NewDronePupaeProc			
		WorkerIHbeesDevProc			
			AffProc		
		DronesDevProc			
		BroodCareProc			
			CountingProc		
			MitesReleaseProc		
			MitesReleaseProc		
			MitesReleaseProc		
			MitesReleaseProc		
		NewIHbeesProc			
		NewDronesProc			
		MiteProc			
			CreateMiteOrganisersProc		
			CountingProc		
			MitesInvasionProc		
			MitePhoreticPhaseProc		
			MiteDailyMortalityProc		
			MiteOrganisersUpdateProc		
		BeekeepingProc			
		DrawIHcohortsProc			
		GenericPlotClearProc			

		Start_IBM_ForagingProc			
			ForagersDevelopmentProc		
			NewForagersProc		
			ForagingRoundProc		
				FlowerPatchesUpdateProc	
				Foraging_start-stopProc	
				Foraging_searchingProc	
				Foraging_collectNectarPollenProc	
				Foraging_flightCosts_flightTimeProc	
					Tfeed_on_honey_stores_foragers_ETOX (ETOX 2021)
					TupdateInternalExposureNectar_ETOX (ETOX 2021)
				Foraging_mortalityProc	
				Foraging_dancingProc	
				Foraging_dancingProc	
			WriteToFileProc		
			ForagersLifespanProc		
		CountingProc			
		PollenConsumptionProc			
		HoneyConsumptionProc			
		TupdateInternalExposureNectar_ETOX (ETOX 2021)			
			Tfeed_on_honey_stores_foragers_ETOX (ETOX 2021)		
			Tfeed_on_honey_stores_cohorts_ETOX (ETOX 2021)		
		Tplot_exposureandeffects_ETOX (ETOX 2021)			
			PlotSetupProc (ETOX update)		
		DoPlotsProc			
			DrawForagingMapProc		
			GenericPlottingProc		
	WriteToFileProc				

4. Design Concepts

No changes have been made to the Design Concepts.

5. Initialisation

The initialisation is very similar to the previous versions, except that variables defining the substances are saved as *substances-own* instead of global variables and that the parameterisation of these variables may come from a *Substance_File*.

6. Input data

Two changes have been made here: a new *Substance_File* has been created and a new format for the already existing *Input_File* has been defined.

The Substance_File

The *Substance_File* is a text file (e.g. “SubstanceFile.txt”) that can provide the parameter values of various pesticides. It consists of a header defining the 13 column names and a number of lines, each line defining one substance:

Header	**Example line 1**	**Unit**
substanceName	„Substance1“	
etox_DT50_Substance	100	[d]
etox_Forager_Oral_LD50_Substance	1000	[µg/bee]
etox_Forager_Oral_slope_Substance	100	
etox_Forager_contact_LD50_Substance	0.6	[µg/bee]
etox_Forager_contact_slope_Substance	1.08	

etox_Larvae_Oral_LD50_Substance	0.0014	[µg/bee]
etox_Larvae_Oral_slope_Substance	1.6	
etox_Forager_ImmediateMortality_Substance	FALSE	
etox_degradation_honey_Substance	FALSE	
etox_DT50_honey_Substance	60	[d]
etox_FF_Nursebees_Nectar_Substance	0.25	
etox_FF_Nursebees_Pollen_Substance	1	

The *Input_File*

The *Input_File* has been in use for the original BEEHAVE version already. Additionally to the original file format, which can still be used if no pesticide applications take place, a new, shorter format has been defined. The new *Input_File* format describes pesticide applications at food sources. Compared to the original file format, it was changed in two aspects:

1. The new file format provides on additional information regarding applications of toxicants
2. The definition of resource availability has been simplified. In the original format, resource availability is described explicitly for each patch and day. In the new format, for each patch a single flowering period is given, during which a constant amount of daily nectar and pollen is provided. Outside the flowering period, the patch does not provide resources. This change shortens the file to a single data line per *flowerPatch*.
   Advantage of this change is a significant reduction of file complexity for most cases of BEEHAVE application and a considerable reduction in startup time, when reading landscapes from file. It can also be used, when the BEEHAVE$_{ecotox}$ module is switched off.

The new format is a text file with a header, defining the 24 column names (see table below) and one line per each *flowerPatch*. Most of the landscape defining information is identical to the original file format, except for the two new variables *startDay* and *stopDay*, defining the beginning and the end of the annual flowering period. These are followed by eight columns, describing the pesticide application in a food source:

Header	**Example flowerPatch**	**Unit**
ID	0	
oldPatchID	0	
patchType	„RedPatch“	
distance_m	500	[m]
xcor	500	
ycor	0	
size_sqm	500000	[m²]
quantityPollen_g	1000	[g]
concentration	1.5	[mol/l]
quantityNectar_l	20	[l]
calc_DetectProb	0.2	
model_DetectProb	-999	
NectarGathering_s	1200	[s]
PollenGathering_s	600	[s]

startDay	1	[day of year]
stopDay	365	[day of year]
ETOX_ApplicationList_patch	[50 100]	[days of year]
ETOX_ExposurePeriodsList_patch	[10]	[day(s) of year]
ETOX_PPPConcNectar_patch	990	[µg/kg]
ETOX_PPPConcPollen_patch	26631	[µg/kg]
ETOX_PPPContact_patch	0.3	[kg/ha]
ETOX_WaterVolume_patch	10000	[ml]
ETOX_WaterConc_patch	0	[µg/l]
ETOX_RUD_patch	21	[(ha*mg)/(kg*kg)]

ETOX_ApplicationList_patch is a list (i.e. in brackets), with each number defining the start day of an annual pesticide application. If the list is empty (i.e. []), no application takes place. *ETOX_ExposurePeriodsList_patch* describes the duration of each application. If only one number is present, this value applies to all applications. If more than one number is provided, the number of durations have to match the number of applications in *ETOX_ApplicationList_patch*. Other variables are as defined in the original BEEHAVE$_{ecotox}$ module.

7. Submodels and alterations

Here, only meaningful changes to the code are listed, not mere editing, simplifications of code, or assertions.

Setup

```
to Setup
  ...
 if ETOX_Fixed_PPPnectar:pollen = true
 [
   ;  ETOX UPDATE 2023:
   ask flowerPatches ;  ETOX UPDATE 2023
       [ set ETOX_PPPConcPollen_patch ETOX_PPPConcNectar_patch * ETOX_N-P-factor ] ; [µg/kg]
 ]

 if SubstanceApplied != "No applications" [ CreateSubstancesProc ]  ;  ETOX UPDATE 2023
```

The pesticide concentration in nectar is no longer saved in the global variables *ETOX_PPPConcNectar_Red* (..*Green*) but in the *flowerPatches-own* variable *ETOX_PPPConcPollen_patch*.

The updated BEEHAVE$_{ecotox}$ version also allows to run the model without applying a *substance* via the new option "No applications" in *SubstanceApplied*.

DailyUpdateProc

Multiple applications of a single compound are now possible. When the current application starts, its end date (*ETOX_StopAppDay_patch*) is determined, as well as the start date (*ETOX_NextAppDay_patch*) and number (*ETOX_Nth_Application* starts) of the next application. (Note that counting of *ETOX_Nth_Application* starts with 0 for the first application in each year):

```
ifelse ReadInfile = false
[
  ask flowerPatches
  [ ; flower patches are set to the max. amount of nectar and pollen possible today:
    set quantityMyl FlowerPatchesMaxFoodAvailableTodayREP who "Nectar"
    set amountPollen_g FlowerPatchesMaxFoodAvailableTodayREP who "Pollen"
  ]
]
[ ; ETOX UPDATE 2023: update PPP application counter and dates of next application:
  ask flowerPatches
  [
    if day = ETOX_NextAppDay_patch and ETOX_ApplicationList_patch != [] ; not called, if no applications are defined
    [
      set ETOX_StopAppDay_patch item ETOX_Nth_Application ETOX_ApplicationList_patch + ETOX_ExposurePeriod_patch ; contamination will end here
                                                          ; unless another application takes place in the meanwhile (or flowerPatch stops blossoming)
      ifelse ETOX_Nth_Application + 1 < length ETOX_ApplicationList_patch
      [ set ETOX_Nth_Application ETOX_Nth_Application + 1 ]
      [ set ETOX_Nth_Application 0 ]
      set ETOX_NextAppDay_patch item ETOX_Nth_Application ETOX_ApplicationList_patch
      if length ETOX_ExposurePeriodsList_patch > 1 ; if only 1 value is provided, this value is applied to all applications..
      [
        if length ETOX_ExposurePeriodsList_patch != length ETOX_ApplicationList_patch ; ..otherwise, there needs to be a value for each application
        [
          user-message "Number of applications specified in ETOX_ApplicationList_patch and ETOX_ExposurePeriodsList_patch do not match (INPUT_FILE)"
          set BugAlarm true
        ]
        set ETOX_ExposurePeriod_patch item ETOX_Nth_Application ETOX_ExposurePeriodsList_patch
      ]
    ] ; end application day
  ] ; end ETOX UPDATE 2023
]
```

In the *DailyUpdateProc* procedure, the amount of nectar and pollen available at a patch is determined. This can either be defined via the GUI (for maximal two food sources) or via an input file. Two file formats are accepted for this input file: the original one defining resource availability for each day of a year or a new, simplified format, where each food source is defined by a single line with a flowering period between *floweringStartDay* and *floweringStopDay* (see chapter 6:

```
if ReadInfile = true
[
  ifelse etoxFileFormat?  = true  ;  ETOX UPDATE 2023
  [
    ask flowerpatches
    [
     ifelse Day < floweringStartDay or Day > floweringStopDay
     [
       set quantityMyl 0
       set amountPollen_g 0
     ]
     [
       set quantityMyl quantityMaxMyl
       set amountPollen_g amountMaxPollen_g
     ]
    ]
  ]
```

Not allowing negative values in the *HoneyEnergyStore*, which was added in the BEEHAVE$_{ecotox}$ module, unintentionally disabled starvation. This has now been corrected by letting colonies di as soon as the honey stores reaches 0:

```
  if HoneyEnergyStore <= 0  ;  ETOX UPDATE 2023: added "=" to enable starvation
  [
    if ColonyDied = false
    [
      output-print word "Starvation! Colony died on Day " ticks
    ]
    set ColonyDied true
```

WorkerLarvaeDevProc

Replacing GUI variables by substances-own variables:

```
if ETOX_PPPOralDose > 1E-20
[
  set ldx 1 - (1 / (1 + (ETOX_PPPOralDose / [etox_Larvae_Oral_LD50_Substance] of substance SubstanceID)
              ^ [etox_Larvae_Oral_slope_Substance] of substance SubstanceID)) ; Dose-response relationship for larvae mortality
                                                                              ; from exposure by honey
]
```

DroneLarvaeDevProc

Replacing GUI variables by substances-own variables like in *WorkerLarvaeDevProc*.

WorkerIHbeesDevProc

Replacing GUI variables by substances-own variables like in *WorkerLarvaeDevProc*.

DronesDevProc

Replacing GUI variables by substances-own variables like in *WorkerLarvaeDevProc*.

CreateFlowerPatchesProc

This procedure creates *flowerPatches* defined via the user interface. The new *flowerPatches-own* variables *ETOX_NextAppDay_patch*, *ETOX_ExposurePeriod_patch* and *ETOX_RUD_patch* have to be set, using information from the GUI:

```
create-flowerPatches N_FLOWERPATCHES
  [
    ...

    set ETOX_NextAppDay_patch ETOX_AppDay ;  ETOX UPDATE 2023
    set ETOX_ExposurePeriod_patch ETOX_ExposurePeriod ;  ETOX UPDATE 2023
    set ETOX_RUD_patch ETOX_RUD  ;  ETOX UPDATE 2023
  ]
```

FlowerPatchesUpdateProc

The *INPUT_FILE* can now have two different formats, the original one and a new, shorter format. Here, the type of format is recognized by the number of columns to be able to correctly read in the data for handling times:

```
if ReadInfile = true
    [
      set TodaysSinglePatchList item who TodaysAllPatchesList
      let colShift 0 ;  ETOX UPDATE 2023 to be able to deal with both - old and new - file formats
      if length TodaysSinglePatchList = 15 [ set colShift 1 ]
      if length TodaysSinglePatchList != 15
         and length TodaysSinglePatchList != 24  ; length 15: old (long) file format, length 24: new ETOX format
      [
          show length TodaysSinglePatchList
          user-message "Change in INPUT_FILE: Update FlowerPatchesUpdateProc!"
          set BugAlarm true
      ]
      ifelse ConstantHandlingTime = true
        [ ; IF CONSTANT handling time:
          set handlingTimeNectar item 12 TodaysSinglePatchList
           ;  item 12: handling time nectar
          set handlingTimePollen item 13 TodaysSinglePatchList
        ] ;   item 13: handling time pollen
        [
          ; ELSE: if handling time is NOT constant:
          if quantityMyl > 0   ; nectar handling time
          [
            set handlingTimeNectar (
              item (12 + colShift) TodaysSinglePatchList) *                                ; Old: item 13, New: item 12: NectarGathering_s
              ((item (9 + colShift) TodaysSinglePatchList) * 1000 * 1000) / quantityMyl    ; New: item 10, item 9: quantityNectar_l
          ]

          if amountPollen_g > 0   ; pollen handling time
          [
            set handlingTimePollen
            (item (13 + colShift) TodaysSinglePatchList)                          ; Old: 14, New: item 13: PollenGathering_s;
            * ((item (7 + colShift) TodaysSinglePatchList) / amountPollen_g)      ; Old: 8, New: item 7: quantityPollen_g
          ]
        ]
    ] ; if ReadInfile = true
```

To allow the model to run without a pesticide application, the option "No applications" was added to *SubstanceApplied*. This results in a never ending spin-up phase:

```
to TExposure_at_patch_ETOX
  ;Within this procedure, the application of pesticides is conducted, as well as, the concentration of the pesticide
  ;  ETOX UPDATE 2023: heavily changed to apply data from INPUT_FILE, saved in flowerPatches!
  let memoSpinupPhase ETOX_SpinupPhase
  if SubstanceApplied = "No applications" [ set ETOX_SpinupPhase 9999999999999 ] ; if no sustance is applied,
                                                                                  ; spin-up phase lasts forever
```

...

```
    ] ; END if etox_year
  ] ; flowerPatches
  set ETOX_SpinupPhase memoSpinupPhase
end
```

In order to avoid error messages, "No applications" has to be defined as a *substance* in the *SUBSTANCE_FILE* (with all values either being set to 0 or to FALSE).

Create_Read-in_FlowerPatchesProc

Depending on the file format used, *TodaysSinglePatchList* and *TodaysAllPatchesList* have to be set correctly:

```
;;  ETOX UPDATE 2023:
let i 0
repeat N_FLOWERPATCHES
[
  ; todays data for ALL N_FLOWERPATCHES flower patches are saved in a
  ; new, shorter list (= todaysAllPatchesList)
  ifelse etoxFileFormat?  = false   ;  ETOX UPDATE 2023
   [
     ...
   ]
   [
     set TodaysSinglePatchList (item i AllDaysAllPatchesList) ;  ETOX UPDATE 2023
     set TodaysAllPatchesList AllDaysAllPatchesList ; for the short file format
   ]
```

The same is true for all other variables which are informed by *INPUT_FILE*. For each *flowerPatch* created, *Counter* is increased by 365 in the old file format or by 1 in the new file format. Also, there is a shift in the columns between old and new file format by 1 column (e.g. *distanceToColony* is in column 5 in the old format and in column 4 in the new format), which is taken into account via *j* (equals 0 or 1):

```
let j 0
ifelse etoxFileFormat?  = false
 [ set counter counter + 365 ]
 [
   set i i + 1
   set j 1
 ]
create-flowerPatches 1
[
  set oldPatchID item (2 - j) TodaysSinglePatchList
    ; refers to patch number of crop maps from a landscape module,
    ; an optional external tool to read in and analyse maps of food patches
  set patchType item (3 - j) TodaysSinglePatchList ; e.g. Oilseed rape
  set distanceToColony item (4 - j) TodaysSinglePatchList  ; [m]
  set xcorMap item (5 - j) TodaysSinglePatchList ; x coordinate
  set ycorMap item (6 - j) TodaysSinglePatchList ; y coordinate
  set size_sqm item (7 - j) TodaysSinglePatchList  ; patch area [m^2]
  set amountPollen_g item (8 - j) TodaysSinglePatchList ; [g]
  set nectarConcFlowerPatch item (9 - j) todaysSinglePatchList  ; [mol/l]
  set quantityMyl  (item (10 - j) TodaysSinglePatchList) * 1000 * 1000 ; [microlitres]
  let calcDetectProb item (11 - j) TodaysSinglePatchList
  ; calculated in "2_BEEHAVE_FoodFlow"-Tool on basis of distance
  ; (if this input file is created by "BEEHAVE_FoodFlow")
  let modelledDetectProb item (12 - j) TodaysSinglePatchList
    ; modelleded in "3_BEEHAVE_LANDSCAPE" with individual scouts
    ; exploring a 2-dim landscape
```

Additionally to the landscape information provided in the old *INPUT_FILE* format, the new file format also provides information about pesticide applications:

```
    if etoxFileFormat?
    [
      set floweringStartDay item 14 TodaysSinglePatchList ;  ETOX UPDATE 2023
      set floweringStopDay  item 15 TodaysSinglePatchList ;  ETOX UPDATE 2023
      if SubstanceApplied != "No applications"  ;  ETOX UPDATE 2023
      [
        set ETOX_ApplicationList_patch  item 16 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_ExposurePeriodsList_patch item 17 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_PPPConcNectar_patch item 18 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_PPPConcPollen_patch item 19 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_PPPContact_patch item 20 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_WaterVolume_patch item 21 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_WaterConc_patch item 22 TodaysSinglePatchList ;  ETOX UPDATE 2023
        set ETOX_RUD_patch item 23 TodaysSinglePatchList ;  ETOX UPDATE 2023
      ]
      set quantityMaxMyl quantityMyl
      set amountMaxPollen_g amountPollen_g
    ]
    set ETOX_Nth_Application 0
    set shape "flower"
    set size 1 + (sqrt size_sqm) /  1000
    setxy (distanceToColony / 300) 3

    if ETOX_ExposurePeriodsList_patch != 0  [ set ETOX_NextAppDay_patch 999 ] ; i.e. no application will take place
    if ETOX_ExposurePeriodsList_patch != [] and ETOX_ExposurePeriodsList_patch != 0 ; i.e. if at least 1 application of
                                ; at least 1 substance is defined (will be 0 if SubstanceApplied = "No applications")
    [
      set ETOX_ExposurePeriod_patch item 0 ETOX_ExposurePeriodsList_patch
      set ETOX_StopAppDay_patch item ETOX_Nth_Application ETOX_ApplicationList_patch + ETOX_ExposurePeriod_patch
      set ETOX_NextAppDay_patch item 0 ETOX_ApplicationList_patch ; i.e. the first number in a nested list
      if ETOX_NextAppDay_patch = ETOX_StopAppDay_patch [ set ETOX_NextAppDay_patch 0 ]
    ]
  ]  ; end create flowerPatches
]
```

Finally, reversion of *TodaysAllPatchesList* is only applied for the old file format:

```
  ]  ; END of "repeat N_FLOWERPATCHES"
  FlowerPatchesUpdateProc
  if etoxFileFormat?  = false [set TodaysAllPatchesList reverse TodaysAllPatchesList]
     ; to correct the reversed order, caused by the fput command (but not for new file format!)
end;
```

Foraging_searchingProc

A bug has been removed from the original BEEHAVE (2014) code, which resulted in too low numbers for the counters *NectarFlightsToday* and *PollenFlightsToday*, as successful recruits and scouts had not been taken into account. Now these variables are updated, whenever the *activity* is set to “bringingNectar” (respectively “bringingPollen” for pollen foragers).

For nectar counts:

```
ask foragerSquadrons with [ activity = "searching" ]
[
  ...
        ifelse ( pollenForager = false ... ) ; i.e. nectar foragers
        [
          set NectarFlightsToday NectarFlightsToday + SQUADRON_SIZE  ;  ETOX UPDATE 2023
          set activity "bringingNectar"  ; then the scout becomes a successful nectar forager
          ...
        ]
]
```

(and similarly for pollen scouts)

For nectar recruits:

```
ask foragerSquadrons with ; ask recruited NECTAR foragers:
   [ activity = "recForaging" ]; forager is recruited
     and knownNectarPatch >= 0 ; it knows a patch where it is recruited to
     and pollenForager = false ] ; and it is looking for nectar
[ ; the flights of recruited bees are counted:
  ...
  ifelse random-float 1 < FIND_DANCED_PATCH_PROB
  [
      ...
      set NectarFlightsToday NectarFlightsToday + SQUADRON_SIZE  ;  ETOX UPDATE 2023
      set activity "bringingNectar"
      ...
  ]
  ...
]
```

(similarly for pollen recruits)

Foraging_collectNectarPollenProc

Another bug has been removed here from the code section on exposure accrual during pollen collection: Now:
`[ETOX_PPPContactDose] of flowerPatch knownPollenPatch`
instead of
`[ETOX_PPPContactDose] of flowerPatch knownNectarPatch`

```
ask foragerSquadrons with [ activity = "bringingPollen" ]
  [
    ...
    [
       ifelse ETOX_ContactSum ; ;  ETOX UPDATE 2023 Bug removed: knownPollenPatch!
       [set ETOX_PPPContactDose (ETOX_PPPContactDose + [ETOX_PPPContactDose] of flowerPatch knownPollenPatch)] ;  ETOX UPDATE 2023
       [set ETOX_PPPContactDose (ETOX_PPPContactDose + [ETOX_PPPContactDose] of flowerPatch knownPollenPatch) / 2] ;  ETOX UPDATE 2023
    ]
    ...
  ]
```

Foraging_mortalityProc

Replacing GUI variables by substances-own variables:

```
; Acute toxicity of Foragers during flight.
 if [etox_Forager_ImmediateMortality_Substance] of substance SubstanceID = true ;  ETOX UPDATE 2023
 [;
   let ldx 0
   ask foragerSquadrons
   [
     if ETOX_PPPOralDose < 0 [ set ETOX_PPPOralDose 1]
     set Ldx 1 - (1 / (1 + (ETOX_PPPOralDose / [etox_Forager_Oral_LD50_Substance] of substance SubstanceID) ;  ETOX UPDATE 2023
                                             ^ [etox_Forager_Oral_slope_Substance] of substance SubstanceID))
     if ldx > ETOX_rdm_survival_oral
     [
       set ETOX_foragers_died ETOX_foragers_died + squadron_size
       die
     ]
     set ldx 1 - (1 / (1 + (ETOX_PPPContactDose / [etox_Forager_contact_LD50_Substance] of substance SubstanceID) ;  ETOX UPDATE 2023
                                             ^ [etox_Forager_contact_slope_Substance] of substance SubstanceID ))
     if ldx > ETOX_rdm_survival_contact
     [
       set ETOX_foragers_died ETOX_foragers_died + squadron_size
       die
     ]
   ]
 ]
```

ForagersLifespanProc

Replacing GUI variables by substances-own variables:

```
if ETOX_PPPContactDose > 0
[
  set ldx 1 - (1 / (1 + (ETOX_PPPContactDose / [etox_Forager_contact_LD50_Substance] of substance SubstanceID)
                                             ^ [etox_Forager_contact_slope_Substance] of substance SubstanceID))
```

ReadFileProc

The BEEHAVE$_{ecotox}$ version of this procedure was replaced by the original BEEHAVE *ReadFileProc*. If the new *INPUT_FILE* format is used, a larger number of columns has to be read in:

```
to ReadFileProc
 ; reads data in from file, copied from: Netlogo: Library:
  ; Code Examples: "File Input Example"

  ifelse ( file-exists? INPUT_FILE )
    ; We check to make sure the file exists first
    [
      set AllDaysAllPatchesList []
        ; IF: data are saved in a list (list still empty)

      file-open INPUT_FILE
      let dustbin file-read-line        ; first line of input file with headings is read - but not used for anything
      set etoxFileFormat? false  ;  ETOX UPDATE 2023
      if member? "startDay" dustbin [ set etoxFileFormat? true ]
      ifelse etoxFileFormat? = true  ; etoxFileFormat: only 1 line per patch with 24 (instead of 16) columns
      [
        while [ not file-at-end? ] ; NEW version (ETOX (short) input file format)
          [
            set AllDaysAllPatchesList sentence AllDaysAllPatchesList
               (list (list file-read file-read file-read file-read file-read
                           file-read file-read file-read file-read file-read
                           file-read file-read file-read file-read file-read
                           file-read
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           file-read ;  ETOX UPDATE 2023: additional column(s)
                           ))
           set N_FLOWERPATCHES ((length AllDaysAllPatchesList))
          ]
      ]
```

CreateSubstancesProc

This is a new procedure for the updated BEEHAVE$_{ecotox}$ module. First, a safety check is done to make sure that "Defined via GUI" is actually an option (and has not been renamed by the user). If the *substance* is defined via the GUI, the necessary data are gathered from the interface and saved in *dataLinelist*. Then *CreateSubstancesSetParametersProc* is called using the data in *dataLinelist*, where the substance is actually created.
If the substance is defined via an input file, *ReadSubstanceFileProc_ETOX* is called to first read in the *SUBSTANCE_FILE* and then call *CreateSubstancesSetParametersProc*. Finally, *SubstanceID* saves the ID (*who*) of the *substance* in use:

```
to  CreateSubstancesProc    ;  ETOX UPDATE 2023 NEW PROCEDURE
 let ypos -4 ; position of substance sign on GUI

 let testSubstanceApplied SubstanceApplied ; to make sure "Defined via GUI" is actually an option..
 set SubstanceApplied "Defined via GUI"    ; (if not, an error message will pop up now..)
 set SubstanceApplied testSubstanceApplied ; original setting is restored

 ifelse SubstanceApplied = "Defined via GUI"
 [                                ;  IF true: a single substance is created from parameters defined on GUI:
   let dataLinelist (list SubstanceApplied etox_DT50 etox_Forager_Oral_LD50 etox_Forager_Oral_slope etox_Forager_contact_LD50
                          etox_Forager_contact_slope etox_Larvae_Oral_LD50 etox_Larvae_Oral_slope etox_Forager_ImmediateMortality
                          etox_degradation_honey etox_DT50_honey etox_FF_Nursebees_Nectar etox_FF_Nursebees_Pollen)
   CreateSubstancesSetParametersProc dataLineList ypos
 ]
 [
   ReadSubstanceFileProc_ETOX ypos  ;  ELSE, one or more substances are created from data provided in SUBSTANCE_INPUT file
   if count substances with [ substanceName = SubstanceApplied ] = 0
     [ user-message (word "Selected substance (chooser 'SubstanceApplied': '" SubstanceApplied
                          "') is not defined in SUBSTANCE_FILE ('" SUBSTANCE_FILE "')")  ]
 ] ; end ELSE
 if count substances != 1 [print (word "Wrong number of substances! N substances: " count substances) set BugAlarm TRUE stop]
 set SubstanceID [who] of one-of substances  ; ...so we can set SubstanceID
end
```

CreateSubstancesSetParametersProc

This is a new procedure for the updated BEEHAVE$_{ecotox}$ module, where the substance is created, based on information from *myDataLineList*. Depending on how *myDataLineList* was created, it is either a list (when created from GUI) or a nested list (when created from *SUBSTANCE_FILE*). If the latter is the case, the first (and only) item in this list contains the required data. For each line in *SUBSTANCE_FILE* a new *substance* is created. However, as only one *substance* can be applied at the moment, they are immediately removed again, except for the single *substance* where *substanceName* equals *SubstanceApplied*:

```
to CreateSubstancesSetParametersProc [ myDataLineList ypos ] ;  ETOX UPDATE 2023 NEW PROCEDURE ; ypos: position of substance sign on GUI
 create-substances 1
 [
   set substanceName item 0 myDataLineList ; if SubstanceApplied is NOT = "Defined via GUI" then myDataLineList is a nested list
                                           ; (e.g. [["Substance1"] [100] [1000] [100] [0.6] [1.08] [0.0014] [1.6]])
     if is-list? substanceName [ set substanceName item 0 substanceName ]
   ; if SubstanceApplied is NOT set to the current substance, the substance is removed:
   if SubstanceApplied != substanceName [ die ] ; only one substance allowed
   set etox_DT50_Substance item 1 myDataLineList
     if is-list? etox_DT50_Substance [ set etox_DT50_Substance item 0 etox_DT50_Substance ]
   set etox_Forager_Oral_LD50_Substance item 2 myDataLineList
     if is-list? etox_Forager_Oral_LD50_Substance [ set etox_Forager_Oral_LD50_Substance item 0 etox_Forager_Oral_LD50_Substance ]
   set etox_Forager_Oral_slope_Substance item 3 myDataLineList
     if is-list? etox_Forager_Oral_slope_Substance [ set etox_Forager_Oral_slope_Substance item 0 etox_Forager_Oral_slope_Substance ]
   set etox_Forager_contact_LD50_Substance item 4 myDataLineList
     if is-list? etox_Forager_contact_LD50_Substance [ set etox_Forager_contact_LD50_Substance item 0 etox_Forager_contact_LD50_Substance ]
   set etox_Forager_contact_slope_Substance item 5 myDataLineList
     if is-list? etox_Forager_contact_slope_Substance [ set etox_Forager_contact_slope_Substance item 0 etox_Forager_contact_slope_Substance ]
   set etox_Larvae_Oral_LD50_Substance item 6 myDataLineList
     if is-list? etox_Larvae_Oral_LD50_Substance [ set etox_Larvae_Oral_LD50_Substance item 0 etox_Larvae_Oral_LD50_Substance ]
   set etox_Larvae_Oral_slope_Substance item 7 myDataLineList
     if is-list? etox_Larvae_Oral_slope_Substance [ set etox_Larvae_Oral_slope_Substance item 0 etox_Larvae_Oral_slope_Substance ]
   set etox_Forager_ImmediateMortality_Substance item 8 myDataLineList
     if is-list? etox_Forager_ImmediateMortality_Substance [ set etox_Forager_ImmediateMortality_Substance item 0 etox_Forager_ImmediateMortality_Substance]
   set etox_degradation_honey_Substance item 9 myDataLineList
     if is-list? etox_degradation_honey_Substance [ set etox_degradation_honey_Substance item 0 etox_degradation_honey_Substance ]
   set etox_DT50_honey_Substance item 10 myDataLineList
     if is-list? etox_DT50_honey_Substance [ set etox_DT50_honey_Substance item 0 etox_DT50_honey_Substance ]
   set etox_FF_Nursebees_Nectar_Substance item 11 myDataLineList
     if is-list? etox_FF_Nursebees_Nectar_Substance [ set etox_FF_Nursebees_Nectar_Substance item 0 etox_FF_Nursebees_Nectar_Substance ]
   set etox_FF_Nursebees_Pollen_Substance item 12 myDataLineList
     if is-list? etox_FF_Nursebees_Pollen_Substance [ set etox_FF_Nursebees_Pollen_Substance item 0 etox_FF_Nursebees_Pollen_Substance ]
   set label substanceName
   set label-color black
   set color white
   set shape "pesticidejar"
   set size 5
   setxy 40 ypos
 ]
end
```

ReadSubstanceFileProc_ETOX

This is a new procedure for the updated BEEHAVE$_{ecotox}$ module, which is called by *CreateSubstancesProc* and reads in the data for creating all *substances* defined in *SUBSTANCE_FILE*.

After opening *SUBSTANCE_FILE*, the *substance* data are read in line by line, calling *CreateSubstancesSetParametersProc* each time to create the specified *substance*:

```
to ReadSubstanceFileProc_ETOX [yposInit] ;  ETOX UPDATE 2023 NEW PROCEDURE
  ifelse ( file-exists? SUBSTANCE_FILE )
  [
    file-open SUBSTANCE_FILE
    let header file-read-line
    if header != "substanceName  \tetox_DT50_Substance  \tetox_Forager_Oral_LD50_Substance  \tetox_Forager_Oral_slope_Substance..." |
      [ user-message "WARNING! Header of SUBSTANCE_FILE has been changed!" ]

    ; now read substance data:
    while [not file-at-end?]
    [
      let nextLineList []
      repeat 13
      [
        set nextLineList lput (list file-read) nextLineList ; 13 columns
      ]
      let ypos yposInit - count Substances * 6
      if ypos < -50 [ set ypos -50 ]
      CreateSubstancesSetParametersProc nextLineList ypos
    ]
    file-close  ; closes file
  ] ; end "ifelse"
  [
    user-message "There is no such SUBSTANCE_FILE in current directory!"
  ]
end
```

TExposure_at_patch_ETOX

This procedure had to be modified to be able to use data from the GUI or from the *INPUT_FILE*. If flowerPatches are defined via the GUI, only two patches can be created, a “GreenField” and a “RedField”. Their variables are explicitely defined on the interface.

```
to TExposure_at_patch_ETOX
  ;Within this procedure, the application of pesticides is conducted, as well as, the concentration of the pesticide in nectar and pollen
  ; in the patch are calculated over time
  ;  ETOX UPDATE 2023: heavily changed to apply data from INPUT_FILE, saved in flowerPatches!

  ask flowerPatches ;runs one time for each flowerPatch in pseudo-random order
  [
    ; ETOX UPDATE 2023:
    ifelse ReadInfile = true ; ETOX UPDATE 2023:
    [ set ETOX_WaterVolume_Fullday ETOX_WaterVolume_patch ]; [mL]
    [
      if patchType = "GreenField" [set  ETOX_WaterVolume_Fullday ETOX_WaterVolume_Green] ;[mL]
      if patchType = "RedField"   [set  ETOX_WaterVolume_Fullday ETOX_WaterVolume_Red] ; [mL]
    ]
    let etox_year floor ticks / 365 ;calculates the years already simulated
    let etox_PPPNectarConc_kg 0
```

Breakdown of pesticide after the last application has now been enabled by adding the new condition:

...OR ETOX_P_PPPNectarconc + ETOX_P_PPPPollenConc + ETOX_PPPContactDose > 0:

```
    if (etox_year > ETOX_SpinupPhase and etox_year < ETOX_ExposurePhase + ETOX_SpinupPhase)
       OR ETOX_P_PPPNectarconc + ETOX_P_PPPPollenConc + ETOX_PPPContactDose > 0  ;  ETOX UPDATE 2023 otherwise, PPP contamination remains unchanged
                                                                                 ; after end of exposure period
    [; no exposure during the spinup phase and after the exposure phase to allow a stable colony to develop in the spinup phase and a recovery phase
     ; after exposure
       if day = ETOX_NextAppDay_patch and (etox_year > ETOX_SpinupPhase) and (etox_year < ETOX_ExposurePhase + ETOX_SpinupPhase) ; ETOX UPDATE 2023
       [; Set the pesticide concentration in the nectar of the patches as given in the inputs in [µg ai/kg nectar]
          ifelse ReadInfile = true ; ETOX UPDATE 2023:
          [
            set etox_PPPNectarConc_kg ETOX_PPPConcNectar_patch
            set ETOX_P_PPPPollenConc ETOX_P_PPPPollenConc + ETOX_PPPConcPollen_patch / 1000  ;convert into [µg/g] = [µg/kg] / 1000 ;  ETOX UPDATE 2023
                                                                                             ; WITH summing up PPP concentration
                                                ; Calculates the Dose per bee [µg/bee] from the Application rate [kg/ha] using the RUD [(ha*mg)/(kg*kg)]
                                                ; with the assumption that an adult bee is 0.1 g
            set ETOX_PPPContactDose ETOX_PPPContactDose + ETOX_PPPContact_patch * ETOX_RUD_patch * 0.1  ;[µg] =  [kg/ha] * [(ha*mg)/(kg*kg)] * [g]
                                                                                                        ; ETOX UPDATE 2023: summing up
            if ETOX_Water_foraging = true [ set  ETOX_WaterConc_Fullday (ETOX_WaterConc_patch / 1000)]  ;[µg/mL] = [µg/L] / 1000
          ]
```

Summing up of pesticide concentration when applied over multiple years can now happen as well:

```
          [   ; ELSE: use data from GUI
            ; Set the pesticide concentration in the nectar of the patches as given in the inputs in [µg ai/kg nectar]
            ;  ETOX UPDATE 2023: similar to original ETOX version, BUT: PPP are summed up (if application lasts for at least 1 year)
            if patchType = "GreenField" [set etox_PPPNectarConc_kg ETOX_PPPConcNectar_Green ]
            if patchType = "RedField"   [set etox_PPPNectarConc_kg ETOX_PPPConcNectar_Red ]

            if patchType = "GreenField" [set ETOX_P_PPPPollenConc ETOX_P_PPPPollenConc + ETOX_PPPConcPollen_Green / 1000]  ;convert into [µg/g] = [µg/kg]
                                                                                                                          ; / 1000
            if patchType = "RedField"   [set ETOX_P_PPPPollenConc ETOX_P_PPPPollenConc + ETOX_PPPConcPollen_Red / 1000]    ;convert into [µg/g] = [µg/kg]
                                                                                                                          ; / 1000

            ;Calculates the Dose per bee [µg/bee] from the Application rate [kg/ha] using the RUD [(ha*mg)/(kg*kg)] with the assumption that an adult bee
                                                                                                                                    ; is 0.1 g
            if patchType = "GreenField" [set ETOX_PPPContactDose ETOX_PPPContactDose + ETOX_PPPContact_Green * ETOX_RUD * 0.1 ] ;[µg] =  [kg/ha]
                                                                                                                                 ; * [(ha*mg)/(kg*kg)] * [g]
            if patchType = "RedField"   [set ETOX_PPPContactDose ETOX_PPPContactDose + ETOX_PPPContact_Red * ETOX_RUD * 0.1 ]   ;[µg] =  [kg/ha]
                                                                                                                                 ; * [(ha*mg)/(kg*kg)] * [g]

            if ETOX_Water_foraging = true
            [
              if patchType = "GreenField" [set ETOX_WaterConc_Fullday ETOX_WaterConc_Fullday + (ETOX_WaterConc_Green / 1000) ] ;[µg/mL] = [µg/L] / 1000
              if patchType = "RedField"   [set ETOX_WaterConc_Fullday ETOX_WaterConc_Fullday + (ETOX_WaterConc_Red / 1000)]    ;[µg/mL] = [µg/L] / 1000
            ]
          ]
          let ETOX_Densityfactor 0.1047 ; it is a coefficient in the linear regression formula y = -x * 0.1047 + 1. y has units [L/kg]
                                        ; and x (nectarConcFlowerPatch) [mol/L]
          ;recalculates the concentration in the patch from µg ai/kg into µgAI/L
          let ETOX_PPPNectarconc_L etox_PPPNectarConc_kg / (1 - ETOX_Densityfactor * nectarConcFlowerPatch) ; [µg ai/L]  = [(µg ai/kg)/ (L/kg)]
                                                                                                           ; as y [L/kg] = 1 - ETOX_Densityfactor * nectarConcFlowerPatch
          ;recalculates the concentration in the patch from µg ai/L into µgAI/mol
          let ETOX_PPPNectarconc_mol ETOX_PPPNectarconc_L / nectarConcFlowerPatch  ;[µg/mol] = [µg/l * l/mol]
          ;recalculates the concentration in the patch from µg ai/mol into µgAI/kJ
          set ETOX_P_PPPNectarconc ETOX_P_PPPNectarconc + ETOX_PPPNectarconc_mol / (ENERGY_SUCROSE * 1000 * 1000)   ; [µg/kJ] = [µg/mol * mol/kJ]
                                                                                  ; [kJ/ mol] = [kJ/micromol * 1000 * 1000]  ;  ETOX UPDATE 2023: Summing up added
      ]
```

If the exposure period is over, the concentration will be set to 0 – otherwise it further disintegrates:

```
      ; if exposure period is over:
      ifelse
           (ReadInfile = TRUE and ETOX_ApplicationList_patch != [] and day = ETOX_StopAppDay_patch and ETOX_StopAppDay_patch != ETOX_NextAppDay_patch)
                                          ;  ETOX UPDATE 2023 if applications are defined, they end today and a new application doesn' start today
           OR (ReadInfile = FALSE and day >= ETOX_AppDay + ETOX_Exposureperiod) ;  ETOX UPDATE 2023 if GUI data is used (only 1 application)
           OR (ReadInfile = FALSE and ticks = (ETOX_SpinupPhase * 365 + (ETOX_ExposurePhase - 1) * 365 + ETOX_AppDay + ETOX_ExposurePeriod))
                                                                ; spinup years + exposure - 1 full years + exposure time in last year
      [
        set ETOX_P_PPPNectarconc 0
        set ETOX_P_PPPPollenConc 0 ;  [µg/g]
        set ETOX_PPPContactDose 0
        if ETOX_Water_foraging = true
           [ set ETOX_WaterConc_Fullday 0 ] ; [µg/mL]
      ]
      [  ; ELSE:
        set ETOX_P_PPPNectarconc ETOX_P_PPPNectarconc * exp (- ln(2) / [ETOX_DT50_Substance] of substance SubstanceID ) ; ( ETOX UPDATE 2023)
                                 ; The disappearance of the pesticide in the nectar of the crop following a single first-order kinetic
        set ETOX_P_PPPPollenConc ETOX_P_PPPPollenConc * exp (- ln(2) / [ETOX_DT50_Substance] of substance SubstanceID ) ; ( ETOX UPDATE 2023)
        set ETOX_PPPContactDose ETOX_PPPContactDose * exp (- ln(2) / [ETOX_DT50_Substance] of substance SubstanceID ) ; ( ETOX UPDATE 2023)
        if ETOX_Water_foraging = true
           [ set ETOX_WaterConc_Fullday ETOX_WaterConc_Fullday * exp (- ln(2) / [ETOX_DT50_Substance] of substance SubstanceID )] ; ( ETOX UPDATE 2023)]
           ; The disappearance of the pesticide in the water source following a single first-order kinetic
      ]
    ] ; END if etox_year
  ] ; flowerPatches
end
```

Finally, the code for plotting information (*set-current-plot*...) at the end of the procedure has now been moved to *Tplot_exposureAndEffects_ETOX procedure*.

TupdateInternalExposureNectar_ETOX

Replacement of GUI variables by substance-own variables

Tplot_exposureandeffects_ETOX

Simplification of the code by using the new procedure *PlotSetupProc*, e.g.:

```
ask flowerPatches
[
  set-current-plot "ETOX_Plot_Water_Volume_at_Patch"
    PlotSetupProc
    plotxy ticks ETOX_WaterVolume_Fullday ; [mL]
  set-current-plot "ETOX_Plot_WaterConc_at_Patch"
    PlotSetupProc
    plotxy ticks  ETOX_WaterConc_Fullday * 1000  ; [µg/mL] -> [µg/L]
]
```

PlotSetupProc

This is a new procedure for the updated BEEHAVE$_{ecotox}$ module. It is called by *flowerPatches* in *Tplot_exposureandeffects_ETOX* and creates a temporary plot pen, choosing its colour dependent on the ID (*who*) of the calling *flowerPatch*:

```
to PlotSetupProc  ;  ETOX UPDATE 2023 determine plot pen and colour for a tidier code in Tplot_exposureandeffects_ETOX
 create-temporary-plot-pen (word patchType "_" who)
 ;set-plot-pen-color color
 let myColor (who * 30) + 5 + 2 * floor (who / 14)  ; to use full color scheme and avoid repetition
                                                    ; (there are 14 main colors in Netlogo)
 if myColor mod 10 > 7.5 or myColor mod 10 < 2.5 [ set myColor myColor + 3] ; to avoid too light or too dark colours
 if count flowerPatches = 2
 [
   let colorList [ "Red" "Green" ]
   foreach colorList [ if member? ? patchType  [ set myColor read-from-string ?] ]
 ]
 set-plot-pen-color myColor
 set-plot-pen-mode 0 ; 0: lines
end
```

DefaultProc_ETOX

This is a new procedure for the updated BEEHAVE$_{ecotox}$ module. It is called by *DefaultProc* and by the buttons “ETOX Default” and “ETOX Version Test”. It sets the parameters defined on the interface to their default value. In contrast to *DefaultProc*, it also sets the ecotox paramteres and switches the BEEHAVE$_{ecotox}$ module on, using the *substance* specification from the

interface (*SubstanceApplied* set to "Defined via GUI"). Some of the original parameters (e.g. *DISTANCE_R*, *Weather*) are set to different values than in *DefaultProc*:

```
to DefaultProc_ETOX

  ;; previous BEEHAVE parameters, CHANGED:
  set DISTANCE_R 500  ; was: 1500
  set MAX_km_PER_DAY 5099  ; was: 7299
  set SeasonalFoodFlow false  ; was: true
  set SHIFT_G 40 ; was: -40
  set SHIFT_R -30  ; was: 30

  ; NEW  ETOX UPDATE 2023:
  set SubstanceApplied "Defined via GUI"
  set SUBSTANCE_FILE "SubstanceFile.txt"
```

BEEHAVE$_{ecotox}$ parameters are set to their default values:

```
; new for ETOX (2022):
set Adult_mortality_mode "BEEHAVE default" ; "BEEHAVE + Stress (Simone-Finstrom 2016)"
set ETOX_AppDay 189  ; only relevant, if flowerPatches are not read in
set ETOX_contactexp_oneday false
set Etox_ContactSum false
set ETOX_degradation_honey false
set ETOX_DT50 1000
set ETOX_DT50_honey 60
set ETOX_ExposurePeriod 8 ; only relevant, if flowerPatches are not read in
set ETOX_ExposurePhase 3
set ETOX_FF_Nursebees_Nectar 0.25
set ETOX_FF_Nursebees_Pollen 1
set ETOX_Fixed_PPPnectar:pollen true
set ETOX_Forager_contact_LD50 0.6
set ETOX_Forager_contact_slope 1.08
set ETOX_Forager_ImmediateMortality false
set ETOX_Forager_Oral_LD50 1000
set ETOX_Forager_Oral_Slope 100
set ETOX_Larvae_Oral_LD50 0.0014
set ETOX_Larvae_Oral_slope 1.6
set ETOX_N-P-factor 26.9
set ETOX_PPPConcNectar_Green 990
set ETOX_PPPConcNectar_Red 990
set ETOX_PPPConcPollen_Green 26631
set ETOX_PPPConcPollen_Red 26631
set ETOX_PPPContact_Green 0.3
set ETOX_PPPContact_Red 0.3
set ETOX_RUD 21  ; only relevant, if flowerPatches are not read in
set ETOX_SpinupPhase 0
set ETOX_Water_foraging false
set ETOX_WaterConc_Green 0
set ETOX_WaterConc_Red 0
set ETOX_WaterVolume_Green 10000
set ETOX_WaterVolume_Red 10000
```

```
set  exp_added-fondant1 10
set  exp_added-fondant2 15
set  exp_added-fondant3 20
set  exp_added-fondant4 0
set  exp_added-fondant5 0
set  exp_added-fondant6 0
set  exp_added-fondant7 0
set  exp_added-fondant8 0
set  exp_added-fondant9 0
set  exp_Age-of-Queen 300
set  exp_feeding-day1 170
set  exp_feeding-day2 175
set  exp_feeding-day3 180
set  exp_feeding-day4 0
set  exp_feeding-day5 0
set  exp_feeding-day6 0
set  exp_feeding-day7 0
set  exp_feeding-day8 0
set  exp_feeding-day9 0
set  exp_feeding-schedule false
set  exp_harvest-day1 20
set  exp_harvest-day2 21
set  exp_harvest-day3 0
set  exp_harvest-day4 0
set  exp_Honey-harvest false
set  exp_HoneyNotNegative true
set  exp_init_eggs 500
set  exp_init_eggs_drone 9
set  exp_init_honey 10
set  exp_init_IHbees 10000
set  exp_init_larvae 3000
set  exp_init_larvae_drone 145
set  exp_init_pollen 2
set  exp_init_pupae 7000
set  exp_init_pupae_drone 50
set  exp_input_initial_conditions false
set  exp_Max-Age-of-queen 600
set  exp_Relocation-Day1 212
set  exp_Relocation-Day2 0
set  exp_remaining-honey1 10
set  exp_remaining-honey2 3
set  exp_remaining-honey3 0
set  exp_remaining-honey4 0
set  exp_StartDay 212
set  exp_StressDuration 14
set  exp_X-Days 365
set  exp-broodSwarmingTH 17000
set  exp-fractionSwarm 0.6
set  WaterNeed? false
```

The other parameters listed are identical to the ones in *DefaultProc* and set to the same values.

Code verification

In order to proof that the update of the BEEHAVE$_{ecotox}$ landscape module did only affect the setup and not the actual model runs, the original BEEHAVE$_{ecotox}$ version was compared with the updated version. A two patch scenario was used for comparison, as the original version is not capable of using more than two *flowerPatches*. For the updated version, the model was either parametrised by inserting the parameter values in the GUI interface or by reading them in from an *INPUT_FILE* (using the new file format).

In both cases, the same parameterisation was used, which is the initial one of the published BEEHAVE$_{ecotox}$ model:

```
set  ETOX_AppDay 189  ; only relevant, if flowerPatches are not read in
set  ETOX_contactexp_oneday false
set  Etox_ContactSum false
set  ETOX_degradation_honey false
set  ETOX_DT50 1000
set  ETOX_DT50_honey 60
set  ETOX_ExposurePeriod 8 ; only relevant, if flowerPatches are not read in
set  ETOX_ExposurePhase 3
set  ETOX_FF_Nursebees_Nectar 0.25
set  ETOX_FF_Nursebees_Pollen 1
set  ETOX_Fixed_PPPnectar:pollen true
set  ETOX_Forager_contact_LD50 0.6
set  ETOX_Forager_contact_slope 1.08
set  ETOX_Forager_ImmediateMortality false
set  ETOX_Forager_Oral_LD50 1000
set  ETOX_Forager_Oral_Slope 100
set  ETOX_Larvae_Oral_LD50 0.0014
set  ETOX_Larvae_Oral_slope 1.6
set  ETOX_N-P-factor 26.9
set  ETOX_PPPConcNectar_Green 990
set  ETOX_PPPConcNectar_Red 990
set  ETOX_PPPConcPollen_Green 26631
set  ETOX_PPPConcPollen_Red 26631
set  ETOX_PPPContact_Green 0.3
set  ETOX_PPPContact_Red 0.3
set  ETOX_RUD 21  ; only relevant, if flowerPatches are not read in
set  ETOX_SpinupPhase 0
set  ETOX_Water_foraging false
set  ETOX_WaterConc_Green 0
set  ETOX_WaterConc_Red 0
set  ETOX_WaterVolume_Green 10000
set  ETOX_WaterVolume_Red 10000
```

All scenarios were run with 100 replicates over one year. The graphs below plot the mean of various output variables ± standard deviation:

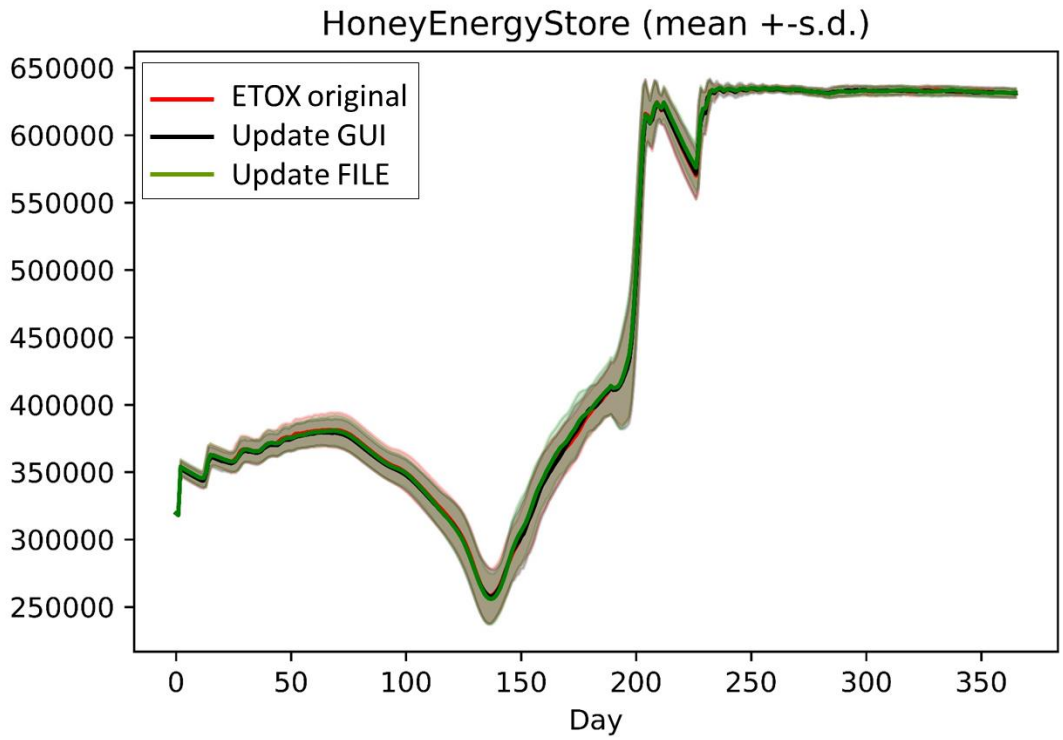

Figure 1a) Mean honey energy store [kJ] ± S.D. over one simulated year, comparing the original BEEHAVE$_{ecotox}$
("ETOX original") with the new BEEHAVE$_{ecotox}$ landscape module, when parameterisation was either taken from the interface ("Update GUI") or from an input file ("Update File").

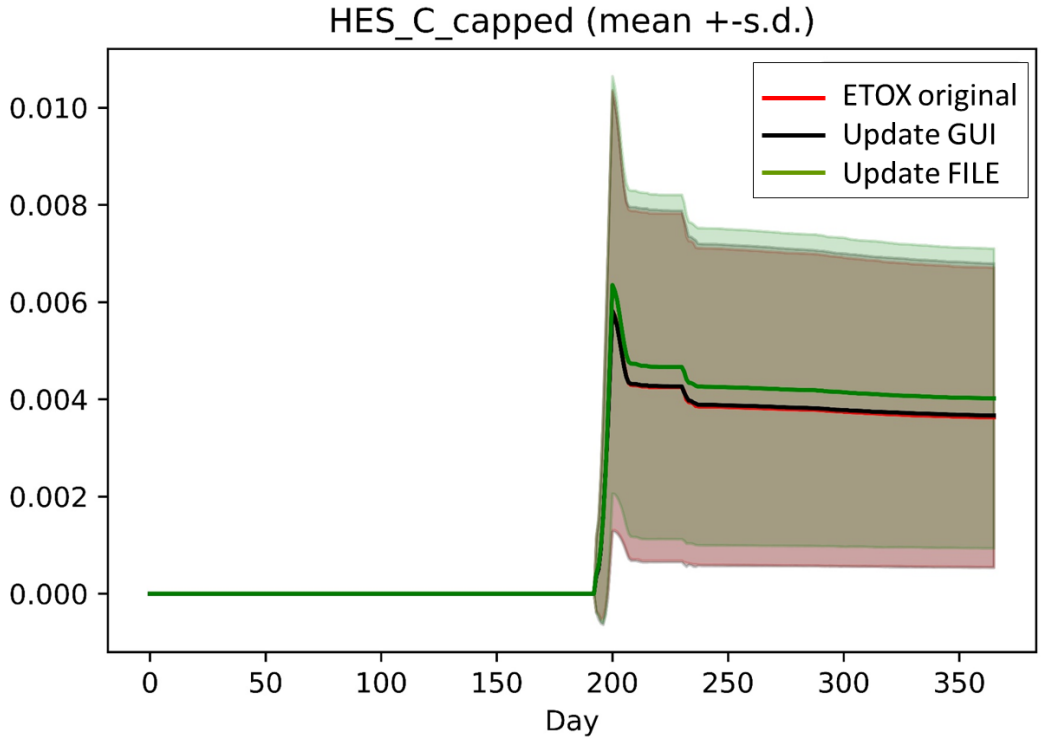

Figure 1b): Pesticide concentration in capped honey stores

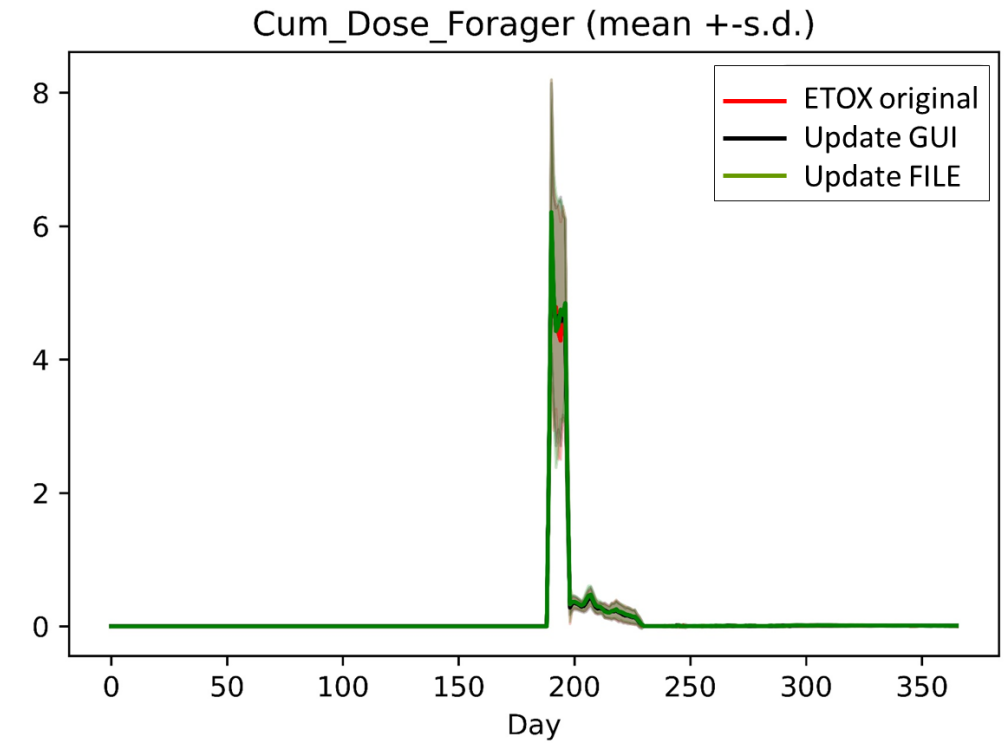

Figure 1c): Cumulative daily dose for all foragers

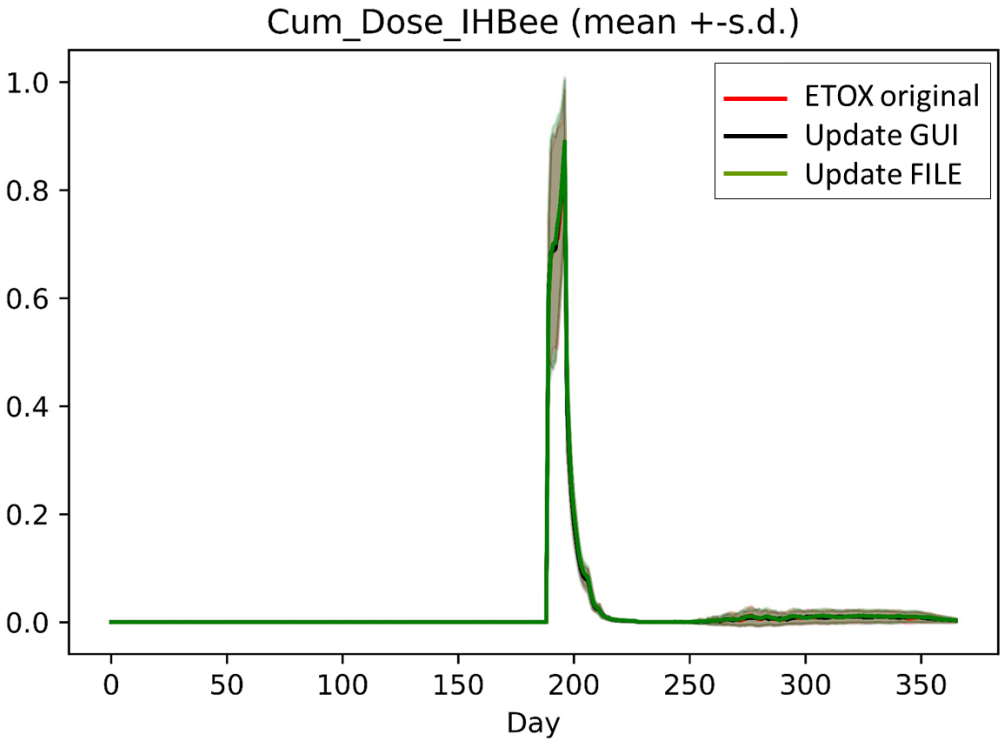

Figure 1d): Cumulative daily dose for all in-hive bees

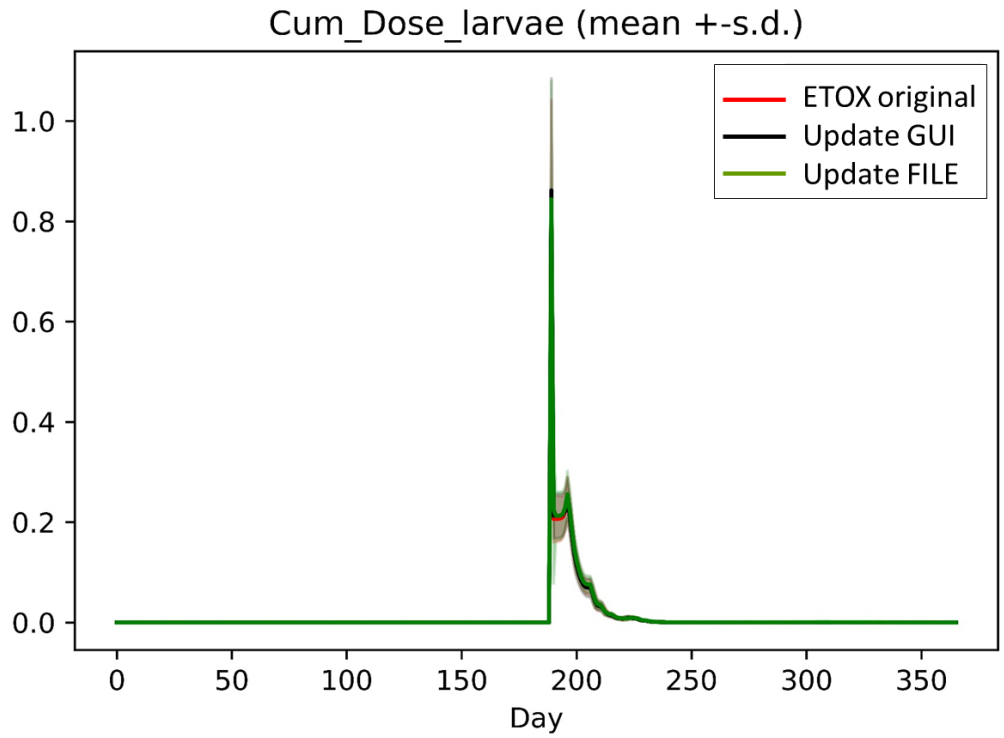

Figure 1e): Cumulative daily dose for all larvae

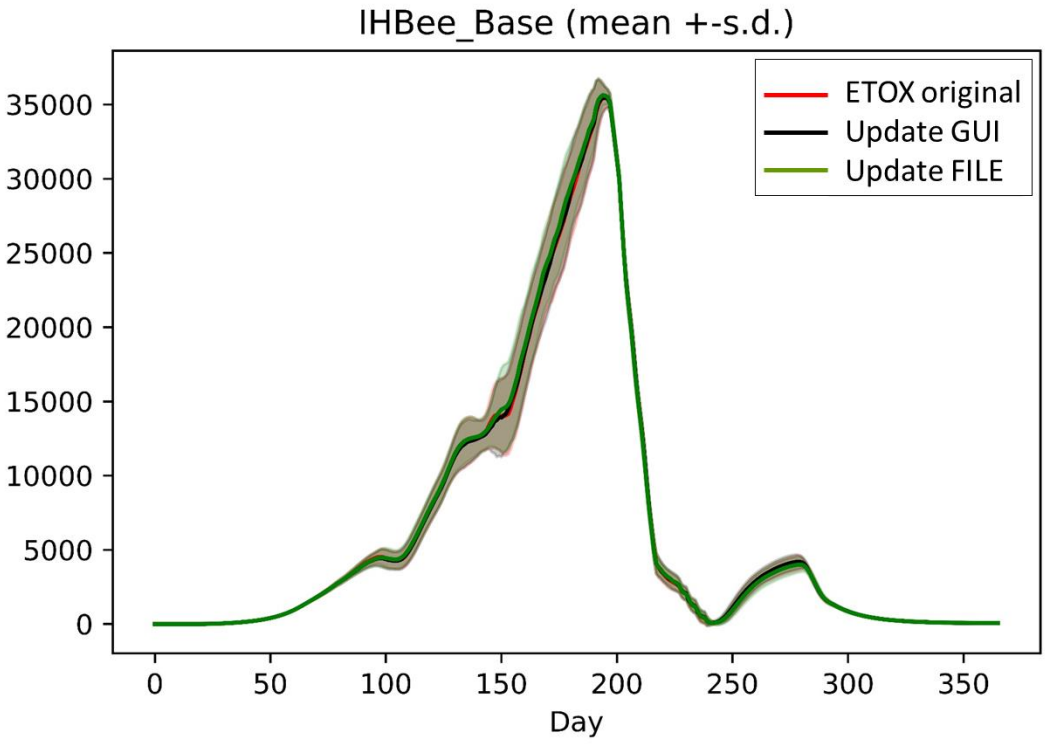

Figure 1f): The number of in-hive bees present (before mortality is applied)

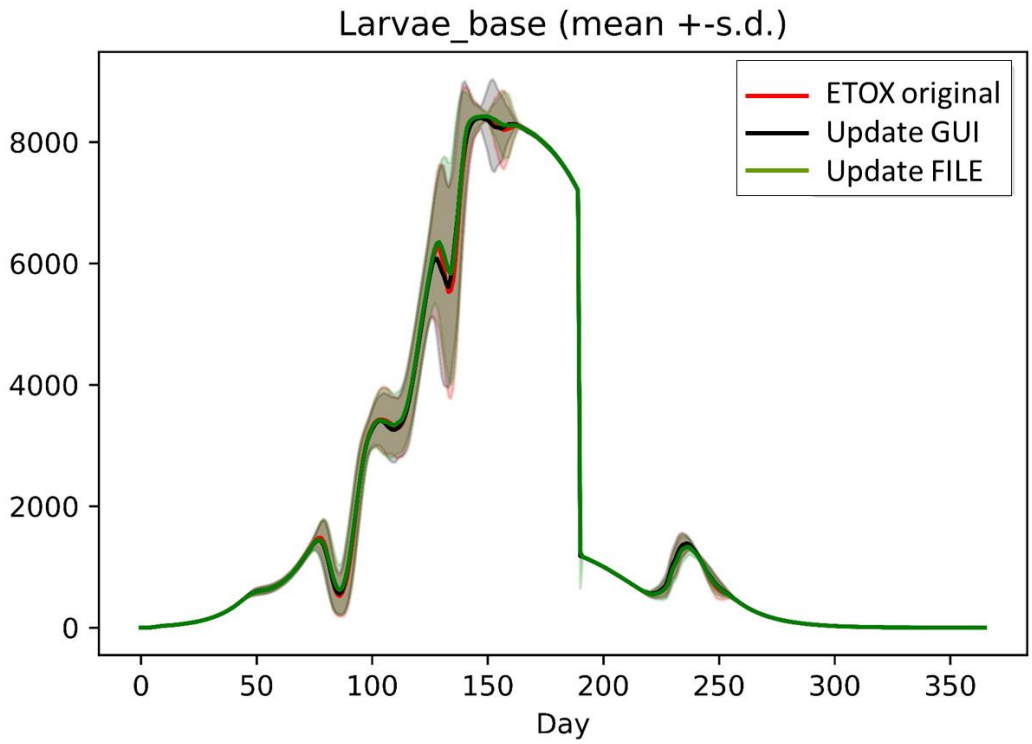

Figure 1g): The number of larvae present (before mortality is applied)

In all plots, the mean values of all three scenarios are very close together and well within the standard deviations, showing that no significant change has been made to the model performance. The variation between the runs is caused by stochastic effects due to differences in the pseudo-random numbers.

Example simulation: agricultural landscape

With the “Rothamsted landscape” provided with the original BEEHAVE publication (Becher et al. 2014) a test simulation with the new BEEHAVE$_{ecotox}$ landscape module was conducted.

The map (Fig. 2) consisted of a 10 x 7.5km landscape composed of three artificial habitat (crop) types (“red”, “blue”, “yellow”) with 115 fields altogether. The virtual colony (depicted beehive) was placed in the centre of the landscape.

Figure 2: Habitat map of an agricultural landscape in hertfordshire, UK.

Food resources for the artificial habitat types were set as follows:

patchType	N patches	Conc. nectar [M]	Nectar [ml/m2]	Pollen [mg/m2]	NectarGathering_s	PollenGathering_s	startDay	stopDay
BlueField	24	1,5	0,125	2	1200	600	30	302
YellowField	38	1,5	0,5	2,5	1200	600	121	179
RedField	53	2	0,125	10	1200	600	100	248

BEEHAVE$_{ecotox}$ setup for the treated habitat type was:

ETOX_ApplicationList_patch	ETOX_ExposurePeriodsList_patch	ETOX_PPPConcNectar_patch	ETOX_PPPConcPollen_patch	ETOX_PPPContact_patch
[121]	[58]	990	26631	0,3

ETOX_WaterVolume_patch	ETOX_WaterConc_patch	ETOX_RUD_patch
10000	0	21

Four scenarios, either treating the red, blue or yellow fields or no fields at all as control were run with 20 replicates. Shown are the means of various output variables ± standard deviation for each of the scenarios. Panel titles indicate the patch type (colour) which had been treated. The purpose of the simulation was to show that the new BEEHAVE$_{ecotox}$ landscape module can be applied to realistic landscapes. As the habitat types as well as the pesticide applications

were artificial, the results do not represent a realistic scenario but only show the model functionality.

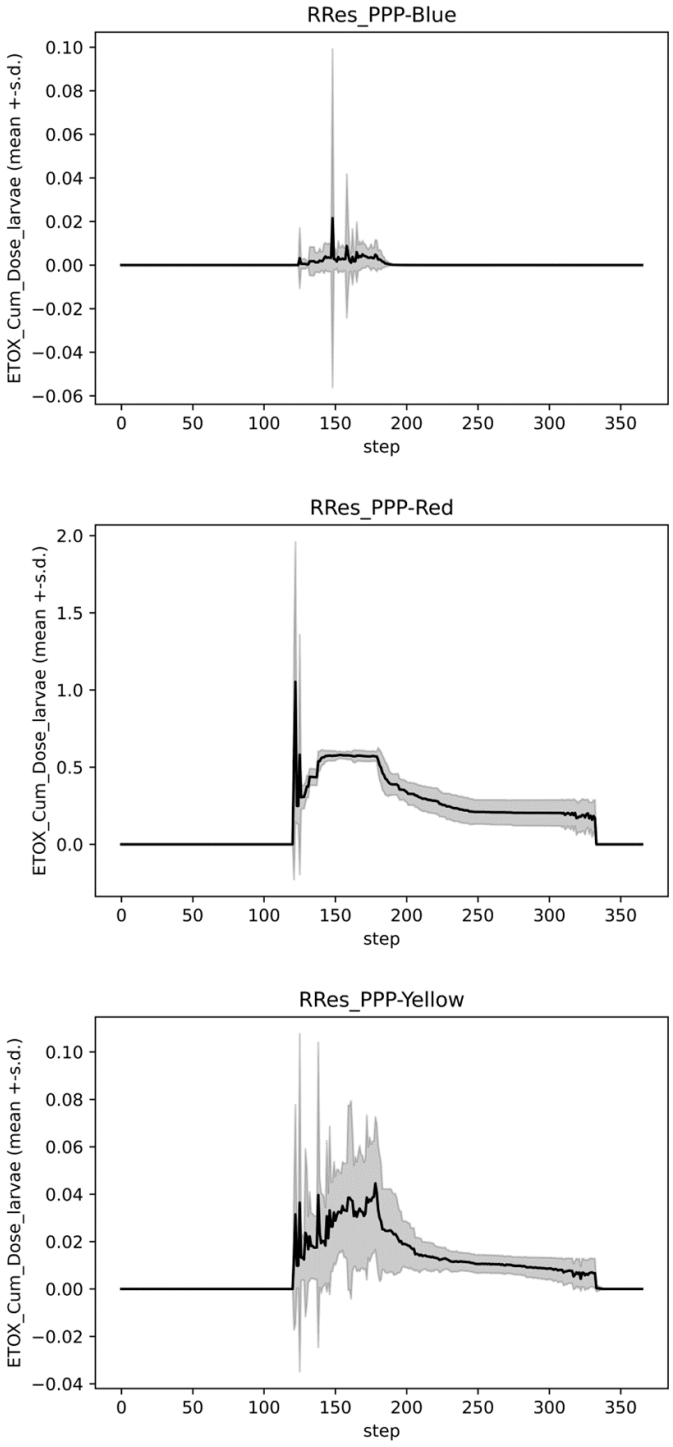

Figure 3a): Mean cumulative daily dose (±S.D.) for all larvae when either the yellow, the blue or the red fields have been treated

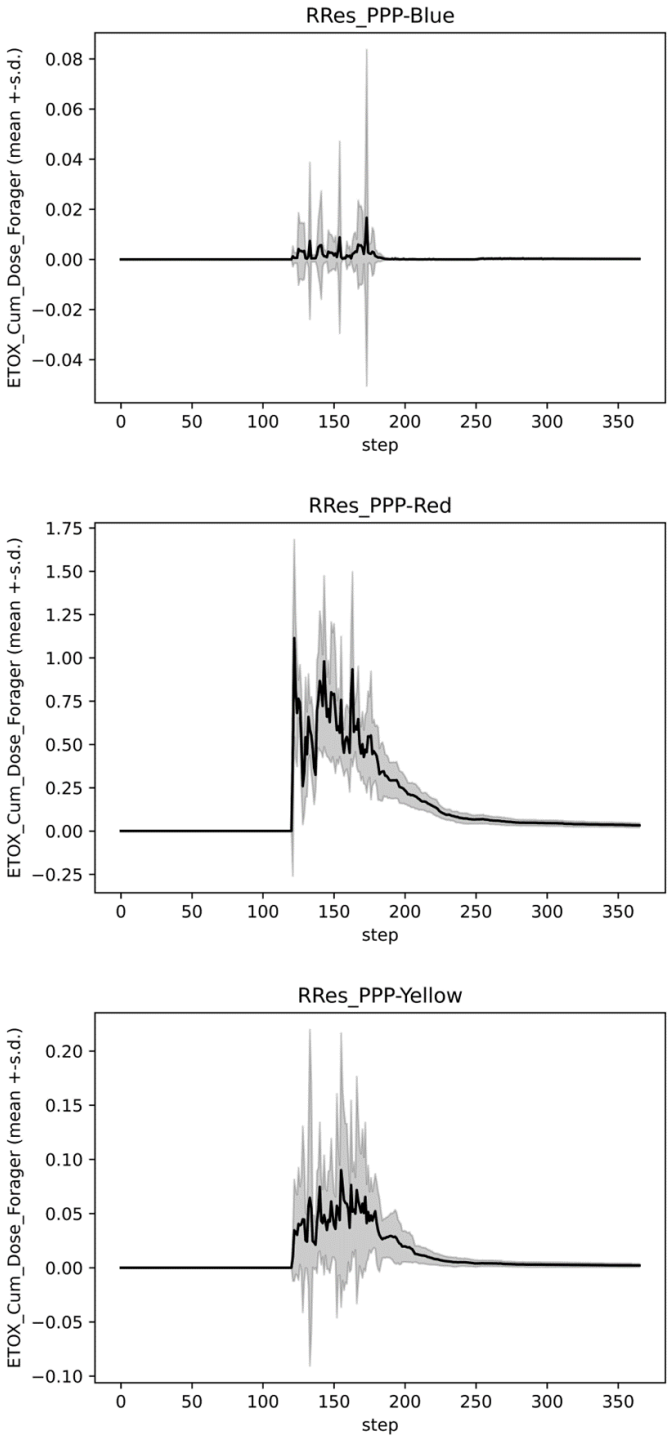

Figure 3b): Mean cumulative daily dose (±S.D.) for all foragers

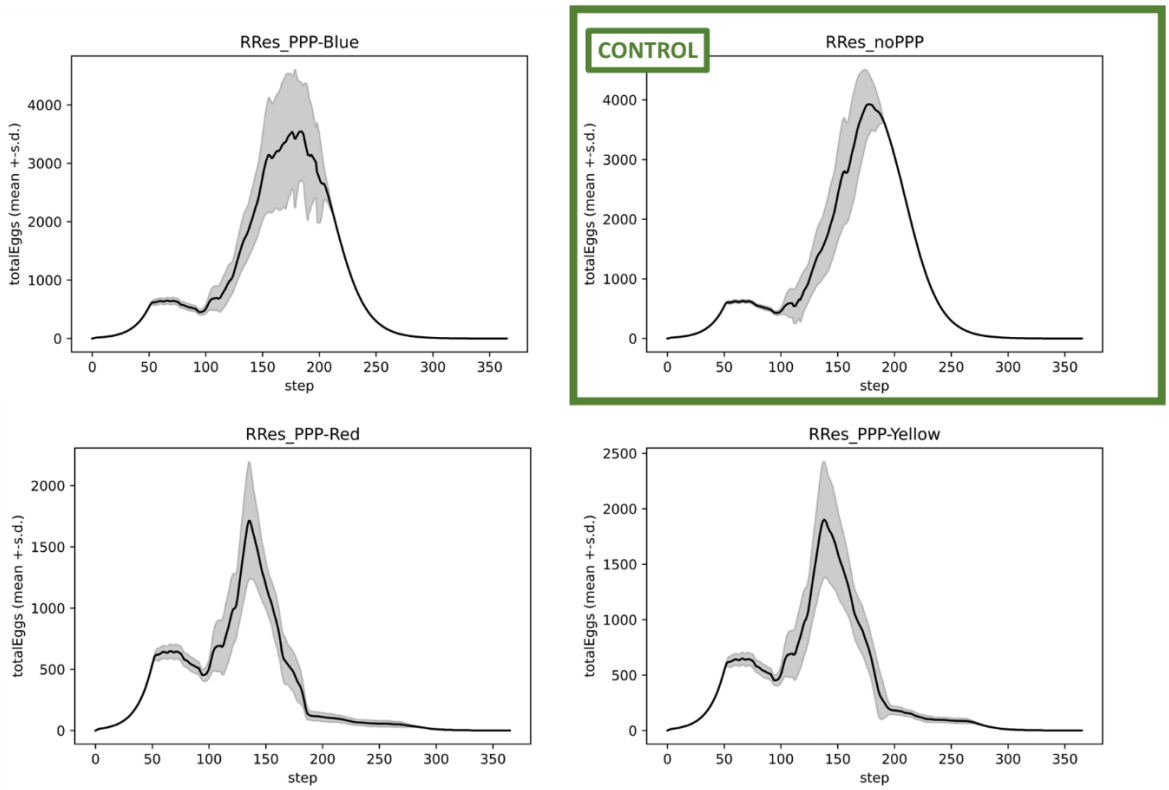

Figure 3c): Mean number of eggs (±S.D.) present in the colony

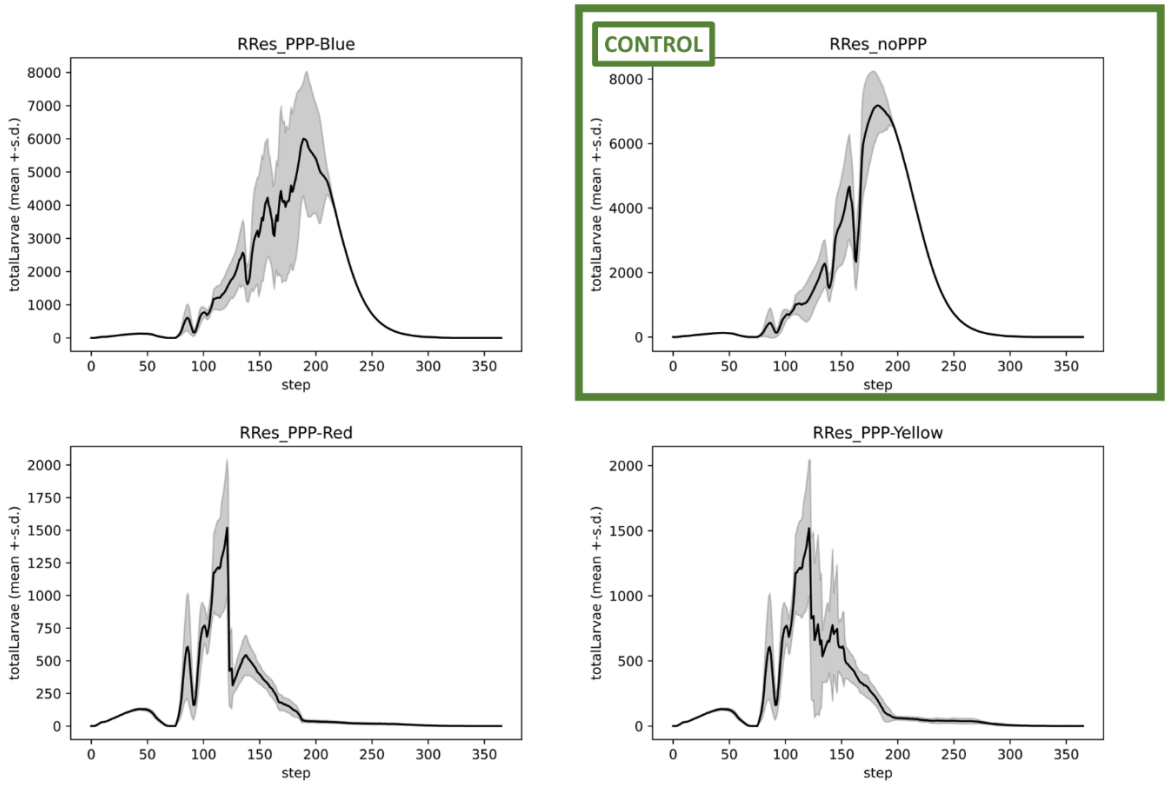

Figure 3d): Mean number of larvae (±S.D.) present in the colony

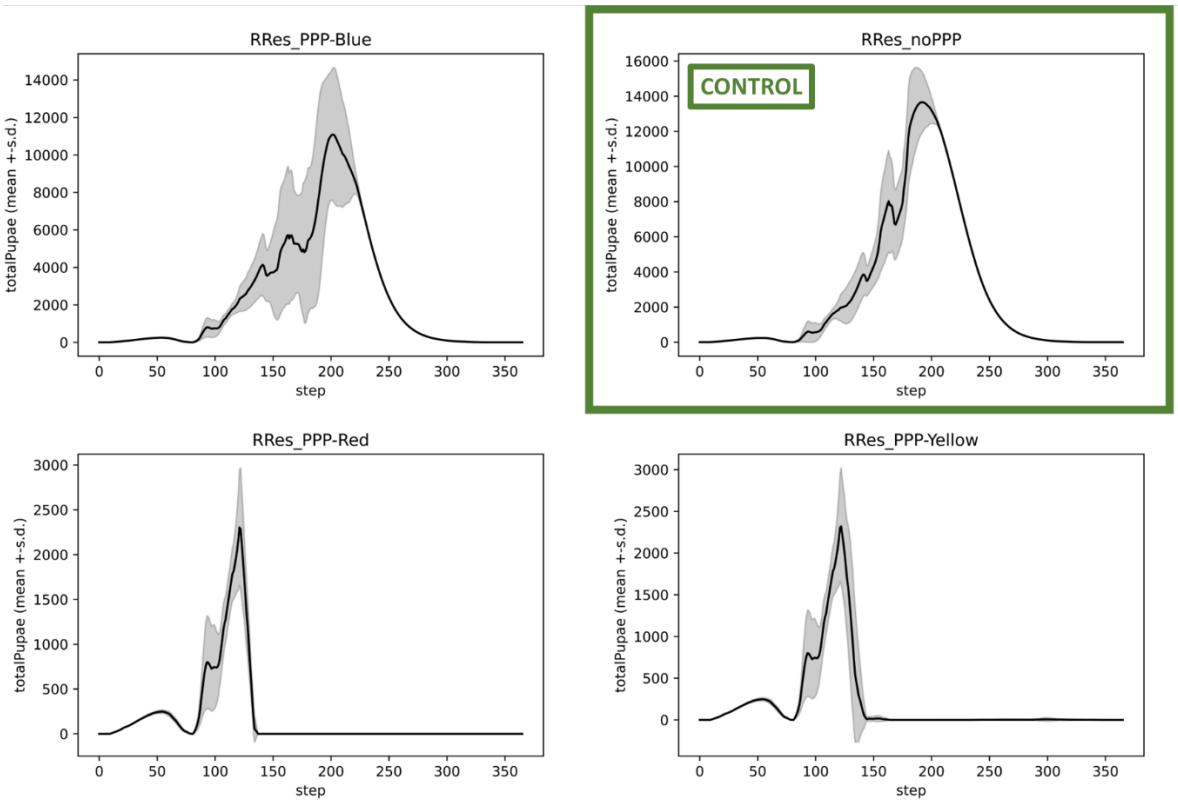

Figure 3e): Mean number of pupae (±S.D.) present in the colony

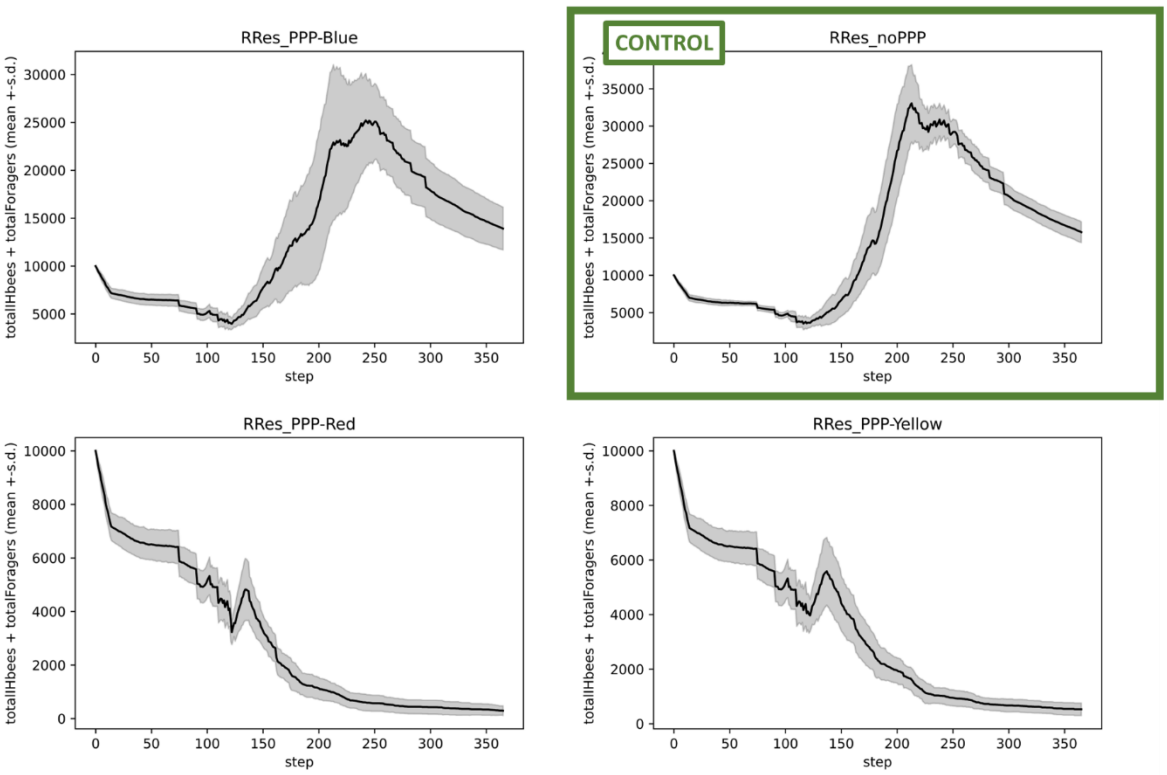

Figure 3f): Mean number of worker bees (±S.D.) present in the colony

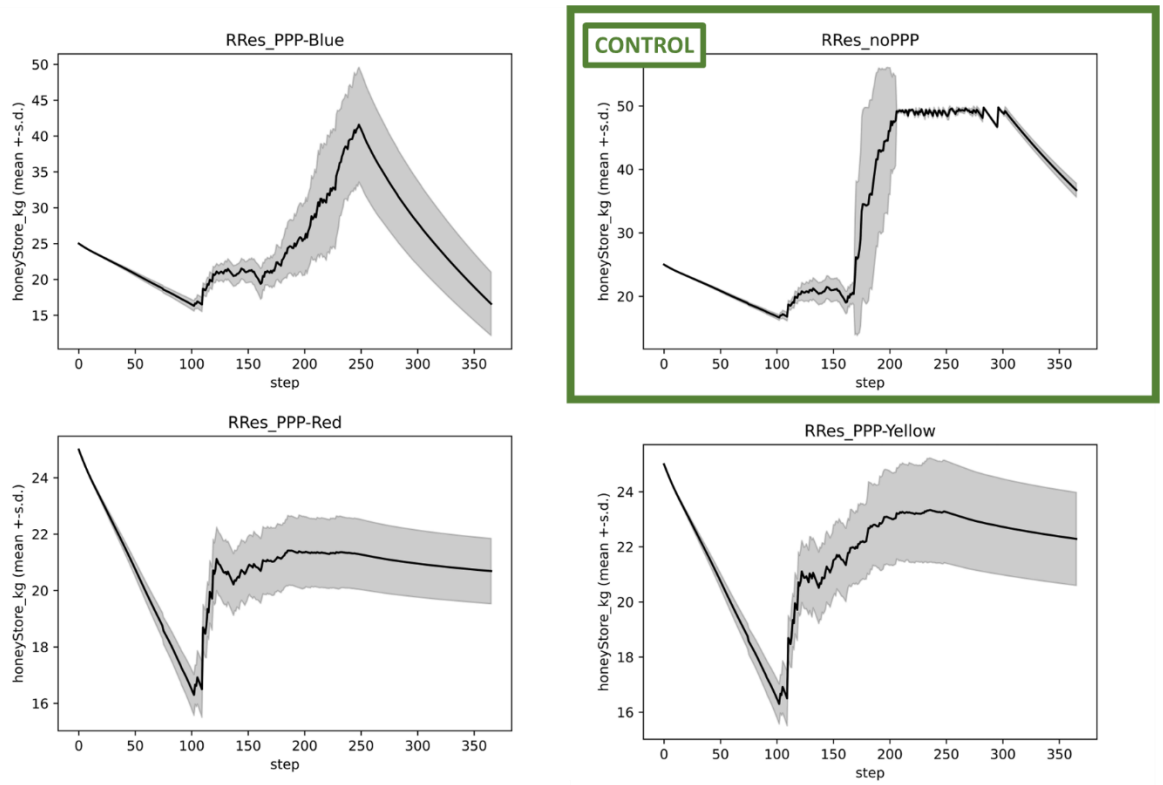

Figure 3g): Mean of honey [kg] (±S.D.) stored in the colony

References

Becher, M.A., Grimm, V., Thorbek, P., Horn, J., Kennedy, P.J., & Osborne, J. L. (2014). BEEHAVE: A systems model of honeybee colony dynamics and foraging to explore multifactorial causes of colony failure. Journal of Applied Ecology, 51(2), 470-482. https://doi.org/10.1111/1365-2664.12222.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S.K., Huse, G., Huth, A., Jepsen, J.U., Jørgensen, C., Mooij, W.M., Müller, B., Pe'er, G., Piou, C., Railsback, S.F., Robbins, A. M., et al. (2006). A standard protocol for describing individual-based and agent-based models. Ecological Modelling, 198(1–2), 115-126. https://doi.org/10.1016/j.ecolmodel.2006.04.023.

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J.G., Giske, J., & Railsback, S.F. (2010). The ODD protocol: A review and first update. Ecological Modelling, 221(23), 2760-2768. https://doi.org/10.1016/j.ecolmodel.2010.08.019.

Preuss, T.G., Agatz, A., Goussen, B., Roeben, V., Rumkee, J., Zakharova, L., & Thorbek, P. (2022). The BEEHAVEecotox Model—Integrating a Mechanistic Effect Module into the Honeybee Colony Model. Environmental Toxicology and Chemistry, 41(11), 2870-2882. https://doi.org/10.1002/etc.5467.

Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University. https://ccl.northwestern.edu/netlogo/.