

PROGRAM MAMO

Models for Avian Management Optimization

User guide, 1st edition

Alban Guillaumet, Eben Paxton

TABLE OF CONTENTS

Preface

Chapter 1

A QUICK INTRODUCTION TO MAMO

1.1 Overview

1.2. The R language

Chapter 2

THE BASIC MAMO FUNCTION: *mamo*

2.1 The arguments of *mamo*.....

2.1.1 Arguments of the 'Spatial structure' class

2.1.2 Arguments of the 'Time frame' class

2.1.3 Arguments of the 'Initial conditions' class

2.1.4 Arguments of the 'Survival' class

2.1.5 Arguments of the 'Reproduction and habitat quality' class	
2.1.6 Arguments of the 'Malaria' class	
2.1.7 Arguments of the 'Movement' class	
2.1.8 Arguments of the 'Other options' class	
2.2 Running <i>mamo</i>	
2.2.1 First use	
2.2.2 Subsequent use	
2.3 <i>mamo</i> output	
Chapter 3	
CALIBRATING MAMO	
3.1 The <i>f.calibr</i> function	
3.2 Running <i>f.calibr</i>	
3.3 The 'sensitivity envelop'	
3.3.1 The function <i>f.envp.single</i>	

3.3.2 Outputs of <i>f.envp.single</i>	
---	--

Chapter 4	
-----------------	--

SIMULATION STUDY

4.1 The <i>f.run</i> function	
-------------------------------------	--

4.2 Analysis of the simulation study	
--	--

4.2.1 The <i>f.plot.univar</i> function	
---	--

4.2.2 The <i>f.plot.bivar</i> function	
--	--

4.2.3 The <i>f.plot.composite</i> function	
--	--

Literature cited	
------------------------	--

Preface

Because Hawaii is the most remote archipelago of oceanic islands on earth, it has the highest percentage of land-based endemic birds anywhere in the world. Unfortunately, Hawaiian birds have suffered terrible losses since the arrival of people 800 - 1200 years ago: entire groups have disappeared such as flightless rails, waterfowl and a whole passerine family (Hawaiian honeyeaters), while others have lost much of their initial diversity. For instance, only 17 species (one-third) of Hawaiian honeycreepers remain, 10 of which are threatened with imminent risk of extinction, being listed as 'Endangered' or 'Critically Endangered' by the International Union for Conservation of Nature.

We developed and present in this manuscript a R-based program called MAMO that can evaluate the relative importance of different threats on forest birds as well as the effectiveness of various management actions to mitigate the threats at Hakalau Forest national Wildlife Refuge in Hawaii and beyond. The program is designed to become an adaptive management tool for managers, allowing non-modelers to update information, design simulations, and obtain meaningful output. The name of our software 'MAMO' is an acronym of 'Models for Avian Management Optimization', based on the name of the extinct Hawaiian honeycreeper mamoa (*drepanis sp.*).

The program comes with no guarantee. Its complexity is an asset to investigate a variety of scientific and applied questions, but it is also a weakness in the sense that we were unable to test its behavior in all the possible conditions. Despite our best efforts, bugs are not only possible, they are perhaps likely. In addition, the code is not always optimal in terms of both code efficiency and homogeneity, it is not always as simple as it could be for users, and it is not as flexible as it could be (for instance, hard code modification is necessary to introduce new data sets in the present version). Despite these potential shortcomings, we believe that MAMO fills an important gap in the current tool set available to researchers and managers in Hawaii. It already

produced exciting new insights into the dynamics and conservation of its wonderful native birds, and the program is amenable to future improvements.

None of this would have been possible if fifteen years ago the programmer (Alban Guillaumet) had not been introduced to the magic world of computer programming by Khalid Belkhir, researcher at the University of Montpellier II in France. Khalid, here is another opportunity for me to thank you for your support and generosity.

Chapter 1

A QUICK INTRODUCTION TO MAMO

1.1 Overview

Taken together, chapters 2-4 will allow the user to replicate a simulation study investigating the consequences of climate change and two potential management actions on the population dynamics of a vulnerable and iconic Hawaiian forest bird, the ‘Iiwi (*Vestiaria coccinea*; hereafter Iiwi).

Chapter 2 is devoted to the comprehensive description of the core function of the program MAMO, *mamo*; *mamo* is the function actually simulating the life-history of individuals belonging to a set of populations arranged on a spatial grid and potentially connected by seasonal migrations and dispersal.

While *mamo* can be called on its own, other wrapping functions such as *f.calibr* and *f.run* are generally used to call *mamo* and perform a batch of simulations exploring different *mamo* parameter combinations.

Chapter 3 concerns the calibration of *mamo* parameters. We describe the tools we developed (including *f.calibr*) to determine a set of *mamo* parameter values capable of replicating the current pattern of distribution of the study species (in this case, Iiwi) along an elevational gradient. This step is critical in order to obtain meaningful output when predicting the impact of

future climatic conditions or a given management action. In essence, if we cannot predict the present, how can we expect predicting the future?

In chapter 4, we will present the tools available (including *f.run*) to run a series of simulations covering the full spectrum of scenarios that we want to investigate while accounting for parameter uncertainty *via* what we call the 'sensitivity envelop'; in other words, chapter 4 is about designing and conducting a simulation study. A series of graphical tools are also presented in order to help the user interpreting the simulation results.

Additional information on the project, including the mathematical underpinning of MAMO, are described in a scientific paper that we will frequently refer to in this manual, and called the 'IIWI paper' [Guillaumet, A., Kuntz, W., Samuel, M. and Paxton, E. *Ecological Monographs*, In press. Altitudinal migrations and the future of an iconic Hawaiian honeycreeper in response to climate change and management]. A pdf of this paper will be included as soon as possible. In addition, occasional reference will be made to a second paper in preparation called the 'HAKALAU paper' extending the analyses presented here to address management strategies at the community level, when each avian species in the community respond idiosyncratically to management actions [Guillaumet, A., and Paxton, E. In preparation. A modeling framework to optimize management strategies applied to a diverse Hawaiian forest bird community]. Covering 15,390 ha on the windward slope of Mauna Kea, Hakalau is the largest protected area of mid- to high elevation forest in the state of Hawai'i. The 8 native forest bird species of Hakalau Forest National Wildlife Refuge are: Hawai'i 'Elepaio (*Chasiempis s. sandwichensis*), hereafter called ELEP, 'Ōma'o (*Myadestes obscurus*) (OMAO), 'Apapane (*Himatione sanguinea*) (APAP), 'I'iwi (IIWI), Hawai'i 'Amakihi (*Hemignathus virens*) (HAAM), 'Akiapōlā'au (AKIP), Hawai'i 'Ākepa (AKEP), and Hawai'i Creeper (HCRE).

1.2. The R language

R is a free open source software for data analysis, statistical computing, and graphics (Sanchez 2013). It is also a functional programming language, ideal for the creation and manipulation of functions. We designed MAMO as the equivalent of a R package, containing reusable R functions together with the documentation necessary to understand how to use them. We certainly entertain the idea that MAMO truly becomes a R package in the future.

Although learning R is not required to use MAMO, it is very helpful and we strongly recommended it. Self-teaching is an option, and we provide you below with a few lectures available online. Good luck!

<https://www.stat.berkeley.edu/~specter/Rcourse.pdf>

https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf

Chapter 2

THE BASIC MAMO FUNCTION: *mamo*

The function *mamo* is at the core of the program. As we previously mentioned, it is the one actually simulating the life-history of individual e-birds. Please refer to the 'IIWI paper' for a description of the mathematics underlying the function.

2.1 The arguments of *mamo*

Arguments are grouped into several classes, such as spatial structure and time frame, purely for descriptive and practical purpose. No special property is attached to any class. In the sections below, we present each argument which can be modified by the user using the following components: definition of the argument, range of values it can take as well as an example, and a facultative caution section to highlight potential misuse, unexpected interaction with other parameter values, etc.

2.1.1 Arguments of the 'Spatial structure' class

<i><u>nr</u></i>

Definition: Number of rows of the spatial grid. Each row represents a single elevational band.

Value: $\in \mathbb{N}^+ = \{1, 2, \dots\}$.

Ex: $nr = 10$

Caution:

1) Simulation time is related to the dimension of the grid.

nc

Definition: Number of columns of the spatial grid.

Value: $\in \mathbb{N}^+ = \{1, 2, \dots\}$.

Ex: $nc = 2$

Caution:

1) Simulation time is related to the dimension of the grid.

2) In order to reduce the simulation time, it is tempting to reduce the number of columns when no ecological process varies along the x -axis. It must be noted, however, that having a single column will result in a greater 'underestimation' of the number of pairs in the cells at the extremity, because they have a single adjacent cell, while other cells have two adjacent cells. This has implications for the number of migrants they receive. This effect, which would be artefactual if the habitat available is not unidimensional, is attenuated as the number of columns increase. In practice, we recommend using 2 or 3 columns at least.

grad

Definition: The elevational range of the spatial grid, starting by the highest elevation, in meters.

Used for producing figures and calculating summary statistics.

Value:

Ex: `grad = c(1900, 1000)`

Caution:

<u><i>unit</i></u>

Definition: Dimension of the patch, which is a square of side = *unit* Km.

Note: In practice for our analyses on windward Mauna Kea, *unit* = 1 so that a square of 1 Km² covers an elevational band of approximately 100 m.

Value:

Ex: `unit = 1`

Caution:

1) The value of *unit* seems to affect the capacity of the "fast.risky" option of *calc.gamma.d* to generate meaningful outputs. Always check the output parameter 'test.disp.breed' when using "fast.risky" (see chapter 2.3)

MAMO does not require the user to enter spatially explicit data for the arguments of other classes. Four options are possible, and MAMO will automatically spread the values entered into each of the spatial grid cell.

Unless stated otherwise, our examples for the remaining arguments correspond to a spatial grid (called *sg*) of 4 patches distributed over 2 rows and 2 columns. In this case, A and C are at higher elevation than B and D; as in R columns are filled in first, A corresponds to patch 1 (i.e. *sg*[1]), B to patch 2, C to patch 3 and D to patch 4.

[A] [C]

[B] [D]

1) When a single value is entered, the single value is spread into each grid cell.

Ex: for the parameter $t.b$, the value $t.b = 242$ will yield the matrix:

[242] [242]

[242] [242]

2) When a vector of length nr is entered, the vector is distributed over the nc columns.

Ex: for the parameter $t.b$, the value $t.b = c(242, 300)$ will yield the matrix:

[242] [242]

[300] [300]

3) A vector of length $nr \times nc$ can be entered.

Ex: for the parameter $t.b$, the value $t.b = c(242, 300, 150, 355)$ will yield the matrix:

[242] [150]

[300] [355]

4) A matrix can be entered.

Ex: for the parameter $t.b$, the value $t.b = \text{matrix}(c(242, 300, 150, 355), nr = 2, nc = 2)$ will yield the matrix:

[242] [150]

[300] [355]

2.1.2 Arguments of the 'Time frame' class

T

Definition: Number of years of simulation.

Value: $\in \mathbb{N}^+ = \{1, 2, \dots\}$.

Ex: $T = 60$

Caution:

1) Near-equilibrium dynamics may not been achieved with significantly shorter T . Graphical and numerical tools are available to help choose a particular value (see chapter 2.2.1).

T_m

Definition: Malaria (and rat predation) are introduced in the simulations at year $T_m + 1$. Previous simulation time aim at reaching an equilibrium, in particular concerning the proportion of each age class.

Value: $\in \mathbb{N}^+ = \{1, 2, \dots\}$.

Ex: $T = 5$

Caution:

1) T_m must be $\leq T$. If $T_m = T$, all simulation are run without malaria (and rat predation).

2) We recommend choosing T_m such as:

i) $T_m \geq 2$ (at least)

ii) $T - T_m > 50$

<u>t.b</u>

Definition: Duration of the breeding season (in number of days).

Value:

Range = [1, 365] ($\in \mathbb{N}^+$)

Ex: t.b = 365 is used for non-migratory species.

Caution:

1) Two distinct non-breeding seasons have been recognized for IIWI ('IIWI paper') and, presumably, APAP which shares a similar ecology. When the user chooses a value of t.b < 365, the current version of *mamo* switches to a 3-season model, with one breeding and two non-breeding seasons. Hence, at least one day per non-breeding season should be allowed, excluding the value t.b = 364.

2) Although *mamo* was designed to accommodate variation in phenology across the gradient, no specific testing of the model behavior was performed in such conditions.

<u>f.nb.1</u>

Definition: proportion of the non-breeding season (lasting 365 - t.b) spent in the first non-breeding patch.

Value:

1) Range =]0,1[for migratory species

Ex: f.nb.1 = 0.5

Theoretically, any value can be chosen for non-migratory species, since the parameter is irrelevant, although we recommend using f.nb.1 = NA.

Caution:

1) In the current *mamo* version, only one- or three-season designs are allowed, so *f.nb.1* cannot equal zero or one (both corresponding to two seasons).

<u><i>min.fledg</i></u>

Definition: Lower bound of fledging time (number of days after beginning of breeding season)

Value:

Range = $[1, t.b]$ ($\in \mathbb{N}^+$) if reproduction.malaria = "simple"

Ex: min.fledg = 100

Caution:

1) If reproduction.malaria = "complex", Range = $[1, t.b-14]$ ($\in \mathbb{N}^+$); make sure min.fledg < peak.fledg \times t.b < t.b - 14

<u><i>peak.fledg</i></u>

Definition: Peak of fledging time

Note: if either reproduction.malaria = "simple" or SD.fledg = 0, the fledging date is calculated as peak.fledg \times t.b; otherwise, the fledging date is derived from a truncated normal distribution with mean peak.fledg \times t.b.

Value:

Range = $]0, 1[$

Ex: peak.fledg = 2/3

Caution:

SD.fledg

Definition: standard deviation around *peak.fledg*

Value:

$\in R_0^+$ (0 and positive values)

Ex: *SD.fledg* = 20

Caution:

1) When reproduction.malaria = "complex" and *SD.fledg* \neq 0, the fledging time is calculated based on a truncated normal distribution with min = min.fledg, mean = *peak.fledg* \times t.b, max = t.b -14, and standard deviation = *SD.fledg*. Preliminary assessments suggested that the actual mean of the distribution differed notably from the expected mean (*peak.fledg* \times t.b) when *SD.fledg* was large; further debugging necessary.

2.1.3 Arguments of the 'Initial conditions' class

init.1

Definition: Initial number of 1-year old individuals per patch

Value: *init.1* $\in \mathbb{N}^+$

Ex: *init.1* = 50

Caution:

1) We recommend starting simulations at or near carrying capacity, by choosing e.g. *init.1* = *init.2* = half of the carrying capacity. Note that *init.1* = K.b / 2 cannot be used to initialize MAMO. If K.b = 200, then use *init.1* = 100 as well as *init.2* = 100.

2) If initial values are chosen too far from carrying capacity, it may take too long to build up numbers and the dynamics measured at T might still be transient.

<u><i>init.2</i></u>

Definition: Initial number of individuals ≥ 2 -year old per patch

Value: $\text{init.2} \in \mathbb{N}^+$

Ex: $\text{init.2} = 50$

Caution:

1) We recommend starting simulations at or near carrying capacity, by choosing e.g. $\text{init.1} = \text{init.2} = \text{half of the carrying capacity}$. Note that $\text{init.2} = K.b / 2$ cannot be used to initialize MAMO. If $K.b = 200$, then use $\text{init.1} = 100$ as well as $\text{init.2} = 100$.

2) If initial values are chosen too far from carrying capacity, it may take too long to build up numbers and the dynamics measured at T might still be transient.

2.1.4 Arguments of the 'Survival' class

<u><i>s.ad</i></u>

Definition: Baseline annual survival rate of adults (not including the effect of malaria or rat predation)

Value:

Range = $[0,1]$

Ex: $s.ad = 0.8$

Caution:

1) The parameter correspond to true survival, and not apparent survival (See e.g. Gilroy et al. 2012 for definitions).

<u><i>rat.s</i></u>

Definition: proportion of breeding females killed by rats; *rat.s* is such that the actual survival probability of females is multiplied by the coefficient (1-rat.s).

Value:

Ex: $rat.s = 0.05$

Caution:

1) We assume that $rat.s \leq rat.f$ for calibration purpose. Parameter combinations not respecting this inequality are automatically discarded during the calibration run. However, no such automatic constraint is present when running a single run of MAMO, so the user can actually choose $rat.s > rat.f$ (potentially biologically irrelevant).

<u><i>s.juv</i></u>

Definition: Baseline survival rate of juveniles from fledging to next year's breeding season (not including the effect of malaria or rat predation)

Value:

Range = [0,1]

Ex: $s.juv = 0.4$

For simplicity, we recommend using a value equivalent to $s.ad / 2$ (e.g., Gardali et al. 2003)

Caution:

1) Note that $s.juv = s.ad / 2$ cannot be used to initialize MAMO. If $s.ad = 0.7$, then use $s.juv = 0.35$.

2.1.5 Arguments of the 'Reproduction and habitat quality' class

<u><i>fec</i></u>

Definition: Baseline (malaria- and density-independent) number of female offspring fledged per female ≥ 2 -year old during a single breeding season.

Value: $fec \in \mathbb{R}^+$

Ex: $fec = 1.54$

Caution:

1) During the calibration process, we assume that $fec.1 \leq fec$ (other run are automatically discarded). This inequality is likely to be true for Hawaiian forest birds (e.g., Woodworth & Pratt 2009) and other birds/passerines. However, no such automatic constraint is present when running a single run of MAMO.

<u><i>fec.1</i></u>

Definition: Baseline (malaria- and density-independent) number of female offspring fledged per female 1-year old during a single breeding season.

Value: $fec.1 \in \mathbb{R}^+$

Ex: $\text{fec.1} = 0.49$

Caution:

1) During the calibration process, we assume that $\text{fec.1} \leq \text{fec}$ (other run are automatically discarded). This inequality is likely to be true for Hawaiian forest birds (e.g., Woodworth & Pratt 2009) and other birds/passerines. However, no such automatic constraint is present when running a single run of MAMO.

<u><i>rat.f</i></u>

Definition: proportion of juveniles killed by rats

Value:

Ex: $\text{rat.f} = 0.07$.

Caution:

1) We assume that $\text{rat.s} \leq \text{rat.f}$ for calibration purpose. Parameter combinations not respecting this inequality are automatically discarded during the calibration run. However, no such automatic constraint is present when running a single run of MAMO, so the user can actually choose $\text{rat.s} > \text{rat.f}$ (potentially biologically irrelevant).

<u><i>K.b</i></u>

Definition: Carrying capacity (maximum number of breeding pairs) of a single patch.

Value: $K.b \in \mathbb{N}^+$

Ex: $K.b = 580$

Caution:

1) The behavior of the model has not been tested at very low density ($K.b < 4$), and some erratic or unexpected behavior may arise; further developments will be necessary to account for irregular landscapes including patches with $K.b = 0$

thr.DD

Definition: Threshold (number of pairs per patch) above which automatic density-dependent reduction in fecundity occurs (see 'IIWI paper' for details).

Value: $thr.DD \in \mathbb{N}^+$

Ex: $thr.DD = 64$

We recommend using a value equivalent to $K.b / 2$ (e.g., Frederiksen et al. 2001)

Caution:

1) $thr.DD \leq K.b$

2) Note that $thr.DD = K.b / 2$ cannot be used to initialize MAMO. If $K.b = 580$, then use $thr.DD = 290$.

K.nb.1

Definition: patch quality (food abundance) during the 1st period of the non-breeding season

Value:

Ex: $K.nb.1.2003 = c(2.45556, 3.23333, 4.01111, 4.78889, 5.56667, 6.05000, 9.81667, 17.52500, 19.12000, 17.33333)$ corresponding to $grad = c(1900, 1000)$.

Caution:

1) although the function *mamo* can accommodate any *K.nb.1* value, other functions involved in calibration and simulation studies currently can only recognize the *K.nb.1* values included in the

data file associated with the program (called 'm3_data.r'); greater flexibility may be allowed in future MAMO versions.

K.nb.2

Definition: patch quality (food abundance) during the 2nd period of the non-breeding season

Value:

Ex: `K.nb.2 = c(4.06815, 7.07556, 10.08296, 13.09037, 16.09778, 19.16667, 7.47778, 4.55000, 4.16667, 2.36667)` corresponding to `grad = c(1900, 1000)`.

Caution:

reproduction.malaria

Definition: This parameter offers two similar yet distinct algorithms to account for the fact that newly infected e-birds should have, everything else being equal, a lower reproductive success than other categories.

Value: "simple" or "complex"

Note: the complete life-history information of all individuals, including their lifetime patch occupancy and reproductive output, can be stored and obtained at the end of the simulation using the "complex", but not the "simple", algorithm.

Ex: `reproduction.malaria = "simple"`

Caution:

1) We recommend using the "simple" version because it is much faster and has provided us with very similar results in the simulations conducted so far.

2.1.6 Arguments of the 'Malaria' class

alpha.b

Definition: daily probability of infection of a susceptible e-bird during the breeding season

Value:

Ex: $\alpha.b = c(0.00000e+00, 0.00000e+00, 3.21323e-06, 6.42646e-06, 1.40851e-03, 3.37271e-03, 5.89904e-03, 8.98748e-03, 1.29177e-02, 1.34780e-02)$ corresponding to $\text{grad} = c(1900, 1000)$.

Caution:

1) Currently available data sets, *alpha.b* (above) and *alpha.b.2100*, have been estimated for the period going from September 1st to April 30th (corresponding to $t.b = 242$, used for IIWI and APAP).

alpha.nb.1

Definition: daily probability of infection of a susceptible e-bird during the 1st period of the non-breeding season

Value:

Ex: $\alpha.nb.1 = c(0.00000e+00, 0.00000e+00, 1.61291e-08, 3.22581e-08, 7.02270e-04, 1.68541e-03, 2.94945e-03, 4.49439e-03, 6.46068e-03, 6.74157e-03)$ corresponding to $\text{grad} = c(1900, 1000)$.

Caution:

1) Currently available data sets used for IIWI and APAP, *alpha.nb.1* (above) and *alpha.nb.1.2100*, have been estimated for the May-June period corresponding to $f.nb.1 = 0.5$

(technically, they were actually estimated for the May 1st to July 1st period due to rounding code used, which could be changed in future versions).

<u><i>alpha.nb.2</i></u>

Definition: daily probability of infection of a susceptible e-bird during the 2nd period of the non-breeding season

Value:

Ex: $\alpha.nb.2 = c(0.00000e+00, 0.00000e+00, 6.92295e-05, 1.38459e-04, 2.92186e-03, 6.84632e-03, 1.19118e-02, 1.81184e-02, 2.60054e-02, 2.71084e-02)$ corresponding to $grad = c(1900, 1000)$.

Caution:

1) Currently available data sets used for IIWI and APAP, *alpha.nb.2* (above) and *alpha.nb.2.2100*, have been estimated for the July-August period corresponding to $f.nb.1 = 0.5$ (technically, they were actually estimated for the July 2nd to August 31st period due to rounding code used, which could be changed in future versions).

<u><i>Sm.ac</i></u>

Definition: Probability of survival to acute malaria infection.

Note: if they survive the first, acute infection, e-birds are assumed to be immune to subsequent infections

Value:

Range = [0,1]

Ex: $Sm.ac = 0.16$

Caution:

1) Current version of *mamo* does not allow spatial variation in this parameter.

2.1.7 Arguments of the 'Movement' class

<u><i>gamma.mov</i></u>

Definition: parameter controlling the extent of seasonal ('migratory') movements

Note: the larger the *gamma.mov* value, the greater the propensity to 'migrate' and stay (including at night) near higher quality patches. We therefore regard *gamma.mov* as an index of migration propensity

Value:

Range = [-10,10]

Can take a continuum of values between -10 (no 'net' movement, i.e. roosting always in the breeding patch) and +10 which corresponds to an ideal-free distribution (no resistance to movement / distance does not impede movement). For intermediate values, birds tend to spend the night closer to their breeding patch than expected based on resources alone due to the combined effects of partial resistance to movements and / or partial commuting.

Ex: *gamma.mov* = 0.541, for which the probability to reach a given (equivalent) patch is multiplied by 3/4 every time distance increases by one km.

Caution:

1) Current version of *mamo* does not allow spatial variation in this parameter.

2) The current version only applies to IIWI and APAP (as determined by $t.b < 365$; see argument *t.b* for further explanations). For other species, any value can be entered, but we recommend using `gamma.mov = NA`.

calc.gamma.d

Definition: Algorithm used to calculate the hidden parameter *gamma.d*, which is the parameter of the negative exponential distribution used to shuffle the dispersing fraction *fraction.disp* into neighboring patches; *fraction.disp* depends in particular (but not only - patch dimension also is important) on the parameters *fidelity.ad* for adults and *m.natal* for juveniles (see also 'IIWI paper' and chapter 2.3 for more details)

Value:

Currently, can take two values: `calc.gamma.d = "fast.risky"` or `"slow.robust"`

Caution:

1) As the name implies, "slow.robust" is slow but should always give a meaningful result. Limited testing suggests that the much faster "fast.risky" works well when *unit* = 1, but not in general (further developments required). Always check the output parameter *test.disp.breed* when using "fast.risky" (see chapter 2.3)

n.sim.disp

Definition: The number of dispersal events simulated to estimate *fraction.disp*, the proportion of e-birds leaving a given patch. Each dispersal event is a random draw based on *fidelity.ad* (for adults), or *m.natal*, *SD.natal* and *psi.DD* (for juveniles).

Value: $n.sim.disp \in \mathbb{N}^+$

Ex: `n.sim.disp = 10000`

Caution:

- 1) We recommend to use values ≥ 1000 , in order to have a decent sample size.

<u><i>R.ter</i></u>

Definition: Dimension of the breeding territory (radius of a disk in Km).

Value:

Range = $R.ter \in \mathbb{R}^+$

Ex: $R.ter = 0.0234$

Caution:

- 1) Current version of *mamo* does not allow spatial variation in this parameter
- 2) *R.ter* is generally estimated at relatively high density (see the 'IIWT' and 'HAKALAU' papers for details). Density-dependence could be accounted for in future versions.

<u><i>fidelity.ad</i></u>

Definition: probability for an adult to breed in year $t+1$ in the same breeding territory as in year t .

Value:

Range = $[0,1]$

Ex: *fidelity.ad* = 0.93 could be used for ELEP (VanderWerf 2004).

Caution:

- 1) Current version of *mamo* does not allow spatial variation in this parameter.

- 2) Density-dependence and public-information could be accounted for in future versions.
- 3) Since all 8 Hawaiian forest bird species considered so far may start breeding when 1 year old (Woodworth & Pratt 2009, Table 8.2), 1-year old birds are regarded as adults.

m.natal

Definition: The mean natal dispersal distance in Km.

Note: we use random draws from a log-normal distribution (X) to calculate natal dispersal distances, with location and scale parameters (μ and σ) chosen such as $E(X) = m.natal$ and $SD(X) = SD.natal$ using the formulas: $\mu = \ln\left(\frac{m.natal^2}{\sqrt{m.natal^2 + SD.natal^2}}\right)$ and $\sigma = \sqrt{\ln\left(1 + \frac{SD.natal^2}{m.natal^2}\right)}$.

Value: $m.natal \in R_0^+$ (0 and positive values)

Ex: $m.natal = 0.3$

Caution:

- 1) Current version of *mamo* does not allow spatial variation in this parameter.

SD.natal

Definition: The standard deviation of natal dispersal distance in Km.

Note: we use random draws from a log-normal distribution (X) to calculate natal dispersal distances, with location and scale parameters (μ and σ) chosen such as $E(X) = m.natal$ and $SD(X) = SD.natal$ using the formulas: $\mu = \ln\left(\frac{m.natal^2}{\sqrt{m.natal^2 + SD.natal^2}}\right)$ and $\sigma = \sqrt{\ln\left(1 + \frac{SD.natal^2}{m.natal^2}\right)}$.

Value: $SD.natal \in R_0^+$ (0 and positive values)

Ex: $SD.natal = 0.3$

Caution:

1) Current version of *mamo* does not allow spatial variation in this parameter.

psi.DD

Definition: Coefficient affecting juvenile dispersal rates due to density-dependence. Mean and standard deviation of natal dispersal distance are multiplied by *psi.DD* when local density is superior to *K.b*.

Value: $[1, \infty)$

Ex: $psi.DD = 1$ (no effect); $psi.DD = 1.5$ (mean natal dispersal distance is multiplied by 1.5)

Caution:

1) must be ≥ 1 , but reasonably small values should be used (e.g., well inferior to 10)

2.1.8 Arguments of the 'Other options' class

add.cline

Definition: Logical: should a sigmoid curve (cline) be fitted to density values along the elevational gradient?

Value: Calling the elevational gradient *grad*, the cline is fitted using the formula:

$a + \frac{b-a}{1 + e^{-K \cdot (grad - c)}}$, where the constants *a*, *b*, *K* and *c* are obtained using the *nls* function in R.

Ex: `add.cline = TRUE` (or FALSE)

Caution:

1) The algorithm may fail to estimate the parameters of the cline, for instance if density pattern is not sigmoidal, but also unpredictably when the pattern seems compatible with a sigmoid. To avoid *mamo* to execute an error action, we use the `warnonly = TRUE` option of the *nls* function (fitting the sigmoid), which will return a warning message in case of convergence failure. However, even if convergence has not been reached, *nls* will provide an estimate for the cline parameters that will be automatically added to the *mamo* plot (see chapter 2.2) if `add.cline = TRUE`. More flexible algorithms, such as GAM (generalized additive model), may be implemented in future versions.

2.2 Running *mamo*

2.2.1 First use

For the sake of the demonstration, running the function *mamo* can be decomposed into a 7-step process at first use. Afterwards, only 3 steps will be necessary.

1) Open a R workspace, e.g. by double-clicking on the icon of the R Graphical User Interface (RGui).

2) Install the following R packages: *truncnorm*, *compiler*, *pastecs*, *lme4*, *plotrix*, *rsm*, *ade4*, *ggplot2* and *msm* by copying and pasting the following code (in green). Some of these packages

are not necessary to run *mamo* per se, but they will be useful for additional use of the MAMO software.

```
install.packages("truncnorm")
```

```
install.packages("compiler")
```

```
install.packages("pastecs")
```

```
install.packages("lme4")
```

```
install.packages("plotrix")
```

```
install.packages("rsm")
```

```
install.packages("ade4")
```

```
install.packages("ggplot2")
```

```
install.packages("msm")
```

3) Tell R where to find the source code corresponding to MAMO. We created a folder called 'MAMO' containing all the source code and some additional elements that will be discussed later. In our case, 'MAMO' is in a folder called 'Programs' that we created on the C hard drive of the computer. Of course, you would have to modify the pathway if you created the MAMO file in a different folder.

```
# Location of the source files
```

```
setwd("C:/Programs/MAMO")
```


4) Read the R code that will load the required R libraries.

```
# load libraries
```

```
source("m2_libraries.r")
```

5) Read the R code that will load the required data sets.

```
# load data sets
```

```
source("m3_data.r")
```

6) Read the R code corresponding to the *mamo* function

```
# load the mammo function
```

```
source("m4_mamo.r")
```

7) Run mammo

```
# run an example of MAMO
```

```
mamo.ex = mammo(
```

```
# Spatial structure
```

```

nr = 10, nc = 2, grad = c(1900, 1000), unit = 1,

# Time frame

T = 60, Tm = 5, t.b = 242, f.nb.1 = 0.5, min.fledg = 100, peak.fledg = (2/3), SD.fledg = 0,

# Initial conditions

init.1 = 300, init.2 = 300,

# Survival (ad = annual, juv = from fledging to breeding age)

s.ad = 0.729, rat.s = 0, s.juv = (0.729/2),

# Reproduction and habitat quality

fec = 3, fec.1 = 3, rat.f = 0, K.b = 600, thr.DD = 300, K.nb.1 = K.nb.1.2004, K.nb.2 = K.nb.2,
reproduction.malaria = "simple",

# Malaria parameters (daily except Sm.ac)

alpha.b = alpha.b, alpha.nb.1 = alpha.nb.1, alpha.nb.2 = alpha.nb.2, Sm.ac = 0.13,

# Movements

gamma.mov = 0.541, calc.gamma.d = "fast.risky", n.sim.disp = 10000, R.ter = 0.0234, fidelity.ad
= 0.95, m.natal = 0.3, SD.natal = 0.3, psi.DD = 1,

# Other options

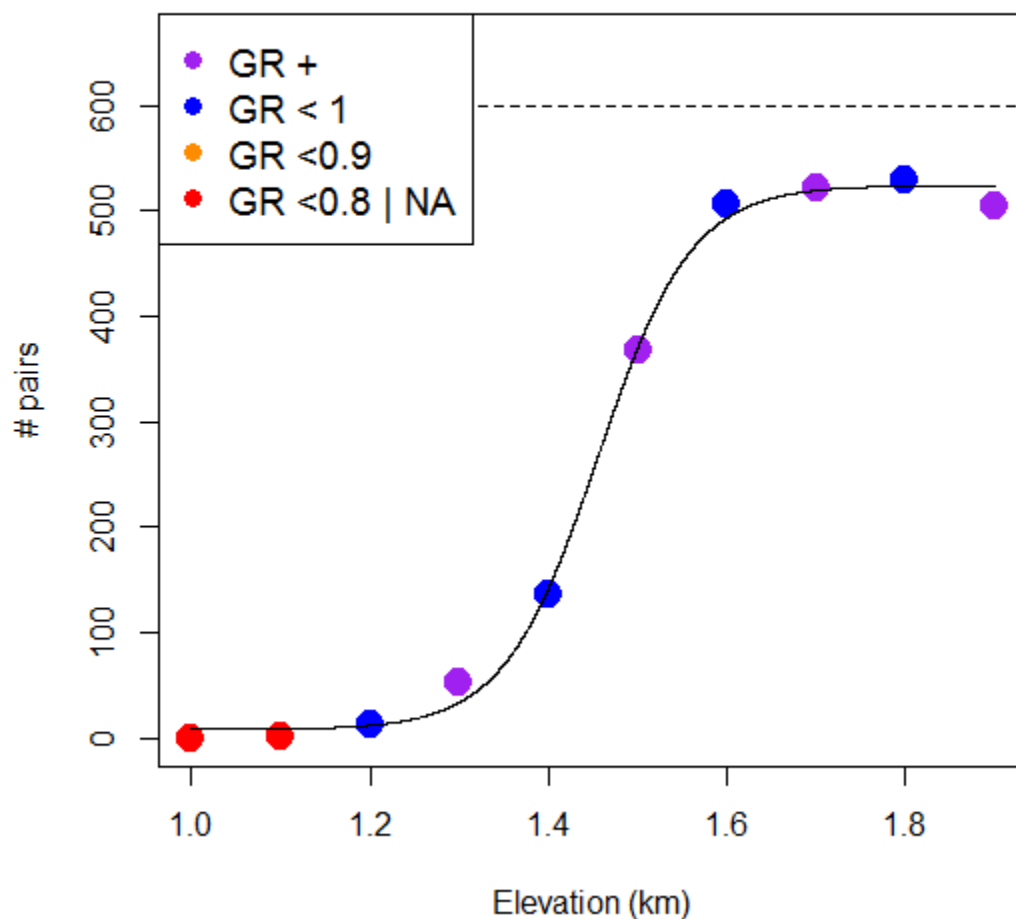
add.cline = TRUE)

```

Details of the simulation are stored into the R object *mamo.ex* that we created. In particular, *mamo.ex\$np* contains the simulated vector of population size (the number of pairs in a square patch of side = unit Km) along the elevational gradient measured by the vector *mamo.ex\$grad* (elevation in Km). We will get back on *mamo* output in more details later. For now, we note that a run of *mamo* automatically produces a figure, plotting *np* as a function of *grad*. Population size

at each elevation is represented by a circle of one of the following colors: purple when the growth rate ($GR = \frac{N_t}{N_{t-1}}$), as averaged over the last three years of simulation, is ≥ 1 , blue when $0.9 \leq GR < 1$, orange when $0.8 \leq GR < 0.9$, and red when $GR < 0.8$ or $GR = \text{NA}$. The latter happens for instance when population size during any of the last three years is zero. If the populations we simulated were sampled are at or near equilibrium, and are significantly above zero, we generally expect a mixture of purple and blue points (see Fig below). Finally, the carrying capacity in the absence of malaria (parameter $K.b$) is shown as a dashed black line. Note that if you run the same piece of code again, a slightly different figure will be produced, and so your figure should be slightly different than ours. This is due to demographic stochasticity.

Fig. 2.2.1-1



2.2.2 Subsequent use

Now that some required R packages have been installed, only 3 steps are necessary to run the function *mamo*.

1) Open a R workspace, e.g. by double-clicking on the icon of the R Graphical User Interface (RGui).

2) Load all the necessary source code, including libraries, data sets, and all MAMO functions. The simplest way to do this is opening the MAMO file called 'm1_SCRIPT.r' (hereafter called SCRIPT file) in a text editor such as Notepad++, and copy and paste in R the code at the beginning of the file, called BLOCK 1, delimited by `setwd("C:/Programs/MAMO")` (beginning) and `source("m10_f.plot.bivar.r")` (end).

These two steps are to be repeated every time a new MAMO session is conducted, no matter what exactly is performed (calibration, analyzing outputs, etc).

3) Run *mamo*

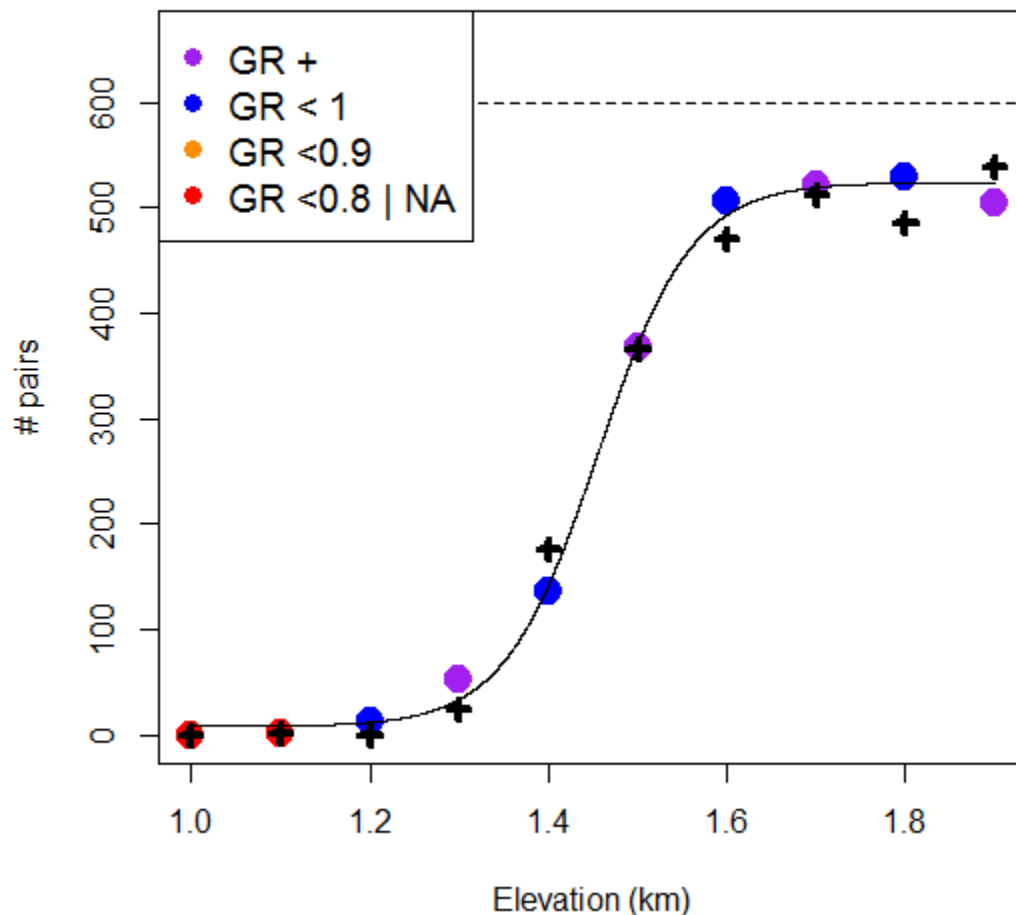
Now R is ready to run the previous example of MAMO, whose code can be found in the BLOCK 2 of the SCRIPT file.

The *mamo* parameter values used in this example aimed at replicating the current distribution of IIWI along the elevational gradient along the Hamakua coast. It is therefore possible to visually

estimate the fit between observed and simulated data by plotting the observed data points using black crosses with the following code (please see the file called 'm3_data.r' for details regarding the IWI data (y.obs.IWI.1)). As you can see in the Fig. below, the simulation run makes a good job at replicating the observed distribution.

```
points(seq(1.9, 1.0, length.out = 10), (y.obs.IWI.1*100), pch = 3, col = "black", cex = 1, lwd = 4)
```

Fig. 2.2.2-1



You can try changing a single parameter of the model and see how this affects the prediction. For instance, consider changing *fec.1* from 3 to 0, which implies that first-year birds do not

reproduce at all. We reproduce the code of BLOCK 2 below, with the change highlighted by orange color and enlarged font size. As you can see in the Figure that follows (and hopefully, in your R console!), this time the fit is quite not as good. As could be anticipated, there is a deficit in the number of predicted birds based on this new parameter value.

BLOCK 2

run an example of MAMO

```
mamo.ex = mamo(
```

```
# Spatial structure
```

```
nr = 10, nc = 2, grad = c(1900, 1000), unit = 1,
```

```
# Time frame
```

```
T = 60, Tm = 5, t.b = 242, f.nb.1 = 0.5, min.fledg = 100, peak.fledg = (2/3), SD.fledg = 0,
```

```
# Initial conditions
```

```
init.1 = 300, init.2 = 300,
```

```
# Survival (ad = annual, juv = from fledging to breeding age)
```

```
s.ad = 0.729, rat.s = 0, s.juv = (0.729/2),
```

```
# Reproduction and habitat quality
```

```
fec = 3, fec.1 = 0, rat.f = 0, K.b = 600, thr.DD = 300, K.nb.1 = K.nb.1.2004, K.nb.2 = K.nb.2,  
reproduction.malaria = "simple",
```

```
# Malaria parameters (daily except Sm.ac)
```

```
alpha.b = alpha.b, alpha.nb.1 = alpha.nb.1, alpha.nb.2 = alpha.nb.2, Sm.ac = 0.13,
```

```
# Movements
```

```
gamma.mov = 0.541, calc.gamma.d = "fast.risky", n.sim.disp = 10000, R.ter = 0.0234, fidelity.ad  
= 0.95, m.natal = 0.3, SD.natal = 0.3, psi.DD = 1,
```

```
# Other options
```

```
add.cline = TRUE)
```

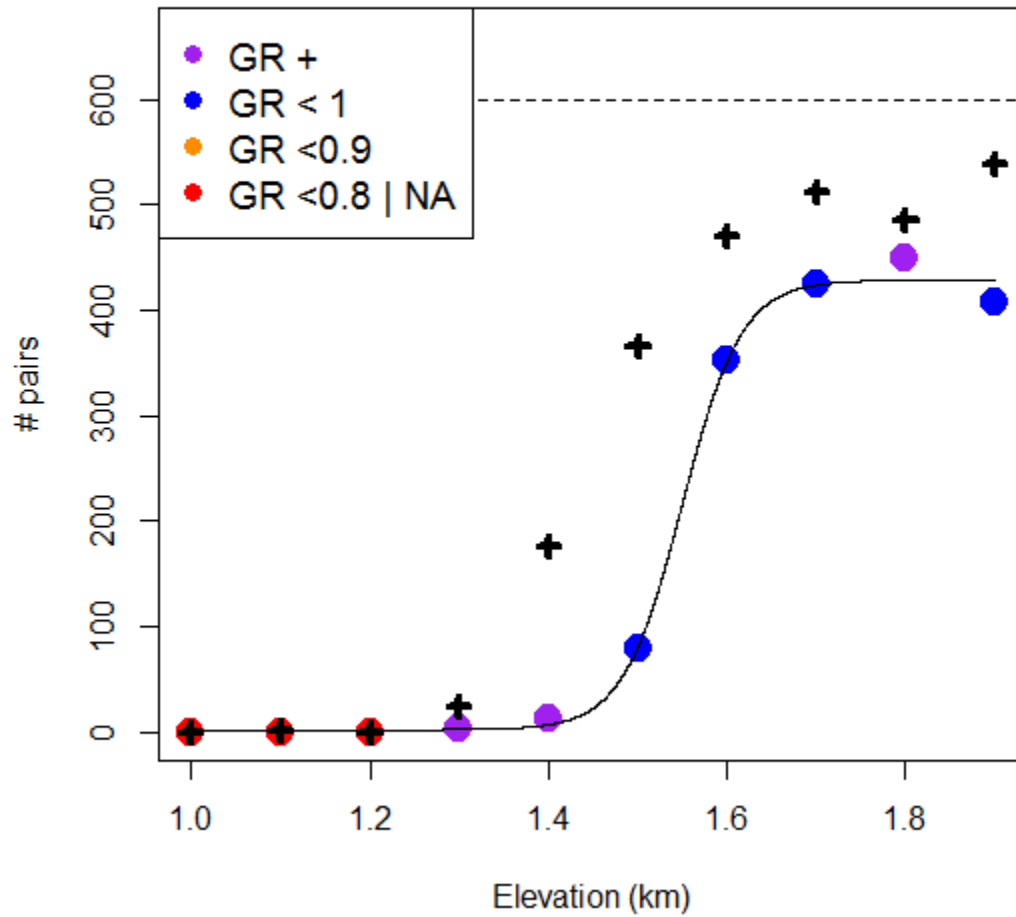
```
# see output
```

```
mamo.ex
```

```
# Add real data points for comparison
```

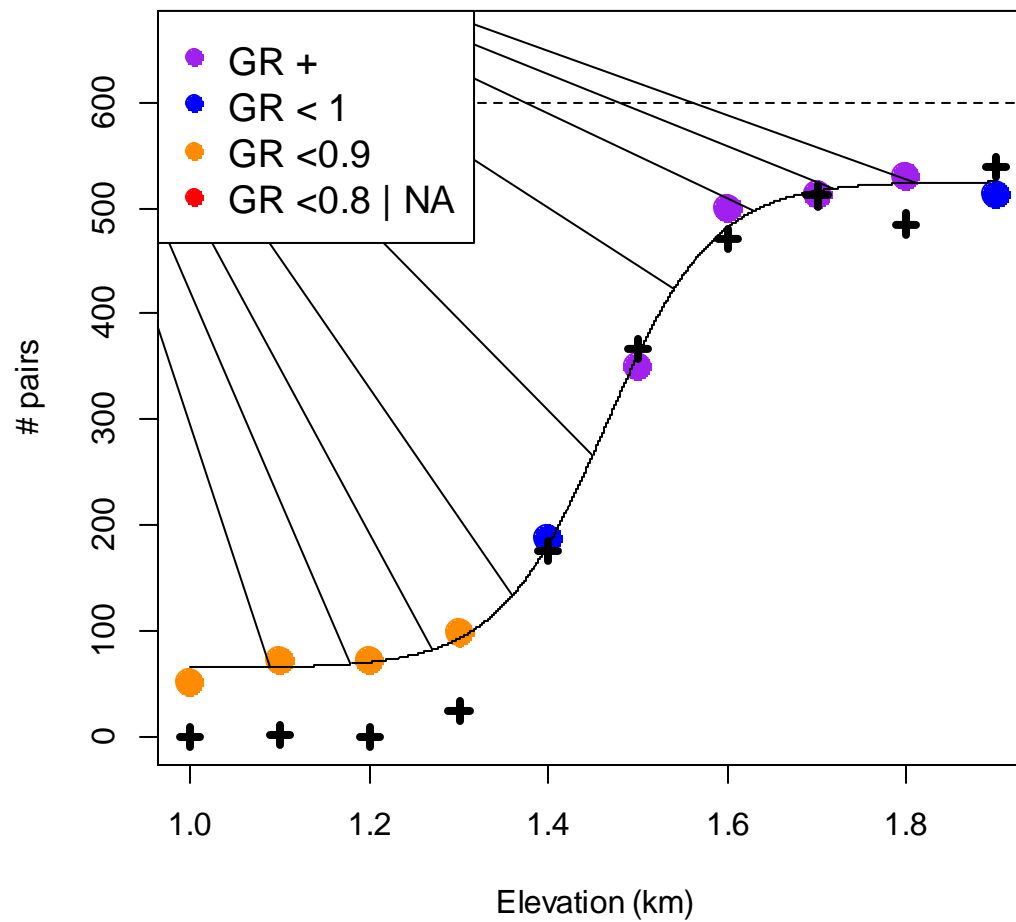
```
points(seq(1.9, 1.0, length.out = 10), (y.obs.IIWI.1*100), pch = 3, col = "black", cex = 1, lwd =  
4)
```

Fig. 2.2.2-2



Finally, here is a last example where a single parameter has been changed, from $T = 60$ to $T = 10$. As you can see in the Figure below (2.2.2-3), the populations in the upper sites are already at or near equilibrium, but the populations in the lower ranges, the most affected by malaria, are not and are still decreasing to their actual equilibrium (see Fig. 2.2.2-1 for equilibrium value).

Fig. 2.2.2-3



2.3 *mamo* output

In addition to the automatic plot described above, the function *mamo* returns a list of variables and their value.

As a rule, the input variable and their values are returned by *mamo*. This is useful to keep track of the input values producing a given result. For some of them, the input value is returned without any modification.

This includes *nr*, *nc*, *unit*, *T*, *Tm*, *reproduction.malaria*, *Sm.ac*, *gamma.mov*, *calc.gamma.d*, *n.sim.disp*, *R.ter*, *fidelity.ad*, *m.natal*, *SD.natal*, *psi.DD*, and *add.cline*.

For the rest of the arguments, the actual output is generally modified from a number or a vector into a matrix (see chapter 2.1.1 for an explanation). The arguments that can vary in space fall into this category.

This includes *t.b*, *t.nb*, *t.nb.2*, *min.fledg*, *peak.fledg*, *SD.fledg*, *init.1*, *init.2*, *s.ad*, *rat.s*, *s.juv*, *fec*, *fec.1*, *rat.f*, *K.b*, *thr.DD*, *K.nb.1*, *K.nb.2*, *alpha.b*, *alpha.nb.1*, *alpha.nb.2*.

The argument *grad* is transformed into a vector of length *nr* using the formula $\text{seq}(\text{grad}[1], \text{grad}[2], \text{length} = \text{nr}) / 1000$. For instance, a gradient entered as argument as *grad* = c(1900, 1000) in meters will be returned as the following vector (1.9, 1.8, 1.7, 1.6, 1.5, 1.4, 1.3, 1.2, 1.1, 1.0) in Km.

Finally, a set of variables are new:

s.ad.d and *s.juv.d* are the daily survival rates of adult and juveniles; *s.ad.d* is calculated over the entire year (i.e., $s.ad.d = s.ad^{1/365}$), while *s.juv.d* is calculated over a period of $(365 - \text{peak.fledg} \times t.b)$ days.

K.nb.1.high, *K.nb.1.mid*, *K.nb.1.low*, *K.nb.2.high*, *K.nb.2.mid*, and *K.nb.2.low* are the corresponding value of *K.nb.1* or *K.nb.2* at high (1800 m), mid (1500m) and low (1200 m) elevation, respectively. Note that these values are returned only if $\text{grad}[1] \geq 1.9$ and $\text{grad}[\text{nr}] \leq 1.1$. Otherwise, NA is returned.

alpha.b.low, *alpha.nb.1.low*, and *alpha.nb.2.low*, are the corresponding value of the daily probability of malaria infection of a susceptible e-bird ('alpha') at the low (1200 m) elevation band during the breeding season, first period of the non-breeding season, and second period of

the non-breeding season, respectively. Note that these values are returned only if $\text{grad}[1] \geq 1.9$ and $\text{grad}[\text{nr}] \leq 1.1$. Otherwise, NA is returned.

gamma.d.breed is the estimate of the hidden parameter *gamma.d* for the $(\text{nr} \times \text{nc})^{\text{th}}$ patch. Hence *gamma.d.breed* is the parameter of a negative exponential distribution estimated in order to distribute *fraction.disp* leaving the $(\text{nr} \times \text{nc})^{\text{th}}$ patch into the remaining $\text{nr} \times \text{nc} - 1$ patches; essentially for debugging, not management, purpose.

test.disp.breed is a parameter used for debugging purpose, checking for one patch that the fraction of breeders that left the patch after breeding dispersal is about equal to the value of the hidden variable *fraction.disp*; *test.disp.breed* therefore tests the ability of the negative exponential with parameter *gamma.d* to shuffle the right amount of dispersers into neighboring patches. **Contrary to *gamma.d.breed*, it is essential that users check that this parameter has a value close to zero (typically $\sim 10^{-9}$ - 10^{-5}) if *calc.gamma.d* = "fast.risky" has been chosen.** If any *test.disp.breed* value in the simulation set is larger than say 10^{-3} , we advise to re-run the analyses using *calc.gamma.d* = "slow.robust".

The variables *disp.breed* and *disp.natal* give the patch-specific transition probability matrices based on natal and breeding dispersal distances, respectively, while accounting for patch and grid dimensions and resource distribution; for instance, *disp.breed*[[1]][1] is the probability for an adult of patch 1 to remain in its former breeding patch next year, *disp.breed*[[1]][3] is the probability that it instead breeds in patch 3, and *disp.breed*[[4]][1] is the probability for an adult of patch 4 to disperse into patch 1. The variable *disp.natal.DD* gives the (higher) natal dispersal probabilities that apply when *psi.DD* > 1 and the breeding population is above *thr.DD*. For 'migratory' species, the variables *disp.nb.1* and *disp.nb.2* give the patch-specific transition probability matrices based on the 'migratory propensity' parameter *gamma.mov*, patch and grid dimensions and resource distribution during the non-breeding seasons; for instance, *disp.nb.1*[[1]][2] is the probability for breeding adult or juvenile of patch 1 to spend the first part of the non-breeding season (nb.1) in patch 2.

The two variables *mal.y* and *mal.a* are only valuable for internal routines and could be removed as outputs from future *mamo* versions.

The next set of variables, namely $n.l$, $n.ad$, $n.pairs$ and $n.juv$, are arrays of dimension nr , nc , and T , containing for each year from 1 to T the matrix of the number of first-year, adults, pairs and juveniles at the beginning (all except $n.juv$) or end ($n.juv$) of the breeding season, respectively. For instance, $n.pairs[,10] = n.l[,10] + n.ad[,10]$ gives the number of females (or pairs) at the beginning of the 10th breeding season in each patch of the spatial grid; $n.pairs[1,1,20]$, which can also be called by $n.pairs[,20][1]$, is the number of pairs at $T = 20$ in the first patch.

To obtain the elevation-specific average number of pairs at the end of the simulation (variable np , the average for a single patch), the number of breeding pairs was averaged over the last three years of simulation and over the nc columns of the grid, in order to attenuate the effect of demographic stochasticity; np values obtained at an elevation of 1800 (1750-1850), 1500 and 1200 m are called $np.high$, $np.mid$ and $np.low$, respectively. The variables $r.hm$ and $r.ml$ are simply the ratio between $np.high$ and $np.mid$ on one hand, or $np.mid$ and $np.low$ on the other hand, respectively. The total number of pairs in the final population (all patches) was $np.metapop = nc \times \sum_{k=1}^{nr} np_k$.

For illustration purposes, a sigmoid curve was fitted to np values along the elevational gradient $grad$ using the formula: $a + \frac{b-a}{1 + e^{-K \cdot (grad - c)}}$, where the constants a , b , K and c were obtained using the `nls` function in R. The output variables `cline.2`, `cline.5`, and `cline.8` are the values of $grad$ for which np , as modelled by the sigmoid, is 20%, 50% and 80% of its maximum, respectively; note that `cline.5` is the center c of the cline (inflexion point).

As we previously mentioned for the plot produced by a *mamo* run, we calculate for each patch the growth rate ($GR = \frac{N_t}{N_{t-1}}$), as averaged over the last three years of simulation. The growth rate is next averaged across patches in the same elevation to produce the output variable gr ; the variables $gr.high$, $gr.mid$, and $gr.low$ are simply gr values at 1800, 1500 and 1200 m, respectively. Note that NA values can be obtained (e.g., if a denominator value is 0).

Finally, $m.elev$ is the mean elevation in the gradient: $m.elev = \frac{\sum_{i=1}^{nr} grad_i}{nr}$, while $m.elev.np$, $m.elev.K.nb.1$, $m.elev.K.nb.2$, $m.elev.gr$, $m.elev.alpha.b$, $m.elev.alpha.nb.1$, and $m.elev.alpha.nb.2$ are weighted average measures of elevation along the gradient, with weights np , $K.nb.1$, $K.nb.2$,

gr , $alpha.b$, $alpha.nb.1$ and $alpha.nb.2$, respectively. For instance, $m.elev.np = \frac{\sum_{i=1}^{nr} grad_i \times np_i}{\sum_{i=1}^{nr} np_i}$ and $m.elev.gr = \frac{\sum_{i=1}^{nr} grad_i \times gr_i}{\sum_{i=1}^{nr} gr_i}$. Weighted averages indicate the location of the center of mass; for instance, in presence of malaria, we expect $m.elev.np > m.elev$ because more pairs will be present at high (refuge) than low elevation.

Chapter 3

CALIBRATING MAMO

3.1 The *f.calibr* function

Calibration consists in determining a set of *mamo* parameters values capable of replicating the current pattern of distribution of the study species along an elevational gradient. This will be helpful in order to obtain meaningful predictions when, for instance, we want to predict the impact of a management action, or future climatic conditions.

In that aim, a new function called *f.calibr* is introduced. Essentially, *f.calibr* will call and run *mamo* as many times as needed to explore the desired range of parameter combinations. The results will be stored in a folder that the user has created. The argument *output.dir* is used to indicate *f.calibr* its location.

Ex: `output.dir = "C:/Programs/MAMO/CALIBRATION/run/IWI.1_1"`

Raw results of the calibration will be stored in this folder in two different ways. First, a subset of *mamo* results are stored in a text file called *t.sim.txt*. Each row of the text file will contain the results of a single simulation run. **Importantly, in the current version of MAMO, an empty file called *t.sim.txt* must be created by the user and put into the destination folder (*output.dir*) before the calibration starts.** At the end of each calibration run, *t.sim.txt* will be called in and results will be added. Such an empty file is available in GitHub, but downloading it may be problematic (due to its size?). If you face problems downloading this file, you can create

yourself a tab-delimited text file that you will call *t.sim.txt* and copy into your 'output.dir' folder, with empty cells filled by NA, 50,000 rows and 77 columns named as follows: [total.sim, batch, sim, design, nr, nc, grad.max, grad.min, unit, T, Tm, t.b, t.nb.1, t.nb.2, min.fledg, peak.fledg, SD.fledg, init.1, init.2, s.ad, s.ad.d, rat.s, s.juv, s.juv.d, fec, fec.1, rat.f, K.b, thr.DD, K.nb.1.high, K.nb.1.mid, K.nb.1.low, K.nb.2.high, K.nb.2.mid, K.nb.2.low, reproduction.malaria, alpha.b.low, alpha.nb.1.low, alpha.nb.2.low, Sm.ac, gamma.mov, calc.gamma.d, n.sim.disp, R.ter, fidelity.ad, m.natal, SD.natal, psi.DD, add.cline, gamma.d.breed, test.disp.breed, disp.breed.p1, disp.natal.p1, disp.natal.DD.p1, disp.nb.1.p1, disp.nb.2.p1, np.high, np.mid, np.low, np.metapop, gr.high, gr.mid, gr.low, r.hm, r.ml, cline.5, cline.2, cline.8, m.elev, m.elev.np, m.elev.K.nb.1, m.elev.K.nb.2, m.elev.gr, m.elev.alpha.b, m.elev.alpha.nb.1, m.elev.alpha.nb.2, data.fit].

Second, an R file containing the complete *mamo* output will be produced for each calibration run, named s1.1.rdata for the first, s2.1.rdata for the second, etc. The argument *n.sim* of *f.calibr* allows the user to run each parameter combination more than once, in order to test the effect of demographic stochasticity (e.g., *n.sim* = 2). In that case, the files s1.2.rdata and s2.2.rdata also will be produced. Finally, another R file is produced synthesizing *mamo* outputs for the *n.sim* calibration run of each parameter combination. In our example, they would be called s1.rdata, and s2.rdata, respectively. Like for the text file, only a subset of *mamo* output are stored in this latter R file. Double-clicking on any R file opens the R program. The results can be accessed by typing *x* in the console.

Another critical argument of *f.calibr* is called *d*, a data set established beforehand which gives *f.calibr* important information regarding the parameter values to be explored. In our example, *d* = `read.table("C:/Programs/MAMO/CALIBRATION/Starting parameters/param_calib.IIWI.1.txt", header = T, sep = "\t", dec = ".")`. This apparently complex formula simply tells *f.calibr* to read the text file called 'param_calib.IIWI.1.txt' located in the folder 'Starting parameters', with the added indications that the text file contains header, decimals are "." as opposed to ",", etc. Please download the file 'param_calib.IIWI.1.txt' from GitHub and copy it into the appropriate folder.

First, the file `param_calib.IIWI.1.txt` contains the input value of parameters meant to be fixed across the calibration runs, namely *t.b*, *f.nb.1*, *min.fledg*, and *peak.fledg*. Of course, the assumption that these parameters are known without uncertainty could be relaxed in future versions of MAMO.

The file `param_calib.IIWI.1.txt` also contains the range of parameter values that we are willing to explore for another set of 13 variables such as *rat.s* (*rat.s* measures the extra mortality of breeding females due to rat predation). Specifically, the table contains the minimum and maximum value the user regard as likely. An argument specific to *f.calibr* is associated to each of these 13 variables, called *n.rat.s* for *rat.s*, *n.s.ad* for *s.ad* and so on. This specific argument tells *f.calibr* how many values of each parameter we want to examine during calibration. For instance, if *n.rat.s* = 1, only the minimum value of *rat.s* will be used; if *n.rat.s* = 2, both the minimum and maximum values of *rat.s* will be used. Beyond 2, a regular sequence of values from min to max is constructed; e.g., if *n.rat.s* = 3, the minimum, maximum and average (of min and max) values of *rat.s* will be used.

In more details, the set of 13 variables involved in the calibration process can be separated in three groups. For a first group of 7, namely *rat.s*, *rat.f*, *K.b*, *R.ter*, *fidelity.ad*, *m.natal*, and *psi.DD*, the calibration process is identical to what we just described for *rat.s* (called Proc.1).

For *K.nb.1*, the default argument of *f.calibr* is *K.nb.1* = list(*K.nb.1.2003*, *K.nb.1.2004*, *K.nb.1.avg*). If *n.K.nb.1* = 3, all three candidate values for *K.nb.1*, namely *K.nb.1.2003*, *K.nb.1.2004* and *K.nb.1.avg* will be examined in turn. Instead, if *n.K.nb.1* = 2, the minimum and maximum values in the file '`param_calib.IIWI.1.txt`' will be used. No other option is allowed.

For the final set of 5 variables in the third group, namely *s.ad*, *fec*, *fec.I*, *Sm.ac*, and *gamma.mov*, another specific *f.calibr* parameter is associated to each of them, respectively called *input.direct.s.ad*, *input.direct.fec*, etc.... These 'input.direct' parameters are logical, i.e. they can take the value TRUE or FALSE. For instance, if *input.direct.s.ad* = FALSE, then Proc.1 applies and the number of *s.ad* values examined will depend on *n.s.ad*. If *input.direct.s.ad* = TRUE, however, the values examined are determined by yet another *f.calibr* parameter, called *s.ad.direct* in this case (for *fec*, the corresponding parameter is *fec.direct*, etc.). The reason for adding this level of complexity is that for these variables, a regular sequence of values between min and max may not be appropriate, due to non-linear effects for instance (e.g., *gamma.mov*).

The majority of remaining arguments are identical to the ones described for *mamo*, with the following exceptions:

- *sp* is the species considered; so far, it can be any of the eight native forest bird species that breed at Hakalau National Wildlife Refuge in Hawaii: the Hawai'i 'Elepaio (*Chasiempis s. sandwichensis*) is a monarch flycatcher (Monarchidae), and the 'Ōma'o (*Myadestes obscurus*) is a thrush (Turdidae); the remaining six species are Hawaiian honeycreepers: the 'Apapane (*Himatione sanguinea*), 'Iiwi, Hawai'i 'Amakihi (*Hemignathus virens*), 'Akiapōlā'au, Hawai'i 'Ākepa, and Hawai'i Creeper. Corresponding acronyms for *sp* are "ELEP", "OMAO", "APAP", "IIWI", "HAAM", "AKIP", "AKEP" and "HCRE", respectively.
- *y.obs* is the observed vector of species distribution along the considered elevational gradient (as defined by the argument *grad*), that is taken from the data file called 'm3_data.r'. It will be used to choose among the set of calibration runs the subset that best predict the observed data.
- *paired.s.ad.fec* is logical (TRUE or FALSE); if FALSE, *n.s.ad* × *n.fec* combinations of parameters are run for these two parameters (i.e., all possible combinations); if TRUE, only paired values are run. For instance, if *input.direct.s.ad* = *input.direct.fec* = TRUE and *n.s.ad* = *n.fec* = 2, the only combinations allowed will be *s.ad.direct*[1]-*fec.direct*[1] and *s.ad.direct*[2]-*fec.direct*[2]. This is useful if we want the calibration run to be shorter, assuming that the maximum annual fecundity (*fec*) and adult survival (*s.ad*) are linked in avian species, and that only certain combinations are reasonable (see the 'IIWI paper' Appendix S4 for details).

- *alpha.1* is a data set described in the data file; It is the daily probability of infection of a susceptible e-bird staying in the same patch during the whole year (applicable to non-migratory species).
- *design* and *batch* are two arguments required by *simul_mamo*, an ancillary function 'transmitting the information' between *f.calibr* and *mamo*. In the current version of MAMO, use of default values for *design* and *batch* ("simple" and 1, respectively) is required.

3.2 Running *f.calibr*

The code of BLOCK 3 of the SCRIPT file is reproduced below. **It will perform the calibration step of the 'IIWI paper' once we change the code of the first row from *run_f.calibr = FALSE* to *run_f.calibr = TRUE*.** This line of code is here to ensure that the user will not inadvertently run the calibration again if she/he does not desire so, which would automatically starts to erase the results previously obtained. **Importantly, the user has to wait until completion of all the simulation before opening the *t.sim.txt* file, otherwise calibration will fail. Review the sleep mode of your computer** as the process took us 31 hours and 9 minutes on a DELL LATITUDE E6230 with a 2.8 GHz Intel Core 3rd Generation i5-3360M Processor. You can monitor the progress of the calibration by looking into the IIWI_1 folder you created, as .R files start to be created and accumulate from s1.rdata onwards. In this case, the calibration should end after the file s5184.rdata has been created. Your R window should be active as well; a total of 5184 Figures will be produced throughout the calibration process (as expected since ultimately we are conducting *mamo* runs), although they will not be saved.

BLOCK 3

CALIBRATE MAMO FOR A SINGLE SPECIES: THE ALTITUDINAL MIGRANT IIWI
(*'IIWI paper'*)

```
run_f.calibr = FALSE # CHANGE TO run_f.calibr = TRUE to launch the calibration
```

```
#---
```

```
if(run_f.calibr == TRUE) {
```

```
  f.calibr(
```

```
    # species
```

```
    sp = "IIWI", output.dir = "C:/Programs/MAMO/CALIBRATION/run/IIWI.1_1",
```

```
    y.obs = y.obs.IIWI.1,
```

```
    d = read.table("C:/Programs/MAMO/CALIBRATION/Starting  
parameters/param_calib.IIWI.1.txt", header = T, sep = "\t", dec = "."),
```

```
    # Spatial structure
```

```
    nr = 10, nc = 2, grad = c(1900, 1000), unit = 1,
```

```
    # Time frame
```

```
    T = 60, Tm = 5, SD.fledg = 0,
```

```
    # Survival (ad = annual, juv = from fledging to breeding age)
```

```
    n.s.ad = 3, input.direct.s.ad = TRUE, s.ad.direct = c(0.729, 0.78, 0.877), n.rat.s = 1,
```

```
    # Reproduction and habitat quality
```

```
    n.fec = 3, input.direct.fec = TRUE, fec.direct = c(3, 2.444, 1.5), n.fec.1 = 3, input.direct.fec.1 =  
    TRUE, fec.1.direct = c(3, 2.444, 1.5),
```

```

paired.s.ad.fec = TRUE, n.rat.f = 1, n.K.b = 3, K.nb.1 = list(K.nb.1.2003, K.nb.1.2004,
K.nb.1.avg), n.K.nb.1 = 3, K.nb.2 = K.nb.2,

reproduction.malaria = "simple",

# Malaria parameters (daily except Sm.ac)

alpha.b = alpha.b, alpha.nb.1 = alpha.nb.1, alpha.nb.2 = alpha.nb.2, alpha.1 = alpha.1, n.Sm.ac =
3, input.direct.Sm.ac = TRUE,

Sm.ac.direct = c(0.02, 0.07, 0.13),

# Movements

n.gamma.mov = 4, input.direct.gamma.mov = TRUE, gamma.mov.direct = c(-10, 0.159, 0.541,
10), calc.gamma.d = "fast.risky",

n.sim.disp = 10000, n.R.ter = 1, n.fidelity.ad = 2, n.m.natal = 2, n.psi.DD = 2,

# Other options

add.cline = FALSE,

# Simulations

n.sim = 1, design = "simple", batch = 1)

}

```

3.3 The 'sensitivity envelop'

3.3.1 The function *f.envp.single*

Once the calibration runs are completed, we need to identify the ones that best predict the observed data (the so-called 'sensitivity envelop'). This is the role of the function *f.envp.single*.

The function *f.envp.single* has seven arguments; most have been previously described for *mamo* or *f.calibr*, with the following exceptions:

- *output.dir* this time describes not where the results of the calibration will be stored, but where they *are* stored.
- *col* is the color used to plot the observed data (represented by crosses); simulated data is always shown in pink.
- *n.envp* is the number of best run the user wants to include in the envelop. Runs are ranked using a least-squares approach, wherein $fit \propto -\sum_{i=1}^{nr} (obs_i - sim_i)^2$.

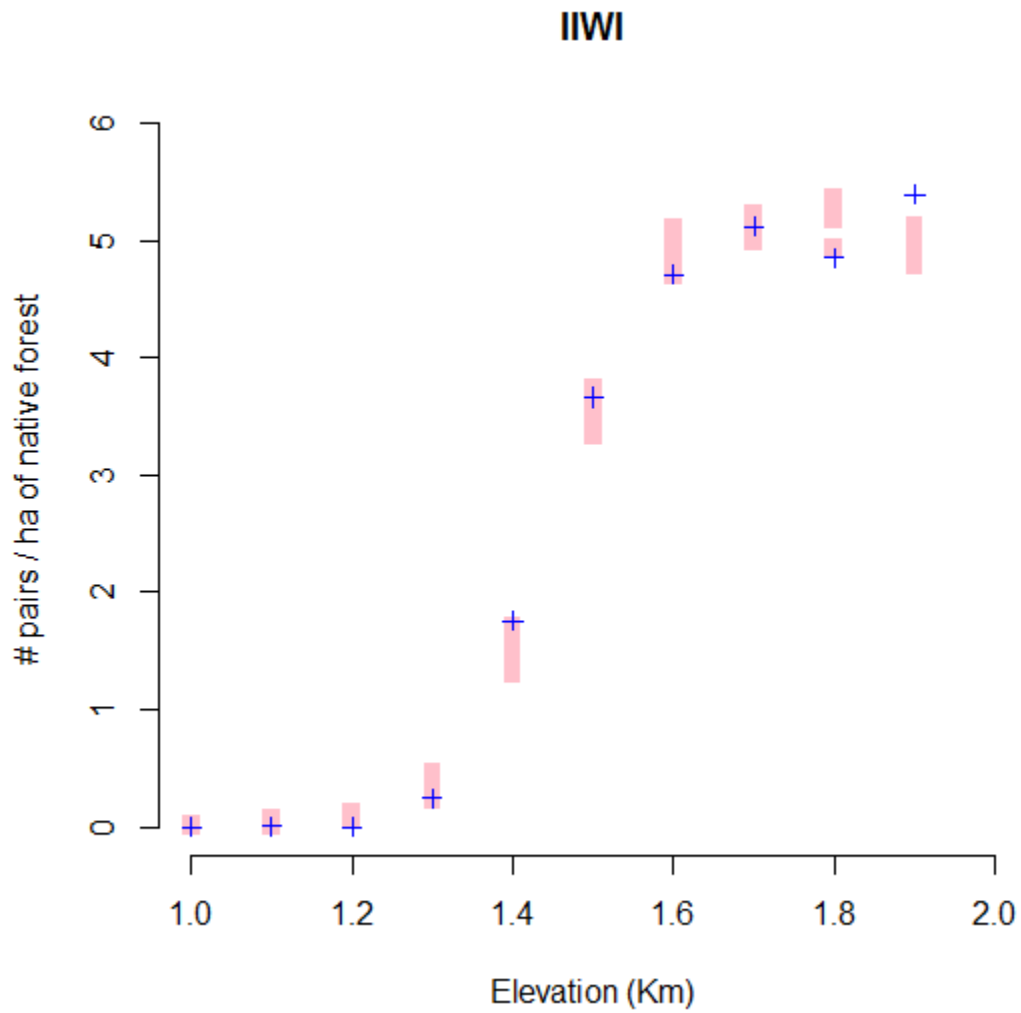
Running the function in the case of the 'IIWI paper' can be done as follows (code also found in BLOCK 4 of the SCRIPT file)

```
IIWI.1_1.calibr = f.envp.single(  
  
sp = "IIWI",  
  
output.dir = "C:/Programs/MAMO/CALIBRATION/run/IIWI.1_1",
```

```
y.obs = y.obs.IIWI.1, n.envp = 10, col = "blue",  
  
d = read.table("C:/Programs/MAMO/CALIBRATION/Starting  
parameters/param_calib.IIWI.1.txt", header = T, sep = "\t", dec = ".")  
  
)
```

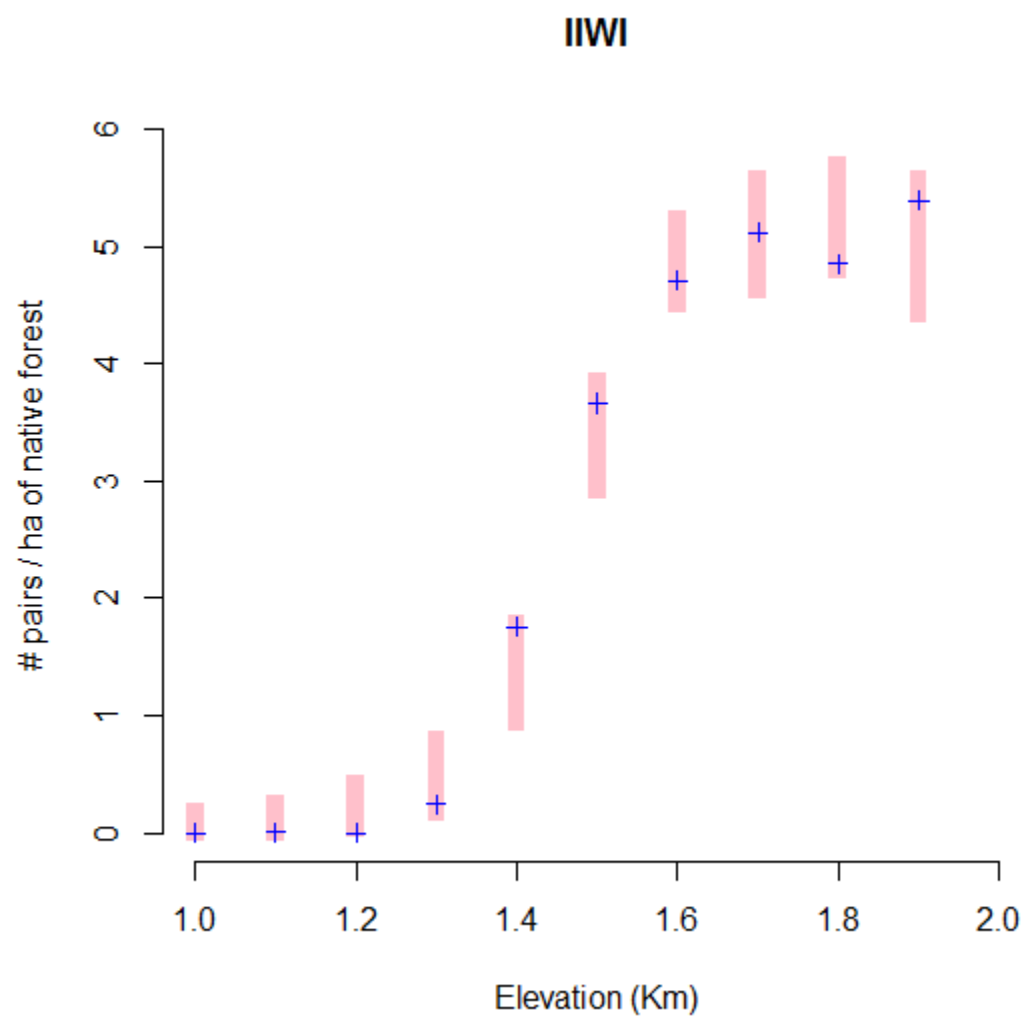
A Figure is automatically produced once we execute the function (See Fig. 3.3.1-1 below). Note once again that a slightly different Figure should be obtained if you run the calibration step from scratch, due to demographic stochasticity. Furthermore, the actual envelop will probably contain a different set of run, as we experienced ourselves by repeating several times the calibration run independently. This is to be expected.

Fig. 3.3.1-1



Of course, if you are increasing the number of runs to be included in the 'envelop', an increase of the envelop's variance is to be expected. This is shown in the Figure 3.3.1-2 below obtained by changing the argument *n.envp* from 10 to 100.

Fig. 3.3.1-2



3.3.2 Outputs of *f.envp.single*

In addition to producing a Figure that can help us visualize whether the calibration step produced a reasonable fit with observed data (see Fig. 3.3.1-1), the function *f.envp.single* returns a list of four variables stored in the object we created when running the function which, in this case, we called *IWI.1_1.calibr* (see above). Normally the user will not have to deal directly with these outputs, so we will only present them briefly here.

1) The variable *d.sim* essentially corresponds to the calibration file called *t.sim.txt*. It can be seen in R by using the *edit* function using the code:

```
edit(IWI.1_1.calibr$d.sim)
```

Normally, the data set should contain 5184 rows if the code of BLOCKS 1-4 has been used without modifications until now.

2) The variable *envp* contains the *n.envp* rows of *d.sim* that best predicted the observed data:

```
edit(IWI.1_1.calibr$envp)
```

3) The variable *stat.envp* contains information on the set of 13 variables that were potentially subjected to calibration:

```
edit(IWI.1_1.calibr$stat.envp)
```

For each of them, five statistics are reported:

- i) *mean.envp* is the mean value of the parameter in the envelop
- ii) *sd.envp* is the standard deviation of the parameter in the envelop
- iii) *min.envp* is the minimum value of the parameter in the envelop
- iv) *max.envp* is the maximum value of the parameter in the envelop
- v) *exp.mean* is the expected average value of the parameter in the whole calibration run

4) The output variable d is the modification of the input variable d accounting for *stat.envp*. Specifically, for each of the 13 variables subjected to calibration, if $\text{mean.envp} \geq \text{exp.mean}$, it implies that best-fitting runs tend to have a relatively high value for this parameter. If we are interested to run more calibration runs after the first round, we would be tempted to discard the low-end values of this parameter to concentrate in the regions where the match with reality is better. We therefore modify d such as the minimum value for the parameter now corresponds to *exp.mean*, while the maximum value does not change. Conversely, if $\text{mean.envp} < \text{exp.mean}$, we modify d such as the maximum value for the parameter now corresponds to *exp.mean*, while the minimum value does not change. The modified data set d can now be used as input for the function *f.calibr* to run yet another round of calibration, as will be described in another paper.

In this example, we are not interested in running a second round of calibration, so we can create the final 'sensitivity envelop' object called *envp.IIWI.1_1* using the code:

```
envp.IIWI.1_1 = IIWI.1_1.calibr$envp
```

Chapter 4

SIMULATION STUDY

4.1 The *f.run* function

Now that we have found a set of parameter conditions capable of replicating the current pattern of IWI distribution (our 'sensitivity envelop'), we can start asking questions concerning the impact of climate change or potential management actions. This is the role of the function *f.run*, to run a series of simulations covering the full spectrum of scenarios that we want to investigate while accounting for parameter uncertainty *via* the 'sensitivity envelop'.

The *f.run* function is akin to the *f.calibr* function. Both are calling *mamo* repeatedly via the ancillary *simul_mamo* function, and both store the results in a similar way: 1) a subset of *mamo* results are stored in a text file called *t.sim.txt* located in the *output.dir* folder created by the user, as described previously; 2) an R file containing the complete *mamo* output will be produced for each run, named *s1.1.rdata* for the first, *s2.1.rdata* for the second, etc. The argument *n.sim* of *f.run* allows the user to run each parameter combination more than once, in order to test the effect of demographic stochasticity (e.g., *n.sim* = 2). In that case, the files *s1.2.rdata*, *s2.2.rdata*, etc. also will be produced. Finally, another R file is produced synthesizing *mamo* outputs for the *n.sim* run of each parameter combination. In our example, they would be called *s1.rdata*, and *s2.rdata*, respectively. Like for the text file, only a subset of *mamo* output are stored in this latter R file. Double-clicking on any R file opens the R program. The results can be accessed by typing *x* in the console.

Unlike *f.calibr*, however, *f.run* will also yield a *de novo* text file called *factors.txt* containing for each run, the value of a number of variables characterizing the scenarios investigated. In the current version of MAMO, these variables are limited to the following:

1) 'RISK' for malaria transmission risk; in the current version of MAMO, five scenarios are investigated when the argument of *f.run* called *n.RISK* = 5. Actual transmission risk is expected to increase from *RISK* = 1 to 5, but we cannot exclude situations where this actually would not be the case (see below for details):

i) *RISK* = 1 ↔ malaria transmission risk is zero all year round at all elevations; simulate pre-malaria conditions.

ii) *RISK* = 2 ↔ malaria transmission risk corresponds to the present climatic conditions, but it is partly alleviated by management aiming at reducing transmission risk; such management is described by the *f.run* argument called *mg.RISK*; *mg.RISK* is a vector describing the effect of malaria reduction at each elevation. For instance, assuming *grad* = c(1900, 1000) and *nr* = 10, *mg.RISK* will be a vector of 10 elements starting by the effect of management at 1900 m. In our case, we want to simulate the effect of a 50% reduction of malaria transmission risk at elevations ≥ 1600 m. This is accomplished by choosing *mg.RISK* = c(rep(0.5, 4), rep(1, 6)).

iii) *RISK* = 3 ↔ corresponds to *RISK* = 2 but with no management.

iv) *RISK* = 4 ↔ corresponds to future, not present, climatic conditions and management action like for *RISK* = 2

v) *RISK* = 5 ↔ corresponds to *RISK* = 4 but without management action.

Note that if you pick say *n.RISK* = 3, only the scenarios i) to iii) will be explored; if *n.RISK* = 2, only the scenarios i) to ii), etc. In the current version of MAMO, it is not possible to choose any particular subset, or introduce a new scenario; of course, more flexibility could be allowed in future versions.

2) 'AC' which stands for acute malaria mortality, the probability of dying from the first, so-called 'acute', malaria infection. While the value of the *mamo* parameter *Sm.ac* is determined by the calibration process (in fact, we have *n.envp* different *Sm.ac* estimates, which can be identical or not), we cannot exclude that the species will evolve towards a greater malaria tolerance in the future. Accordingly, it is possible to test a number *n.AC* of alternative scenarios, where the current estimate of *Sm.ac* is multiplied by a factor called *evol.AC*. For instance, if *n.AC* = 2 and *evol.AC* = c(2, 1), the first *n.envp* run will be conducted with the *Sm.ac* estimate multiplied by 2, while the next *n.envp* run will be conducted with the *Sm.ac* estimate multiplied by 1. Because *Sm.ac* is the probability of surviving acute malaria infection, *evol.AC* = 2 corresponds to our 'guess' for the future value (after evolution), while *evol.AC* = 1 corresponds to the current estimate, or a scenario with no evolution of malaria tolerance in the future (due to lack of suitable genetic variation for instance). Note that by ranking the values in *evol.AC* in decreasing order, as we did here, we ensure that malaria mortality risk (our factor 'AC') increases from AC = 1 to *n.AC*. This choice is important to remember for correctly interpreting AC values in the analyses.

Accounting for each and every possible value in the 'sensitivity envelop' is the approach used by *f.run* to account for the uncertainty in the demographic estimates. Let's illustrate this for *Sm.ac*. For instance, if *n.envp* = 2 and *Sm.ac* (in the envelop) = c(0.07, 0.13), then four run will be conducted with *Sm.ac* = 0.14 and 0.26 first (corresponding to *evol.AC* = 2), followed by *Sm.ac* = 0.07 and 0.13 (corresponding to *evol.AC* = 1). Of course, different *Sm.ac* values traded off with different values of other demographic parameters to produce similar calibration results, so for each run, the appropriate 'envelop' value is used as well for these other parameters.

3) RAT.S and RAT.F are two columns of *factors.txt* reporting the effort on reducing rat predation on breeding females and juveniles, respectively. We assumed in the current version of MAMO that reducing rat predation would affect similarly the *mamo* parameters *rat.s* and *rat.f* so RAT.S always = RAT.F. The number of management actions concerning rat predation is

controlled by the parameter *n.RAT*. The actual management actions are indicated by *mg.RAT*. For instance, if we want to test two scenarios for a spatial grid characterized by *grad* = c(1900, 1000) and *nr* = 10, one scenario which is *status quo* and the other one where we reduce rat predation on adults and juveniles by 30 % at elevations ≥ 1600 m, we can write a list of two vectors as follows:

```
mg.RAT = list( rep(1, 10), c(rep(0.3, 4), rep(1, 6)) )
```

In the status quo scenario, the 'sensitivity envelop' values for *rat.s* and *rat.f* will be multiplied by 1 at all 10 elevations, while in the second scenario, a 70 % reduction will be applied for elevations ≥ 1600 m. Note that when the 'sensitivity envelop' value for *rat.s* or *rat.f* is zero, the management action will have no effect. In building *mg.RAT*, we recommend to always order the management actions from least (left) to most (right), so that *RAT.S* (and *RAT.F*) increase from *RAT.S* = 1 to *n.RAT*. For instance, if we wanted to test a third scenario where rat predation is reduced by 80 % at elevations ≥ 1400 m, we would write:

```
mg.RAT = list( rep(1, 10), c(rep(0.3, 4), rep(1, 6)), c(rep(0.2, 6), rep(1, 4)) )
```

The vector *rep(1, 10)* would be associated with *RAT.S* = *RAT.F* = 1 in *factors.txt*, the vector *c(rep(0.3, 4), rep(1, 6))* with *RAT.S* = *RAT.F* = 2, and the vector *c(rep(0.2, 6), rep(1, 4))* with *RAT.S* = *RAT.F* = 3.

4) *RES.1* and *RES.2* are two columns of *factors.txt* reporting management actions aimed at increasing the resources available to altitudinal migrants during the first (*K.nb.1*) and second (*K.nb.2*) period of the non-breeding season, respectively. Unlike *RAT.S* and *RAT.F*, a unique strategy can be applied to each season. Again, we recommend ordering the management actions from left (least) to right (most) so that *RES.1* and *RES.2* increase from *RES.1* = 1 to *n.K.NB.1* and from *RES.2* to *n.K.NB.2*, respectively. For instance, if we want to test 3 management scenarios for *nb.2* and no management action for *nb.1*, with a spatial grid characterized by *grad* = c(1900, 1000) and *nr* = 10, we can use the following values of *f.run* arguments:

```
n.K.NB.1 = 1
```

```
mg.K.NB.1 = rep(1, 10) # no management
```

```
n.K.NB.2 = 3
```

```
mg.K.NB.2 = list( rep(1, 10), c(rep(1.5, 3), rep(1, 7)), c(rep(2, 3), rep(1, 7)) )
```

Hence for nb.2, the second part of the non-breeding seasons, we have a *status quo* for scenario 1, resources will be multiplied by 1.5 at elevations ≥ 1700 m for scenario 2, and resources will be multiplied by 2 at elevations ≥ 1700 m for scenario 3.

5) ENVP and SIM are the last two columns of *factors.txt* reporting the rank of the 'sensitivity envelop' for the former (from 1 to *n.envp*, 1 being the parameter combination which produced the best fit with observed data), and the replicate number for each unique parameter combination (from 1 to *n.sim*) for the latter. These data will be useful to conduct subsequent statistical analyses.

Most of the other arguments of *f.run* are shared with the functions described previously, so they will not be detailed here, with the following exceptions:

- *envp* is a new argument that takes the data frame resulting from the calibration step and called 'sensitivity envelop'. In our ongoing example, we have `envp = envp.IIWI.1_1`.
- for the *K.nb.1* argument, it is currently required to use the default, i.e. `K.nb.1 = list(K.nb.1.2003, K.nb.1.2004, K.nb.1.avg)`
- we introduce here the equivalent for the future (2100) of the malaria transmission risk variables, i.e. *alpha.b.2100*, *alpha.nb.1.2100*, *alpha.nb.2.2100*, and *alpha.1.2100*, counterparts of *alpha.b*, *alpha.nb.1*, *alpha.nb.2*, and *alpha.1*, respectively.

We are now ready to run the simulation study of the 'IIWI paper" by using the following code (BLOCK 5 of the SCRIPT file reproduced below). Remember to change `run_f.run` from FALSE to TRUE otherwise the code will not be read (precaution against inadvertent launch of a run which would erase previously obtained results). As for the calibration step, remember to **wait**

until completion of all the simulation before opening the *t.sim.txt* file, otherwise run will fail. Changes in the sleep mode of the computer may be taken as the process took us 3 hours and 29 minutes on a DELL LATITUDE E6230 with a 2.8 GHz Intel Core 3rd Generation i5-3360M Processor. You can monitor the progress of the simulation study by looking into the IIWI_1 folder you created within the RUN folder of MAMO, as .R files start to be created and accumulate from s1.1.rdata onwards. In this case, the study should end after the file *factors.txt* has been created. Your R window should be active as well during the study; a total of 900 figures will be produced throughout the study, although they will not be saved ($n.RISK = 5 \times n.AC = 3 \times n.K.nb.2 = 3 \times n.sim = 2 \times n.envp = 10$ - the latter not an explicit *f.run* argument, but accessible to the function via the *envp* argument-).

#BLOCK 5

Run management scenarios based on the 'sensitivity envelop' for the 'IIWI paper'

run_f.run = FALSE # CHANGE TO run_f.run = TRUE to launch the simulation study

#---

if(run_f.run == TRUE) {

IIWI.1_1.run = f.run(

species-specific

sp = "IIWI",

envp = envp.IIWI.1_1,


```

output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",

y.obs = y.obs.IIWI.1,

d = read.table("C:/Programs/MAMO/CALIBRATION/Starting
parameters/param_calib.IIWI.1.txt", header = T, sep = "\t", dec = "."),

# Spatial structure

nr = 10, nc = 2, grad = c(1900, 1000), unit = 1,

# Time frame

T = 60, Tm = 5, SD.fledg = 0,

# Survival

n.RAT = 1,

mg.RAT = rep(1, 10),

# Reproduction and habitat quality

K.nb.1 = list(K.nb.1.2003, K.nb.1.2004, K.nb.1.avg),

n.K.NB.1 = 1,

mg.K.NB.1 = rep(1, 10), # no management

K.nb.2 = K.nb.2,

n.K.NB.2 = 3,

mg.K.NB.2 = list( rep(1, 10), c(rep(1.5, 3), rep(1, 7)), c(rep(2, 3), rep(1, 7)) ),

reproduction.malaria = "simple",

# Malaria parameters (daily except Sm.ac)

n.RISK = 5, mg.RISK = rep(0.5, 10),

```

```

alpha.b = alpha.b, alpha.nb.1 = alpha.nb.1, alpha.nb.2 = alpha.nb.2, alpha.1 = alpha.1,

alpha.b.2100 = alpha.b.2100, alpha.nb.1.2100 = alpha.nb.1.2100, alpha.nb.2.2100 =
alpha.nb.2.2100, alpha.1.2100 = alpha.1.2100,

n.AC = 3, evol.AC = c(3, 2, 1),

# Movements

calc.gamma.d = "fast.risky",

n.sim.disp = 10000,

# Other options

add.cline = FALSE,

# Simulations

n.sim = 2, design = "simple", batch = 1

)

}

```

4.2 Analysis of the simulation study

Now that the simulation run have been completed, it is time for exploring the results. In the following chapters, we will present a number of tools to visualize and analyze the results. The different blocks of code in the SCRIPT file allows the user to reproduce the figures of the 'IIWI paper'.

4.2.1 The *f.plot.univar* function

This function combines all or a fraction of individual *mamo* run corresponding to a simulation study into a single plot representing the density (number of pairs at the end of the simulation per hectare; based on the output *mamo* variable *np*, the patch-average elevation-specific number of pairs at the end of the simulation) as a function of the input *mamo* variable *grad* (patches elevation).

As usual, the argument *output.dir* tells the function where the data can be found. The next two arguments, *var.excl* and *val excl*, allow the user to exclude the fraction *val.excl* of the data corresponding to *var.excl*. An example will be given later. Next is a series of arguments concerning the plot characteristics. The argument *ylab* is self-explanatory (label for the y-axis); *col.obs* allows the user to choose the color for the non-simulated (observed) data *y.obs*, plotted together with the simulations if the argument *add.y.obs* = TRUE; *margin.up* allows the user to modify the range of the y-axis; several attempts may be needed to find the optimal value of *margin.up* that do not leave data outside the plot, and not too big an empty range on top. Ideally, this should be entirely automatic but the current version of MAMO is manual for technical reasons that will not be discussed here. So please be warned: make at least one attempt where *margin.up* is big (say > 2) to make sure you know where your data stop before picking a (smaller, including possibly negative) value of *margin.up* that produces a sharp figure. The argument *main* concerns the title of the plot, but there is a catch: if *main* = NA, no title is added to the plot; otherwise, the title will start with *main* but will also contain the values chosen for the fixed parameters of the plot. As its name implies, *f.plot.univar* only allows to explore the variation of a single variable. This variable is selected by the argument *wch.plot*; *col.cat* allows the selection of a specific color for each category of the variable selected; e.g., if *wch.plot* = "RISK", five categories (from 1 to 5) have been simulated (we used *n.RISK* = 5 in our

simulation study); the colors in *col.cat* will be attributed in that same order, i.e. *col.cat*[1] is the color ascribed to represent RISK = 1, etc. In addition, if *wch.plot* = "RISK", then the user has to select *val.RISK* = NA, while all other variables have to be fixed to a certain value; for instance, *val.AC* = 3, *val.RAT* = 1, *val.RES.1* = 1, *val.RES.2* = 1, which implies we take the simulation set with no evolution of malaria tolerance (remember, we used *evol.AC* = c(3,2,1) in our simulation study; *val.AC* = 3 therefore corresponds to *evol.AC* = 1, i.e. present value of malaria survival is not augmented - being multiplied by 1-); we also take the simulation set with no increase of resources during the second part of the non-breeding season (*val.RES.2* = 1 corresponds to *mg.K.NB.2*[[1]] in the simulation study, i.e. *rep*(1, 10)); because we did not explore management scenarios such as reduction of rat predation or increasing resources during the first part of the non-breeding seasons, the only value possible for the arguments *val.RAT* and *val.RES.1* is 1 (*n.RAT* = 1 and *n.K.NB.1* = 1 in our simulation study). Finally, the last set of *f.plot.univar* arguments concerns the legend which will be added to the Figure at coordinates *leg.x* and *leg.y*, with the text, R symbols, line type and colors defined by *leg.txt*, *leg.pch*, *leg.lty* and *leg.col*, respectively. Note that the legend must include information for observed data if *add.y.obs* = TRUE (default).

The Figure is produced automatically when running *f.plot.univar*. Individual squares are simulated data points, thin lines are sigmoid clines fitted for individual simulation run, and broad lines are sigmoid clines fitted for average population size at each elevation. As we mentioned in the chapter 2.1.8, the algorithm we are using (*nls* R function) may fail to estimate the parameters of the cline, for instance if density pattern is not sigmoidal, but also unpredictably when the pattern seems compatible with a sigmoid. To avoid an error action to be executed, which would prevent the Figure to be produced, we used the *warnonly* = TRUE option of *nls*, which will return a warning message in case of convergence failure. This way, even if convergence has not been reached, *nls* will provide estimate for the cline parameters. In the case of *f.plot.univar*, such an 'imperfect' cline is added to the plot, but a warning message will be produced. More flexible algorithms, such as generalized additive models, might be implemented in future MAMO versions to replace *nls*.

The function *f.plot.univar* also returns two output elements. First, *d.subset* is the subset of the data resulting from the combination of *t.sim.txt* and *factors.txt* that helped producing the Figure (but note that in the current version of MAMO, this data set is not sufficient; *f.plot.univar* also needs to access and use the R data files produced during the simulation study). Second, *f.plot.univar* also returns *test.mamo* which is the maximum value of the *mamo* output *test.disp.breed* generated during the simulation study; *test.mamo* should have a value close to zero (typically $\sim 10^{-5}$); if it is larger than say $\sim 10^{-3}$, it may be advisable to re-run the simulation study using `calc.gamma.d = "slow.robust"` (will take much longer!).

Below is reproduced the code from BLOCK 6 of the SCRIPT file as an example of *f.plot.univar* use, yielding Fig. 4.2.1-1 below (which is identical to Fig. 5 of the 'IIWI paper').

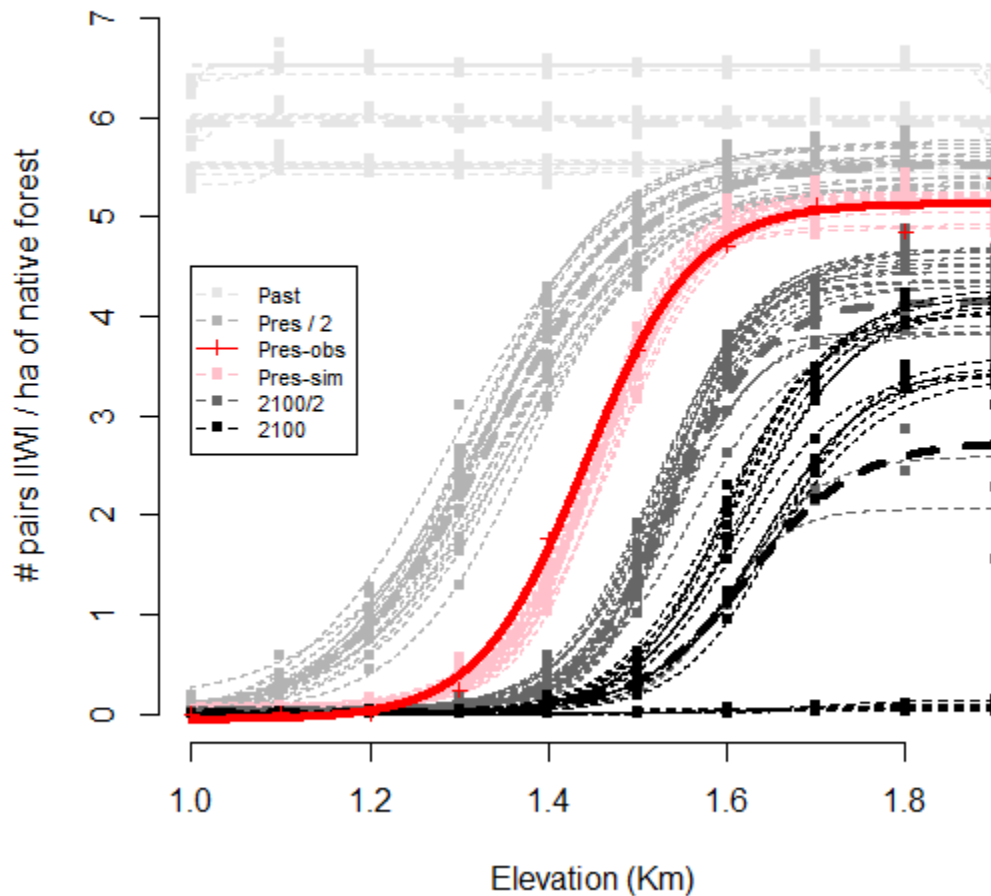
```
# BLOCK 6
```

```
# Fig. 5 of the 'IIWI paper'
```

```
f.plot.univar(
  output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",
  var.excl = NA, val.excl = NA,
  ylab = "# pairs IIWI / ha of native forest", main = NA, col.obs = "red", margin.up = 0,
  y.obs = y.obs.IIWI.1,
  wch.plot = "RISK", col.cat = c("grey90", "grey70", "pink", "grey40", "black"),
  val.RISK = NA, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,
  leg.x = 1, leg.y = 4.5, leg.text = c("Past", "Pres / 2", "Pres-obs", "Pres-sim", "2100/2", "2100"),
```

```
leg.pch = c(15,15,3,15,15,15), leg.lty = c(2,2,1,2,2,2), leg.col = c("grey90", "grey70", "red",
"pink", "grey40", "black")
)
```

Fig. 4.2.1-1

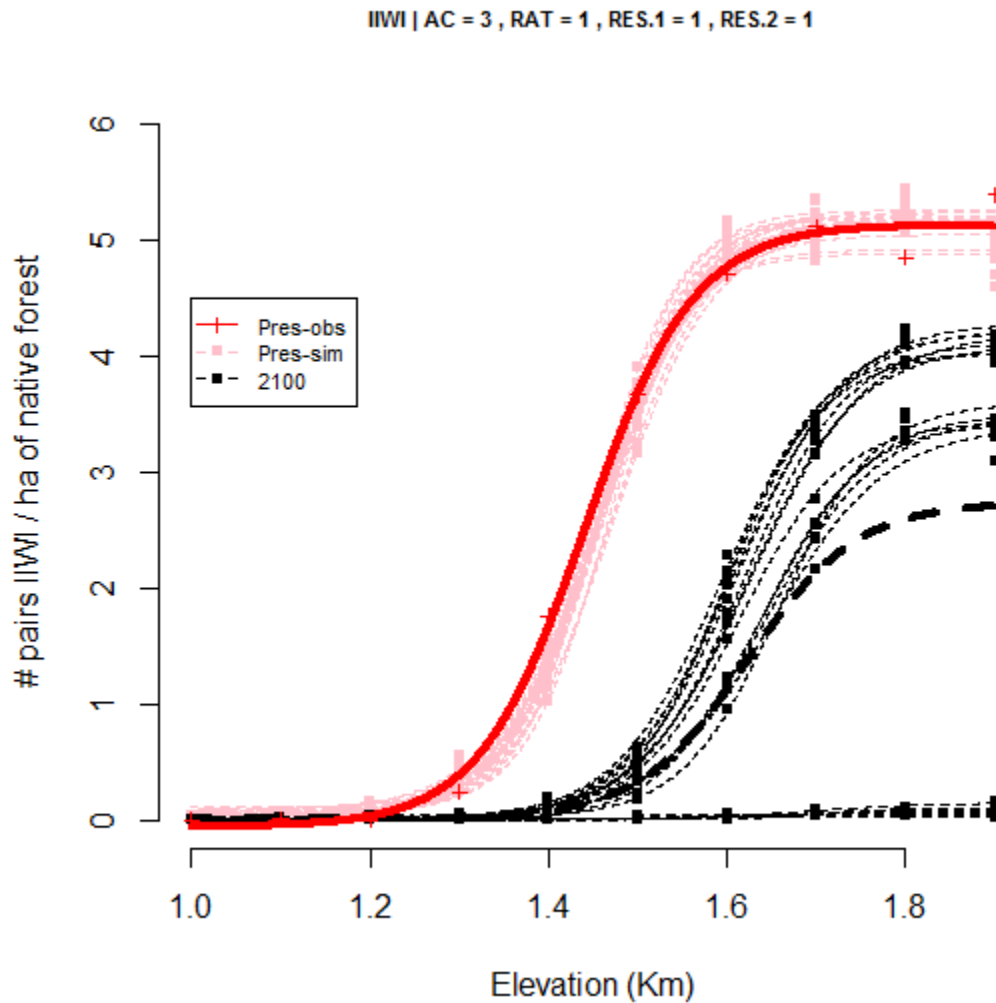


Below is another example of use of the function *f.plot.univar*. We want to basically reproduce Fig. 4.2.1-1, except we want to keep only the fraction of the simulation study corresponding to

present and future climatic conditions without management for reducing malaria transmission risk, corresponding to the variable RISK (of *factors.txt*) = 3 or 5. In addition, we change the argument *main* from NA to "IIWI"; accordingly a title is added which starts by 'IIWI' and displays our choice of parameter values for other, 'fixed' parameters. Here is the corresponding code, as modified from BLOCK 6:

```
f.plot.univar(  
  
  output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",  
  
  var.excl = rep("RISK", 3), val.excl = c(1, 2, 4),  
  
  ylab = "# pairs IIWI / ha of native forest", main = "IIWI", col.obs = "red", margin.up = 0,  
  
  y.obs = y.obs.IIWI.1,  
  
  add.y.obs = TRUE,  
  
  wch.plot = "RISK", col.cat = c("pink", "black"),  
  
  val.RISK = NA, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,  
  
  leg.x = 1, leg.y = 4.5, leg.text = c("Pres-obs", "Pres-sim", "2100"),  
  
  leg.pch = c(3,15,15), leg.lty = c(1,2,2), leg.col = c("red", "pink", "black")  
  
)
```

Fig. 4.2.1-2



4.2.2 The *f.plot.bivar* function

Like *f.plot.univar*, this function combines all or a fraction of individual *mamo* run corresponding to a simulation study into a single plot. Unlike *f.plot.univar*, however, *f.plot.bivar* is not limited to plotting the output *mamo* variable *np* (patch-average elevation-specific number of pairs at the end of the simulation) as a function of the input *mamo* variable *grad* (patches elevation). Instead,

it can plot any quantitative or semi-quantitative variable y (the argument for the response variable on the y -axis) as a function of not one but two explanatory factors, namely x (the argument for the explanatory variable on the x -axis) and the offset factor o . Note that for plotting purpose *f.plot.bivar* is essentially a wrapper of the function *raw.means.plot* of the R package *plotrix*. Accordingly, it plots both raw data in the background (open symbols) and factor means in the foreground (filled symbols) to provide a more accurate visualization of the underlying distribution.

Some *f.plot.bivar* arguments are identical to the ones of *f.plot.univar*, including *output.dir* (location of the data), *var.excl* (variable(s) to exclude) and its companion *val.excl* (which value(s) of the variable(s) to exclude). The arguments *val.RISK*, *val.AC*, *val.RAT*, *val.RES.1* and *val.RES.2* also have the same function; but this time since there are two explanatory factors, identified by the arguments x and o , the two 'value' arguments corresponding to x and o must be set to NA (e.g., if $x = \text{"AC"}$ and $o = \text{"RISK"}$, then *val.AC* = NA and *val.RISK* = NA). Unlike *f.plot.univar*, *f.plot.bivar* has an argument called *take.subset* (logical). If *take.subset* = TRUE (default) and the remaining 'value' arguments (in our example, *val.RAT*, *val.RES.1* and *val.RES.2*) are not set to NA, then a subset of the data containing only the chosen values will be kept and plotted (e.g., *val.RAT* = 1, *val.RES.1* = 1 and *val.RES.2* = 1, corresponding to no management action against rats and no increase of nectar resources). The argument *main* also has the same function as in *f.plot.univar*, i.e. no title will be printed when set to NA, and a title starting by *main* and indicating the values taken by the three factors different than x or o otherwise. Note that if *take.subset* = FALSE and *main* is not set to NA, the 3 non-explanatory factors will have to be set manually to NA otherwise the title (but not the plot) may be misleading. Finally, be aware that in the current version of MAMO (as mentioned earlier; see chapter 4.1 concerning the *f.run* function), there is no 'RAT' column/factor in the simulation study output; instead, two columns named *RAT.S* and *RAT.F* are produced. Therefore, the user has to choose either one of them as the x or o factor (e.g., $x = \text{"RAT.S"}$) in order to visualize the effect of rat management (currently, *RAT.S* and *RAT.F* are always identical as we assume that rat management does not affect differently adult and fledging survival). Future versions of

MAMO could achieve a simpler and more coherent design, for instance by merging permanently the columns RAT.S and RAT.F into a single 'RAT' column within '*factors.txt*'.

The remaining *f.plot.bivar* arguments are *ylab* and *xlab*, the labels on the y- and x-axis, respectively; *title.o* is the title given to the legend on the right of the Figure concerning the offset explanatory factor if the argument *legend.o* = TRUE (default); *l.pos* is a numeric vector of length 2 indicating the position of the legend; if not specified, it is automatically determined.

The outputs of *f.plot.bivar* are:

- 1) the plot produced automatically;
- 2) *d.subset*, the subset of the data resulting from the combination of *t.sim.txt* and *factors.txt* necessary and sufficient to produce the Figure;
- 3) *test.mamo* which is the maximum value of the *mamo* output *test.disp.breed* generated during the simulation study; *test.mamo* should have a value close to zero (typically $\sim 10^{-5}$); if it is larger than say $\sim 10^{-3}$, it may be advisable to re-run the simulation study using *calc.gamma.d* = "slow.robust" (will take much longer!);
- 4) a print on the console of the results of two mixed-effect linear models testing the influence of the explanatory factors *x* and *o* on the response factor *y* while accounting for the rank in the 'sensitivity envelop' and replicate number of each unique parameter combination (columns 'ENVP' and 'SIM' of *factors.txt*, respectively, treated as random factors - intercepts only-):
 - i) full model including interaction between *x* and *o*;
 - ii) additive model between *x* and *o* (no interaction)

As a first example of use, we reproduced below the part of BLOCK 7 that creates Fig. 6 of the 'IIWI paper' (= our Fig. 4.2.2-1 below). The rest of BLOCK 7 coding, which allows the user to extract some statistics used in the paper, such as the fraction of IIWI remaining as compared to the pre-malaria era, will not be discussed here as it does not relate to *f.plot.bivar*.

```

# BLOCK 7

# Fig. 6 of the 'IIWI paper'

z = f.plot.bivar(

output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",

var.excl = NA, val.excl = NA,

take.subset = TRUE, val.RISK = NA, val.AC = NA, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,

y = "np.metapop", ylab = "# pairs IIWI in metapopulation", main = NA,

x = "AC", xlab = "Malaria mortality",

o = "RISK", title.o = "RISK", legend.o = TRUE, l.pos = NA

)

# add observed data point

np.metapop.obs = sum(y.obs.IIWI.1* 100) * 2

# we multiply by 100 y.obs.IIWI.1 to get # pairs / Km2 at each elevation (from # pairs / ha)

# because unit = 1 and 1 Km2 covers ~ 100 m in elevation (see IIWI paper), sum gives the total

#number of pairs along the gradient ( 1 single column)

# because nc = 2, we multiply by 2 to obtain the observed metapopulation abundance

points(3, np.metapop.obs, col = "red", pch = 3, cex = 1.8, lwd = 2)

```

```
# differentiate the different categories of the parameter gamma.mov in the future

k = 0.2

di = z$d.subset

di = di[di$AC == 3 & di$RISK == 5,]

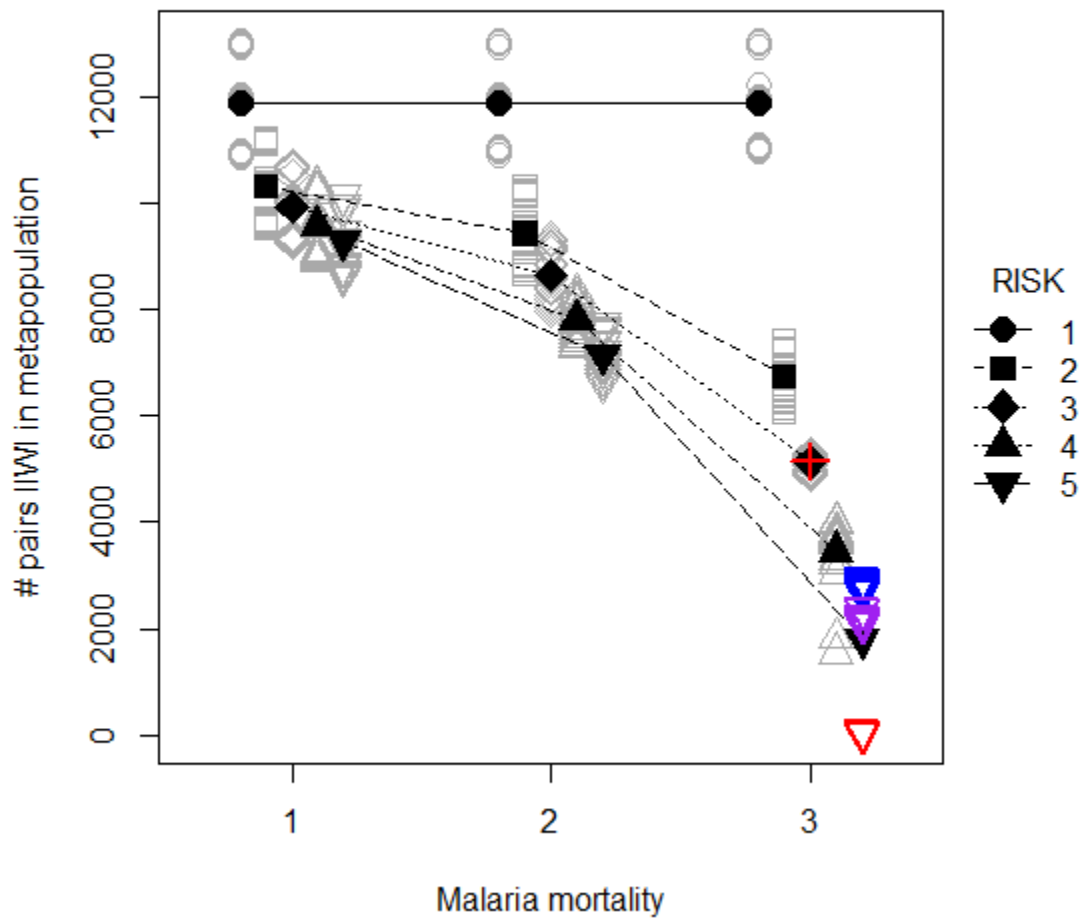
di.m1 = di[di$gamma.mov == 0.159,]; di.m2 = di[di$gamma.mov == 0.541,]; di.m3 =
di[di$gamma.mov == 10,]

points(di.m1$AC+k, pch = 25, cex = 1.8, di.m1$np.metapop, col = "blue", lwd = 2)

points(di.m2$AC+k, pch = 25, cex = 1.8, di.m2$np.metapop, col = "purple", lwd = 2)

points(di.m3$AC+k, pch = 25, cex = 1.8, di.m3$np.metapop, col = "red", lwd = 2)
```

Fig. 4.2.2-1



As our second example of use, we reproduced below the part of BLOCK 8 that creates Fig. 7 of the 'IIWI paper' (= our Fig. 4.2.2-2 below). The rest of BLOCK 8 coding, which allows the user to extract some statistics such as the fraction of IIWI remaining as compared to the pre-malaria era, will not be discussed here as it does not relate to *f.plot.bivar*.

```
# BLOCK 8
```

```
# Fig. 7 of the 'IIWI paper'
```

```
layout(matrix(c(1:3), 3, 1)); par(mar = c(3.6,3.6,0,0)+0.5, cex.main = 1)
```

```
i = f.plot.bivar(
```

```
output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",
```

```
var.excl = NA, val.excl = NA,
```

```
take.subset = TRUE, val.RISK = NA, val.AC = NA, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,
```

```
y = "np.high", ylab = "# pairs IIWI - 1800m", main = NA,
```

```
x = "AC", xlab = "",
```

```
o = "RISK", title.o = "RISK", legend.o = TRUE, l.pos = NA
```

```
)
```

```
# add observed data point
```

```
np.1800.obs = y.obs.IIWI.1[2] * 100
```

```
points(3, np.1800.obs, col = "red", pch = 3, cex = 1.8, lwd = 2)
```

```
# differentiate the different categories of the parameter gamma.mov in the future
```

```
k = 0.2
```

```
di = i$d.subset
```

```

di = di[di$AC == 3 & di$RISK == 5,]

di.m1 = di[di$gamma.mov == 0.159,]; di.m2 = di[di$gamma.mov == 0.541,]; di.m3 =
di[di$gamma.mov == 10,]

points(di.m1$AC+k, pch = 25, cex = 1.8, di.m1$np.high, col = "blue", lwd = 2)

points(di.m2$AC+k, pch = 25, cex = 1.8, di.m2$np.high, col = "purple", lwd = 2)

points(di.m3$AC+k, pch = 25, cex = 1.8, di.m3$np.high, col = "red", lwd = 2)

#---

j = f.plot.bivar(

output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",

var.excl = NA, val.excl = NA,

take.subset = TRUE, val.RISK = NA, val.AC = NA, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,

y = "np.mid", ylab = "# pairs IIWI - 1500 m", main = NA,

x = "AC", xlab = "",

o = "RISK", title.o = "RISK", legend.o = TRUE, l.pos = NA

)

# add observed data point

np.1500.obs = y.obs.IIWI.1[5] * 100

points(3, np.1500.obs, col = "red", pch = 3, cex = 1.8, lwd = 2)

```

```
#---
```

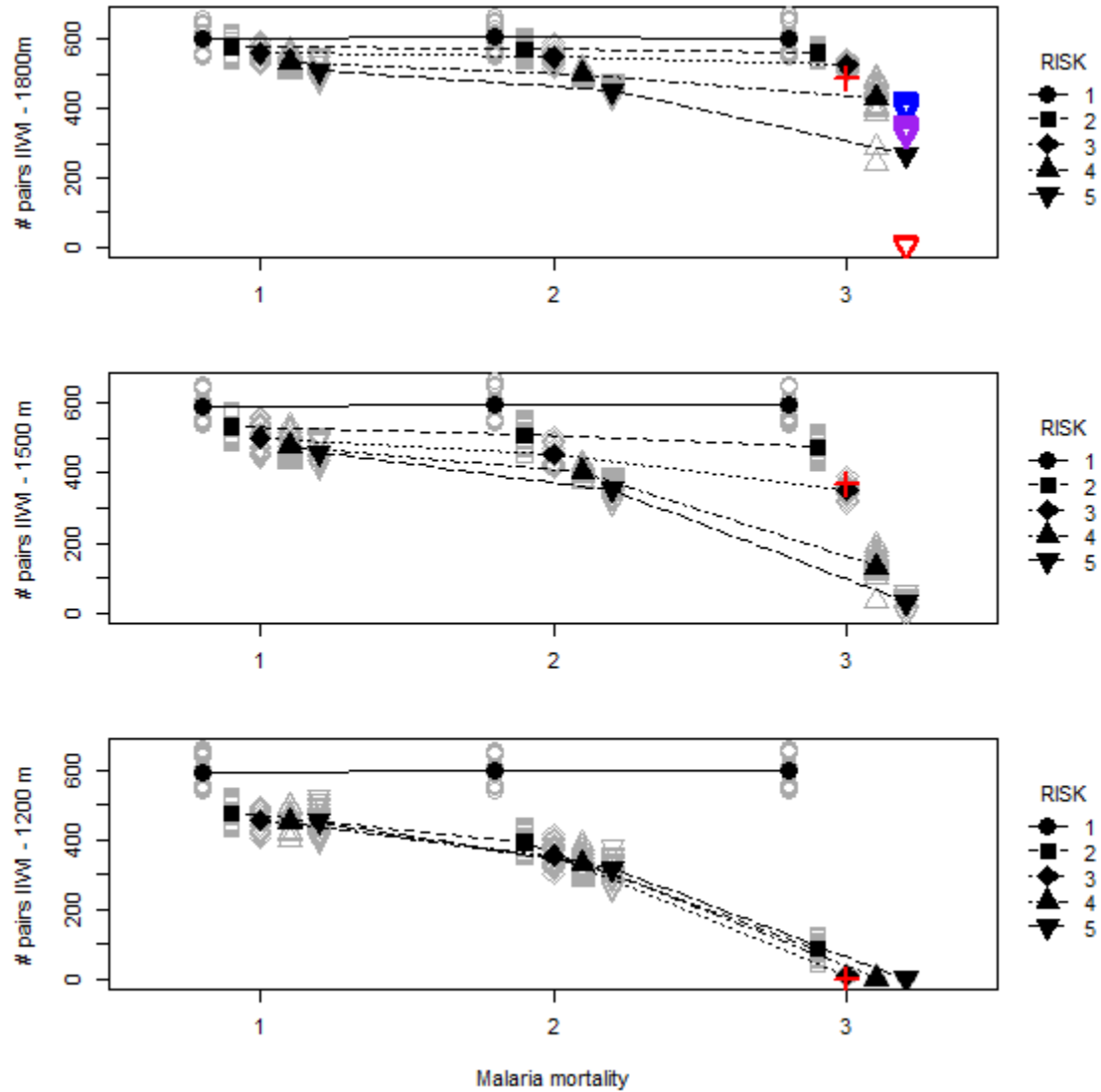
```
k = f.plot.bivar(  
  output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",  
  var.excl = NA, val.excl = NA,  
  take.subset = TRUE, val.RISK = NA, val.AC = NA, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,  
  y = "np.low", ylab = "# pairs IIWI - 1200 m", main = NA,  
  x = "AC", xlab = "Malaria mortality",  
  o = "RISK", title.o = "RISK", legend.o = TRUE  
)
```

```
# add observed data point
```

```
np.1200.obs = y.obs.IIWI.1[8] * 100
```

```
points(3, np.1200.obs, col = "red", pch = 3, cex = 1.8, lwd = 2)
```


Fig. 4.2.2-2



4.2.3 The *f.plot.composite* function

This function is related to (and modified from) *f.plot.univar*; like this function, *f.plot.composite* aims at combining all or a fraction of individual *mamo* run corresponding to a simulation study

into a single plot representing the density (number of pairs at the end of the simulation per hectare) as a function of elevation. Two main differences are:

- 1) unlike *f.plot.univar*, *f.plot.composite* only plots the sigmoid clines fitted for average population size at each elevation; simulated data points and sigmoid clines fitted for individual simulation run are not shown. Letting go of the variability and uncertainty allows the user to focus on the expected effect size of a given management action;
- 2) *f.plot.composite* has a new argument called *create.plot* (logical). If *create.plot* = TRUE, the figure is created from scratch; but if *create.plot* = FALSE, only the clines are added to a pre-existing plot. Hence, while *f.plot.univar* only allows to explore the variation of a single variable, it is possible to explore two or more variables by repeatedly calling *f.plot.composite*, the first time with the argument *create.plot* = TRUE, and thereafter with the argument *create.plot* = FALSE (see example below).

Most of the remaining arguments are identical between *f.plot.univar* and *f.plot.composite*, including *output.dir*, *var.excl*, *val.excl*, *ylab*, *main*, *margin.up*, *val.RISK*, *val.AC*, *val.RAT*, *val.RES.1*, *val.RES.2*, *leg.x*, *leg.y*, *leg.text*, *leg.pch*, *leg.lty*, and *leg.col*.

Because *f.plot.composite* focuses on comparative simulation results, it does not allow plotting the observed data so the following arguments have been discarded: *add.y.obs*, *col.obs*, *y.obs*.

Finally, four more arguments are specific to *f.plot.composite*:

- 1) if *fig.title* is not set to NA, then the figure title is *fig.title*; otherwise, it will be defined by *main* (see *main* description in *f.plot.univar* for specifics); hence, *fig.title* can override *main*;
- 2) *add.cline.cat* is a vector telling *f.plot.composite* whether it should fit a cline for a specific value of the argument *wch.plot*; for instance, let's assume *wch.plot* = "RISK" with five categories (1 to 5); if *add.cline.cat* = c("FALSE", rep("TRUE", 4)), we tell *f.plot.composite* to fit a cline when *RISK* > 1, but not when *RISK* = 1; it may make sense because in the latter case we simulated pre-malaria condition; assuming like we did a spatially constant carrying capacity, no spatial cline is to be expected. Hence, when *add.cline.cat* = FALSE, instead of a cline

f.plot.composite adds an horizontal line corresponding to the mean value of the response variable;

3) *add.leg* allows the user to decide whether the legend should be added to the plot;

4) *lty.mean* is the line type to be used for the clines (same value as the R *lty* parameter: 1 = solid, 2 = dashed, 3 = dotted, etc.)

As an example, we may want to explore the consequence of increasing nectar resources during the second part of the non-breeding season (coded by *wch.plot* = "RES.2" and *val.RES.2* = NA), but not only at a given, fixed time period (as would be the case with *f.plot.univar*), but comparatively at different levels of malaria transmission risk:

i) in the present (without management): *val.RISK* = 3; first calling of *f.plot.composite*

ii) in 2100 but after reducing the risk of malaria transmission risk by 50 %, so-called '2100/2': *val.RISK* = 4; second calling of *f.plot.composite*

iii) in 2100 but without management: *val.RISK* = 5; third calling of *f.plot.composite*

The code producing this Figure (Fig. 4.2.3-1 = Fig. 8 of the 'IIWI paper') is given in BLOCK 9 of the SCRIPT file and reproduced below. Since *wch.plot* = "RES.2", each time *f.plot.composite* is called, 3 different levels of resource management are plotted, namely no increase of nectar (control), a 50% increase at high elevations (1700 to 1900 m) or a 100 % increase at the same high elevations.

```
# BLOCK 9
```

```
# Fig. 8 of the 'IIWI paper'
```

```
f.plot.composite(  
  
  output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",  
  
  ylab = "# pairs IIWI / ha of native forest", fig.title = NA, main = NA, margin.up = -0.5,  
  
  wch.plot = "RES.2", col.cat = c("black", "grey70", "grey90"), add.cline.cat = rep("TRUE", 3),  
  
  val.RISK = 3, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = NA,  
  
  add.leg = TRUE,  
  
  leg.x = 1, leg.y = 5,  
  
  leg.text = c("Present", "Present (RES×1.5 | high)", "Present (RES×2 | high)", "2100/2", "2100/2  
(RES×1.5 | high)", "2100/2 (RES×2 | high)", "2100", "2100 (RES×1.5 | high)", "2100 (RES×2 |  
high)"),  
  
  leg.pch = NA, leg.lty = c(rep(1,3), rep(2,3),rep(3,3)), leg.col = c(rep(c("black", "grey70",  
"grey90"), 3)),  
  
  create.plot = TRUE, lty.mean = 1  
  
)
```

```
f.plot.composite(  
  
  output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",  
  
  ylab = "# pairs IIWI / ha of native forest", fig.title = NA, main = NA, margin.up = -0.5,  
  
  wch.plot = "RES.2", col.cat = c("black", "grey70", "grey90"), add.cline.cat = rep("TRUE", 3),
```

```

val.RISK = 4, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = NA,

add.leg = FALSE,

create.plot = FALSE, lty.mean = 2

)

f.plot.composite(

output.dir = "C:/Programs/MAMO/RUN/IIWI.1_1",

ylab = "# pairs IIWI / ha of native forest", fig.title = NA, main = NA, margin.up = -0.5,

wch.plot = "RES.2", col.cat = c("black", "grey70", "grey90"), add.cline.cat = rep("TRUE", 3),

val.RISK = 5, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = NA,

add.leg = FALSE,

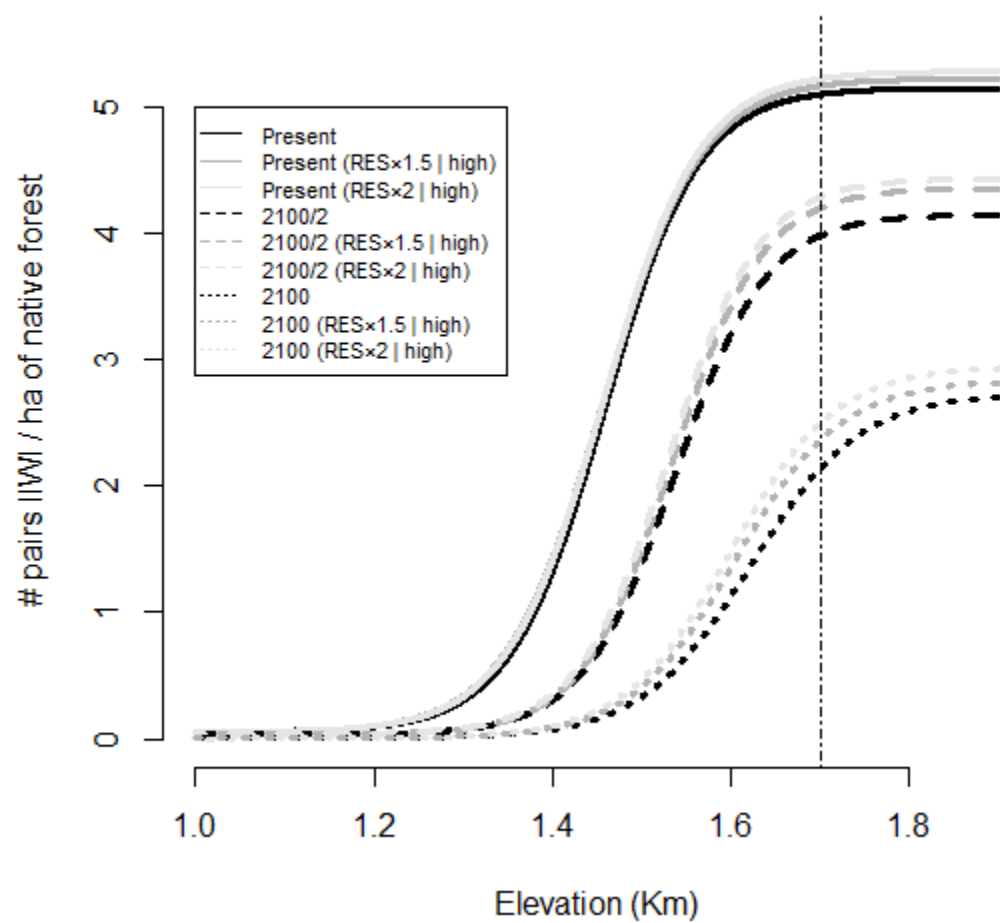
create.plot = FALSE, lty.mean = 3

)

abline(v = 1.7, lty = 4)

```

Fig. 4.2.3-1



Literature cited

Frederiksen, M., J. -D. Lebreton, and T. Bregnballe. 2001. The interplay between culling and density-dependence in the great cormorant: a modelling approach. *Journal of Applied Ecology* 38: 617-627.

Gardali, T., D. C. Barton, J. D. White, and G. R. Geupel. 2003. Juvenile and adult survival of Swainson's Thrush (*Catharus ustulatus*) in coastal California: Annual estimates using capture-recapture analyses. *Auk* 120: 1188–1194.

Gilroy J. J., T. Virzi, R. L. Boulton and J. L. Lockwood. 2012. A new approach to the “apparent survival” problem: estimating true survival rates from mark–recapture studies. *Ecology* 93: 1509–1516.

Sanchez, G. (2013) PLS Path Modeling with R. Trowchez Editions. Berkeley, 2013.
<http://www.gastonsanchez.com/PLS Path Modeling with R.pdf>

Woodworth, B.L. and Pratt, T.K. 2009. Life history and demography of Hawaiian forest birds. - In: Pratt, T.K., Atkinson, C.T., Banko, P.C., Jacobi, J. and Woodworth, B.L. (eds). *The Conservation Biology of Hawaiian Forest Birds: Implications for Island Avifauna*, Yale University Press, New Haven, pp. 194-233.

VanderWerf, E. 2004. Demography of Hawaii Elepaio: variation with habitat disturbance and population density. *Ecology* 85: 770-783.