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 *mamo* ……………..……………………...

2.2.1 First use …………………………………………...

2.2.2 Subsequent use ……………………………………...

2.3 *mamo* output ……………………...……………….

Chapter 3 …………………………….……………...

CALIBRATING MAMO

3.1 The *f.calibr* function ……………………………......

3.2 Running *f.calibr* ……………………...…………….

3.3 The 'sensitivity envelop' …………………….............

3.3.1 The function *f.envp.single* ……………………...............

3.3.2 Outputs of *f.envp.single* ……………………..................

Chapter 4 ……………………...…………………….

SIMULATION STUDY

4.1 The *f.run* function ……………………...…………..

4.2 Analysis of the simulation study……………………...

4.2.1 The *f.plot.univar* function ……………………...………

4.2.2 The *f.plot.bivar* function …………………….................

4.2.3 The *f.plot.composite* function ……………………...……

Chapter 5 ……………………...…………………….

MAMO AT THE COMMUNITY LEVEL

5.1 Calibration ……………………...…………………

5.1.1 First calibration round ……………………...…………

5.1.2 'Sensitivity envelop' corresponding to the first calibration round ...

5.1.3 Second calibration round ……………………...……….

5.1.4 Obtaining the final 'sensitivity envelop' using *f.envp.double* ……

5.2 Simulation study ……………………...……………

5.3 Create data set for community-level analysis ………….

5.3.1 First step: species-specific data sets ……………………....

5.3.2 Second step: community-level data set ……………………

5.4. Statistical analysis ……………………...………….

5.4.1 Mixed-effect linear models ……………………...……...

5.4.2 Effect size ……………………...…………………..

5.4.3 Graphical presentation of results …………………….......

5.5. Complementary species-specific analyses ……………

5.5.1 Using principal component analyses (PCA) ………………...

5.5.2 Using *f.plot.composite* ……………………...…………

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 mamo (*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 ‘I‘iwi (*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.

Finally, in chapter 5 we will use the techniques presented earlier for a single species (IIWI) and apply them to the 8 native forest bird species of Hakalau Forest National Wildlife Refuge, namely 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). 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 Hawaiʻi.

Because the current version of MAMO assumes that interspecific competition has no significant impact on the species in the community, the calibration and run procedures are species-specific and therefore follow exactly the same logic as previously, although we will introduce a two-step calibration procedure aiming at reducing computation time.

Hence, simulation studies for each species can be analyzed independently, following the approach described in chapter 4, and we will give here a few examples of application of previously used techniques. However, a species-specific approach becomes unpractical when the number of species and management scenarios increase, as different species respond differently to each management action. This is why we also developed an integrative approach that allows users to estimate community-level (rather than species-specific) response to climate change or management practices. The weight of each species in the community-level average is defined by the user, and can for instance reflect conservation status.

Additional information on the project, including the mathematical underpinning of MAMO, are described in two scientific papers that we will frequently refer to in this manual, and called the 'IIWI paper' (corresponding to chapters 2-4) and the 'HAKALAU paper' (chapter 5). A link to these two papers will be included in future versions of MAMO as soon as they have been accepted for publication.

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/~spector/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

***nr***

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

Value: N+ = {1, 2, ...}.

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: N+ = {1, 2, ...}.

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:

***unit***

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* × *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 = 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: N+ = {1, 2, ...}.

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).

***Tm***

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

Value: N+ = {1, 2, ...}.

Ex: T = 5

Caution:

1) Tm must be ≤ T. If Tm = T, all simulation are run without malaria (and rat predation).

2) We recommend choosing *Tm* such as:

i) Tm ≥ 2 (at least)

ii) T - Tm > 50

***t.b***

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

Value:

Range = [1, 365] ( 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.

***f.nb.1***

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).

***min.fledg***

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

Value:

Range = [1, *t.b*] ( N+) if reproduction.malaria = "simple"

Ex: min.fledg = 100

Caution:

1) If reproduction.malaria = "complex", Range = [1, *t.b*-14] ( N+); make sure min.fledg < peak.fledg × t.b < t.b - 14

***peak.fledg***

Definition: Peak of fledging time

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

Value:

Range = ]0,1[

Ex: peak.fledg = 2/3

Caution:

***SD.fledg***

Definition: standard deviation around *peak.fledg*

Value:

(0 and positive values)

Ex: SD.fledg = 20

Caution:

1) When reproduction.malaria = "complex" and SD.fledg ≠ 0, the fledging time is calculated based on a truncated normal distribution with min = min.fledg, mean = peak.fledg × 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 × 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 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.

***init.2***

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

Value: init.2 N+

Ex: init.2 = 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.2 = 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.

2.1.4 Arguments of the 'Survival' class

***s.ad***

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).

***rat.s***

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 ≤ 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).

***s.juv***

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

***fec***

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

Value: *fec* R+

Ex: fec = 1.54

Caution:

1) During the calibration process, we assume that fec.1 ≤ 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.

***fec.1***

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

Value: *fec.1* R+

Ex: fec.1 = 0.49

Caution:

1) During the calibration process, we assume that fec.1 ≤ 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.

***rat.f***

Definition: proportion of juveniles killed by rats

Value:

Ex: rat.f = 0.07.

Caution:

1) We assume that rat.s ≤ 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).

***K.b***

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

Value: K.b 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 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 ≤ 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 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 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).

***alpha.nb.2***

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).

***Sm.ac***

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

***gamma.mov***

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*, whichis 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 *fidelilty.ad* (for adults), or *m.natal*, *SD.natal* and *psi.DD* (for juveniles).

Value: *n.sim.disp* N+

Ex: n.sim.disp = 10000

Caution:

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

***R.ter***

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

Value:

Range = R.ter 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 'IIWI' and 'HAKALAU' papers for details). Density-dependence could be accounted for in future versions.

***fidelity.ad***

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 ) chosen such as E(X) = *m.natal* and SD(X) = *SD.natal* using the formulas: and .

Value: *m.natal*  (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 ) chosen such as E(X) = *m.natal* and SD(X) = *SD.natal* using the formulas: and .

Value: *SD.natal* (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, ∞)

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:

, 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 mamo function

source("m4\_mamo.r")

7) Run mamo

# 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 = 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

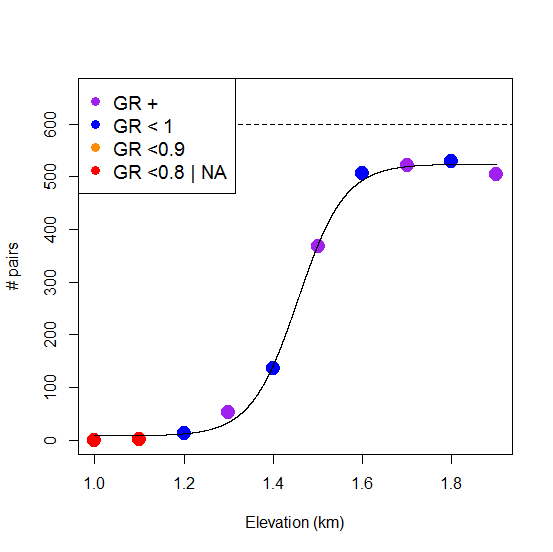
gamma.mov = 0.541, calc.gamma.d = "fast.risky", n.sim.disp = 10000, R.ter = 0.0234, fidelity.ad = 0.95, m.natal = 0.3, SD.natal = 0.3, psi.DD = 1,

# 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 (as averaged over the last three years of simulation, is ≥ 1, blue when 0.9 ≤ GR < 1, orange when 0.8 ≤ GR < 0.9, and red when GR < 0.8 or GR = 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("m14\_f.data.community.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).

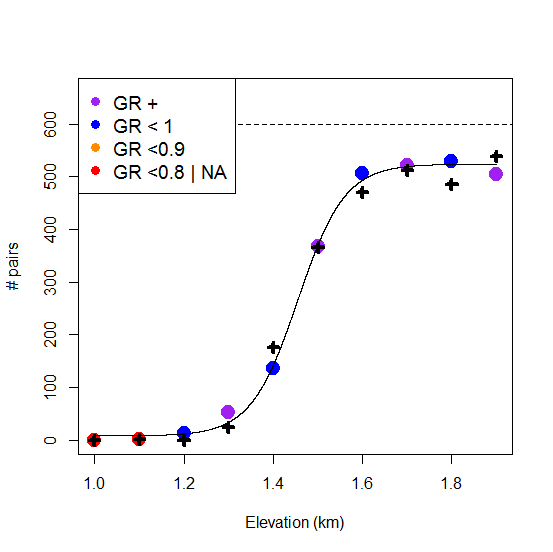
3) Run mamo

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 IIWI data (y.obs.IIWI.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.IIWI.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)

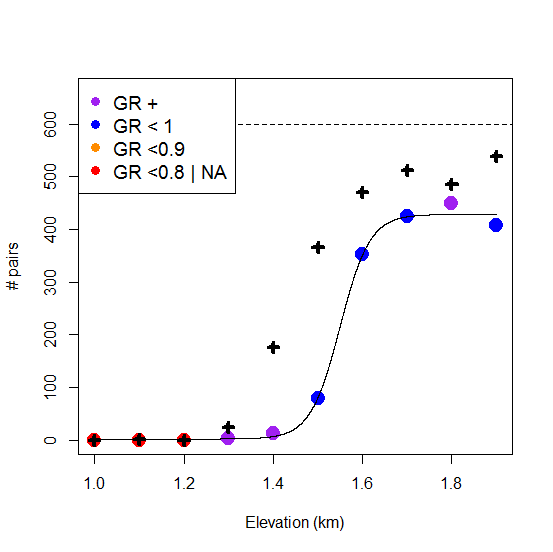
# see output

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 seq(grad[1], grad[2], length = 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.ad1/365), while *s.juv.d* is calculated over a period of (365 - *peak.fledg* × *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 grad[1] ≥ 1.9 and grad[nr] ≤ 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 grad[1] ≥ 1.9 and grad[nr] ≤ 1.1. Otherwise, NA is returned.

*gamma.d.breed* is the estimate of the hidden parameter *gamma.d* for the (nr×nc)th patch. Hence *gamma.d.breed* is the parameter of a negative exponential distribution estimated in order to distribute *fraction.disp* leavingthe (nr×nc)th patch into the remaining nr×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 ~ 10-9-10-5) if calc.gamma.d = "fast.risky" has been chosen.** If any *test.disp.breed* value in the simulation set islarger 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 g*ives 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.1*, *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.1[,,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

For illustration purposes, a sigmoid curve was fitted to *np* values along the elevational gradient *grad* using the formula: , 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 (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: , 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, and. 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/IIWI.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 the folder C:/Programs/MAMO/CALIBRATION/run/aaa\_empty table. Hence, the simplest (and recommended) way to create the output folder of the calibration run is to copy the folder 'aaa\_empty table', paste it, and rename it. In our example, since we want to name the output folder 'IIWI.1\_1' (as output.dir = "C:/Programs/MAMO/CALIBRATION/run/IIWI.1\_1"), we would change the folder name 'aaa\_empty table - Copy' by 'IIWI.1\_1'.

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.

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.1*, *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 Wildrife 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*), ‘I‘iwi, 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.sad × 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 C3 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. Changes in the sleep mode of the computer should be taken** 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.* Another function called *f.envp.double* can be used to combine two calibration runs into a single one, but this will be presented in chapter 5.

The function *f.envp.single* has seven arguments, and the majority have previously been 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 .

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",

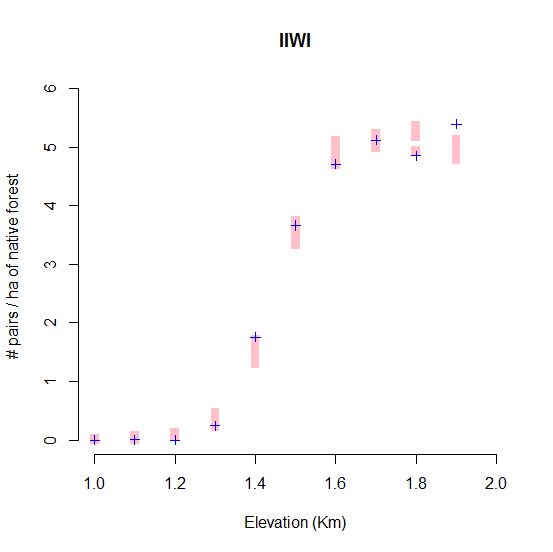
y.obs = y.obs.IIWI.1, n.envp = 10, col = "blue",

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

)

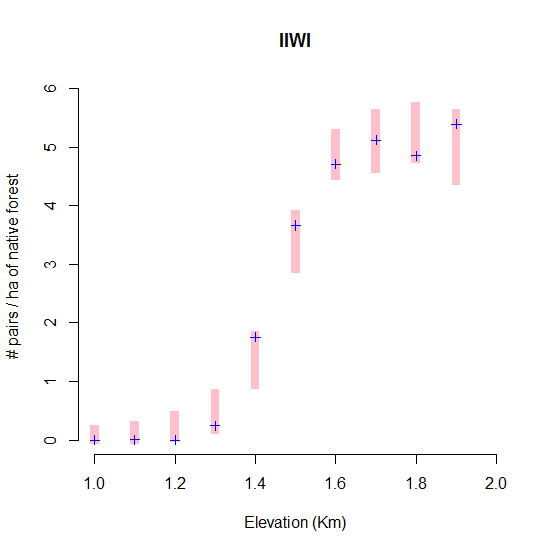
A Figure is automatically produced once we execute the function (See Fig. 3.3.1-1 below). Note once again that a slightly different Figure should be obtained if you run the calibration step from scratch, due to demographic stochasticity. Furthermore, the actual envelop will probably contain a different 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 IIWI.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* essentiallycorresponds to the calibration file called *t.sim.txt*. It can be seen in R by using the *edit* function using the code:

edit(IIWI.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(IIWI.1\_1.calibr$envp)

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

edit(IIWI.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 mean.envp ≥ 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 mean.envp < 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 we will see in chapter 5.

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 IIWI 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; the simplest way to create the output folder is to copy the folder named 'aaa\_empty table' located in the RUN folder of MAMO, paste it, and rename it by a name of your choice. It contains the required empty *t.sim.txt* file; 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 × n.AC = 3 × n.K.nb.2 = 3 × n.sim = 2 × 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 with 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 havea value close to zero (typically ~ 10-5); if itislarger than say ~ 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. 4 of the 'IIWI paper').

# BLOCK 6

# Fig. 4 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"),

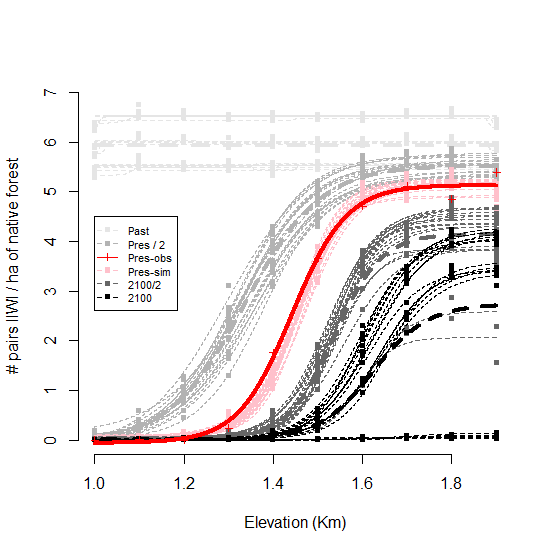
val.RISK = NA, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,

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"),

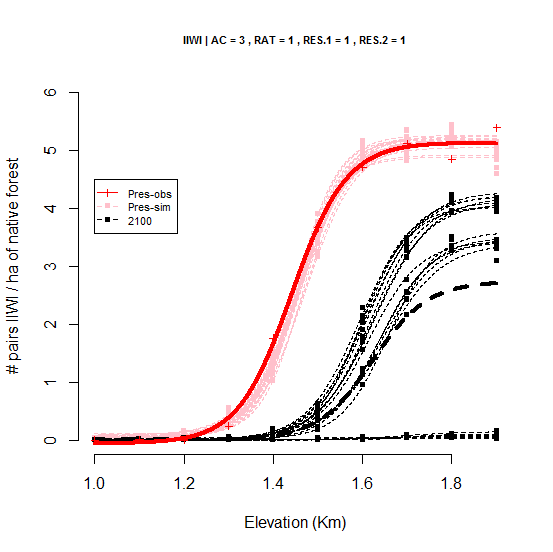
val.RISK = NA, val.AC = 3, val.RAT = 1, val.RES.1 = 1, val.RES.2 = 1,

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 = "AC" and o = "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 = "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 havea value close to zero (typically ~ 10-5); if itislarger than say ~ 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. 5 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. 5 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,]

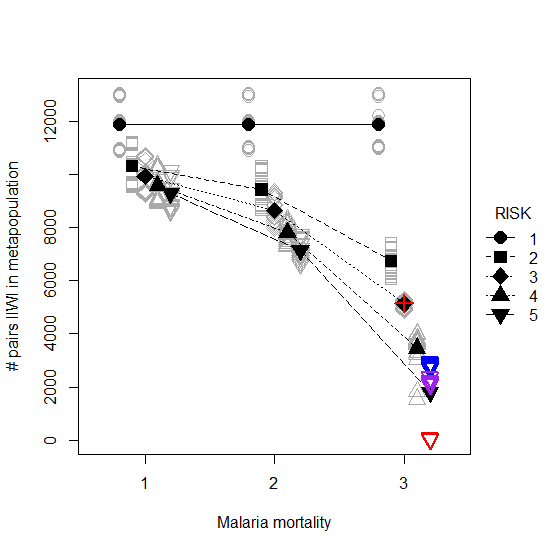
di.m1 = di[di$gamma.mov == 0.159,]; di.m2 = di[di$gamma.mov == 0.541,]; di.m3 = di[di$gamma.mov == 10,]

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. 6 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. 6 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

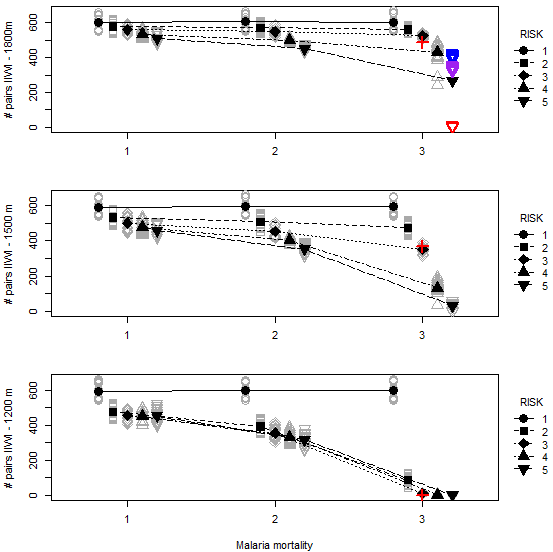
)

# 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. 7 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. Other examples of use of *f.plot.composite* will be given in Chapter 5.

# BLOCK 9

# Fig. 7 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,

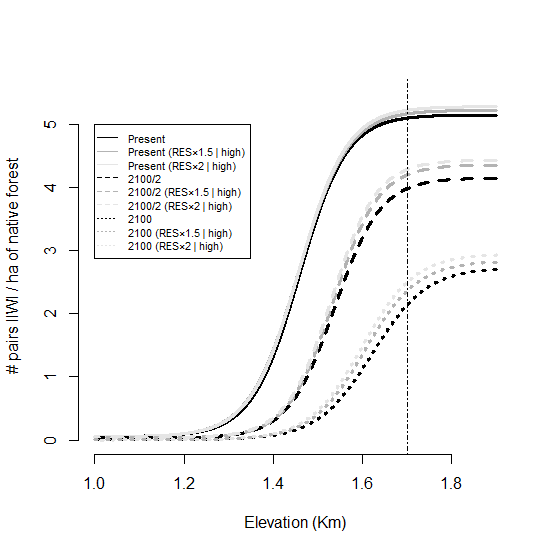
add.leg = FALSE,

create.plot = FALSE, lty.mean = 3

)

abline(v = 1.7, lty = 4)

**Fig. 4.2.3-1**



Chapter 5

MAMO AT THE COMMUNITY LEVEL

As previously mentioned, in chapter 5 we will first use the techniques presented earlier for a single species (IIWI) and apply them to the 8 native forest bird species of Hakalau Forest, namely AKIP, AKEP, HCRE, IIWI, ELEP, OMAO, APAP, and HAAM.

Because the current version of MAMO assumes that interspecific competition has no significant impact on the species in the community, the calibration and run procedures will follow exactly the same logic as previously, although we will introduce a two-step calibration procedure applicable when a larger number of species and parameters need to be calibrated (we simulate a whole community rather than a single species, many species have been studied far less intensively than IIWI so uncertainty is bigger, and new parameters concerning rat predation are introduced; see the 'HAKALAU paper' for details).

The principle is simple:

1) the first round (using *f.calibr*) is used to explore the boundaries of each parameter (minimum and maximum allocated values) and their combinations;

2) using *f.env.single* allows to identify which boundary tends to yield a better fit with observed data;   
3) this better boundary is used together with an intermediate value (minimum + maximum / 2) in yet another round of calibration of the new boundaries and their combinations (using *f.calibr*);

4) the two rounds of calibration each yield a 'sensitivity envelop' using the function *f.envp.single*; the final 'sensitivity envelop' is obtained by combining these two envelops and extracting the *n.envp* best using the function *f.envp.double* that will be presented in this chapter.

Altogether, assuming that *n* parameters need to be calibrated, a total of 2×2n = 2n+1 combinations will be explored using the two-step calibration procedure, as compared to 3n if all combinations were explored simultaneously using the minimum, intermediate and maximum parameter values (hereafter, one-step procedure). The reason to implement the two-step procedure is to save computation time while exploring the subset of parameter combination that reasonably offers the best chance of yielding a good fit between observed and simulated data; for instance if n = 10, the two-step calibration procedure would require running 2,048 *mamo* run, while the comprehensive one-step version would require 59,049 run (> 28 times more). In future versions of MAMO, it should be relatively straightforward to implement a multi-step calibration procedure (*x* steps, with x ≥ 2). The number of steps *x* such as the multi-step procedure becomes equivalent to the one-step procedure in terms of computational time (i.e., 2n-1+x = 3n) is given by . For instance, if n = 10, we can perform as many as 6.85 steps (i.e. 6, almost 7). The great interest of the multi-step calibration procedure, however, is that it allows to explore a much wider range of the parameter space than the one-step equivalent, because at each step, we define and explore a new parameter value which is the average between the boundaries of the previous step (see above). Furthermore, the process is 'intelligent' in the sense that it chooses at each step to focus in the region of greater fit.

Simulation studies for each species can be run and analyzed independently, following the approach described in chapter 4, and we will give here only a few examples of application of previously used techniques such as *f.plot.composite*, as well as an example of a more comparative approach relying on principal component analysis (chapter 5.5).

However, because species-specific approaches rapidly become unpractical when the number of species in the community or the number of management scenarios increase (as each species may respond idiosyncratically to each management action), we also developed an integrative approach that allows users to estimate community-level (rather than species-specific) response to climate change or management practices (chapters 5-3 and 5-4). The weight of each species in the community-level average is defined by the user, and can for instance reflect conservation status.

5.1 Calibration

5.1.1 First calibration round

Block 11 of the SCRIPT file contains the code pertaining to the first round of calibration using the previously described *f.calibr* function (chapter 3.1). The code specifying calibration for the IIWI, reproduced below, is very similar to the code used earlier, with the following exceptions:

1) output.dir is obviously different;

2) y.obs = y.obs.IIWI instead of y.obs.IIWI.1; in order to ensure the homogeneity of the data sets for each species, y.obs.IIWI was obtained using a different method than y.obs.IIWI.1 (see the 'HAKALAU paper' for details)

3) d = read.table("C:/Programs/MAMO/CALIBRATION/Starting parameters/param\_calib.HAKALAU.txt", header = T, sep = "\t", dec = ".")

instead of

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

The file param\_calib.HAKALAU.txt was used instead of the file param\_calib.IIWI.1.txt for the same reason we described for *y.obs* (similar treatment of all species)

4) *n.gamma.mov* was taken here to be 3 (instead of 4), and gamma.mov.direct = c(0.159, 0.541, 10), instead of c(-10, 0.159, 0.541, 10); in order to save computation time, the '-10' value which appeared less relevant for IIWI (see 'IIWI paper' for details) was excluded here both for IIWI and APAP

5) we used n.s.ad = 2 (instead of 3) and input.direct.s.ad = FALSE instead of TRUE; we did the same for *n.Sm.ac*, *n.fec* and *n.fec.1* (and their associated 'input.direct' parameters); we also used paired.s.ad.fec = FALSE instead of TRUE

6) we used n.rat.s = 2 (instead of 1; this parameter was no longer assumed to be known); we did the same for *n.rat.f* and *n.R.ter*.

# Excerpt of BLOCK 11 concerning IIWI calibration

f.calibr(

# species

sp = "IIWI", output.dir = "C:/Programs/MAMO/CALIBRATION/run/iiwi-p1",

y.obs = y.obs.IIWI,

d = read.table("C:/Programs/MAMO/CALIBRATION/Starting parameters/param\_calib.HAKALAU.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 = 2, input.direct.s.ad = FALSE, n.rat.s = 2,

# Reproduction and habitat quality

n.fec = 2, input.direct.fec = FALSE, n.fec.1 = 2, input.direct.fec.1 = FALSE, paired.s.ad.fec = FALSE,

n.rat.f = 2, n.K.b = 2, 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 = 2, input.direct.Sm.ac = FALSE,

# Movements

n.gamma.mov = 3, input.direct.gamma.mov = TRUE, gamma.mov.direct = c(0.159, 0.541, 10), calc.gamma.d = "fast.risky",

n.sim.disp = 10000, n.R.ter = 2, 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

)

The process for IIWI alone took 46 hours and 51 minutes on a DELL LATITUDE E6230 with a 2.8 GHz Intel Core 3rd Generation i5-3360M Processor. In total 10,368 individual *mamo* run were necessary to complete this first round of calibration.

The calibration code in BLOCK 11 for IIWI and APAP on one hand and the 6 remaining non-migratory species on the other hand is actually very similar, as differing parameters such as *t.b* (duration of the breeding season) are assumed to be fixed and encoded directly in the param\_calib.HAKALAU.txt file of the *d* argument of *f.calibr*. The main difference is that the parameters *n.gamma.mov* and *n.K.nb.1* have to be set to 1 for non-migratory species (their actual values do not matter, as these parameters are irrelevant to sedentary species). As an example of sedentary species, we reproduce below the code for HAAM (note that future telemetry studies may reveal seasonal movements in species currently believed to be sedentary). Importantly, note that the first 3 parameters of the function, namely *sp*, *output.dir* and *y.obs* are species-specific and need to be changed for each species (e.g., sp = "IIWI" becomes sp = "HAAM"):

# Excerpt of BLOCK 11 concerning HAAM calibration

f.calibr(

# species

sp = "HAAM", output.dir = "C:/Programs/MAMO/CALIBRATION/run/haam-p1",

y.obs = y.obs.HAAM,

d = read.table("C:/Programs/MAMO/CALIBRATION/Starting parameters/param\_calib.HAKALAU.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 = 2, input.direct.s.ad = FALSE, n.rat.s = 2,

# Reproduction and habitat quality

n.fec = 2, input.direct.fec = FALSE, n.fec.1 = 2, input.direct.fec.1 = FALSE, paired.s.ad.fec = FALSE,

n.rat.f = 2, n.K.b = 2, K.nb.1 = NA, n.K.nb.1 = 1, K.nb.2 = NA, 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 = 2, input.direct.Sm.ac = FALSE,

# Movements

n.gamma.mov = 1, input.direct.gamma.mov = FALSE, calc.gamma.d = "fast.risky",

n.sim.disp = 10000, n.R.ter = 2, 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

)

The process for HAAM alone took 2 hours and 02 minutes on a DELL LATITUDE E6230 with a 2.8 GHz Intel Core 3rd Generation i5-3360M Processor. In total 1536 individual *mamo* run were necessary to complete this first round of calibration.

5.1.2 'Sensitivity envelop' corresponding to the first calibration round

As we explained in chapter 3.3.2 and in the preamble of this chapter, estimating the 'sensitivity envelop' of a calibration run using the function *f.envp.single* also returns a modified data set *d* that can be used as input for the function *f.calibr* to run a second round of calibration.

Below is an excerpt of BLOCK 12 of the SCRIPT file corresponding to the calibration for HAAM. We create an object called haam.p1 which yields a plot of the first calibration round, and also produces several outputs including the envelop (haam.p1$envp) and the modified data set (haam.p1$d; although the user should not routinely access these outputs directly, we note that in haam.p1$d, only the row corresponding to the argument *sp* of *f.envp.single*, i.e. "HAAM", has been modified in comparison to the original which, in this case, is param\_calib.HAKALAU.txt).

haam.p1 = f.envp.single(

sp = "HAAM", output.dir = "C:/Programs/MAMO/CALIBRATION/run/haam-p1",

y.obs = y.obs.HAAM, n.envp = 10, col = "blue",

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

5.1.3 Second calibration round

Continuing our example of two-step calibration based on HAAM, we provide below the excerpt of code from BLOCK 13 that run a new set of calibration run based on the outcome of the first one, i.e. substituting the *f.calibr* argument:

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

by

d = haam.p1$d

Note that here again, a different *output.dir* argument has to be established (and the corresponding folder created); otherwise, the second round would simply erase and replace the results of the first round; e.g., replace

output.dir = "C:/Programs/MAMO/CALIBRATION/run/haam-p1"

by

output.dir = "C:/Programs/MAMO/CALIBRATION/run/haam-p2"

# Excerpt of code from BLOCK 13 - calibration of HAAM parameters (second round)

f.calibr(

# species

sp = "HAAM", output.dir = "C:/Programs/MAMO/CALIBRATION/run/haam-p2",

y.obs = y.obs.HAAM,

d = haam.p1$d,

# 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 = 2, input.direct.s.ad = FALSE, n.rat.s = 2,

# Reproduction and habitat quality

n.fec = 2, input.direct.fec = FALSE, n.fec.1 = 2, input.direct.fec.1 = FALSE, paired.s.ad.fec = FALSE,

n.rat.f = 2, n.K.b = 2, K.nb.1 = NA, n.K.nb.1 = 1, K.nb.2 = NA, 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 = 2, input.direct.Sm.ac = FALSE,

# Movements

n.gamma.mov = 1, input.direct.gamma.mov = FALSE, calc.gamma.d = "fast.risky",

n.sim.disp = 10000, n.R.ter = 2, 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

)

Once the run is completed, we can obtain the second 'sensitivity envelop' by modifying the code for haam.p1 as follows:

# Excerpt of code from BLOCK 14 - 'sensitivity envelop' of HAAM (second round)

haam.p2 = f.envp.single(sp = "HAAM", output.dir = "C:/Programs/MAMO/CALIBRATION/run/haam-p2",

y.obs = y.obs.HAAM, n.envp = 10, col = "blue",

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

5.1.4 Obtaining the final 'sensitivity envelop' using *f.envp.double*

We now have access to the *n.envp* best run of the first calibration round and the *n.envp* best run of the second calibration round as well (we'll assume for the sake of the demonstration that both *n.envp* values are identical, but this is not mandatory). Basically, using the *n.envp* argument of *f.envp.double*, we will take the subset *n.envp* (which again, can be different from the previous two), out of 2 × n.envp, that provided the best fit with the observed data while ensuring that no two run with identical parameter combinations are included to avoid pseudoreplication.

The other *f.envp.double* arguments are fairly straightforward: *sp* is the name of the species, *y.obs* is the observed data for the species, *col* the color to be used to plot observed data, *envp.1* is the envelop based on the first calibration round, *d1* the path to the folder containing results of the first calibration round, and *envp.2* and *d2* are the equivalent of *envp.1* and *d1*, respectively, for the second calibration round.

The single output of *f.envp.double* is the final 'sensitivity envelop', although of course the calibration plot is automatically produced just like in *f.plot.single*. Below is reproduced the BLOCK 15 code of the SCRIPT file that produces Fig. 5.1.4-1 (= Fig. 1 of the 'HAKALAU paper).

# BLOCK 15

# Fig. 1 - Calibration: get the final 'sensitivity envelop' for each species

layout(matrix(c(1:8), 4, 2)); par(mar = c(2.2,2.2,0,0)+0.5, cex.main = 1)

envp.akip = f.envp.double(

sp = "AKIP",

envp.1 = akip.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/akip-p1",

envp.2 = akip.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/akip-p2",

n.envp = 10, y.obs = y.obs.AKIP, col = "blue")

envp.akep = f.envp.double(

sp = "AKEP",

envp.1 = akep.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/akep-p1",

envp.2 = akep.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/akep-p2",

n.envp = 10, y.obs = y.obs.AKEP, col = "blue")

envp.hcre = f.envp.double(

sp = "HCRE",

envp.1 = hcre.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/hcre-p1",

envp.2 = hcre.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/hcre-p2",

n.envp = 10, y.obs = y.obs.HCRE, col = "blue")

envp.iiwi = f.envp.double(

sp = "IIWI",

envp.1 = iiwi.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/iiwi-p1",

envp.2 = iiwi.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/iiwi-p2",

n.envp = 10, y.obs = y.obs.IIWI, col = "blue")

envp.elep = f.envp.double(

sp = "ELEP",

envp.1 = elep.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/elep-p1",

envp.2 = elep.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/elep-p2",

n.envp = 10, y.obs = y.obs.ELEP, col = "blue")

envp.omao = f.envp.double(

sp = "OMAO",

envp.1 = omao.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/omao-p1",

envp.2 = omao.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/omao-p2",

n.envp = 10, y.obs = y.obs.OMAO, col = "blue")

envp.apap = f.envp.double(

sp = "APAP",

envp.1 = apap.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/apap-p1",

envp.2 = apap.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/apap-p2",

n.envp = 10, y.obs = y.obs.APAP, col = "blue")

envp.haam = f.envp.double(

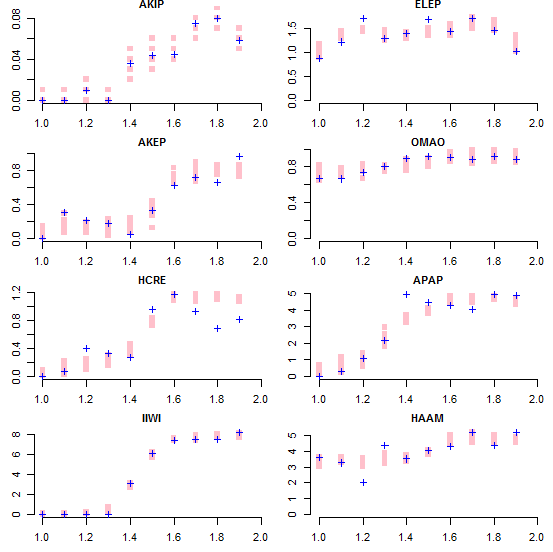
sp = "HAAM",

envp.1 = haam.p1$envp, d1 = "C:/Programs/MAMO/CALIBRATION/run/haam-p1",

envp.2 = haam.p2$envp, d2 = "C:/Programs/MAMO/CALIBRATION/run/haam-p2",

n.envp = 10, y.obs = y.obs.HAAM, col = "blue")

**Fig. 5.1.4-1**



5.2 Simulation study

Our objective is to compare the impact of 3 different management actions implemented alone or in combination with each other, both in the present and the future (based on the climatic conditions expected for the year 2100):

1) an increase of available habitat by adding a forested band from 1900 to 2000 m;

2) a constant-effort reduction of malaria transmission risk by 50 % on a forested band covering the top 400 meters of the elevational gradient, i.e. at elevations ≥ 1600 m when no habitat management is performed (gradient up to 1900 m) and elevations ≥ 1700 m otherwise (gradient up to 2000 m);

3) a reduction of rat predation on adults and fledglings by 70 % on a forested band covering the top 400 meters of the elevational gradient.

Altogether, we will therefore compare 8 management scenarios, from the first one where none of the 3 actions are implemented, to the last one where all 3 are implemented simultaneously.

Using the same value for our elevational gradient as in the 'IIWI paper', i.e. grad = c(1900, 1000), the last two management actions are coded by choosing the following values for *f.run* arguments (see chapter 4.1 for further explanations):

2)

i) n.RISK = 5

ii) mg.RISK = c(rep(0.5, 4), rep(1, 6))

3)

i) n.RAT = 2,

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

The first management action we want to investigate cannot be coded the same way, as in the current version of *f.run* we did not include any argument related to the creation of new habitat (see chapter 4.1). It may not be the best option for future MAMO versions anyway, because the coding of many parameters is affected when the landscape structure is modified.

The method we used here was to make two distinct call to *f.run*. The first call takes cares of the first half of the simulations, where management for reducing malaria risk and rat predation are conducted within the current landscape defined by grad = c(1900, 1000). Taking the example of OMAO, we create a folder called 'omao.10' (because nr = 10) in the RUN folder of MAMO (as usual, containing the empty file called *t.sim.txt*), and run the following code (an excerpt of BLOCK 16 which contains the code necessary to run the simulation study for all eight species):

omao.run.10 = f.run(

# species-specific

sp = "OMAO",

envp = envp.omao,

output.dir = "C:/Programs/MAMO/RUN/omao.10",

y.obs = y.obs.OMAO,

d = read.table("C:/Programs/MAMO/CALIBRATION/Starting parameters/param\_calib.HAKALAU.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 = 2,

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

# 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 = 1,

mg.K.NB.2 = rep(1, 10), # no management

reproduction.malaria = "simple",

# Malaria parameters (daily except Sm.ac)

n.RISK = 5, mg.RISK = c(rep(0.5, 4), rep(1, 6)),

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 = 2, evol.AC = c(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

)

The second *f.run* call takes cares of the second half of the simulations, where management for reducing malaria risk and rat predation are conducted within a new landscape defined by grad = c(2000, 1000). Ultimately, the combination of the results from both calls will make the simulation study complete.

Taking once more the example of OMAO, we create a folder called 'omao.11' in the RUN folder of MAMO and run the following code (again, an excerpt of BLOCK 16 which contains the code necessary to run the simulation study for all eight species):

omao.run.11 = f.run(

# species-specific

sp = "OMAO",

envp = envp.omao,

output.dir = "C:/Programs/MAMO/RUN/omao.11",

y.obs = c(NA, y.obs.OMAO),

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

# Spatial structure

nr = 11, nc = 2, grad = c(2000, 1000), unit = 1,

# Time frame

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

# Survival

n.RAT = 2,

mg.RAT = list( rep(1, 11), c(rep(0.3, 4), rep(1, 7)) ),

# Reproduction and habitat quality

K.nb.1 = list(K.nb.1.2003\_nc.11, K.nb.1.2004\_nc.11, K.nb.1.avg\_nc.11),

n.K.NB.1 = 1,

mg.K.NB.1 = rep(1, 11), # no management

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

n.K.NB.2 = 1,

mg.K.NB.2 = rep(1, 11), # no management

reproduction.malaria = "simple",

# Malaria parameters (daily except Sm.ac)

n.RISK = 5, mg.RISK = c(rep(0.5, 4), rep(1, 7)),

alpha.b = alpha.b\_nc.11, alpha.nb.1 = alpha.nb.1\_nc.11, alpha.nb.2 = alpha.nb.2\_nc.11, alpha.1 = alpha.1\_nc.11,

alpha.b.2100 = alpha.b.2100\_nc.11, alpha.nb.1.2100 = alpha.nb.1.2100\_nc.11, alpha.nb.2.2100 = alpha.nb.2.2100\_nc.11, alpha.1.2100 = alpha.1.2100\_nc.11,

n.AC = 2, evol.AC = c(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

)

As you can see, most parameters of the second call are identical to the first run, with the following exceptions:

1) obviously, *output.dir* is different

2) *nr* becomes 11 instead of 10, and *grad* becomes c(2000, 1000) instead of c(1900, 1000)

3) *y.obs* becomes c(NA, y.obs.OMAO) instead of y.obs.OMAO; basically, every vector of length *nr* needs to be augmented from nr = 10 to nr = 11; in this case, we could use 0 or NA for the first *y.obs* value (corresponding to the new band of forest between 2000 and 1900 m); which one is chosen does not really matter, however, as the comparison between observed and simulated data is not a part of the *f.run* function (unlike *f.calibr*);

4) *mg.RISK* becomes c(rep(0.5, 4), rep(1, 7))

5) *mg.RAT* becomes list( rep(1, 11), c(rep(0.3, 4), rep(1, 7)) )

6) *mg.K.NB.1* and *mg.K.NB.2* become rep(1, 11)

Running BLOCK 16, after changing run\_f.run = FALSE into run\_f.run = TRUE, is all it takes to perform the simulation study. It is a relatively fast process; for instance, both calls for the species that took longest (IIWI) were completed in 4 hours 21 minutes on a DELL LATITUDE E6230 with a 2.8 GHz Intel Core 3rd Generation i5-3360M Processor; the same simulation study took 54 minutes for AKIP.

5.3 Create data set for community-level analysis

5.3.1 First step: species-specific data sets

Using a new function called *f.data.species*, we create for each species a unique data set (the single output of *f.data.species*) that:

1) combines the results of the two *f.run* calls (see chapter 5.2 for the description of the 2 calls); note than each individual *f.run* output itself consists of the combination of two different text files called *t.sim.txt* and *factors.txt* (see chapter 4.1);

2) is augmented by the creation of new variables (described later in this chapter).

The arguments of *f.data.species* are as follows:

1) *sp* is, as usual, the name of the species considered (e.g., "AKIP");

2) *w* is a data frame containing 3 different weighting schemes applied to each species in the community called *w1*, *w2* and *w3*, respectively; the first piece of code of BLOCK 17 in the SCRIPT file (reproduced below) gives an example of how to create *w*, as different weighting schemes could be chosen by the user (but please note that in the current version of MAMO, exactly 3 weighting schemes have to be supplied by the user):

w = as.data.frame(matrix(NA, nr = 8, nc = 3))

rownames(w) = c("AKIP", "AKEP", "HCRE", "IIWI", "ELEP", "OMAO", "APAP", "HAAM")

colnames(w) = c("w1", "w2", "w3")

w[,1] = 1

w[,2] = c(3, 3, 3, 2, 2, 2, 1, 1)

w[,3] = 1 / c(1900, 12000, 14000, 360000, 190000, 170000, 1300000, 810000)

In this case, *w1* is always equal to 1 which means that all species are given the same weight in our inferences independently of their conservation status; in contrast, more weight is given to vulnerable and endangered species when using *w2* (their weight becomes 2 and 3, respectively) and especially, *w3* (in the latter, the weight is inversely proportional to the species' estimated global abundance; see the 'Hakalau paper' for data source);

3) *directory* is the argument used to tell *f.data.species* where the output of the two *f.run* calls are located; using sp = "AKIP" as an example, we can write:

directory = c("C:/programs/MAMO/RUN/akip.10", "C:/programs/MAMO/RUN/akip.11")

4) *vec.hab* is a vector that tells *f.data.species* whether the *f.run* call corresponded to habitat management (vec.hab = 1) or not (vec.hab = 0); based on the example above, *vec.hab* equals c(0,1); had we chosen

directory = c("C:/programs/MAMO/RUN/akip.11", "C:/programs/MAMO/RUN/akip.10"), *vec.hab* would have been c(1,0);

5) *table.risk* tells *f.data.species* how to tease apart the separate effects of malaria transmission risk *per se* and malaria control for future analysis; in this case, we have simulated five different scenarios encoded as follows in the variable 'RISK' of *factors.txt*: RISK = 1 for pre-malaria era, 2 for current era and malaria control, 3 for current era and no management, 4 for future (2100) and malaria control and 5 for future and no management; these levels are reproduced in the 'RISK\_run' column of *table.risk*; the next two columns indicate the appropriate value for the actual risk in the analysis in the absence of management ('RISK\_analysis' varies from 0 in the pre-malaria era to 2 in the future) and the appropriate value for management ('mg\_RISK' varies from 0 or 1);

the second piece of code of BLOCK 17 in the SCRIPT file (reproduced below) gives an example of how to create *table.risk*:

table.risk = as.data.frame(matrix(c(1:5, c(0, 1, 1, 2, 2), c(0, 1, 0, 1, 0)), nc = 3))

colnames(table.risk) = c("RISK\_run", "RISK\_analysis", "mg.RISK")

As previously mentioned, calling *f.data.species* produces a data set containing a number of rows equivalent to the number of *mamo* simulations necessary to complete the simulation study (in this case, 800); in addition to the *f.run* outputs *t.sim.txt* and *factors.txt* binded together, a new set of variables useful for subsequent analysis is produced:

1) *n.st* is the standardized species-specific index of population abundance based on *np.metapop* (the predicted metapopulation size) taking the minimum value 0 for the single *mamo* simulation yielding the smallest *np.metapop* value in the data set (called *n.min*, but not returned) and the maximum value 1 for the *mamo* simulation yielding the greatest *np.metapop* value (*n.max*, not returned either), and calculated as ;

this index will allow us to compare the relative effect of different management actions on different species without being confounded by widely different species-specific abundance;

2) *w1*, *w2* and *w3* are the species-specific weights corresponding to the columns *w1*, *w2* and *w3* of the argument *w*, respectively;

3) *mg.hab* takes the value 0 if the *mamo* run was conducted in the absence of habitat management, 1 otherwise (as defined using the argument *vec.hab*);

4) *risk* refers to malaria transmission risk and varies from 0 (pre-malaria era) to 2 (future - 2100), as defined using the argument *table.risk*;

5) *mg.risk* takes the value 0 if the *mamo* run was conducted in the absence of malaria control, 1 otherwise, as defined using the argument *table.risk*;

6) the variable *ac* derives directly from the *AC* column of *factors.txt* (ac = AC); in this case, ac = 2 (the highest level) corresponds to the current malaria survival rate, or to the future rate in the absence of evolution towards increased malaria resistance or tolerance, while ac = 1 corresponds to a potential survival rate two times higher in the future (see chapter 4.1 for details on *factors.txt* and 5.2 for details on the present simulation study);

7) *mg.rat* is calculated based on *factors.txt*: mg.rat = RAT.S - 1; in this case, *mg.rat* takes the value 0 if the *mamo* run was conducted in the absence of rat control, 1 otherwise;

8) *mg.res.1* is calculated based on *factors.txt*: mg.res.1 = RES.1 - 1; in this case, it always takes the value 0 as all *mamo* run were conducted in the absence of resource management during the first period of the non-breeding season (see chapter 5.2);

9) *mg.res.2* is calculated based on *factors.txt*: mg.res.2 = RES.2 - 1; in this case, it always takes the value 0 as all *mamo* run were conducted in the absence of resource management during the second period of the non-breeding season (see chapter 5.2);

10) *envp* is the equivalent of the column *ENV* of *factors.txt* (envp = ENVP); it reports the rank of the 'sensitivity envelop' from 1 to *n.envp*, 1 being the parameter combination which produced the best fit with observed data;

11) *sim* is the equivalent of the column *SIM* of *factors.txt* (sim = SIM); it is the replicate number for each unique parameter combination (from 1 to *n.sim*).

Below is an excerpt of the third piece of code of BLOCK 17 that creates a species-specific data set for each species; in this case, using AKIP as a case study, we create the data set called 'AKIP' as follows:

AKIP = f.data.species(sp = "AKIP", w = w,

directory = c("C:/programs/MAMO/RUN/akip.10", "C:/programs/MAMO/RUN/akip.11"),

vec.hab = c(0, 1), table.risk = table.risk)

5.3.2 Second step: community-level data set

We now have for each individual species a data set synthesizing the results of the simulation study. In particular, the variable *n.st* is our standardized species-specific index of population abundance; varying between 0 and 1, it is a suitable index to characterize species-specific response to climate change and management practices.

Our next step is to generate a new data set where the response to climate change and management practices is the weighted average of species-specific standardized responses. Such a weighted average can be calculated as ; by definition, our community-level index of population response it is also bounded by 0 and 1.

Since we have introduced three different weighting schemes called *w1*, *w2* and *w3* (see chapter 5.3.1), we can create 3 response variables called *n.w1*, *n.w2* and *n.w3*, respectively. Our first weighting scheme being w1 = 1 species, *n.w1* is simply the arithmetic mean of standardized species-specific indices of population abundance; hence, *n.w1* is the community-level average where each species is weighted equally, independently or its relative abundance or conservation status. For instance, the weight ascribed to AKIP using *w1* is 1, i.e. 1/8 of the total (0.125). This is equivalent to say that we regard our community as being composed of 12.5% of AKIP (and also 12.5% of AKEP, etc.). The response *n.w1* may be relevant if our management objective is to ensure that as many species as possible respond favorably to the proposed management plan. Conversely, the weight of vulnerable and endangered species in the computation of *n.w2* and *n.w3*, and therefore the importance of these species' response in the choice of management actions, is disproportionately augmented. For instance, the weight ascribed to AKIP using *w2* and *w3* is akin to a community composed of 17.6% and 75.5% of AKIP, respectively. For comparison, AKEP and HCRE contribute 12.0% and 10.2% in the computation of *n.w3*, respectively, while the contribution of all the non-endangered species considered together drops to 2.3%. Of course, our approach is flexible, different weighting schemes can be chosen by users (but not at this step; see chapter 5.3.1) depending on how they rate their priority in terms of management and conservation.

In the current version of MAMO, the function *f.data.community* calculating *n.w1*, *n.w2* and *n.w3* needs the following arguments:

1) the species-specific data sets created for each species in the community are called using the arguments AKIP, AKEP, HCRE, IIWI, ELEP, OMAO, APAP and HAAM; e.g., if you called your data set for AKIP *AKIP*, as we did, use AKIP = AKIP;

2) the argument *var\_* tells *f.data.community* which variables of species-specific data sets must be included as well in the community-level output; typically, we want to include the variables related to climate change and management effort potentially impacting the community response, so the default is = c("mg.hab", "risk", "mg.risk", "ac", "mg.rat", "mg.res.1", "mg.res.2", "envp", "sim"); see chapter 5.3.1 for details on these variables.

Reproduced below is the code of BLOCK 18 of the SCRIPT file creating the data set at the community level:

d.c = f.data.community(

AKIP = AKIP, AKEP = AKEP, HCRE = HCRE, IIWI = IIWI,

ELEP = ELEP, OMAO = OMAO, APAP = APAP, HAAM = HAAM,

var\_ = c("mg.hab", "risk", "mg.risk", "ac", "mg.rat", "mg.res.1", "mg.res.2", "envp", "sim")

)

The output of the function *f.data.community* is a single data set, here called *d.c*, containing a number of rows equivalent to the number of *mamo* simulations necessary to complete the simulation study (in this case, 800), the community-level response variables *n.w1*, *n.w2*, and *n.w3* together with all the explanatory variables included in var\_ (i.e., *mg.hab*, *risk*, etc.).

As an aside, in order to allow the user to investigate species-specific responses in parallel to the community-level response if she/he so desires, we also included as output of *f.data.community* all 8 species-specific index of population abundance (i.e., the variable *n.st*). As an output of *f.data.community*, *n.st* is now called *AKIP* (for AKIP), *AKEP* (for AKEP), etc.

5.4. Statistical analysis

**In the current version of MAMO, the R functions and code developed to perform statistical analyses at the community level are strictly applicable only to the simulation study presented in chapter 5.2 and corresponding to the 'Hakalau paper'.** It is a first step towards the production of more general functions applicable to a wider set of simulation scenarios. Ideally, these functions should be general enough that it becomes unnecessary to have recourse to hard code modification, however minor, as hard code modification is perhaps unlikely to be undertaken by managers. For the time being, developed functions (*f.effect.size*, *f.plot.scenarios* and *f.pca*; presented below) have not been made available in the same 'bank of functions' as e.g., *f.calibr* or *f.run*; instead, these new functions are available directly in the SCRIPT file.

Our objectives for the 'Hakalau paper' were threefold:

1) develop a statistical framework for a community-level rather than species-specific approach to conservation;

2) investigate the impact of 3 potential management practices on the community response at Hakalau NWR;

3) determine whether the anticipated climate change will modify the relationship between management practices and the community response.

We assumed that resistance or tolerance to malaria did not evolve (at least, to detectable levels) between now and 2100, which can be regarded as a plausible worst-case scenario (for instance, a lack of genetic differentiation across the elevational gradient was observed for IIWI and APAP despite at least 6 decades of exposure to avian malaria: Foster et al. 2007). Note that our simulation study included a scenario where survival to malaria was doubled in 2100, but this scenario will not be analyzed here.

5.4.1 Mixed-effect linear models

We used mixed-effect linear models to investigate the relationship between the community response (as determined by *n.w1*, *n.w2* or *n.w3*) and management practices (as determined by *mg.risk*, *mg.rat* and *mg.hab*) both in the present and in the future (see chapter 5.3 for description of variables). We assumed additive effects for *mg.risk*, *mg.rat* and *mg.hab*, treated as fixed factors. We did not account for potential interactions between explanatory factors, an assumption that we will assess later on. The possible influence of the particular value of the 'sensitivity envelop' (variable *envp*) and replicate (*sim*) used to perform the simulation was accounted for by treating *envp* and *sim* as random factors and calculating a random intercept for each of them.

Altogether, we run two series of three models called *m1.p* (using *n.w1* as response; present climatic conditions), *m2.p*, *m3.p*, *m1.f*, *m2.f* and *m3.f* (using *n.w3* as response; future climatic conditions), respectively. We reproduce below the code from BLOCK 19 of the script file that runs these models:

# Present

# take the subset of data corresponding to present

d.p = d.c[d.c$ac == 2 & d.c$risk == 1,]

# mixed linear model

m1.p = lmer(n.w1 ~ mg.risk + mg.hab + mg.rat + (1 | envp) + (1 | sim), data = d.p)

summary(m1.p)

shapiro.test(residuals(m1.p))

m2.p = lmer(n.w2 ~ mg.risk + mg.hab + mg.rat + (1 | envp) + (1 | sim), data = d.p)

summary(m2.p)

shapiro.test(residuals(m2.p))

m3.p = lmer(n.w3 ~ mg.risk + mg.hab + mg.rat + (1 | envp) + (1 | sim), data = d.p)

summary(m3.p)

shapiro.test(residuals(m3.p))

# Future

# take the subset of data corresponding to future risk and no evolution of malaria mortality

d.f = d.c[d.c$ac == 2 & d.c$risk == 2,]

# mixed linear model

m1.f = lmer(n.w1 ~ mg.risk + mg.hab + mg.rat + (1 | envp) + (1 | sim), data = d.f)

summary(m1.f)

shapiro.test(residuals(m1.f))

m2.f = lmer(n.w2 ~ mg.risk + mg.hab + mg.rat + (1 | envp) + (1 | sim), data = d.f)

summary(m2.f)

shapiro.test(residuals(m2.f))

m3.f = lmer(n.w3 ~ mg.risk + mg.hab + mg.rat + (1 | envp) + (1 | sim), data = d.f)

summary(m3.f)

shapiro.test(residuals(m3.f))

Although the result of each model can be inspected individually thanks to the *summary* function, we choose to summarize the results into a single table and calculate an effect size for each management action. This is the topic of the next chapter.

5.4.2 Effect size

We created a function called *f.effect.size* with no argument (available in BLOCK 20 of the SCRIPT file), emphasizing the fact that in the current version of MAMO, *f.effect.size* only performs a pre-determined series of tasks corresponding to the simulation study described in 5.2, the data sets created in 5.3 and the statistical models chosen in 5.4.1. Hard code modification would be necessary to make *f.effect.size* capable of handling a greater variety of scenarios and statistical models.

For each of the six models tested, from *m1.p* (Time = present & response = n.w1) to *m3.f* (Time = future & response = n.w3), use of BLOCK 20 of the script file will yield a table called *es* containing the following variables (note that Table 1 of the 'Hakalau paper' is a subset of *es*):

1) *est.int*, *est.risk*, *est.hab*, and *est.rat* are the fixed effect coefficient estimates corresponding to the model global intercept (i.e., community response without management action), and the effect of *mg.risk*, *mg.hab* and *mg.rat*, respectively;

2) *se.int*, *se.risk*, *se.hab*, and *se.rat* are the fixed effect standard errors;

3) *eff.risk*, *eff.hab*, and *eff.rat* are the estimated effect size of each individual management action. An effect size *x* means that the management action yielded a community response equivalent to *x* times the community response in the absence of management. It is calculated as the ratio of coefficients: .

4) *se.eff.risk*, *se.eff.hab*, and *se.eff.rat* are the standard error for effect sizes calculated using the delta method.

5) *CI1.eff.risk* and *CI2.eff.risk* are the lower and upper bound of the 95% confidence interval for the effect size of *mg.risk*, assuming a normal distribution; bounds of the 95 % confidence interval also are provided for *mg.hab* and *mg.rat*.

5.4.3 Graphical presentation of results

We developed a function called *f.plot.scenarios* which can be used to visualize the results of the simulation study. The different management scenarios are plotted on the *x*-axis; in the current version of MAMO, the 8 management scenarios resulting from the simulation study presented in chapter 5.2 are automatically plotted: 000 = no management action performed; m00 = a reduction of malaria transmission risk (in this case, 50 % reduction on a forested band covering the top 400 meters of the elevational gradient); 0r0 = reduction of rat predation on adults and fledglings (in this case, 70 % reduction on a forested band covering the top 400 meters of the elevational gradient); 00h = increase of available habitat (in this case, by adding a forested band from 1900 to 2000 m); the 4 remaining management scenarios are combinations of the latter 3 (e.g., m0h = reduction of malaria transmission risk and habitat increase, but no rat control). Note that hard-code modification of *f.plot.scenarios* would be required to plot alternative scenarios.

The response of the community to the management scenarios is plotted on the *y*-axis using the argument *y*, which can take values "n.w1", "n.w2" or "n.w3"; the argument *y.lim* allows the user to choose the range of *y* (naturally bounded by 0 and 1) most relevant for visualizing results.

Other arguments of *f.plot.scenarios* are:

1) *d.c* is the name of the community data set (in this case, called *d.c*)

2) *es* is the name of the table produced by the *f.effect.size* function (in this case, called *es*);

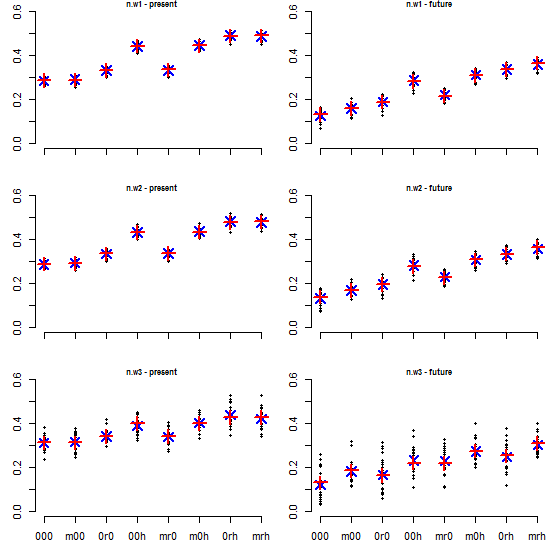
3) *Time* can be "Present" or "Future";

4) a subset of the *d.c* data set can be used for plotting, by choosing a particular value of the *d.c* columns *mg.hab*, *risk*, *mg.risk*, *ac*, *mg.rat*, *mg.res.1,* and *mg.res.2* using the arguments *val.mg.hab*, *val.risk*, *val.mg.risk*, *val.ac*, *val.mg.rat*, *val.mg.res.1*,and *val.mg.res.2*, respectively; for instance, based on the simulation study described in chapter 5.2, val.risk = 1 is the value of malaria transmission risk in the present climatic conditions, and val.ac = 2 is the current malaria survival rate, or the future rate in the absence of evolution towards increased malaria resistance or tolerance (see chapter 5.3.1 for detail of values); conversely, by choosing NA (default), all simulated values will be used for the plot (e.g., if val.risk = NA, included values in our case study would be 0, 1 and 2);

5) *main* is the title (use "" if you don't want any), and *add.x.labels* (logical) tells if labels should be plotted on the *x*-axis.

The code of BLOCK 21 of the SCRIPT file (not reproduced here) yields the equivalent of Figure 2 of the 'Hakalau paper' (= Fig. 5.4.3-1 below). Black dots show the simulated data points, the blue X sign is the empirical mean and the red + sign is the value predicted based on a mixed-effect linear model containing additive effects between the 3 management actions and no interaction (see chapter 5.4.1 for model description). In this case, the match between empirical mean and predicted value is very good, indicating that accounting for interactions between management actions is not necessary.

**Fig. 5.4.3-1**



In addition to the automatic plot, *f.plot.scenarios* returns a data set called *d.s*, the subset of *d.c* used to produce the plot, as well as a vector called *mean.y* containing the empirical mean value of the community response for scenarios ranging from 1 to 8 following the *x*-axis order, i.e., 000, m00, 0r0, 00h, mr0, m0h, 0rh, and mrh.

5.5. Complementary species-specific analyses

In this chapter, we will briefly consider a few complementary approaches to the community-level analyses presented in chapter 5.4. Although species-specific approaches do not have the capacity to produce a single estimate of the impact of management practices on the dynamics of the community, they may be useful to validate the community-level inference or illustrate specific aspects of the community response:

1) a principal component analysis (PCA) allows to examine between-species variability using a subset of statistically important dimensions; note however that the biological interpretation of PCA axes may become more arduous as the number of species and especially, management practices, increase;

2) species-specific response to a given management practice can be plotted on the same graph for all species in the community; this is helpful to identify categories of species responding similarly to a threat and / or its mitigation; again, an increase in the number of species and / or management practices may render this approach unpractical.

5.5.1 Using principal component analyses (PCA)

We created a function called *f.pca* to conduct principal component analyses (basically, *f.pca* is a wrapper function of *dudi.pca* of the ade4 package); *f.pca* is available as the first piece of code of BLOCK 22 of the SCRIPT file. In the current version of MAMO, the arguments of *f.pca* are:

1) the community data set, called by the argument *d.c* (use d.c = d.c if you called the community data set *d.c* as we did); note that *f.pca* will be using the species-specific responses (*AKIP*, *AKEP*, etc..) as opposed to the community response (*n.w1*, *n.w2*, and *n.w3*); see chapter 5.3.2 for composition of *d.c*; in addition to species-specific responses, *f.pca* also considers the management variables of *d.c*, namely *mg.hab*, *mg.risk*, *mg.rat*, *mg.res.1*, and *mg.res.2* in the current version of MAMO;

4) a subset of the *d.c* data set must be used for plotting (note: unlike *f.plot.scenarios* where it was optional), by choosing a particular value of the *d.c* columns *risk* and *ac* using the arguments *val.risk* and *val.ac*,respectively; for instance, based on the simulation study described in chapter 5.2, val.risk = 1 is the value of malaria transmission risk in the present climatic conditions, and val.ac = 2 is the current malaria survival rate, or the future rate in the absence of evolution towards increased malaria resistance or tolerance (see chapter 5.3.1 for detail of values);

3) *nf* is the number of axes to be kept for the principal component analysis (= the *nf* argument of dudi.pca).

The second piece of code of BLOCK 22 (reproduced below) runs two PCA, one for simulations in the present, the second for simulations based on future climatic conditions:

# Conduct PCAs

pca\_p = f.pca(d.c = d.c, val.risk = 1, val.ac = 2, nf = 3)

pca\_f = f.pca(d.c = d.c, val.risk = 2, val.ac = 2, nf = 3)

The third piece of code of BLOCK 22 (reproduced below) relies on the function *fviz\_pca\_var* of the factoextra package to plot the correlation circles; in Fig. 5.5.1-1 below (equivalent to Fig. 3 of the 'Hakalau paper'), A and B correspond to principal component (axes) 1-2 and 1-3 of the first PCA (present), respectively, while C and D correspond to axes 1-2 and 1-3 of the second PCA (future), respectively;

# Fig. 3 of the 'Hakalau paper'

# note that in R the four sections (A-D) have to be obtained separately

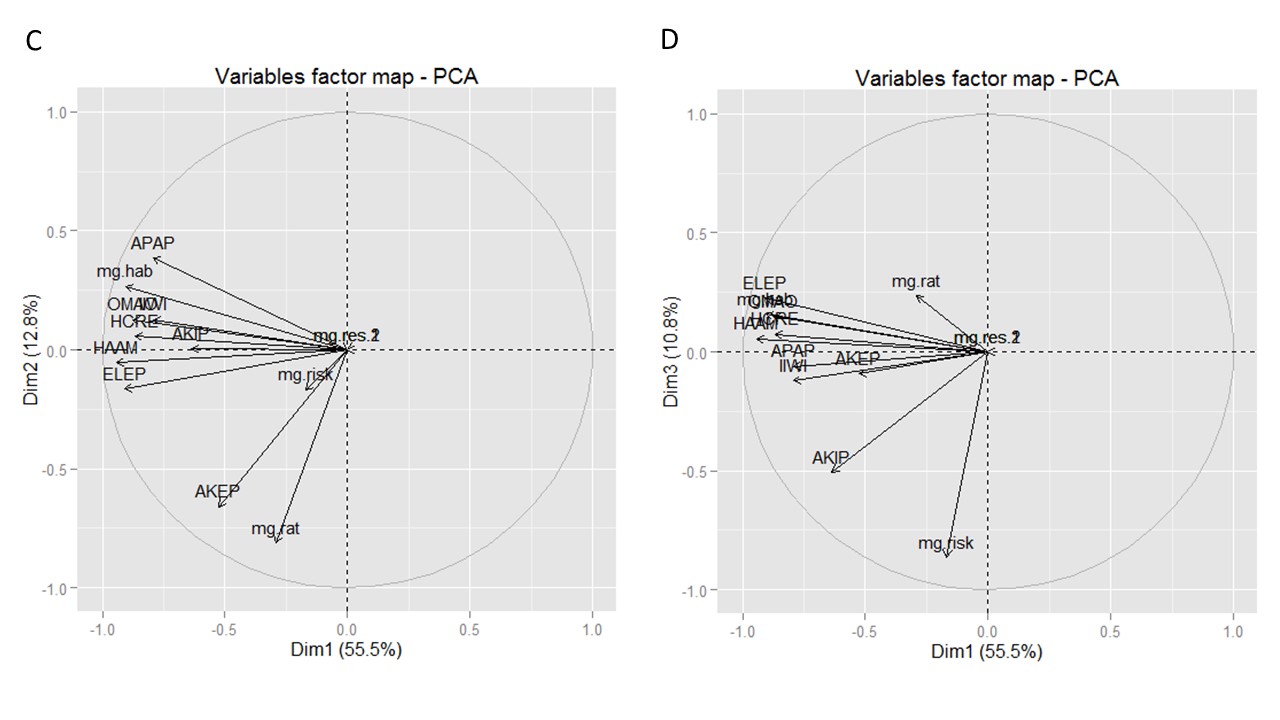
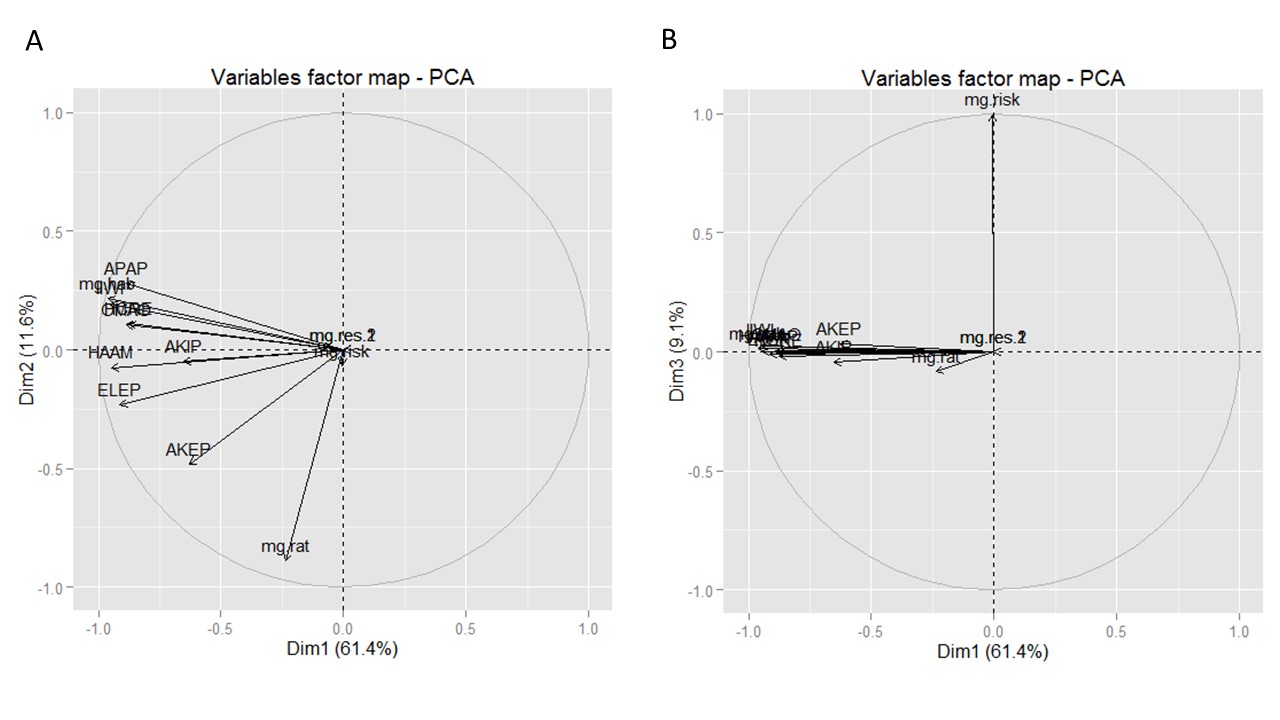
fviz\_pca\_var(pca\_p, axes = c(1, 2)) # section labelled 'A'

fviz\_pca\_var(pca\_p, axes = c(1, 3)) # section labelled 'B'

fviz\_pca\_var(pca\_f, axes = c(1, 2)) # section labelled 'C'

fviz\_pca\_var(pca\_f, axes = c(1, 3)) # section labelled 'D'

**Fig. 5.5.1-1**

****

Finally, the last piece of code of BLOCK 22 (reproduced below) show the correlation between management variables and species-specific responses on one hand, and the principal components (*nf* axes) on the other hand (note: these correlations are the values plotted on the correlation circles); since we did not explore scenarios where resources during the first and second period of the non-breeding season were manipulated, the correlation between *mg.res.1* or *mg.res.2* and the principal components is always null.

# Table 2 of the 'Hakalau paper'

t2 = cbind(round(pca\_p$co, 2), round(pca\_f$co, 2))

t2 = as.data.frame(t2)

colnames(t2) = c("PC1-pres", "PC2-pres", "PC3-pres", "PC1-futur", "PC2-futur", "PC3-futur")

t2

5.5.2 Using *f.plot.composite*

Blocks 23 and 24 of the SCRIPT file contain the code necessary to reproduce the figures 4 and 5 of the 'Hakalau paper', respectively; Fig. 4 concerns species-specific response to change in malaria transmission risk due to management or climate change, while Fig. 5 concerns species-specific response to rat control in the present and future. Please refer to chapter 4.2.3 for description of the *f.plot.composite* function, and to the 'Hakalau paper' for illustration and interpretation of figures.

Literature cited

Foster J. T., B. L. Woodworth, L. E. Eggert, P. J. Hart, D. Palmer, D. C. Duffy, and R. C. Fleischer. 2007. Genetic structure and evolved malaria resistance in Hawaiian honeycreepers. Molecular Ecology 16: 4738-4746.

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](http://www.gastonsanchez.com/PLS%20Path%20Modeling%20with%20R.pdf)

Woodworth, B.L. andPratt,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.