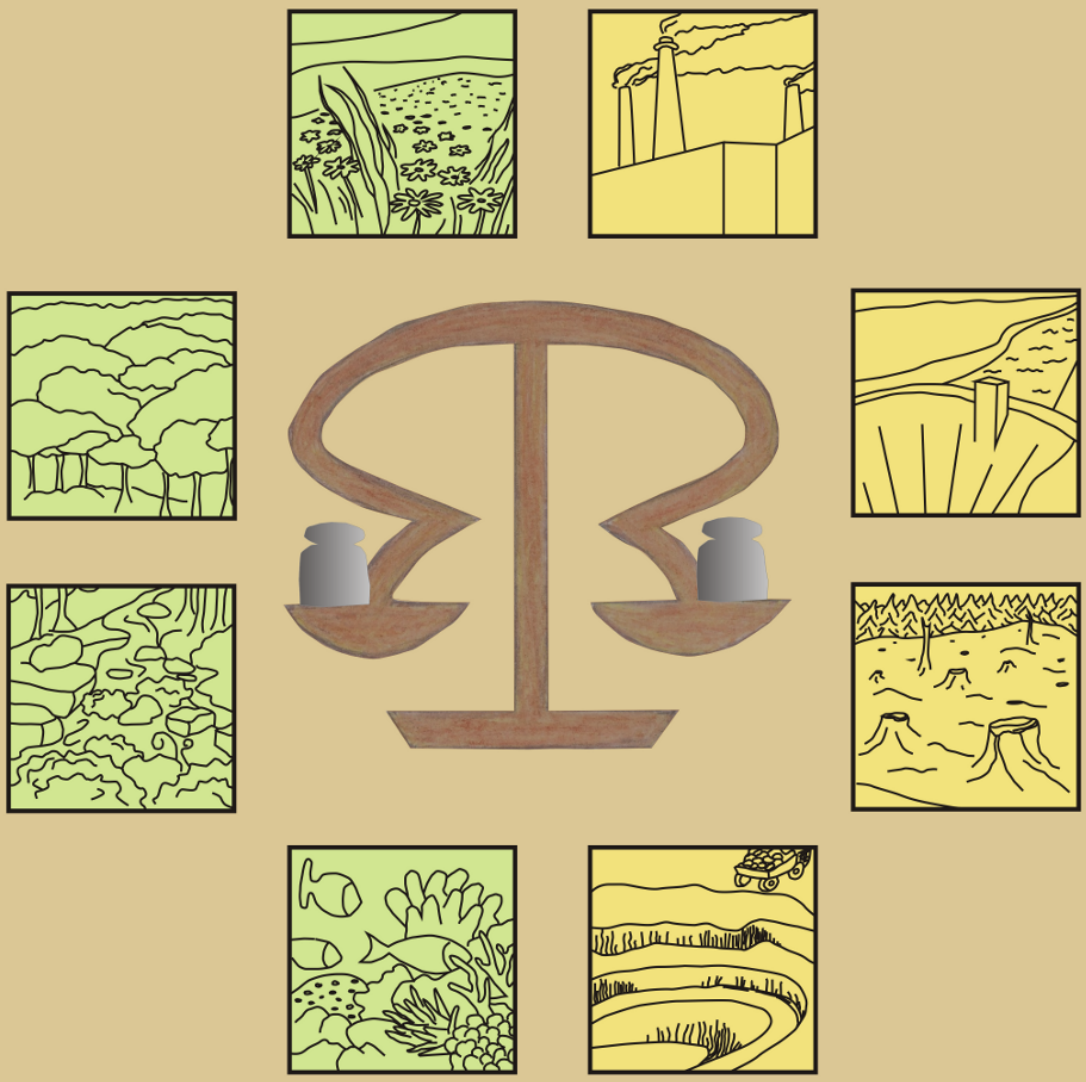
ROBOFF - USER MANUAL

Software for Robust Offsetting, Habitat Restoration, Maintenance and Management



Federico M. Pouzols
Atte Moilanen

version 1.0

RobOff – User Manual
Software for Robust Offsetting, Habitat Restoration, Maintenance and Management

Analysis of alternative land-use options and allocation of conservation resources to multiple actions

RobOff – User Manual Software for Robust Offsetting, Habitat Restoration, Maintenance and Management : Analysis of alternative land-use options and allocation of conservation resources to multiple actions

by Federico M. Pouzols and Atte Moilanen
Technical and language editing: Victoria Veach
Cover page: Aija Kukkala, Atte Moilanen

$Revision: 1.0$

Publication date March 2012

ISBN: 978-952-10-8720-2 (paperback)

ISBN: 978-952-10-8721-9 (PDF)

Development of RobOff was significantly supported by the European Research Council as part of the project Global Environmental Decision Analysis (GEDA) (ERC-StG Grant 260393). We also thank the Finnish Ministry of Environment and the Academy of Finland Centre of Excellence Program 2012-2017 for further support.

Key references about RobOff

Pouzols F. M and A. Moilanen. 2013. RobOff: software for analysis of alternative land-use options and conservation actions. Methods in Ecology and Evolution, in press. DOI: 10.1111/2041-210X.12040.

Pouzols, F. M., Burgman, M., and A. Moilanen. 2012. Methods for allocation of habitat management, maintenance, restoration, and offsetting, when conservation actions have uncertain consequences. Biological Conservation, 153: 41–50.

RobOff: Software for Robust Offsetting, Habitat Restoration, Maintenance and Management

RobOff is a conservation planning framework for allocation of conservation resources to multiple actions, and analysis of alternative land-use options. It has a broad applicability to protection, management, maintenance, restoration, and offsetting. In the RobOff framework, different (conservation and/or development) actions have (uncertain) responses over time for different biodiversity features in different environments. RobOff can evaluate a particular solution and find an optimal allocation of resources for a given setup.

In essence, this software is a decision support tool with an emphasis on uncertainty and time preference, that can solve planning problems with a potentially large set of alternative actions. It aims at solving high-dimensional problems, of the order of thousands or tens of thousands of alternative actions and/or features and environments.

RobOff is currently under active development. Keep an eye on our website for updates: http://www.helsinki.fi/bioscience/consplan

Acknowledgements

Special thanks to John Leathwick, from DOC, New Zealand, for providing the Dam-Forest-River offsetting example setups, and for using, testing and commenting on early versions of RobOff. Great thanks are also due to Mariana Fuentes, from James Cook University, and Heini Kujala, from University of Melbourne, for their feedback and suggestions for improvements to both the software and manual. Thanks as well to Adriano Mazziotta, Mikko Mönkkönen and Janne Kotiaho, from University of Jyväskylä, for providing example data and comments. We are also grateful to all the participants of the first RobOff workshop, held in Melbourne in July/August 2012, for their comments and suggestions. Thanks as well to Brendan Wintle, Terry Walshe, Mark Burgman and other colleagues from the Department of Botany of the University of Melbourne and the Australian Centre of Excellence for Risk Analysis (ACERA), for supporting and hosting the RobOff workshop.

Development of RobOff was significantly supported by the European Research Council as part of the project Global Environmental Decision Analysis (GEDA) (ERC-StG Grant 260393). We also thank the Finnish Ministry of Environment and the Academy of Finland Centre of Excellence Program 2012-2017 for further support.

Table of Contents

List of Figures

List of Tables

Part I. Introduction

Table of Contents

Chapter 1. Introduction

1.1. Aims and purpose

RobOff is a framework and software for conservation planning. It can manage alternative conservation actions and their uncertain effects on biodiversity features in different environments through time, the costs and feasibility of actions, budgetary constraints, time discounting, and robustness requirements.

The RobOff framework and software are specifically intended to complement the many methods that are most appropriate for spatial reserve selection based on static biodiversity patterns (see Section 2.7, “Dealing with connectivity”, p. 38). In RobOff, actions produce different, uncertain responses for features in different environments through time. The focus is specifically on time and uncertainty, and biodiversity pattern is therefore explicitly assumed to be dynamic. To allow for the complexity of problems that arise in habitat maintenance, management, restoration, and offsetting, we make the simplification that any explicitly spatial aspect of analysis is dropped, although workarounds for this simplification are provided. RobOff analyses are intended to answer questions about how much of what kinds of (conservation) measures should be allocated to which environment types. In this role, these results provide an important tool for target setting for systematic conservation planning.

The relative simplicity and flexibility of RobOff should be emphasized. It is key that RobOff builds on uncertain (discrete-time) responses that can be coded into text files with three columns of numbers. Not only is this basic building block very easy to understand, it is very flexible as it allows operation on responses that have no predetermined specific functional form. The uncertainty in responses to actions can be derived as a combination of statistical information (if available) and expert opinion. While setting up analyses using the proposed approach requires specification of a potentially significant amount of information, the very act of collecting the information and thinking about the resource allocation problem will be constructive in terms of understanding the ecology, conservation objectives, and limitations of the particular planning case. Specifying the problem makes the decision process transparent to those who utilize the results.

The RobOff conceptual model has been defined in a way that reduces the amount of information required from the user while still accounting for a number of important factors. Nevertheless, the model is sufficiently flexible so that domain-specific subtleties can be effectively incorporated.

In summary, the analyses defined in the RobOff framework and their implementation in publicly available software make solutions to a significant set of conservation resource allocation problems accessible for the first time to conservation scientists and managers.

For a quick start see Section 1.5, “Software installation and quick start”, p. 9. The RobOff framework is described in depth in Chapter 2, *Framework, Methods and Algorithms*, p. 17. The software and its graphical interface are described in Chapter 3, *The RobOff Software and Command Line Interface*, p. 45 and Chapter 4, *RobOff Graphical User Interface*, p. 87, respectively. Further details on how to use RobOff for different planning needs are provided in Chapter 5, *RobOff analysis setups for common planning needs*, p. 109 and Chapter 6, *Tutorial and examples*, p. 119.

1.2. The RobOff framework in a nutshell

Aim and purpose

- To provide a conservation planning tool that is able to evaluate the outcome of conservation and development actions, and to optimally allocate resources to alternative actions, taking into account their uncertain effects over time on biodiversity features in different environments. RobOff is intended for a wide variety of conservation activities, such as protection, management, maintenance, restoration, and offsetting.

Analyses

- Analysis of uncertainty of conservation value
- Analysis of the time perspective
- Reliability of biodiversity offsets
- Optimal division of resources between conservation actions
- Optimal biodiversity offsetting
- Extraction or area targets

Data

- Definition of environments (habitat types)
- Actions feasible in each environment, including costs and area availability
- Biodiversity features present in each environment
- (Uncertain) responses of features to actions in environments
- Available budget

Features

- Environment (habitat type) and features priorities via weighting
- Methods for dealing with uncertain development of biodiversity features, aiming at robust decisions
- Time discounting (taking time preference into consideration)
- Integrates complementarity and scoring approaches
- Possibility to account for both positive and negative consequences of uncertainty (opportunity and robustness analyses)

- Automated comparison of alternative actions for varying uncertainty conditions

1.3. RobOff inputs and outputs

1.3.1. Inputs

In RobOff, emphasis is placed on two aspects: time and uncertainty. No explicit spatial information (e.g., species distributions, habitat suitability maps, etc.) is required as it operates on per area unit characteristics of habitat types (or environments). RobOff is specifically intended to complement the many methods that are most appropriate for spatial reserve selection based on static biodiversity patterns.

A RobOff setup (a complete definition of a planning problem in the RobOff framework) consists of different types of entities with well defined interactions. The main entities are:

- Environments (which in the simplest case are equivalent to habitat types)
- Actions (including development and compensation/conservation actions)
- Biodiversity features
- Specific responses of features to actions in different environments

In spatial planning tools, data requirements are mainly layers of distributions of features. This usually involves the use of GIS tools. As opposed to this, the essential data required to use RobOff are the responses over time of biodiversity features to actions. Responses are entered as three time-discrete values: best estimate, upper envelope, and lower envelope. This comes down to text files with three numeric columns (or tables with three columns in the graphical interface). Responses describe the evolution of the occurrence levels of biodiversity features for a given action in a certain environment. Consequently, one must first consider what the relevant biodiversity features and environments are and what actions are possible or compulsory.

1.3.2. Outputs

RobOff implements a computational model that aims at evaluating the outcome of sets of conservation and/or development actions under uncertain conditions. Different views on time preference and robustness to uncertainty (such as robust or opportunity values) are supported. RobOff can also find optimal allocations of conservation resources to alternative actions for large problems in reasonable time.

We can distinguish two types of outputs: 1) evaluation of the outcome for a particular allocation of resources to different actions, and 2) optional allocations of resources (i.e., a list specifying the quantity of what resources to allocate to what

actions). In the first case one could evaluate, e.g., a business as usual scenario, an allocation of conservation resources recommended by experts, or an optimal allocation of resources found by using RobOff (second case). In the second case RobOff finds an optimal allocation of resources according to different criteria (including time preference, degree of uncertainty, and robustness requirements).

When evaluating an allocation of resources, RobOff generates diverse outputs in two complementary forms: as text (tables of values) and plots in a graphical interface. The general output of RobOff comprises the following elements:

- Development through time of conservation value and the weak and strong conservation performance ratios across features and/or environments and for the nominal, robust, and opportunity cases. These curves are obtained conditioned on a particular uncertainty horizon for a given time discounting model and rate.
- Time discounted conservation performance ratios as a function of the uncertainty horizon.
- Optimal division of resources between any two given actions, as a function of the uncertainty horizon.

Some of the concepts mentioned here may not be clear at this point if you are using RobOff for the first time. They are described in depth in Chapter 2, *Framework, Methods and Algorithms*, p. 17. These general results consider all the features defined and are produced in three variants:

- All actions and amounts allocated
- No actions at all (do nothing or business as usual scenario)
- A subset of mandatory actions (which can include for instance planned development actions that are unavoidable)

In addition, specific results can be obtained by conditioning the three types of results above to a subset of environmental types, features, or actions.

1.4. A typical RobOff work flow

RobOff has a broad applicability to protection, management, maintenance, restoration, and offsetting. Two parts can be distinguished in a typical RobOff workflow. First, the broad aims of analysis need to be define. Then specific RobOff setups have to be prepared to match the analysis aims.

1.4.1. Specification of aims, ecological model and data

The first step in conservation planning is to define the aim of the computational analysis. Next, one should gather together and organize the required data into one or several RobOff setups. In practice, most of the time invested in a planning

project will likely be used to clearly specify the aims and collect the relevant data, while the computational analysis would take a relatively short time. The steps involved in developing a RobOff analysis are described in more detail below.

1. Make a decision on the broad aims of the planning project. Is it about evaluating the influence of uncertainty or time preference? Is it about evaluating an offset agreement of finding and optimal one? What kind of activities will be considered: protection, management, maintenance, restoration? At first, it is also important to give careful consideration to what biodiversity features and habitats are relevant.

2. Gather and develop the information related to *social and human-related factors and uncertainty*. This includes factors such as costs, availability and feasibility of actions, resources (budget) available, time preference, different criteria to find optimal allocations, etc. What actions are possible and/or compulsory?

3. Develop the *ecological model* (which defines a large part of the RobOff setups that will be generated later on). What actions are relevant for your planning project? What information is available about the responses of features to these actions? What habitat types are relevant and what data is available about them: what is their spatial extent, how much is available for different actions, should or can they be managed as one block or is it not possible because of implementation, administrative, or other reasons? These last questions can help in answering whether habitat types can or should be merged or split into environments (operational blocks of area units to be dealt with as a whole for planning purposes). The degree of uncertainty in the responses of features should be given special attention. It is also key to decide how the importance of different biodiversity features and environments are weighted in relative terms.

4. Develop the data to represent the ecological model and social or human factors. The relevant data can be formatted into text files and/or tables in a graphical interface.

It should be noted that the points above are better accomplished by or in cooperation with experts and different stakeholders. In practice, these stages are part of an iterative process in which the aims, ecological model, and social or human factors must be revisited in the light of new information. In particular, data availability can be a fundamental limiting factor. In fact, before fixing the broad aims and structure of the ecological model it should be ascertained what data is available or can be generated with reasonable effort. In RobOff setups, emphasis is placed on two aspects: time and uncertainty. It is therefore most important to be aware of what time scales are relevant and known, what is the evolution of ecological factors over time, and how to quantify uncertainties in responses.

As a final cautionary note, the relevance and usefulness of results depend strongly on how much data is available. Usual concerns in conservation planning exercises should be kept in mind, including geographical and taxonomic biases. On the bright side, RobOff provides a complete set of tools to account for various forms

of uncertainty. More details about possible steps to define RobOff setups are given in Section 5.1, “General remarks”, p. 109. More details about the RobOff conceptual model can be found in Section 2.2, “RobOff Setups”, p. 20.

1.4.2. Getting a RobOff analysis running

The next points are a step-by-step guide to develop a basic RobOff setup and is intended to get you started. Note, however, that several example setups are included with RobOff. You might want to start by looking at those before developing your own setups. Further information on common planning setups is included in Chapter 5, *RobOff analysis setups for common planning needs*, p. 109.

1. Get a basic analysis running.

    a. Install RobOff and make sure it works correctly by trying one of the examples provided.

    b. Decide your degree of uncertainty (see Section 2.5.1, “Uncertainty Analysis”, p. 25) and concept of time preference (see Section 2.5.4, “Time discounting”, p. 33).

    c. Change these parameters in the graphical interface or in the general settings file (see Section 3.3.1, “General settings file”, p. 50). Check that you are able to generate results (for instance conservation value through time) after these changes.

    d. Try variants of this basic setup by changing the weights of biodiversity features (see Section 3.3.3, “Feature - weights - utility functions file”, p. 59) and environments (see Section 3.3.2, “Environments file”, p. 58). In principle, these first analysis do not require any optimization.

2. Identify a base analysis. After getting some basic RobOff examples running, you need to make a number of decisions regarding the most important entities (both in the ecological model and the human and social factors).

    a. Decide the set of environments (related to the relevant habitat types).

    b. Define the set of actions for every environment and how they are linked to different responses of features. You should also consider their costs and how much area is available for every action.

    c. Define the weights and benefit functions for features and environments.

3. Base analysis and sensitivity analyses. At this point you have a baseline setup from which you will normally generate several variants.

    a. Run the base analysis and have a look at different results, whether in the graphical interface or output text files. You can inspect nominal, opportunity and robust conservation values, sustainability values, etc. You can do this

for the business as usual scenario (i.e., no conservation resources allocated at all).

b. Set a value for the available budget and find an optimal allocation of resources. This can be done for the robust option for example. You can also try to generate the opportunity-optimal solution and compare them.

c. Perform an uncertainty analysis. You need first to define the range of variation of the uncertainty horizon. The output of this analysis will show the sensitivity of results to the degree of uncertainty.

d. Compare pairs of actions with the actions comparison option (or tab in the graphical interface). This comparison will be shown for the range of variation of the uncertainty horizon previously defined.

e. Analyze the sensitivity of results to the available budget. To this end you need to define a range of available budget.

4. Interpretation and post-processing of outputs. Note that RobOff is spatially implicit. Explicit spatial allocation of actions would require to use additional tools such as Zonation. The optimal allocations produced by RobOff can be seen as area targets which can be used as inputs to spatial planning tools (see Chapter 5, *RobOff analysis setups for common planning needs*, p. 109).

5. Evaluation of proposed resource allocations. Quite often at least part of the available conservation resources have been allocated a priori. If you need to evaluate particular predefined allocations, it is possible to include them as mandatory and/or predefined actions with certain amounts allocated to them. You can also use this feature to analyze how the optimal solutions (allocations) change when you forcibly allocate resources to some actions (e.g., following expert recommendations, social or political preferences, etc.).

1.5. Software installation and quick start

If you are using RobOff for the first time, you would typically follow this sequence of steps.

Installation

RobOff can be downloaded from http://www.helsinki.fi/bioscience/consplan, and is available for windows systems as a self-installing binary or zip package.

The windows distribution of RobOff includes the following files:

- The RobOff program (graphical interface): roboffGUI.exe
- The RobOff command line program: roboff.exe
- A user manual (as pdf and html browsable from the graphical interface)
- Example setups

Quick start

First, load some of the examples included in the RobOff distribution. Familiarize yourself with the program options. Some of them are rather intuitive whereas some others will require you to read at least parts of this manual. To change a RobOff setup, use the graphical interface (Chapter 4, *RobOff Graphical User Interface*, p. 87) or follow the detailed description of options and input file formats given in Chapter 3, *The RobOff Software and Command Line Interface*, p. 45.

RobOff setups can be defined via its graphical interface or in a set of input files. To run a RobOff analysis you need to define a complete and consistent setup. This can be done in the graphical interface or via text files. If using text files, at least the following input files are needed:

- General settings file (see Section 3.3.1, “General settings file”, p. 50 for details)
- Environments list file (see Section 3.3.2, “Environments file”, p. 58 for details)
- For each environment, a file with the list of biodiversity features present (see Section 3.3.4, “Set of files: biodiversity features...”, p. 62 for details)
- Files with responses to actions (see Section 3.3.5, “Set of files: responses of biodiversity features”, p. 63 for details)
- File of feature - weights - utility functions (see Section 3.3.3, “Feature - weights - utility functions file”, p. 59 for details)

In the graphical interface there are different tables where all these entities and parameters can be entered interactively.

Note

In numeric fields, whether in the graphical interface or input text files, please use dots as a decimal mark or separator. If your country locale uses commas as the decimal separator, please avoid them and use just dots. Normally, commas will be detected as errors, but there could be unexpected consequences which might be difficult to spot.

Time preference (or discounting model and rate) and degree of uncertainty (and range of variation) are probably the most relevant general settings and should be carefully considered. When you have at least a first guess at these and other settings, You can use the RobOff graphical interface to a) enter them or b) load a setup from files. Once you are able to define a complete setup, you will typically be switching between the three main sections of the graphical interface: 'setup', 'results', and 'optimize'. Alternatively, you will be editing input files and running RobOff analyses from the command line.

The three main sections of the RobOff GUI correspond to three stages between which you can switch at any moment (see Figure 1.1, “Example screen capture of the RobOff GUI: summary of results”, p. 11 and Figure 1.2, “Example screen capture of the RobOff GUI: editing responses in the setup section”, p. 12):

- 'Setup' section: change settings and data model as needed
- 'Result' section: visualize results for predefined or optimized allocations of resources
- 'Optimize' section: define optimization criteria and generate optimal allocations of resources

Figure 1.1. Example screen capture of the RobOff GUI: summary of results

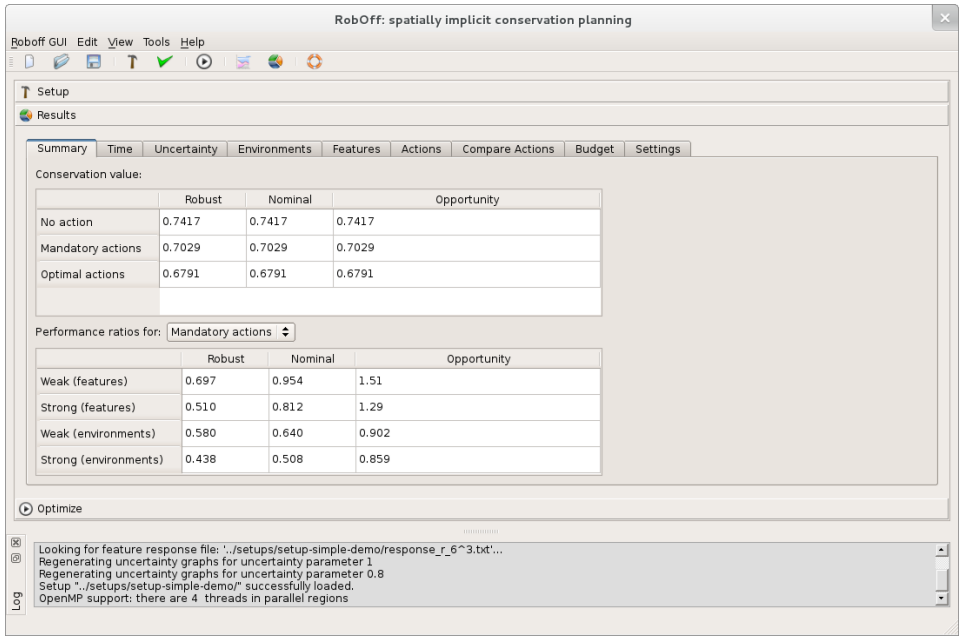
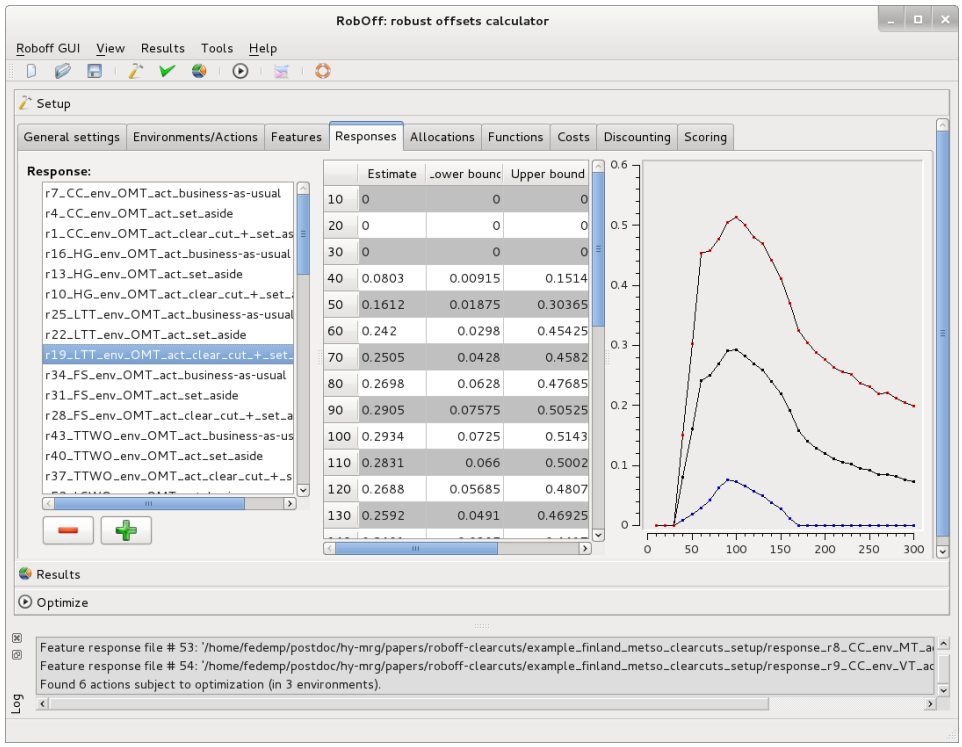


Figure 1.2. Example screen capture of the RobOff GUI: editing responses in the setup section



RobOff outputs can be visualized in its graphical interface, but they are also written into text files. This way it is possible to inspect results with simple text editors, do automated post-processing using scripts or other software, load results into spreadsheet software, etc. Output files are generated into a subdirectory (folder), and includes, among other files:

- A 'Read me' file with basic information about the RobOff analysis
- A summary (table) of main results for different allocations of resources (no allocation, mandatory actions predefined allocation, and optimal allocation, if available)
- A log of how the analysis proceeded, including warnings and notes
- Evolution of conservation value over time
- Conservation performance ratios (i.e., indices of relative sustainability)
- A log from the optimization process and an optimal allocation of resources (i.e., a list of amounts allocated to actions; Optional)
- A pairwise comparison between actions that lists the optimal distribution of resources between them for a varying degree of uncertainty (Optional)

See Section 3.4, “Standard RobOff output”, p. 70 for a detailed description of the output files that RobOff produces.

1.6. Major features

The RobOff framework contains many features that distinguish it from other biodiversity conservation tools. Some major features include:

- Methods for dealing with uncertain development of biodiversity features, aiming at robust decisions
- Time discounting (taking time preference into consideration)
- Integrated use of complementarity and scoring approaches
- Environment (habitat type) and features priorities via weighting
- Automated comparison of alternative actions for varying uncertainty conditions
- Possibility to account for both positive and negative consequences of uncertainty (opportunity and robustness analyses)

Part II. Framework, Methods and Algorithms

Table of Contents

Chapter 2. Framework, Methods and Algorithms

2.1. The RobOff framework

RobOff is a framework and software for conservation resource allocation. What distinguishes RobOff from other frameworks and software systems for conservation planning is its emphasis on:

- Uncertain responses of biodiversity features over time
- Multiple alternative actions, with different effects, costs and feasibility

RobOff focuses on the question of "what to do" rather than on the question of "where". This question is directly relevant for example when investigating the effects of alternative actions in areas that face different threats. RobOff fills a niche in that it concentrates on the uncertain effects through time that alternative (conservation) actions have on different (biodiversity) features in different environments. For references on the RobOff methods and related literature see Section 2.9, “References”, p. 40.

The results produced by RobOff are, in short, of two main types:

- Quantitative evaluation of the outcomes of conservation actions over time (measured as conservation value or sustainability indices).
- Optimal allocations of resources to multiple alternative actions.

Using RobOff requires information on the responses of biodiversity features to different alternative actions. whether from data warehouses, models, or expert knowledge. It is also required to select different criteria and parameters related to preferences and human decision making.

RobOff has been designed as a framework of very broad applicability. It can be applied in very diverse contexts where goals and terminology may be different, but all of which share a common problem structure. Very different types of planning problems can be conceptualized within the RobOff framework. Some examples are summarized in Table 2.1, “Typical or suggested usages of RobOff”, p. 18. These possible and suggested usages are divided into three main categories, the first two of which are education and problem clarification. These are key considerations, since RobOff is a novel framework that complements and differs significantly from traditional reserve selection and related tools. An important component of problem clarification is what can be gained from setting up a RobOff analysis: development of response functions improves understanding about the effects of restoration or management actions. Failure to complete a RobOff setup points out deficiencies in data, models and understanding that is necessary for sound multi-action allocation. If a non-spatial analysis turns out difficult to parameterize, then doing a spatial multi-action analysis will be next to impossible.

Table 2.1. Typical or suggested usages of RobOff. Adapted and extended from (Pouzols & Moilanen 2013).

Type of analysis	Characteristics and tradeoffs analyzed
Education	
Teaching about resource allocation in multi-action problems	How to balance tradeoffs between actions, time preferences, and uncertain responses of biodiversity features.
Insight and problem clarification	
Development of response functions	Development of response functions of features for different actions improves our understanding of restoration and management alternatives.
Identification of data and/or model deficiencies	Failure to develop temporal, non-spatial analysis of conservation actions will reveal deficiencies in our ability to plan restoration or offsetting in a quantitatively well-informed manner.
Management-oriented analysis (simple level)	
Influence of time preference on decisions	Influence of the planning horizon (study period) or time discounting (model and rate) on decisions. Short-term versus long-term goals.
Influence of uncertainty on decisions	Expected value of decision versus associated uncertainty; analysis of opportunities and robustness, or best and worst case; sensitivity of decisions to uncertainty.
Implications of different forms of sustainability	Dependence of decisions on tradeoffs and different views on substitutability between biodiversity features and environments.
Management-oriented analysis (complicated level)	
Balancing of restoration options	Balancing budget allocation between actions to generate an outcome that is balanced and efficient across features and environments.
Biodiversity offsetting	Analysis of best budget allocations to compensatory actions that robustly offset (in a cost-efficient way) specific losses caused by development actions.
Extraction of targets	Focus on finding optimal budget allocation which is converted into targets for (spatial) systematic conservation planning.

The types of analysis defined in the table as management level analyses have relevance for decision making, either via direct generation of management recommendations or via investigation of the robustness of decisions to assumptions. We called simple management level analyses those that focus on the influence of one factor on the decision. Complicated management level

analyses would include allocation of habitat restoration, offsetting and extraction of targets for systematic conservation planning. These analyses involve many or all of the optional components of RobOff analyses described in next sections.

The way conservation value is calculated is described in detail in Section 2.2, “RobOff Setups”, p. 20. We first describe here the motivation, conceptual model (Section 2.2, “RobOff Setups”, p. 20) and output space of the RobOff framework (Section 2.4, “The RobOff output space”, p. 23). Conservation science, systematic conservation planning, spatial conservation prioritization, and conservation resource allocation focus on cost-effective resource allocation. A substantial amount of literature under the keywords *reserve selection* and *site selection* concerns the design and expansion of reserve networks. Most of these studies focus on a binary select-or-not decision (optimization) problem that is applied to implicitly static data describing the distributions' biodiversity features and costs. Software packages specifically developed to address such problems include Marxan, Zonation, ConsNet, and C-Plan.

Clearly, the select-or-not decision problem is a comparatively simple variant of a more complicated decision problem: the "multi-action allocation problem", where the focus is on the allocation of alternative conservation actions. We are interested in habitat maintenance (where the objective is to maintain present biodiversity values), conservation management (where there are specific objectives for features), restoration (where the aim is to return past conservation values that have been lost), and biodiversity offsetting (where the aim is to compensate for ecological damage caused by human activity by managing or rehabilitating alternative sites). The multi-action problem is significantly more complicated and data hungry because actions have a range of costs and responses for features in different areas, making analysis and optimization hard. Ideally, spatial interactions between conservation actions should be taken into account, further complicating the problem. The software Marxan with Zones provides for the allocation of sites to zones that have different conservation treatments (i.e., local representation for a feature depends on the zone the site is assigned to). Generic integer programming approaches have been applied to find optimal habitat restoration strategies.

Uncertainty about what happens to biodiversity features (species, habitats, etc.) through time when landscapes and environmental conditions change and various actions are applied in different locations complicates resource allocation questions. Time is particularly relevant in the context of offsetting, where compensation measures are assigned to ecologically damaging economic activities. Ecological damage is frequently immediate and certain whereas compensation via restoration will appear with a time delay and is not certain.

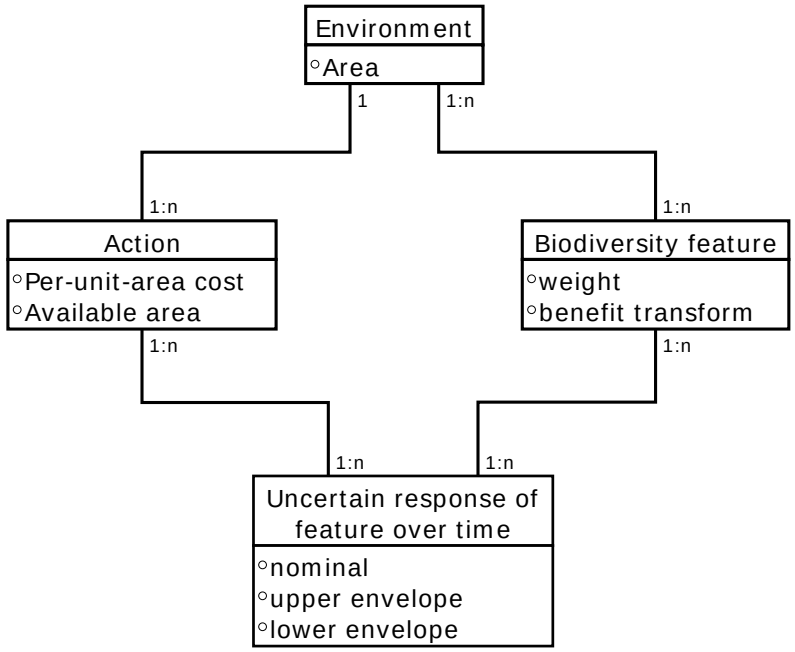
RobOff is intended to complement the many methods mentioned above that are most appropriate for spatial reserve selection based on static biodiversity patterns. We introduce a framework in which actions produce different uncertain responses for features in different environments through time. The focus is specifically on time and uncertainty, and biodiversity pattern is therefore explicitly assumed to be dynamic. To allow for the complexity of problems that arise in habitat maintenance,

management, restoration, and offsetting, we make the simplification that any explicitly spatial aspect of analysis is dropped, although we outline workarounds for this simplification. The present analyses are intended to answer questions about how much of what kinds of (conservation) measures should be allocated to which environment types. In this role, these results provide an important tool for target setting for systematic conservation planning.

2.2. RobOff Setups

In RobOff, a setup is specified to define a particular conservation problem consisting of a number of entities and their interrelations. The conceptual model for these is depicted in Figure 2.1, “Basic components of a RobOff setup”, p. 20 using standard entity-relationship model notation. The core concepts in the setup are environment types, biodiversity features, actions, and the responses of features to actions in different environments.

Figure 2.1. Basic components of a RobOff setup



Calculations that apply to these entities and their interrelations utilize additional information concerning the available budget, costs of actions, availability of areas, conservation benefit transformations, time discounting model, and an uncertainty model. The way these computations are performed and the alternative approaches to optimization are described in the next sections. Here we concentrate on the input data, which are defined as follows:

- An *environment* is an aggregation of sites that in practice are dealt with as a block for planning purposes. The state and evolution of an environment is calculated based on the features present, actions taken, and environment-specific responses of features to actions. In some cases, environments can also correspond to sites or conservation units. Depending on planning objectives and constraints, environments can correspond to habitat types, combinations of habitats, or parts of a habitat. An additional reason to "split" a habitat into multiple environments is the need to define different classes of connectivity (see Section 2.7, “Dealing with connectivity”, p. 38 for an explanation). Whereas in some planning cases environments can directly represent habitat types, in general the correspondence between habitats and environments can be complicated. There are multiple criteria to define environments: differences in the set of possible actions, differences in the features that are present, or differences in the responses of features to actions; connectivity considerations, condition or level of degradation, different threat levels, land ownership issues, borders of administrative units or other socio-political differences, different degrees of uncertainty, etc.

- *Biodiversity features* can be any of those typically considered in conservation planning, including species, vegetation classes, habitat properties, communities, genes, alleles, surrogates, socially relevant factors, etc. These can be defined as simple features or as components for scoring and can be present in different environments at different levels.

- Environment-specific *per-area-unit responses* of features to actions are represented as three time-dependent values: estimated, upper scaling envelope, and lower scaling envelope.

- *Actions in environments* are defined by their costs, the availability of areas where they can be performed, and their implications for features (environment-specific responses). An action can mean any activity (or intervention), or combination of activities related to either development or conservation. The term conservation action is used here to denote multiple and heterogenous types of actions in diverse contexts, such as management, protection, maintenance, restoration, offset, etc. The RobOff software distinguishes three levels or classes of actions (see Section 3.2.1, “Sets of actions”, p. 48): mandatory, preset, and optimal. Mandatory and preset actions are defined a priori by the user, whereas sets of optimal actions are generated automatically for some types of analyses. The onset of an action is implicitly defined by the associated responses. Alternative starting points of a same action should be represented as alternative actions with responses that can potentially differ in more than a delay. See also Section 3.5, “What RobOff does not do directly”, p. 82.

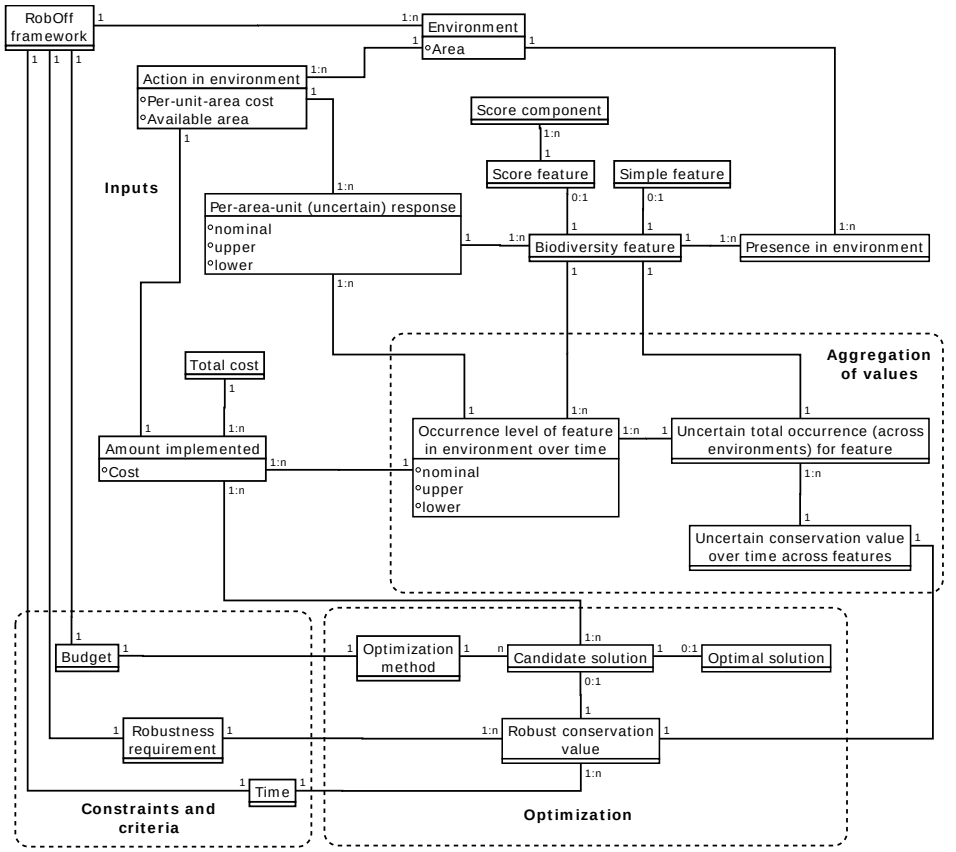
- *Costs* of actions can be specified in three forms: constant values, cost-area functions, or time dependent functions.

- *Implemented amounts* of actions. The amounts and target environments of different actions are assumed to be either mandatorily implemented due to a

priori decisions (e.g., environmentally damaging actions) or are optimized by the software (e.g., compensation actions).

- The *budget* constrains the total cost of the set of actions that can be implemented.
- The uncertainty model and *robustness requirements* allow for accounting of uncertain responses of features to actions. One can evaluate risk/opportunity tradeoffs for some uncertain parameters.

Figure 2.2. RobOff conceptual diagram. Adapted from (Pouzols, Burgman & Moilanen 2012)



The RobOff system records and updates specific characteristics of features in each environment and provides for the linked analysis of features within and across environments.

2.3. Complementarity and scoring

RobOff combines complementarity and scoring, two approaches to conservation that have been seen as fundamentally different. We use a two-step aggregation

process in which some features (e.g., species) are treated as individual biodiversity features in their own right while others are treated as score components that are first aggregated into (uncertain) scores before they are used to assess conservation value. We call the former "simple features" and the latter "scores".

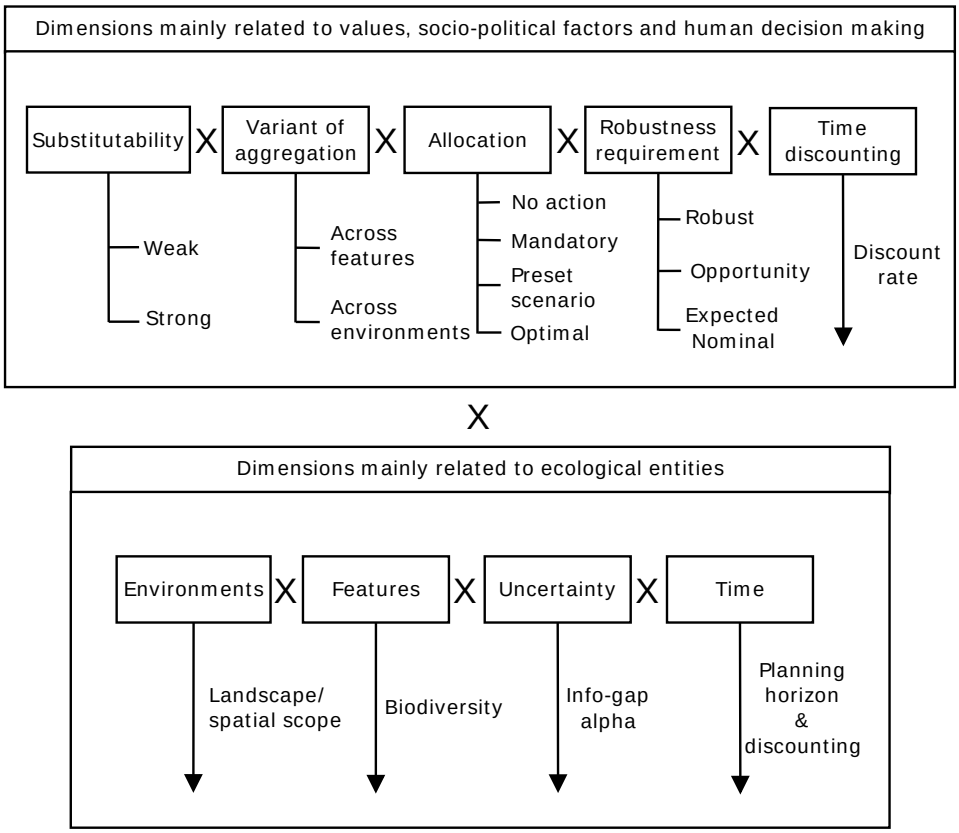
Simple features and scores are entered into complementarity-based computations, thus combining both approaches. We use "complementarity" in the sense that actions and their consequences need to be evaluated jointly, aiming at a well-balanced and cost-effective outcome. An important difference between the two approaches is that score values are typically ranked, whereas complementarity implies iterative computation and typically (or ideally) would require some form of optimization. Also, scoring can be applied without full knowledge of occurrence of biodiversity features across the full landscape, and a score can be computed based on local information, which is an important practical advantage for scores. Scoring is used by many conservation agencies, possibly due to its conceptual simplicity, ease of implementation, and low data demands in the sense that information is only required from the sites of interest, not from the entire landscape.

2.4. The RobOff output space

The inputs to RobOff are descriptions of multiple entities together with their relationships. This includes information about ecological entities, such as biodiversity features and environments, together with the responses of features to actions, and a second class of inputs that are qualitatively different: related to values, socio-political factors, and human decision making, such as costs, substitutability criteria and availability of actions.

While RobOff can produce simple performance indices of conservation value or sustainability, careful evaluation of the consequences of conservation resource allocation requires more in depth analysis and consideration of multiple factors and viewpoints. The outputs generated by RobOff reflect this complexity. In general, the RobOff output space can be seen as the Cartesian product of multiple and heterogeneous dimensions, as shown in Figure 2.3, “Dimensions of the RobOff output space”, p. 24. Some of these dimensions are directly related to ecological entities, such as environments and biodiversity features, while some others lie in the domain of preferences, values and human decision making, such as different time perspectives, different views on substitutability, robustness requirements, or alternative variants of aggregation of value across environments or features.

Figure 2.3. Dimensions of the RobOff output space. Adapted and expanded from (Pouzols & Moilanen 2013).



Note that the figure just shows a conceptually useful classification that is subject to exceptions and different interpretations. For example, the set of actions defined for a particular problem will normally depend on different pieces of ecological information, and conversely the choice of view on substitutability can depend on what features and environments are under consideration. Also, for simplicity some factors are not shown in the figure, such as costs. In the case of costs, one needs to consider the full set of costs of actions possible in every environment (which is described more exactly as a tow-level tree of costs) as one more dimension for the output space. Different results can be obtained for every different set of costs, which can be efficiently visualized in the RobOff GUI.

In this view, RobOff produces multiple outputs that can be considered alternative marginal projections of a high dimensional solution space into a lower dimensional space. RobOff provides the outputs and tools necessary to analyze and visualize the effects of actions along different dimensions. This is possible by processing the results obtained in output files (see Section 3.4, “Standard RobOff output”, p. 70) or, in a more interactively manner, in the results section of the RobOff GUI (see Section 4.3, “Results”, p. 96). Some output dimensions are

categorical and typically require choices among pre-defined sets of options (such as the type of aggregation: across features or across environments), whereas some other dimensions are continuous or numerical (such as the degree of uncertainty) and are typically associated to sensibility analyzes.

2.5. Aggregation of conservation value

In RobOff, conservation value is aggregated across different dimensions: features, environments, actions, and time. Fully aggregated results are useful for a global evaluation of a particular RobOff setup. However, partially aggregated results are needed for a more in depth understanding of the consequences of resource allocation to specific environments or features.

RobOff generates fully aggregated results as well as partially aggregated results which are calculated across either features, environments, actions, time, or a combination of some of these dimensions. The next sections define the RobOff computational model for calculating conservation value. See Section 4.3, “Results”, p. 96 for details on how to use the RobOff graphical interface in order to visualize results across these different dimensions. Alternatively, see Section 3.4, “Standard RobOff output”, p. 70 for details on the output files of RobOff where these results are generated.

2.5.1. Uncertainty Analysis

Uncertainty about the development of biodiversity features through time when landscapes and environmental conditions change and various actions are applied in different locations further complicate resource allocation. The effectiveness of conservation action over time and the impact of development actions are uncertain. The aim is to evaluate the sensitivity of conservation action to this uncertainty. To this end, different approaches can be applied, such as various probabilistic models.

An effective way of addressing uncertainty is to use a non-probabilistic info-gap model formulation of uncertain responses. Info-gap is a decision theory for decision making under severe uncertainty. As a particular case, an envelope-bound info-gap model contains different upper and lower envelopes which delimit the information gap. The uncertain development over time of the representation levels of features is given by upper and lower envelopes around the estimated values. These three sequences (nominal and upper and lower envelopes) need to be specified for each discrete interval.

Info-gap models depend on the horizon of uncertainty, parameter α, which indicates the degree of uncertainty. For $\alpha = 0$, the responses will exactly match the expected values. For $\alpha = 1$, responses span the whole range between the specified envelopes. The envelopes do not need to be symmetric around the nominal values and scale around them as a function of the uncertainty horizon.

With an info-gap formulation, it is possible to account for negative consequences of uncertainty (robustness analysis) as well as positive aspects (opportunity

analysis). An optimal budget allocation can be determined for the robustness or opportunity analyses. This can have a determinant relevance in evaluating the reliability of biodiversity offsetting agreements. Time is particularly relevant in the context of offsetting, where compensation measures are assigned to ecologically damaging economic activities. Ecological damage is frequently immediate and certain whereas compensation via restoration will appear with a time delay and is not certain.

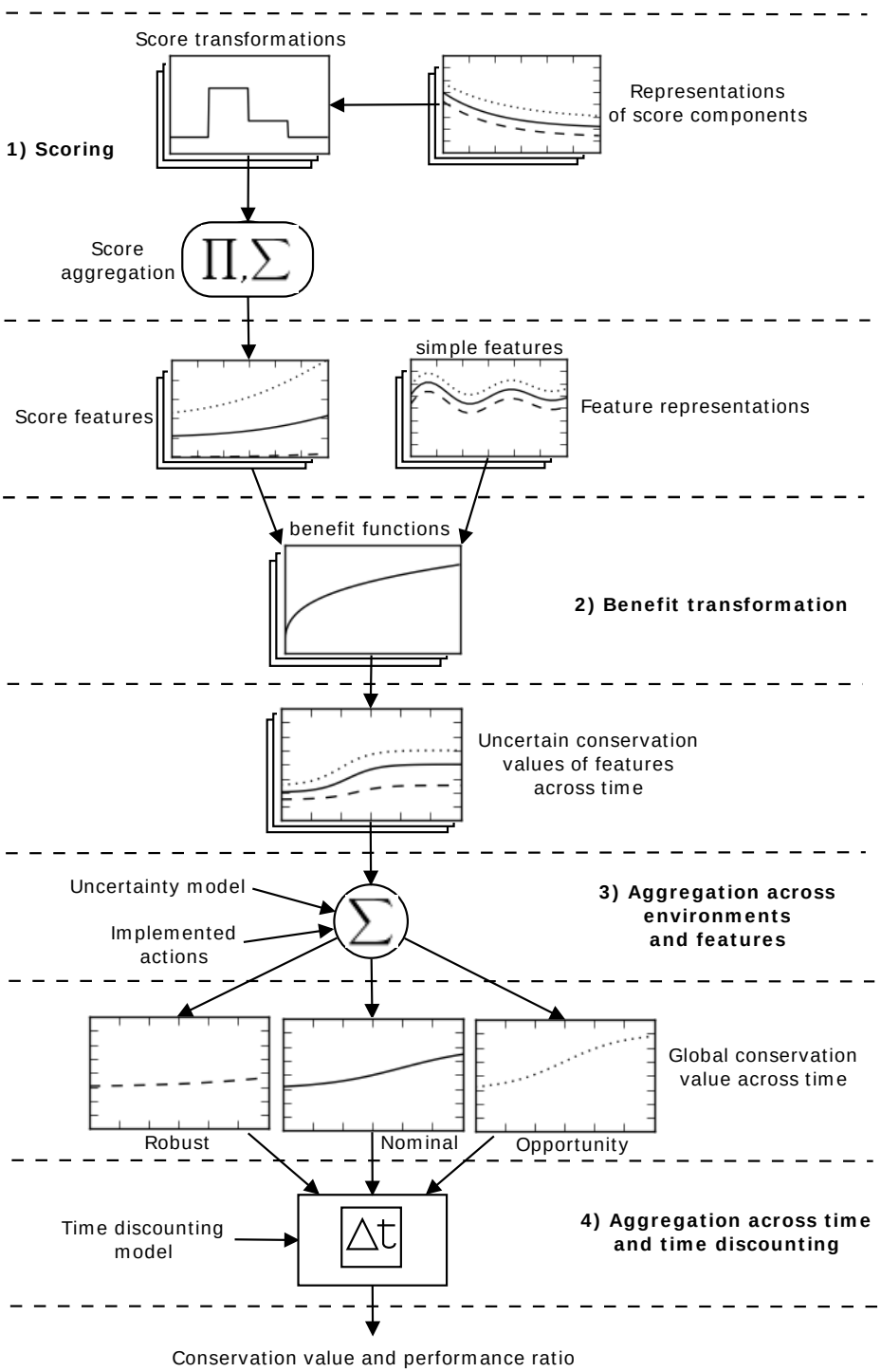
Finally, it should be noted that nothing prevents from estimating the info-gap uncertainty scaling from statistical analysis. The uncertainty in responses to actions can be derived as a combination of statistical information (if available) and expert opinion.

2.5.2. Aggregating conservation value

Conservation value in Roboff is calculated through a number of stages that also generate a global performance index while accounting for uncertainties and time preference. The basic elements required for the computations are the occurrence values of biodiversity features. Discrete time sequences of values for the whole planning horizon must be specified.

A schema of the stages of computation is shown in Figure 2.4, “Flow of aggregation of occurence levels and conservation value in RobOff”, p. 27. These calculations are performed for a proposed set of actions. In stage 1, score component features are evaluated following a scoring approach which includes aggregation into score features. In stage 2, the representation values of score and simple features are transformed into conservation values following a benefit function approach. In stage 3, a global conservation value is calculated by aggregating individual values according to implemented actions and applying an uncertainty model which resolves the uncertainties propagated in stages 1 and 2. Finally, after the time discount stage, scalar values of conservation and performance are obtained.

Figure 2.4. Flow of aggregation of occurence levels and conservation value in RobOff



The mathematical symbols used in this manual are listed in Table 2.2, “Mathematical symbols”, p. 28. Occurrence levels of features are standardized responses that model the relative change in their representation as a consequence of a given action. Representation values are transformed into conservation values using a benefit function with feature-specific weights and functions. By applying specific functions, different mappings from representation of biodiversity features to conservation value can be modeled. In parallel, representations of score features can be transformed to score values by analogous transformations and by an aggregation procedure which produces score features. These are further aggregated with simple biodiversity features in the next stages, effectively integrating scoring and complementarity approaches in a unified approach to conservation resource allocation.

Table 2.2. Mathematical symbols

Symbol	Description
j	Subscript for biodiversity features
i	Sub/superscript for environments
k	Subscript for actions
$\left\{f_j\right\}_{j=1}^{nf}$	Set of features present across all the environments
$\left\{E_i\right\}_{i=1}^{ne}$	Set of environments that make up the landscape
$g_j(.)$	Benefit transformation for feature f_j
w_j	Weight for feature f_j
q_j^i	Condition of the jth feature in environment i
$\left\{a_k^i\right\}_{k=1}^{K^i}$	Set of actions defined for environment E_i
s_k^i	Area (amount of area units) where action a_k^i is performed
$o_{k,j}^i$	Occurrence level of feature f_j in environment E_i when action a_k^i is undertaken. If feature f_j is not present, $o_{k,j}^i = 0 \forall\ k$
r_j^i	Representation value of feature f_j in environment E_i
$\check{o}_{k,j}^i, \check{r}_j^i$	Info-gap expected occurrence level and representation of feature f_j in environment E_i, (discrete-time sequence)
$u_{k,j}^i, u_j^i$	Info-gap upper envelope for occurrence level and representation

Symbol	Description
$l^i_{k,j}$, l^i_j	Info-gap lower envelope for occurrence level and representation
V_j	Conservation value of feature f_j for a given set of actions and amounts implemented
N_{weak}	Weak conservation performance index. Two variants: across features, and across environments. Three robustness requirements: robust, nominal, and opportunity
N_{strong}	Strong conservation performance index. Two variants and three robustness requirements (see N_{weak}).
c^i_k	Cost of action k in environemnt i

Table (Table 2.3, “Key terms for understanding conservation value aggregation in RobOff”, p. 29) lists some key terms needed to understand conservation value aggregation in RobOff. The terms come from various research areas and traditions, and have frequently been used in ambiguous ways.

Table 2.3. Key terms for understanding conservation value aggregation in RobOff

Term	Meaning in RobOff
Occurrence level or value	Level of occurrence of a biodiversity feature. May refer, for example, to observed abundance, richness, probability of occurrence, habitat suitability, or values at different levels, such as genetic diversity or species richness. In the context of ecosystem services the level of occurrence is related to the concept of stock. In RobOff, the term occurrence level is used to denote the local occurrence values of features (i.e., per area unit values).
Representation	Occurrence levels integrated (summed) across the environment (or all environments where the feature occurs).
Conservation value	Benefit converted from representation via a benefit function transform, such as convex increase with diminishing returns or target functions. In the context of ecosystem services, conservation value would correspond (or be related to) the concept of flow.
Retention	In conservation biology, one original definition of retention is what remains in the landscape without conservation action. Here, we use retention in a slightly expanded sense, meaning what remains of a feature in the whole landscape under a certain set of actions. Note that retention applies to representation and conservation value.
Substitutability, interchangeability	In ecological economics these terms refer to the potential for the susbstitution of certain forms of capital for different forms

Term	Meaning in RobOff
	of capital. The major forms of capital are: social, human, natural, and manufactured. In particular, it is an important debate to what extent natural capital can be substituted for man-made capital. This concept is key for differentiating between weak and strong sustainability.
Weak sustainability	This term originates from the field of ecological economics and refers to a view of sustainability in which economy is sustainable if the total capital does not decline. Total capital includes all forms of capital (manufactured, human, natural, and social). This implies that is it acceptable to interchange one kind of capital for another (i.e., all forms of capital are substitutable). Weak sustainability is also referred to as *economic sustainability*. In RobOff this interpretation of sustainability is extended to biodiversity features, meaning that features (and their conservation values) are interchangeable. This means that weak conservation values and performance indices are average values across features.
Strong sustainability	An alternative interpretation of sustainability that establishes that natural capital must be sustained. Or in other words, it is not acceptable to trade natural capital for other forms such as manufactured captital. This is also called *ecological sustainability*.In the RobOff framework this notion is extended to all forms of conservation value (i.e., all biodiversity features must be sustained). It follows that strong conservation values and performance indices are minimun values across features.

It is assumed that two actions cannot overlap geographically, or in other words actions in RobOff can be bundles of actions that take place in a same location. If in actuality two given actions overlap, this should be modeled as three non-overlapping actions: one action, the other action, or both simultaneously. For each environment at least one action must be specified that corresponds to the case where neither development nor conservation actions are undertaken (i.e., no human intervention takes place). Different actions have diverse effects on biodiversity features. Typically the effectiveness of conservation actions over time and the impact of development actions are uncertain. The aim is to evaluate the sensitivity of conservation actions to this uncertainty around the estimated outcome. To this end, uncertainty conditions are addressed by an info-gap model formulation of uncertain responses. We use an envelope-bound info-gap model with different upper and lower envelopes which delimit the information gap model.

The uncertain development over time of the representation levels of features is specified by lower and upper envelopes around the estimated values. These

three sequences need to be specified for each discrete interval in the planning horizon. Info-gap models depend on the horizon of uncertainty, parameter α, which indicates the degree of uncertainty. For $\alpha = 0$, the responses will exactly match the expected values. For $\alpha = 1$, responses span the whole range between the specified envelopes. Generally, as α increases, responses will proportionally deviate from the expected values within an expanding interval. The deviation is not necessarily symmetrical. In a general info-gap formulation, for a representation level r^i_j, the envelope-bound model is $\mathcal{U}(\alpha, r^i_j, u^i_j, l^i_j) = \{\tilde{r}(t) : \tilde{r}^i_j - \alpha(\tilde{r}^i_j - l^i_j) \le \tilde{r}^i_j + \alpha(u^i_j - \tilde{r}^i_j)\}, \alpha \ge 0, r(t) \ge 0$. Note that both the upper and lower bounds scale with α, and can be as high or low as desired.

The computational model of RobOff has been defined in a way that reduces the amount of information required from the user while still accounting for important factors such as time and uncertainty. Nevertheless, the model is sufficiently flexible so that domain-specific subtleties can be effectively incorporated. It is, for instance, possible to enter the same actual feature several times as different factors in the model, possibly in different environments, and either as a simple or score feature. This flexibility can be exploited to incorporate interactions into RobOff models (as described in Section 5.6, “Interactions”, p. 115).

If only nominal values are considered (i.e., there is no uncertainty at all), then all occurrence values are nominal values: $r^i_j = \tilde{r}^i_j$, $o^i_j = \tilde{o}^i_j$. We will first consider this case for simplicity. The calculations involve the info-gap model and its uncertainty horizon, time preference, environments, features, actions, and areas where these are performed. The simplified pseudo-code for the RobOff computations of the conservation value across features is as follows:

For each feature $f_j \in \{f_j\}_{j=1}^{nf}$ in the landscape:

  For each environment $E_i \in \{E_i\}_{i=1}^{ne}$ in the landscape:

    a) Given a proposed set of amounts (areas), s^i_k, where actions are performed, the total representation of feature f_j in environment E_i is:

$$r^i_j(t) = \sum_{k=1}^{K^i} s^i_k o^i_{k,j}(t)$$

    b) The conservation value in this environment is $V^i_j(t) = w_j g_j(r^i_j(t))$

  Endfor

  The retention of representation of feature f_j is $R_j(t) = \sum_{i=1}^{ne} r^i_j(t)$

  The conservation value for feature f_j is $V_j(t) = w_j g_j(R_j(t))$

Endfor

The total conservation value, across all features, is $V(t) = \sum_{j=1}^{n_f} V_j(t)$

An important practical consideration is that, in the implementation of RobOff described in the next chapters, conservation value across features or environments is by default divided by the sum of feature weights or environment weights, respectively. If all the feature or environment weights are 1 this is equivalent to the number of features or environments. Hence the total conservation values produced by RobOff are in fact (weighted) average conservation values across features or environments. This allows for example, to compare conservation values obtained for different setups or different variants of a same setup that may have a different number of features.

From the values derived above, the relative performance for feature f_j (i.e., a ratio between the conservation value obtained for the selected actions and the value obtained when no action is performed), is

$$N_j = \frac{V_j(t)}{V_{j,\varnothing}(t)}.$$

These ratios are measures of the conservation value of the total retention in the environments considered. $V_{j,\varnothing}(t)$ and denotes the conservation values obtained when no human intervention takes place (i.e., business as usual scenario), or, in other words, when only the “do nothing” action is performed everywhere throughout the landscape. Note that the aggregation process described here involves a number of non-linearities: responses of features, benefit functions, etc.

2.5.3. Weak and strong sustainability

We can define a performance ratio of sustainability, relating what is obtained for a particular allocation of resources against what would be obtained if no action is undertaken (business as usual scenario). Four options can be distinguished depending on whether sustainability is considered in a weak or strong sense and whether it is calculated across features or environments.

Strong sustainability implies that kind must be replaced with kind, whereas weak sustainability implies that some particular type of loss can be compensated by gains of a different kind. Measures of strong sustainability, like the performance ratio defined here, emphasize the non-substitutability or non-interchangeability of biodiversity features.

Weak sustainability across features

$$N^{\text{feat}}_{\text{weak}}(t) = \frac{\sum_{j=1}^{n_f} V_j(t)}{\sum_{j=1}^{n_f} V_{j,\varnothing}(t)} = \frac{V(t)}{V_{\varnothing}(t)}$$

Strong sustainability across features

$$N^{\text{feat}}_{\text{strong}}(t) = \min_j \frac{V_j(t)}{V_{j,\varnothing}(t)}$$

Weak sustainability across environments

If we denote the aggregation of the environment specific conservation values of all the features present in environment E_i with $V^i(t)$, the performance ratios for environments are as follows:

$$N^{\text{env}}_{\text{weak}}(t) = \frac{\sum_{i=1}^{n_e} V^i(t)}{\sum_{i=1}^{n_e} V^i_{\varnothing}(t)}$$

Strong sustainability across environments

$$N^{\text{env}}_{\text{strong}}(t) = \min_i \frac{V^i(t)}{V^i_{\varnothing}(t)}$$

Note that strong sustainability as calculated here can be seen as a variant of minimax optimization where minimum retention is maximized (or maximum loss is minimized).

2.5.4. Time discounting

Evaluation of offsetting situations and conservation action in general should also consider when compensation is attained. To this end, RobOff integrates various models for time discounting. For any of the considered criteria (weak or strong) and variants (across features or across environments), the last stage in the conservation value aggregation process consists of applying a time discounting model, which produces a scalar conservation performance ratio:

$$N_{\text{discounted}} = \frac{\sum_t \delta(t) N^{\text{variant}}_{\text{criterion}}(t)}{\sum_t \delta(t)}$$

where $\delta(t)$ are discount factors that follow a given discounting model. The most common models are hyperbolic, quasi-hyperbolic, and exponential, all of which are supported in RobOff. For a concise review of time discounting models please refer to (Green and Myerson, 2004). These depend on the discount rate parameter which has to be set depending on time preference.

2.5.5. Robust and opportunity performance indices

The calculations and ratios defined above correspond to the case where there is no uncertainty, such that the uncertainty horizon of the info-gap model is $\alpha = 0$, $o^i_{k,j} = \check{o}^i_{k,j}$, and $r^i_{k,j} = \check{r}^i_{k,j}$. In order to account for the sensitivity of the models to uncertainty, the representation values in the algorithm above are replaced with their corresponding info-gap models derived from their expected values, upper and lower envelopes, and a variable uncertainty horizon.

By maximizing or minimizing conservation value in the info-gap models it is then possible to calculate maximal conservation values, (V^+_j, across features; $V^{i,+}$across environments) and minimal conservation values (V^-_j, across features; $V^{i,-}$ across environments).

To calculate the robust version of the performance ratios across features, $V_j(t)$ are replaced with $V^-_j(t, \varnothing, \mathcal{U})$, and $V_{j,\varnothing}(t)$ with $V^+_{j,\varnothing}(t, \varnothing, \mathcal{U})$. $V^+_{j,\varnothing}$ are obtained by maximizing conservation value in the info-gap model, whereas V^-_j are obtained by minimizing in the same interval. That is, the ratios are calculated between the minimum conservation value that can be obtained within the region of uncertainty around the nominal values for a proposed set of actions and the conservation value that could be obtained if no action is performed.

Similar reasoning applies to the ratios for environments. In the simple case that benefit functions are increasing monotone, maximum values are attained for the upper envelope values of representation. For decreasing monotone benefit functions, similar reasoning can be applied. However, in a more general case, these quantities must be calculated by maximization or minimization in an interval. Similarly, for calculating the opportunity version of the performance ratios, $V_j(t)$ is replaced with $V^+_j(t, \alpha, \mathcal{U})$ and $V_{j,\varnothing}(t)$ with $V^-_{j,\varnothing}(t, \alpha, \mathcal{U})$, which likewise require minimization and maximization in intervals.

Robust weak sustainability across features

$$N^{\text{feat,-}}_{\text{weak}}(t) = \frac{\sum_{j=1}^{n_f} V^-_j(t)}{\sum_{j=1}^{n_f} V^+_{j,\varnothing}(t)} = \frac{V^-(t)}{V^+_{\varnothing}(t)}$$

Robust strong sustainability across features

$$N^{\text{feat,-}}_{\text{strong}}(t) = \min_j \frac{V^{-}_{j}(t)}{V^{+}_{j,\varnothing}(t)}$$

Robust weak sustainability across environments

$$N^{\text{env,-}}_{\text{weak}}(t) = \frac{\sum_{i=1}^{n_e} V^{i,-}(t)}{\sum_{i=1}^{n_e} V^{i,+}_{\varnothing}(t)}$$

Robust strong sustainability across environments

$$N^{\text{env,-}}_{\text{strong}}(t) = \min_i \frac{V^{i,-}(t)}{V^{i,+}_{\varnothing}(t)}$$

Finally, opportunity indices are calculated as ratios of maximal value for a proposed allocation of resources against minimal values for the business as usual allocation.

2.6. Optimizing resource allocation

In RobOff, aggregation of (conservation) value is done across environments, features and time. Once a RobOff setup has been established, it is not trivial to find an optimal allocation of resources for the different actions available. In this case, *optimal allocation* refers to an allocation of resources such that the conservation performance ratios defined above are maximal, given all the required parameters and choices of criteria such as weak or strong sustainability. This can be interpreted as the most cost-effective solution (allocation of resources among alternative actions) for a given budget.

The optimization process must also account for the area availability constraints and different costs of actions, and it is subject to the set of mandatory actions. Also, an optimal budget allocation can be determined for the robustness or opportunity immunity functions (robustness or opportunity versions of the performance ratios that will be defined below). In their general form both the robustness and opportunity optimization cases are highly nonlinear problems and exact methods for linear and integer programming are not appropriate. Effectively, various sources of non-linearities are involved in the aggregation of conservation value, including cost functions, responses of features, benefit functions, etc.

Different alternative and complementary optimization methods can be used in the RobOff framework. For problems where the number of actions is lower than about

10, exhaustive grid search methods with local search operators can be employed in order to find a global optimum. However this approach does not scale well for a larger number of actions. For a restricted subset of problems, it would be possible to use convex optimization methods that are exact and scalable. However, this can only be applied when all the benefit functions as well as the constraints are convex (i.e., a straight line segment between two points in the function always lies above the function). For instance one can have a non-convex constraint if the per area unit cost of an action depends on the total area in a non-convex manner. In a general case, stochastic optimization methods such as evolutionary optimization approaches and genetic algorithms in particular are an efficient means for finding satisfactory solutions in reasonable time.

Five different optimization methods are presently implemented in the current version of RobOff. Some general properties of these methods currently are summarized in Table 2.4, “Scalability and optimality of the optimization methods supported by RobOff”, p. 37, which gives an idea of the applicability of the methods to different planning problems, and Table 2.5, “Speed and ease of use of the optimization methods supported by RobOff”, p. 37, which is informative about the practical usability of the software tool. The first method is based on allocations of random amounts of resources to actions picked at random. It is meant as a reference method that provides baseline results to compare against more elaborated methods.

The simplest method is a greedy search algorithm. It is meant as a fast way to obtain a first approximation for large problems, and can be effective and nearly optimal in some cases. In the third method, grid-based exhaustive search, the search space (division of money into actions in environments) is divided into a multidimensional grid at a resolution specified by the user. All feasible solutions as defined by this grid are evaluated. Exhaustive search is reliable, easy to understand, deterministic and straightforward to implement, but suffers from the problem that computation time increases exponentially with the number of dimensions (alternative actions). The fourth optimization method expands the grid-based exhaustive search by adding a local search process that is executed starting from each feasible grid point. The last optimization method implemented in RobOff is stochastic global search using a genetic algorithm, which is applicable to very large and complex problems but which cannot guarantee optimality of solutions. For details on how to use these methods, see Section 3.3.1, “General settings file”, p. 50 and Section 4.4, “Optimization”, p. 103.

Table 2.4. Scalability and optimality of the optimization methods supported by RobOff: general properties. Adapted and expanded from (Pouzols & Moilanen 2013).

Method	Approximate maximum problem size	Optimality
Random	Virtually unlimited	None
Greedy	Virtually unlimited. Hundreds or thousands of action-environment pairs are no issue.	Potentially very suboptimal. Good first and fast approximation in some cases.
Grid-based exhaustive search	Up to 8-10 dimensions (action-environment pairs) within 1 hour.	Optimal for the budget resolution used
Exhaustive search + local search	Similar as with exhaustive search	Optimal for convex problems
Stochastic global search (genetic algorithm)	Tens or hundreds of action-environment pairs within tens of minutes or 1 hour.	Not guaranteed but currently the only practical and effective option for large non-convex problems

Table 2.5. Speed and ease of use of the optimization methods supported by RobOff: general properties. Adapted and expanded from (Pouzols & Moilanen 2013).

Method	Speed of computation	Ease of use
Random	Extremely fast, seconds	High
Greedy	Extremely fast, seconds	High
Grid-based exhaustive search	Fast with small dimensions but becomes exponentially more time consuming with the number of actions	High
Exhaustive search + local search	As with exhaustive search, but an order of magnitude or more slower	High
Stochastic global search (genetic algorithm)	Fast but growing with the number of environments, features and actions. Computation time can be limited. Several repetitions are recommended to verify convergence	Medium

2.7. Dealing with connectivity

The RobOff framework considers alternative conservation actions and their uncertain effects on biodiversity features in different environments through time, costs and feasibility of actions, a budget, time discounting, and robustness requirements. The compromise in making this combination of factors analytically operational was to omit the explicit consideration of space.

If the RobOff model was spatially explicit, the already moderately large dimensionality of the data would have been multiplied by the dimensionality of the landscape. Assuming a GIS-derived, grid-based landscape description, this could result in a data dimension in the order of 10^5 to 10^7 times its present size, making data demands prohibitive. The obvious drawback of the non-spatial representation is that the present analysis will not suggest in which locations conservation action should be undertaken. The spatial element is important. Rather than ignoring it, we can suggest three approaches that provide spatial context for the analyses above:

- The first is to account for connectivity by defining different variants of environments for different categories or degrees of connectivity. One could have different responses in, say, isolated forest fragments, moderately connected forest fragments, and well connected forest areas that are part of a semi-continuous forest landscape. This is a compromise solution that has the implication that data dimension is multiplied by the count of connectivity categories added.

- As a second approach, it is possible to combine the analysis with expert opinion. The present analysis will suggest how much of which kinds of actions should be undertaken in what environments. Local experts can then apply their insight to specify exactly where action should be taken, accounting for any external information about land availability and socio-political considerations.

- Third, RobOff methods can be used in a first stage to come up with area objectives or targets for different actions. Then, in a second stage one can use software for spatially explicit planning, such as Zonation, Marxan, Marxan with Zones or ConsNet to suggest explicit spatial allocations based on potential management scenarios.

Overall, these ways of accounting for connectivity and location provide a practical solution to the dimensionality problem, without requiring a fully explicit spatial model.

It should be noted that in fact there is a gradient of options (the extreme cases would be: a) the whole landscape aggregated into a single environment, and b) a specific environment for every single site). In practice, there is an important difference in that in the second option, areas with different degrees of connectivity are dealt with differently in RobOff calculations (possibly using different responses, etc.). In short, optimization results would be more fine-grain area targets in the second case, with targets specific to areas with certain degrees of connectivity.

2.8. Assumptions and limitations

Conceptual simplifications

RobOff is a spatially implicit conservation planning framework. As such, it does not take into account explicit information about the spatial distribution of biodiversity features and habitats or environments. This is a necessary compromise to reduce data demands and to make it possible to solve allocation problems within reasonable computational resources. See Section 2.7, “Dealing with connectivity”, p. 38 for effective ways of dealing with connectivity.

In addition to space, other factors have been intentionally omitted in the RobOff framework. It does not account for process-based planning and dynamic interactions between features through time. While a conceptual limitation, this is perhaps less relevant in practice, as the data demands of including dynamic interactions would in most cases be prohibitive. In such a case, one would be approaching a parametric dynamic model of landscape and metacommunity dynamics, something that can, albeit with considerable difficulty, be implemented using generic dynamic modelling software. Note however, that it is possible to consider simple forms of interactions in RobOff by defining simple or score features that correspond to interactions (see Section 5.6, “Interactions”, p. 115 for details).

Implementation limitations

In principle there are no hard limitations to the computations described in previous sections. See Section 3.7, “Data limitations and system requirements”, p. 83 for more specific implementation details of the RobOff software.

Memory limitations

RobOff imposes no hard limits on the size of setups. In principle, there is no limitation on the number of environments, features, actions, etc., that a setup can have. Processing thousands or tens of thousands of features and environments should be no issue with current commodity PCs.

Computation time limitations

Computational time will in most cases be the limiting factor. RobOff computation time will generally increase linearly with the number of features and environments. The most constraining factor is the number of alternative actions.

When optimizing the allocation of actions, the number of alternative actions can quickly become the major practical limiting factor. Among the optimization strategies supported by RobOff, exhaustive grid search methods can be useful for a reduced number of alternative actions (typically less than 10). When the number of alternative actions is large you should definitely employ the genetic algorithm optimization strategy. The graphical interface (see Chapter 4, *RobOff*

Graphical User Interface, p. 87) provides feedback that allows you to estimate how long an optimization process will take. It is up to the user to decide to switch to faster optimization strategies if needed. See Section 2.6, “Optimizing resource allocation”, p. 35 for more detatils on the effect of different parameters on the computation time of the supported optimization methods.

2.9. References

The RobOff framework and software

Pouzols F. M and A. Moilanen. 2013. RobOff: software for analysis of alternative land-use options and conservation actions. Methods in Ecology and Evolution, in press. DOI: 10.1111/2041-210X.12040.

Pouzols, F. M., Burgman, M., and A. Moilanen. 2012. Methods for allocation of habitat management, maintenance, restoration, and offsetting, when conservation actions have uncertain consequences. Biological Conservation, 153: 41–50.

Moilanen, A., van Teeffelen, A., Ben-Haim, Y., and S. Ferrier. 2009. How much compensation is enough? A framework for incorporating uncertainty and time discounting when calculating offset ratios for impacted habitat. Restoration Ecology, 17: 470-478.

Benefit function approach to reserve selection

Arponen, A., Heikkinen, R., Thomas, C.D., and A. Moilanen. 2005. The value of biodiversity in reserve selection: representation, species weighting and benefit functions. Conservation Biology, 19: 2009-2014.

Moilanen, A. 2007. Landscape Zonation, benefit functions and target-based planning: unifying reserve selection strategies. Biological Conservation. 134: 571-579.

Laitila, J. and Moilanen, A. 2012. Use of many low-level conservation targets reduces high-level conservation performance. Ecological Modelling. 247: 40-47.

Time preference and time discounting

Green, L. and J. Myerson. 2004. A Discounting Framework for Choice with Delayed and Probabilistic Rewards. Psychological Bulletin, 130: 769-792.

Loewenstein, G. and J. Elster. 1992. Choice Over Time. Russel Sage Foundation, New York, USA.

Spatial prioritization and reserve selection

Moilanen, A., Franco, A.M.A., Early, R., Fox, R., Wintle, B., and C.D. Thomas. 2005. Prioritising multiple use landscapes for conservation: methods for large multi species planning problems. Proceeding of the Royal Society of London, Series B, Biological Sciences, 272: 1885-1891.

Watts, M.E., Ball, I.R., Stewart, R.S., Klein, C.J., Wilson, K., Steinback, C., Lourivald, R., Kircher, L., and H. P. Possingham. 2009. Marxan with Zones: Software for optimal conservation based land- and sea-use zoning. Environmental Modelling & Software, 24: 1513–1521.

Moilanen, A., Meller, L., Leppänen, J., Montesino Pouzols, F., Arponen, A., and H. Kujala. 2012. Zonation spatial conservation planning framework and software v. 3.1, User manual, 287 pp. http://www.helsinki.fi/bioscience/consplan

Conservation resource allocation

Kukkala, A. and A, Moilanen, A. 2013. The core concepts of spatial prioritization in systematic conservation planning. Biological Reviews. 88: 443-464.

Moilanen, A., Wilson, K.A., and H.P. Possingham. 2009. Spatial Conservation Prioritization: Quantitative Methods and Computational Tools. Oxford University Press, Oxford, UK.

Margules, C.R. and S. Sarkar. 2007. Systematic Conservation Planning. Cambridge University Press, Cambridge, UK.

Approaches to conservation: complementarity and scoring

Moilanen, A. 2008. Generalized Complementarity and Mapping of the Concepts of Systematic Conservation Planning. Conservation Biology. 22, 6:1655-1658.

Burgess, N.D., Hales, J.D., Rickets, T.H., and E. Dinerstein E. 2006. Factoring species, non-species values and threats into biodiversity prioritisation across the ecoregions of Africa and its islands. Biological Conservation, 127: 383-401.

Part III. The RobOff Software and Command Line Interface

Table of Contents

Chapter 3. The RobOff Software and Command Line Interface

3.1. Introduction and important general information about files

This chapter describes how to use the RobOff command line. There are two alternative approaches to using the RobOff software: command line or graphical user interface. If you are not familiar with command line tools, or editing plain text files, you should rather skip this section and use the RobOff graphical user interface (see Chapter 4, *RobOff Graphical User Interface*, p. 87). The RobOff GUI provides custom dialogs for editing RobOff setups and different visualization options.

Using RobOff from the command line usually involves creating and editing a set of text files that contain the description of a RobOff setup. The next sections describe:

- What types of analyses can be performed using the RobOff command line tool, including a concrete list of configuration options and command line options
- The types of input files that are either required or optional
- The output files generated by RobOff for different types of analysis

First, we highlight a few important practical issues.

File names

In principle, RobOff imposes no limits on the length and characters used in file names and file paths. Both the slash (/) and backslash (\) characters can be used as directory separators. Be aware, however, that special characters such as spaces, quotation marks, brackets, and punctuation marks in general are a very common source of problems. You will likely experience issues with some operating systems and/or third software. We recommend to avoid such special characters when possible and to use them with care. For example, using the underscore character (_) instead of spaces can save a lot of trouble.

File paths

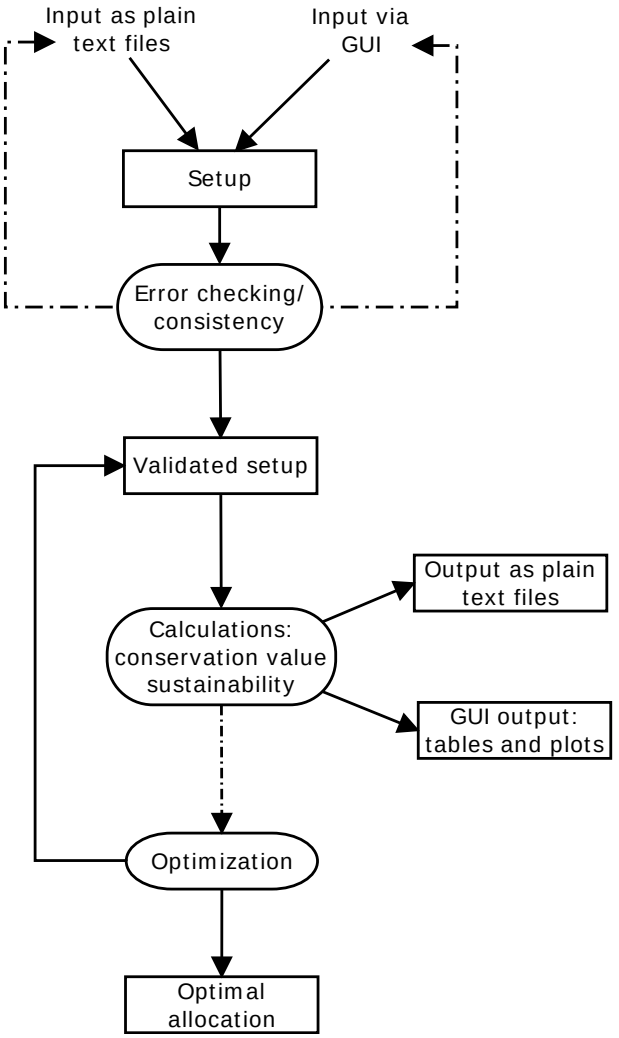
In RobOff setups, all the file paths are interpreted as relative to the location of the main settings file. This rule applies to every single file referred to in any file that is part of a RobOff setup. Always keep in mind that paths are required if you use files located in different directories. For instance, if the response files are located in a `responses` subdirectory or folder, then the responses must be named by their relative path, such as `responses/response_business_as_usual.csv`, `responses/response_restoration1.csv`, etc. Similarly, it is possible to use

full paths, such as `C:/roboff/responses/response_restoration1.csv`, or `/home/foo/roboff/responses/response_restoration1.csv`. Some comprehensive examples can be found in the RobOff distribution, and some of them are described in Chapter 6, *Tutorial and examples*, p. 119.

3.2. Running RobOff from the command line

The general syntax of a RobOff command line is just a sequence of options. Whether you use RobOff from the command line or the GUI, three main stages or functional blocks can be distinguished: *editing a setup*, *visualization of results*, and *optimization of resource allocation*, as summarized in Figure 3.1, “Running RobOff: stages from inputs to outputs”, p. 46.

Figure 3.1. Running RobOff: stages from inputs to outputs



When you use the RobOff command line, you will typically edit input files first. To do so you can use your favorite text editor (or spreadsheet software), but always remember to save files as plain text. The RobOff command line will first check the consistency of the input files you provide. If no errors are found, RobOff builds a validated setup and performs certain calculations (which may include optimization of resource allocation). The results of these calculations are written into output files. In actuality the RobOff command line can be used together with the GUI. You can actually create setups via the GUI and save them for later processing using the RobOff command line. Likewise, results generated using the command line can later be opened and visualized using the GUI. However in this chapter we concentrate on the workflow parts (see Figure 3.1, “Running RobOff: stages from inputs to outputs”, p. 46) that are directly performed using the RobOff command line.

This is the output obtained if you type `roboff` or `roboff -v` as a command:

```
RobOff - software for allocation of conservation effort with multiple actions.
Version 1.0.0rc4

Copyright (C) 2011-2013
Biodiversity Conservation Informatics Group
Center of Excellence in Metapopulation Biology
University of Helsinki
http://www.helsinki.fi/bioscience/consplan

Usage: roboff [OPTIONS]

Options:
   -h, --help                 Print help and exit.
   -v, --version              Print version and exit.
   -V, --verbose              Verbose output.
   -s FILENAME|DIR, --setup   Set setup file/directory.
   -o DIR, --output           Set output directory name.
   -p, --optimize-alloc       Find an optimal allocation of resources.
   -u, --uncertainty          Perform uncertainty analysis of cons. value.
   -k, --compare-actions      Perform comparison of actions.
   -b, --budget-analysis      Perform budget analysis.
   -t FILENAME, --time-disc   Use time discounting file.
                              (overrides time discounting options in setup).
   -a FILENAME, --input-alloc Use allocation file as input.
                              (overrides mandatory and preset allocations).
                              (overrides resource allocation options).
   -f, --per-feature-out      Output feature specific result files.
                              (potentially big)
   -e, --per-environment-out  Output environment specific result files.
                              (potentially big)
   -r INTEGER, --parallel     Number of parallel units/threads.
```

An example of simple RobOff command line could be:

```
roboff -V --setup setup/example.setup --output example-output --optimize-alloc
```

The command line options can be used in any sequence. Some other simple examples are:

```
roboff --setup setup/example2.setup --output example2-output --budget-analysis
```

```
roboff --setup setup/ex3.setup --output ex3-output --budget-analysis --uncertainty --
optimize-alloc
```

The `--output` option specifies a directory name, and RobOff generates at least a handful of files inside output directories. It is strongly advisable to use different directory names for different commands (or RobOff runs) to avoid overwriting files. As it can be seen, certain types of analyses are enabled with specific command line options, including the uncertainty analysis, the comparison of actions, the analysis of sensitivity to budget variations, and the optimization of resource allocations.

After running RobOff, it is always advisable to check the `0.readme.txt` and especially the `2.log.csv` files that are generated into the output directory. These files include notices about suspicious events detected throughout the RobOff calculations as well as warnings and errors regarding inconsistencies in input data.

Other output files generated by RobOff into the output directory include information about the evolution in time of conservation value, ratios of sustainability, and optional results for uncertainty analysis, optimization, comparison of actions, optimized allocation of resources, etc. These are further described in Section 3.4, “Standard RobOff output”, p. 70.

3.2.1. Sets of actions

When using RobOff for resource allocation optimization it is important to keep in mind that three different sets of allocations of resources to actions are considered in the RobOff software. An allocation is simply an amount of area where an action is undertaken in a given environment. As such, individual allocations are specified as three parameters: environment, action, and amount (area extent). The allocation sets are:

- Mandatory: These are forcibly included actions. A typical example would be unavoidable development actions. Another possible case is actions for which resources have already been allocated beforehand.
- Preset: This set is considered for convenience. The resource allocations included in this set are added to the mandatory allocations before performing any optimization. This allows the user to, for example, explore what is the impact in the final optimal allocation of forcibly including certain actions (which may reflect, for example, political preferences, a priori expert opinion, etc.).
- Optimal: This set will be automatically defined by RobOff by maximizing conservation value, given the sets of mandatory and preset actions and the available budget.

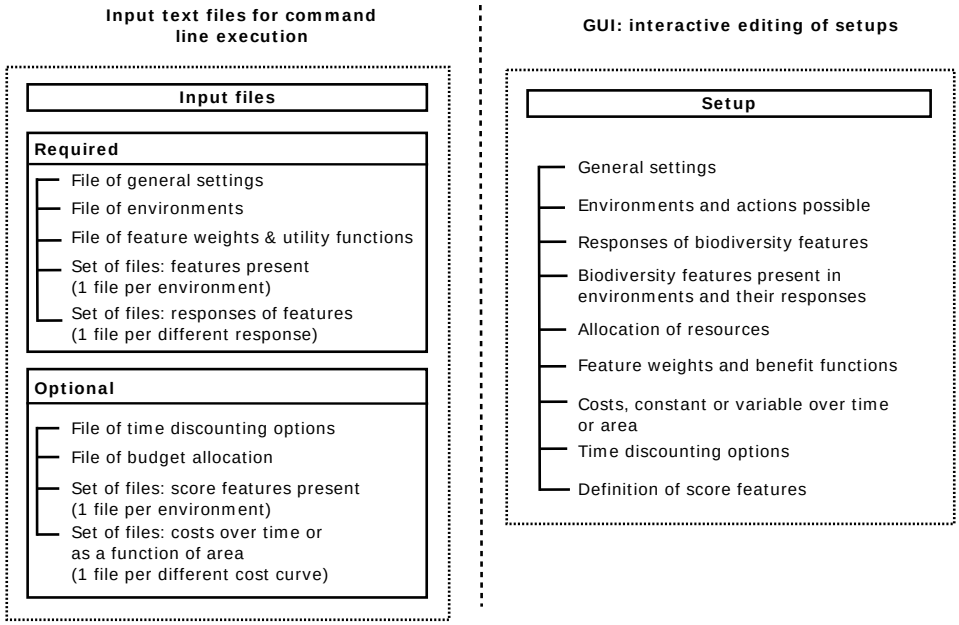
These sets of allocations are specified in the budget allocation file (see Section 3.3, “Input files and settings”, p. 49). When looking for the best set of actions to offset the impact of development, development actions would typically be allocated in the set of mandatory actions. More generally, the sets of mandatory and preset

actions are useful when planning is done incrementally, such as in the case of incremental reserve design.

3.3. Input files and settings

A complete and correct RobOff setup needs to be defined in a set of files before running RobOff from the command line. Some of these files are always required while some others are optional. A diagram that summarizes the different types of files is shown in Figure 3.2, “Set of input files and their equivalent GUI dialogs”, p. 49. These files list the entities and attributes that make up a RobOff setup, such as environments, features and responses. The figure also includes the equivalent GUI dialogs (see Chapter 4, *RobOff Graphical User Interface*, p. 87 for a description of the RobOff GUI), where the same entities can be edited in a more interactive manner.

Figure 3.2. Set of input files and their equivalent GUI dialogs. Adapted and expanded from (Pouzols & Moilanen 2013).



The input files are described by simple and self-documenting examples first, in the following sections. From the examples you should be able to write your own input files, assuming some familiarity with the RobOff conceptual model (see Section 2.1, “The RobOff framework”, p. 17). For further details, a definition of each of the accepted options in each type of file is provided as well.

Note

You will notice that many lines in the following example files start with a sharp (or number or pound) symbol (#). These are comment lines, which

are ignored by RobOff and you can use freely. Anything written after a # is ignored by RobOff. You will also see comment lines like these in output and setup files generated by RobOff.

Note

All these input files must be written as plain text files. In windows systems their extension would typically be `'.csv'`, although this is not strictly necessary. In principle, RobOff imposes no limitation on the names of these files. Their length is not limited and any character set can be used, as long as it is supported by the operating system. However, there is a risk to run into trouble with other software or your operating system if you use disproportionately long or not so common characters including spaces, accents, parenthesis, etc.

3.3.1. General settings file

A example general settings file (`roboff.setup` in the example command line above) is shown below. This file defines all the main properties of a setup, but most of the information is actually provided in additional files.

Each option must be set in one line, using the '=' character to separate the option name and its desired value. The definition list given below specifies the meaning and possible values for every option. To start using RobOff, you do not need to know about most of these. The comments included in the example setup files will give you enough hints on what can be changed and how to change it. The next listings are intended as a complete reference that can be consulted at any time when needed.

```
# RobOff setup file
# beta release

name = simple_example

# Whether to perform robust estimations
robust offsetting = 1     # formerly k.a. 'read uncertainty info'
#allow overwriting = 0

# Uncertainty horizon for  which:
#    - robust results are calculated
#    - allocation of resources is optimized
info-gap alpha = 1.

# Range of uncertainty horizons.
# Three possible formats = start:end OR start:step:end OR alpha1,alpha2,...
# start:end is equivalent to start:0.1:end (step 0.1)
info-gap alpha range = 0.:0.25:2

# Range of planning horizons
# Three possible formats = start:end OR start:step:end OR date1,date2,date3,...
# start:end is equivalent to start:1:end (step 1)
planning horizons = 0:30

# Time discounting options. These can be defined in an optional time discounting file
# Time discounting model: quasi-hyperbolic (default) OR hyperbolic OR exponential
time discounting model = quasi-hyperbolic

# Time discounting rate (in %)
time discounting rate = 2.5

# Number of environments that should be defined in the environments file
environment types = 5

# Total number of respones that should be defined in response files
feature responses  = 1

# File names and prefixes. All these are default values and do not need to be
 specified
# unless changed
environments file = environments
feature weight function type file = feature_weight_functiontypes
per environment files prefix = features_present_
response files prefix = response_
score features files prefix = score_features_present_
variable cost files prefix = cost_units_curve_
budget allocation file = budget_allocation
# data files extension = .csv

# Not needed if default discounting options are ok
# discounting file = discounting.txt

# Budget
budget = 15000000000
# mandatory budget =
# preset budget =

optimization method = genetic algorithm
optimization criterion = weak (features)
# robust (default) OR opportunity OR nominal
optimization robustness requirement = robust
```

Note

In all input files, file paths are relative to the setup directory (location of the main setup file).

Let us now define each of the options (key-value pairs) supported in the general settings files:

`name` — Defines the name of a RobOff setup. This will be used to identify a setup and for reporting. Any text string is accepted.

`info-gap alpha` — Sets the uncertainty horizon (alpha) for info-gap calculations (see Section 2.5.1, “Uncertainty Analysis”, p. 25). This value will be taken as the reference value. It is also possible to specify a range of values (see below).

`info-gap alpha range` — Range for the uncertainty analysis. The range is specified by three numbers separated by colons. The first number is the lowest value, the second value is the step, and the third number is the highest value of the uncertainty horizon. Default: [0,1] by steps of 0.2. Examples: `info-gap alpha range = 0:0.2:1` (default); `info-gap alpha range = 0.:0.25:2`.

`planning horizons` — Sequence of discrete time instants. It can be specified either as a sequence of comma-separated numbers, or using a range as in the `info-gap alpha range` option. The length of the sequence must be equal to the number of columns in the specific response files. If the length of the sequence is shorter, RobOff might stil run but it will emit a warning message. The default value is: 0, 1, 2,... '# columns in response files'. Note that the first value should be interpreted as "units of time ahead of present time", where present time can be also interpreted as the time when values are not discounted (immediate or non-delayed accrual). If for example present time is year 2008 and the first planning step is year 2013, the first value of the planning horizon sequence could be defined as either 5 or 0. Using 5 implies that even the conservation

values accrued in the first year (2013) will be discounted (considering a delay of 5 years from now). Alternatively, to avoid discounting values in 2013 use 0 as first step (which effectively defines 2013 as present time). This consideration is extremely important in practice. If you use 2013 as first time step, all conservation values will be heavily discounted (as obtained more than two thousand years ahead of present time) and most likely results will be nonsensical.

`time discounting model` Defines the discounting model that should be applied. Together with the discounting rate (see below), this defines the relative weights of conservation value over time (see Section 2.5.4, “Time discounting”, p. 33). The following options are available: `quasi-hyperbolic`, `hyperbolic`, and `exponential`. Default: `quasi-hyperbolic`.

`time discounting rate` Time discounting rate is a percentage. Example: `time discounting rate = 2.5` sets a discounting rate of 2.5%. It is also possible to specify a sequence of discounting rates over time (variable discounting rate), see Section 3.3.6, “Time discounting file”, p. 65 for the details.

`environment types` Number of environment types defined. This optional parameter is used to check the consistency of RobOff setups. RobOff will check whether this number is consistent with the total number of environments defined in the global environments file, and whether all the corresponding environment specific (presence of biodiversity feature) files can be found.

`feature responses`

Number of feature responses defined. This optional parameter is used to check the consistency of RobOff setups. RobOff will check that this number is consistent with the total number of different responses that are used in the environments file and that all the corresponding specific response files can be found.

`budget`	Mandatory parameter that specifies the maximum budget available for conservation. Expressed in arbitrary monetary units (as long as the same units are consistently used in every file).
`mandatory budget`	Maximum budget available for mandatory actions. This is an optional parameter. Default: global budget.
`preset budget`	Maximum budget available for preset actions. This is an optional parameter. Default: global budget.
`budget resolution`	Resolution used for allocating the available budget, in percentage. When performing resource allocation optimization, RobOff will always use multiples of this resolution value to allocate fractions of the available budget. Default: 5%.
`budget allocation file`	Name of an optional file specifying the amounts allocated to different actions, see Section 3.3.7, “Budget allocation file”, p. 65 for details. Default value: `budget_allocation`.
`optimization method`	Optimization method. The options currently supported are: • `genetic algorithm` • `exhaustive` • `exhaustive + local search` • `greedy local search` • `random` The default method is `genetic algorithm`. The `genetic algorithm` method is strongly recommended for highly dimensional problems. `exhaustive` will provide exact solutions but can be very time consuming and in practice will be feasible only for problems of reduced dimensionality. `greedy local search` can be useful for quick tests but it is not recommended at all. Finally, `random` will randomly allocate fractions of the budget to different actions, trying to allocate as much as possible of the total budget. Its results can be used as a reference for evaluating the outcome of other optimization methods. Note

	that a random allocation may generate worse results than if no resources are allocated at all. In contrast, all the other methods will provide results that are equal or better than the starting point.
`optimization criterion`	Optimization criterion (or performance index that should be maximized by the optimization process). The options are: • `weak (features)` • `strong (features)` • `weak (environments)` • `strong (environments)` These options correspond to the different ratios of sustainability defined in the RobOff computational model (see Section 2.5, “Aggregation of conservation value”, p. 25)
`optimization robustness requirement`	Optimization robustness requirement. Options: `nominal`, `opportunity`, or `robust`. Default: `robust` (see Section 2.5, “Aggregation of conservation value”, p. 25 and Section 2.6, “Optimizing resource allocation”, p. 35)
`economic discount rate`	Specified as a percentage. Example: `economic discount rate = 2.5` sets a discounting rate of 2.5% per unit of time.
`economic discount model`	Sets the discounting model that should be applied to costs. Together with the discounting rate (see above), this defines the relative weights of costs over time, which are used to calculate the corresponding net present cost. The following options are available: `quasi-hyperbolic`, `hyperbolic`, and `exponential`. Default: `quasi-hyperbolic`.
`enable rescaling of responses`	For simple uses of RobOff this option is not needed. It sets whether to enable rescaling of responses based on the `start_value` and `end_value` fields in files of features present (for details on how to use this feature, see Section 3.3.4, “Set of files: biodiversity features...”, p. 62). Options: `no` (or any other value different to `yes`, default), and `yes`. Unless `yes` is specified the start and

	end value fields are ignored. Note that this option does not change the way RobOff calculates conservation value, it just modifies the values of responses, and it is provided for convenience.
`responses scale`	Scale in which the response values are expressed in the response files. The possible scales are: `range 0 to 1`, `absolute`, and `proportional gain`. Default: `range 0 to 1`. This option controls the way in which benefit functions are applied to occurrence values. The default is in principle the most convenient choice for most problems.

Proper use of the last option, `responses scale`, requires a good understanding of how conservation value is aggregated across features and environments, and benefit functions need to be defined consistently with the choice of scale. For details, see Section 2.5, “Aggregation of conservation value”, p. 25. The default value is in principle a good choice in most cases. This option does not prevent the use of any value in the response files, but the occurrence values specified in the responses will be treated differently. The default, `range 0 to 1`, corresponds to the typical case in which responses are in a certain range of values (and the benefit functions are defined accordingly). The maximum of the range does not need to be 1, but 1 can be used as a reference for good, pristine, satisfactory or similar occurrence levels. What is important is that the same range (or comparable ranges) should be used for the range of variation of the responses of different features in different environments, and the range of values for which benefit functions are defined (for example definitions of benefit functions, see Section 3.3.3, “Feature - weights - utility functions file”, p. 59). By default RobOff will calculate aggregated occurrence levels (representation and retention) averaged across spatial units, features and environments.

Other alternatives for the `responses scales` option are `proportional gain` and `absolute`. If `proportional gain` is used, benefit functions will be calculated on the ratios between response values over time divided by the initial response value. Note that the same effect can be obtained by preprocessing the responses, and that this proportional scale can lead to mathematically indeterminate results if initial values are 0. If `absolute` is selected, occurrence levels will be aggregated as they are specified in the response files, without any rescaling or averaging (as an exception to the algorithms described in section Section 2.5, “Aggregation of conservation value”, p. 25). Responses in absolute scale can be useful if benefit functions are defined for total or global occurrence values, across space and environments. This, however, requires that the spatial range of analysis be known in advance when preparing the benefit functions, and analyses for different problem subsets (or subsets of environments) would require readjusting the benefit functions (to different scales of total occurrencies). Also, responses in absolute scale could be useful for example in problems with a single

feature where occurrence levels are transformed into conservation value by a linear benefit function. In such cases it is possible to obtain conservation values that directly represent occurrence values (such as abundance).

In addition, the following options define the names of different types of input files. These options allow the user to change the names, prefixes and extension (or suffix) of the input files. If you decide to use the default values you do not need to specify any these options at all.

`environments file` — Name of the global environments file. Default: `environments`. See Section 3.3.2, “Environments file”, p. 58 for details on this type of file.

`per environment files prefix` — Prefix of the names of per environment files. RobOff will construct the names of environment specific files (containing a list of features present) by prepending this prefix to the environments names used in the global environments file (see Section 3.3.4, “Set of files: biodiversity features...”, p. 62). Default: `features_present_`

`response files prefix` — Prefix of specific response files. RobOff will look for response files by prepending this prefix to the names of responses used in the environments specific files (see Section 3.3.5, “Set of files: responses of biodiversity features”, p. 63). Default: `response_`. This option can be used, for example, to enforce the same prefix for all the response files (which should make it easier to distinguish response files) and/or to use responses that are located in a certain directory/folder.

`variable cost files prefix` — Prefix of files defining variable (per spatial unit) costs. RobOff will look for this type of file by prepending this prefix to the variable cost names used in the environments file (see Section 3.3.9, “Set of files: costs of actions”, p. 68 for details on the cost of actions files). Default: `cost_units_curve_`

`score features files prefix` — Prefix of files defining score features. One of this type of file can be specified for every environment. RobOff will look for this type of file by prepending this prefix to the

	environment names used in the environments file (see Section 3.3.8, “Set of files: score features”, p. 66 for details on the score features files). Default: `response_`
`data files extension`	String to be used as file extension (or suffix) for the RobOff input and output files. This suffix will be appended to most input files. Default value: `.csv`. Note that this default value (assumed throughout this manual) is intended for better interaction with common spreadsheet software packages (which typically load and save comma-separated values files using the `.csv` extension). If it is changed to `.txt`, then for example the environments file name will be environments.txt rather than the default environments.csv. This extension would make it easier to associate RobOff input files with plain text editors. Note that regardless of the extension used, RobOff always loads and saves files as plain text files.

3.3.2. Environments file

The environments considered in a RobOff setup are specified in a single file that contains one line per environment. The following listing is an example of an environments file:

```
# Environment types file, containing a table/list of environment types.
# Habitat# weight totalArea areaNoAction condition ListOfTriplets(Actions
 perAreaUnitCost areaAvail)
environment 1, 1.0, 25000, 500, 1.0, development action, 0, 150000, benign neglect,
 400, 15000, active restoration/management, 1000, 15000
environment 2, 1.0, 25000, 2000, 1.0, development action 2, 0, 150000, benign neglect
 2,
 200, 20000, active restoration/management 2, 800,  15000
```

Note

Note that fields are separated by commas within each line of the environments files (like in normal comma-separated values (csv) files). This is a general rule in RobOff input files.

Note that the example file contains two effective lines but these are wrapped because they are too long for this manual. In the input text files, the specification of each environment should go in a single line. In addition to the environment name, you need to specify a weight, the total area, the area that is not available for any action, the condition (typically but not necessarily in a scale from 0 to 1), and a

list of actions that are possible. For each action, three parameters must be given: the action name, its cost per unit of area, and the total area extension where the action can be undertaken (available area).

Technical note about character strings

In comma-separated files, spaces at the beginning and at the end of strings are ignored by RobOff. String fields will be automatically trimmed. For example, if you type ", restoration expensive " (note the spaces after the first comma (at the beginning of the character string) and before the second comma (at the end of the character string)), the string will be trimmed to "restoration expensive". This includes normal spaces and tabs. Note that spaces in between words matter. The names "restoration expensive" (single space in between the words) and "restoration expensive" (double space in between the words) are different. Similarly, "restoration_expensive" is different than "restoration__expensive".

This type of file is related to the option `environment types` of the general settings file. The number given in that option should be consistent with the number of environments (rows) specified in the environments file.

Note

In RobOff input files there is no explicit distinction between conservation and development actions. Normally, conservation actions would have positive costs whereas development actions would have zero cost (and be specified as mandatory). Development actions could also have negative costs. However, their definition and interpretation is up to the user.

3.3.3. Feature - weights - utility functions file

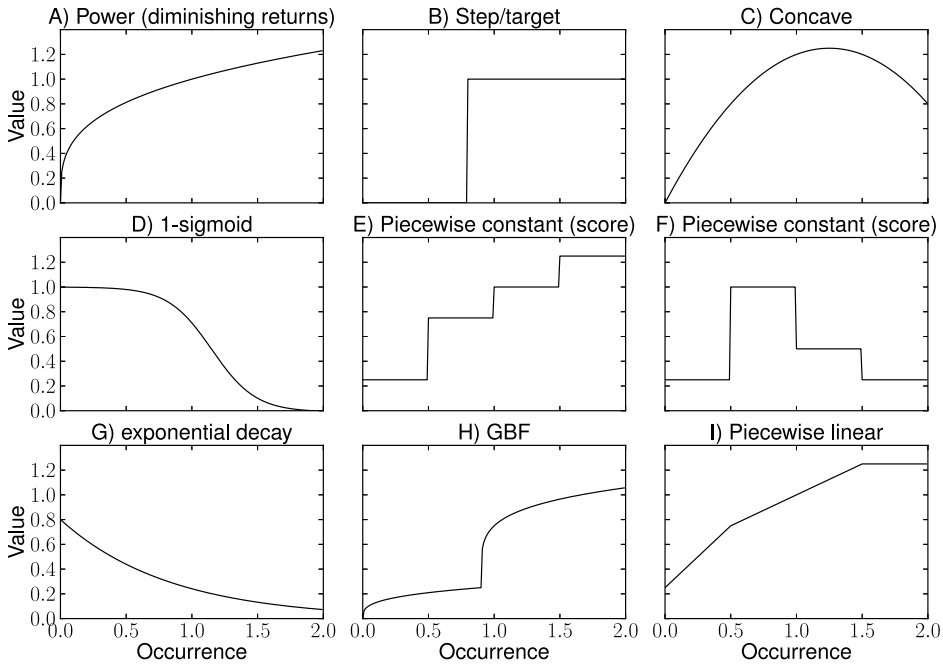
For every feature included in a RobOff setup you need to specify its weight and benefit (or utility) function (see Chapter 2, *Framework, Methods and Algorithms*, p. 17 for more information on the benefit function approach if you are not familiar with it). To see how to indicate that biodiversity features are present in different environments, see Section 3.3.4, “Set of files: biodiversity features...”, p. 62. Here is an example of this type of file:

```
# Types of functions:
#  1 - concave increase with diminishing returns: pow(R,x)
#  2 - Target: 0 if R < x; 1 if R >= x
#  3 - exponential decay: exp(-p*t)
#  4 - gbf (w1, x, T, w2, y)
#  5 - sigmoid
#  6 - inverse sigmoid
#  7 - quadratic polynomial
#  8 - piecew constant
#  9 - piecew linear
# 10 - piecew interp.
# Feature#, weight, function_type, function_parameter(s)_comma_separated_list
f_1, 1.0, 1, 0.3,   # pow
f_2, 1.0, 3, 0.2, 1.0  # exp
```

The example includes a list of supported benefit functions. Note that the number of parameters required depends on the type of function. In any case, the parameters must be provided after the function type code and separated by commas. For the piecewise constant, piecewise linear, and piecewise interpolated functions, the number of parameters is unlimited, and these must be provided as a list of pairs of x,y values (the number of parameters is always even).

Examples of some types of benefit functions are shown in Figure 3.3, “Example benefit functions”, p. 61. The panels in the figure illustrate benefit functions of type power (panel A, function 1), target (B,2), quadratic (C,7), inverse sigmoid (D,6), two different piecewise constant functions (E & F, 8), an exponential decay (G, 3), a generalized benefit function (H, 4), and a piecewise linear function (I, 9). Piecewise constant functions are typically used for score features (see Section 3.3.8, “Set of files: score features”, p. 66, Section 2.3, “Complementarity and scoring”, p. 22 and Section 2.5.2, “Aggregating conservation value”, p. 26). In general, it is possible to define arbitrary shapes with piecewise constant and piecewise linear functions. This feature should be used with care, as the use of piecewise functions with a large number of intervals can slow down RobOff computations significantly. With a number of intervals of approximately ten or less, computations will be as fast as for the simple functions. In any case, for simplicity and speed of computation we suggest using only the simplest possible types of functions unless otherwise required.

Figure 3.3. Example benefit functions



Different types of functions require different parameters (all the functions require at least one). This is an overview of the parameters needed in every case (see the example below for a more detailed explanation):

- *Type 1, concave increase with diminishing returns*, requires the exponent.
- *Type 2, target*, requires the target level.
- *Type 3, exponential decay*, requires two parameters: the (inverse) time constant, usually called lambda, and the initial value.
- *Type 4, generalized benefit function*, requires four parameters: w_1, x, T, w_2 and y. This type of function is in fact a flexible family of functions that was introduced in Zonation 3 (see the Zonation manual, which can be found online at http://www.helsinki.fi/bioscience/consplan, for details). w_1 and w_2 are weights for two parts of the generalized benefit function, T is the target level that divides the two parts, and x and y are exponents corresponding to the two parts.
- *Type 5, sigmoid*, uses the formulation of the logistic function as defined for modeling population growth in ecology. This function requires three parameters, r (growth rate), K (carrying capacity), and P0 (initial value).
- *Type 6, inverted sigmoid*, or 1-sigmoid uses the same parameters as the sigmoid function (type 6).
- *Type 7, quadratic polynomial*, requires the three coefficients of a second-order polynomial, starting from the higher power.

- *Types 8 and 9, piecewise constant and piecewise linear*, respectively, require an unlimited number of parameters, given as pairs of x-y values for every interval.

The following example clarifies the exact meaning of the required parameters and how to specify them in the feature-weights-function file. The examples listed in this file correspond to the panels shown above in Figure 3.3, “Example benefit functions”, p. 61. Note that the same sequence of values is used as the list of parameters for panels E and I, but the resulting functions are quite different from each other.

```
# More examples of utility functions in RobOff
# Feature#, weight, function_type, function_parameter(s)_comma_separated_list

# Panel A, power function
feature with diminishing return, 1.0, 1, 0.3

# Panel B, target function at 0.8 (or 80% if the reference is 1.0)
feature with target, 1.0, 2, 0.8

# Panel C, convex function defined with a quadratic polynomial
# polynomial: -0.8x^2 + 2x + 0
feature qpoly, 1.0, 7, -0.8, 2, 0

# Panel D, 1-sigmoid
# 3 parameters (as in sigmoid): r, K, P0, with r=7, K=1, P0=0.00001
# 1 - P0 * K * exp(r * occurrence) / (K + P0 * exp(r * occurrence) -1)
feature 1-sig, 1.0, 6, 7, 1, 0.00001

# Panel E, piecewise constant function (score)
feature sc1, 1.0, 8, 0, 0.25, 0.5, 0.75, 1.0, 1, 1.5, 1.25

# Panel F, another piecewise constant function (score)
feature sc2, 1.0, 8, 0, 0.25, 0.5, 1, 1.0, 0.5, 1.5, 0.25

# Panel G, exponential: 0.8*exp(-1.2 * occurrence)
feature exp decay, 1.0, 3, 1.2, 0.8

# Panel H, GBF - Generalized benefit function, as in Zonation
# Parameters:  w1=0.25, x=0.3, T=0.9, w2=0.5, y=0.2
feature gbf, 1.0, 4, 0.25, 0.3, 0.9, 0.5, 0.2

# Panel I, Piecewise linear
feature sc2, 1.0, 9, 0, 0.25, 0.5, 1, 1.0, 0.5, 1.5, 0.25
```

3.3.4. Set of files: biodiversity features present in an environment

For every environment, a file must be provided that contains a listing of the biodiversity features that are present in the environment. These files specify not only what biodiversity features are present but also their specific responses to actions. A file of this type must be provided for every environment that appears in the environments file (see Section 3.3.2, “Environments file”, p. 58). If you forget some of them, RobOff will detect it and report the error.

Every row or line specifies characteristics of a different feature. Features not included in a file of this type are considered not to be present in the corresponding

environment. If you forget to include a feature in a per environment file, RobOff will not be able to detect your error.

The (per environment) biodiversity features present files must be named according to this convention: concatenate the `environment files prefix` with the name of the environment and the extension (`'.csv'` by default. For example, if the `environment files prefix` option is set to 'feat_env_' and there are two environments named 'forest' and 'lake', then two files must be provided: 'feat_env_forest.csv' and 'feat_env_lake.csv'. Here is an example of a file of features present in an environment:

```
# For each habitat type, a file with features present and actions that can be
# applied to these features.
# feature#, present_estimate, resp_no_action, list_of_quartets(action#, response#
 end_value, uncert_weight)
f_1, 0.8, r_1^0, devel action, r_1^1, 0.1, 1.0, benign neglect, r_1^2, 1.0, 1.0,
 active restor/mngmnt, r_1^3, 1.3, 1.0,
f_2, 0.4, r_2^0, devel action, r_2^1, 0, 1, benign neglect, r_2^2, 0.8, 1.0,
 active restor/mngmnt, r_2^3, 1.2, 1.0,
```

Note

The parameters `start_value` (per feature) and `end_value` (per response to action) are not used by default *and are not needed for simple uses of RobOff*. To enable it use the option `enable rescaling of responses` needs to be enabled in the general settings file (Section 3.3.1, “General settings file”, p. 50. These parameters and the rescaling option are provided for convenience and are meant to be used with generic response shapes that can be rescaled between a minimum and maximum. Under the assumption that responses are increasing or decreasing curves over time, there are two possibilities: 1) `start_value` is used as minimum and `end_value` as maximum (monotonic increasing response) or 2) vice-versa (monotonic decreasing response). The effect is that the first and last points of responses are shifted to `start_value` and `end_value`, respectively, and the responses are rescaled (or multiplied by an amplitude factor) correspondingly. Note that this option is provided for convenience, and it has the same effect as pre-processing response files and providing different, rescaled response files to RobOff. For responses that are non-monotonic the interpretation of the rescaling is case dependent.

Note

The parameter `uncert_weight` (per specific response) is not currently used

This type of file is related to the option `per environment files prefix` of the general settings file.

3.3.5. Set of files: responses of biodiversity features

The specific responses of biodiversity features to actions are formatted in text files with three columns of numbers (nominal, lower envelope, and upper envelope). The columns especify estimate, lower envelope and upper envelope of uncertain occurrence levels for a feature specific response (see Chapter 2, *Framework, Methods and Algorithms*, p. 17 for details on how RobOff handles uncertain responses). A file of this type must be provided for every different response that appears in the files of type 'biodiversity features present in an environment' (see Section 3.3.4, “Set of files: biodiversity features...”, p. 62). If you forget any of them, RobOff will detect it at the setup error checking stage. The response files must be named according to the following convention: concatenate the `response file prefix` with the name of the action and the extension (`'.csv'` by default). For example, if the `response files prefix` option is set to 'response_' and there are three responses named 'business-as-usual', 'act1', and 'act2', then three files must be provided: 'response_business-as-usual.csv', 'response_act1.csv' and 'response_act2.csv'. Here is an example of response file:

```
# RobOff response file
# Response of feature 1 to action 1
# Columns: estimate, lower envelope, upper envelope
0.800, 0.700, 0.900
0.500, 0.200, 0.750
0.200, 0.000, 0.600
0.100, 0.000, 0.450
0.100, 0.000, 0.300
0.100, 0.000, 0.200
0.100, 0.000, 0.200
0.100, 0.000, 0.200
0.100, 0.000, 0.200
0.100, 0.000, 0.200
# End of response
```

This type of file is related to the option `response files prefix` of the general settings file.

Note

Notice that in the examples above the names of the response files start with 'response_' because that is the default prefix for the names of response files. This prefix is not part of the names of actions and is only used for file names. The prefix can be any valid path. The same applies to the default extension or suffix. As an example, the file 'response_business-as-usual.csv' would typically contain the response values (best estimate and uncertainty envelopes) that correspond to the response called 'business-as-usual'. For more details, see the option `response files prefix`. Note that the prefix is not part of the names of responses. This means that the prefix should not be used in the fields of the environments file (Section 3.3.2, “Environments file”, p. 58) or the files of biodiversity features present in an environment (Section 3.3.4,

“Set of files: biodiversity features...”, p. 62). In these files the name that identifies the response is just like 'business-as-usual'.

3.3.6. Time discounting file

This is an optional type of file that defines options related to time discounting, including discounting model, discounting rate, and the weights for each time interval. The options included in this type of file can all be specified in the general settings file. By defining different alternative time discounting files and using them from the general settings file (or with different `--time-disc` command line options), it is possible to analyze the influence of time discounting on a base-case analysis.

The following example of a time discounting file defines a discounting rate that is variable in time. Different weights of different time intervals are also considered (the 5 first intervals, say years, are discarded whereas the last 5 intervals are equally weighted).

```
# RobOff - time discounting file
# beta release

# Time discounting model: quasi-hyperbolic OR hyperbolic OR exponential
# Default: quasi-hyperbolic
time discounting model = exponential

# Example of discounting rate variable in time
time discounting rate = 3, 2.75, 2.5, 2.25, 2, 1.75, 1.5, 1.25, 1, 0.75

# Example of different weights over time
time discounting weights = 0, 0, 0, 0, 0, 1, 1, 1, 1, 1
```

The `discounting weights` allows easy exploration of different types of preferences, like equal weights throughout time, final discounting, no discounting, etc. The default value for this option is all the weights equal. Note that these weights are applied in addition to the discounting factor which is by no means replaced by the discounting weights. For the details on how the discounting factor is calculated, please see Section 2.5.4, “Time discounting”, p. 33.

Note that the length of the sequences of time discounting rates and time discounting weights must be consistent with the number of time intervals considered in the planning problem, that is, the number of rows in the response files. If the length of the sequence is too short, RobOff will report an error. If it is too long, RobOff will issue a warning message.

3.3.7. Budget allocation file

Here is an example of budget allocation file that includes mandatory and preset allocations of resources:

```
# Budget allocation file. One line per environment with allocated actions
# Environment, action, area_allocated
[mandatory allocation]
environment 1, development action, 8000000
environment 2, development action 2nd, 800000

[preset allocation]
# A few possibilities:
environment 2, benign neglect 2nd, 2000000
#environment 1, active restoration/management, 2000000
#environment 2, active restoration/management 2nd, 6000000
```

An allocation is nothing but three fields: environment, action, and amount (area extent). This type of file is related to the options `budget`, `preset budget`, and `mandatory budget` of the general settings file (see Section 3.3.1, “General settings file”, p. 50).

3.3.8. Set of files: score features

Score features are defined as combinations of simple features (see Figure 2.4, “Flow of aggregation of occurence levels and conservation value in RobOff”, p. 27 for details on how conservation value is aggregated for simple and score features in the RobOff computational model). In order to define how score features are aggregated from simple features it is necessary to:

- Include the score components as features in the biodiversity features present file.
- Provide an additional file (score features file) per each environment where there is at least one score feature.

The (per environment) score features files must be named according to the following convention: concatenate the `score features files prefix` with the name of the environment and the extension (`'.csv'` by default). For example, if the `score features files prefix` is 'score_features_' and there are two environments named 'forest' and 'lake', then the corresponding score features file names are: 'score_features_forest.csv' and 'score_features_lake.csv'. If a score features file is not provided for an environment, RobOff assumes that all the features present in that environment should be treated as simple features.

Score feature files consist of rows or entries containing four columns. Each row defines a score component and how to aggregate it into a score feature. The columns included in every row (score component) are described below.

- The first column indicates what feature is aggregated as a score component. These features must have been listed in the biodiversity features present file. RobOff will use them as score components, not as simple features anymore. If a feature listed here is missing from the corresponding biodiversity features present file, RobOff will report an error.
- The second column is the name of the score feature. RobOff will create a new score feature for every different name found in this second column.

- The third column defines the operator used to aggregate this score component. At the moment, three operators are supported: +, -, and *, which all correspond to basic arithmetic operations.
- The fourth column indicates the component weight. By using different component weights it is possible to generate combinations of simple features weighted differently.

The following listing shows a simple example of score features file:

```
# RobOff - beta release
# Score features file, table/list of score components
f2, score1, +, 0.5
f3, score1, +, 0.7
f4, score1, -, 0.6
```

In this example, a single score feature is defined as a combination of three features (score components): score1 = +0.5 f2 +0.7 f3 -0.6 f4. To calculate an arithmetic mean-like score, use the + operator for all the components. To calculate a geometric mean-like score, use the * operator for all the components. Note that the weights need to be adjusted if you intend to calculate an arithmetic mean. The next example defines two score features: score_add = +f1 - f2 + f3, and score_mult = +f1 -0.5 f2*f5.

```
# RobOff - beta release
# Score features file, table/list of score components
f1, score_add, +, 1.
f2, score_add, -, 1.
f3, score_add, +, 1.
f1, score_mult, +, 1.
f2, score_mult, -, 0.5
f5, score_mult, *, 1.
```

Note that it is valid to use a biodiversity feature as a score component of two or more score features. You can alternatively enter a same feature multiple times with different names (in the biodiversity features present file) and use those different names for different score features (in the score features file).

Note

Remember that when you include a biodiversity feature in the score features file of an environment, it will become a score component and will not be used as a simple feature in that environment (RobOff identifies it as a score component). It is possible to use the same feature both as a simple feature and a score component. You just need to enter the same feature twice or more times, with different names but using the same responses, weights, and benefit functions, which effectively replicates the same feature with different names. Then you can use one or several of them as score components.

3.3.9. Set of files: costs of actions

Costs can be constant values (per area unit cost) or variable. There are two types of variable costs: cost-area curves and time-dependent costs. Constant costs are directly specified in the environments file (Section 3.3.2, “Environments file”, p. 58). However, variable costs require additional files. If the cost of an action is variable, it is specified as a name in the environments file (as opposed to a numeric value in monetary units) and the corresponding file must be provided (one file per different cost). In the case of cost-area curves, costs are defined as functions of the amount of area where an action is performed. In cost-area files two columns of values must be provided, with one (x,y) pair (or (area,cost) pairs) in each row. Cost values are interpolated linearly. The same area and cost units as in other setup files must be used.

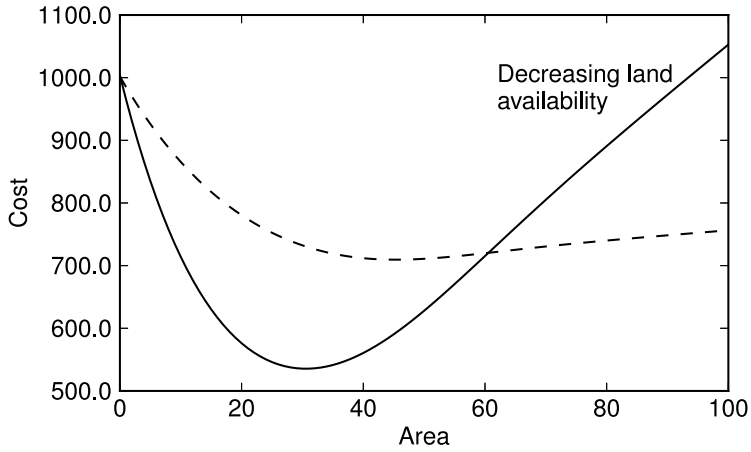
The following is an example of an environments file with some variable costs (as functions of area allocated):

```
# Environment types file, containing a table/list of environment types.
# env_name, weight, total_area, are_no_action, condition, list_of_triplets(action,
 area_unit_cost, area_available)
env1, 1.0, 25000, 500, 1.0, devel1, 0, 150000, restoration, variable_cost1, 15000
env2, 1.0, 25000, 2000, 1.0, devel2 2, 0, 150000, restoration 2, variable_cost2,
 15000
```

Below is an example of a variable (cost-area) cost file. These files describe a cost-area curve and consist of two columns. The first column corresponds to the area value whereas the second column corresponds to the cost in monetary units (x and y coordinates, respectively, in the example Figure 3.4, “Example of (per area unit) cost as a function of area extent”, p. 69). If this file were specifying the variable cost of action 'active restoration 1' (variable_cost1), it should be named 'variable_cost1.csv'

```
# RobOff cost-area file
# Format: area_value, cost_value
0, 50000
5, 45000
10, 30000
15, 25000
20, 18000
25, 16000
30, 15000
35, 14000
```

Figure 3.4. Example of (per area unit) cost as a function of area extent



In contrast, time-dependent cost files consist of one single column, with one row per time interval in the planning horizon (time intervals are implicit, as in response files, and the number of rows must be consistent with the length of the planning horizon). If time-dependent costs are defined, economic discount will be applied by using the model and discount rates specified in the general settings file (options `economic discount rate` and `economic discount rate`, see Section 3.3.1, “General settings file”, p. 50. Alternatively, variable costs can be defined interactively in the GUI (see Section 4.2, “Setup”, p. 89). For an example of variable cost in the GUI, see Figure 4.8, “Editing costs in the RobOff GUI”, p. 94.

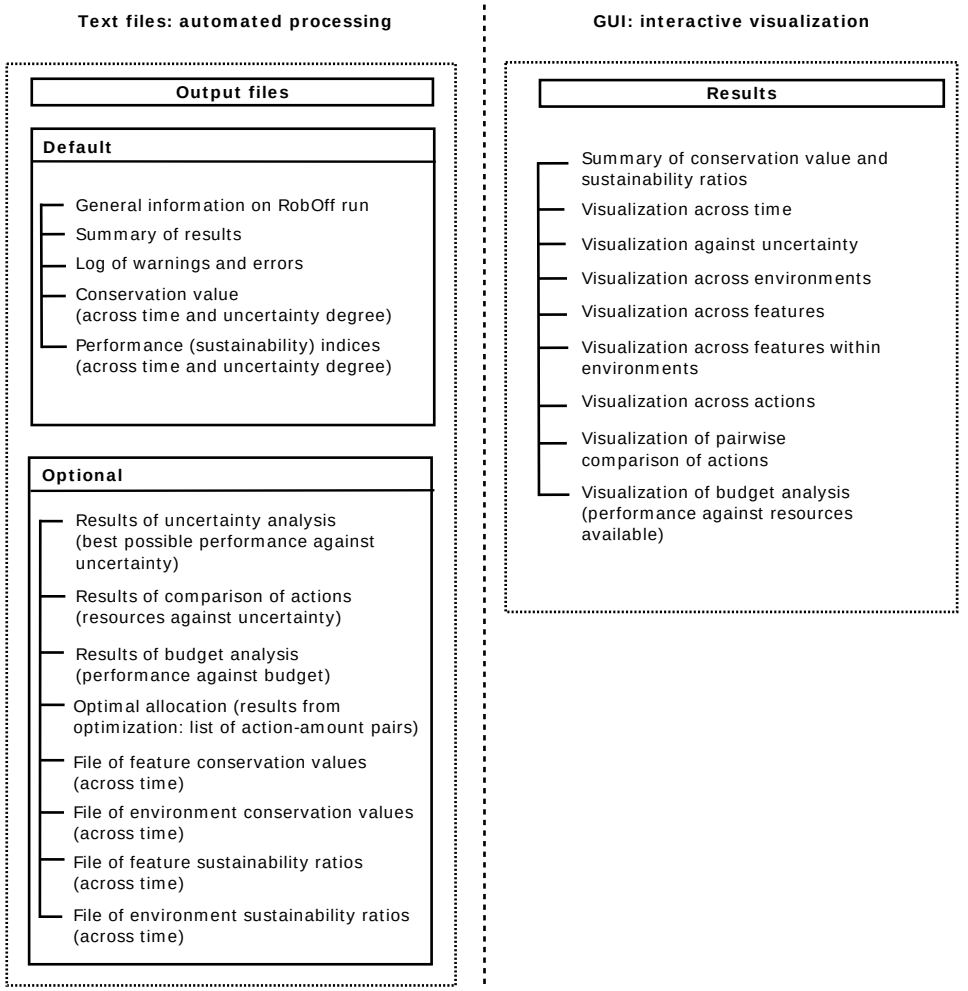
Note

As an exception, the names of cost-area curve and time-dependent cost files cannot be numbers (whether integer or fractional). This is because RobOff will first try to parse a number and if it succeeds the cost field will be interpreted as a constant monetary value. Only names that cannot be parsed as numbers will be considered as names of cost files. For example, if in the cost field of an environments file you indicate the value `30`, it will be interpreted as constant cost 30. RobOff will not look for a file named '30' even if you provide it. A better name if you want to use a variable cost would be `cost30` for example. Using numbers as file names is in general a very error prone practice that we strongly discourage.

3.4. Standard RobOff output

RobOff generates a set of files in the output directory (`-o` or `--output` command line option). Some of them are generated regardless of the options specified whereas some others are optional, depending on what options have been enabled in the command line. A diagram that summarizes the different types of output files is shown in Figure 3.5, “Set of output files and their equivalent GUI dialogs”, p. 70. The figure also includes the equivalent GUI dialogs (see Chapter 4, *RobOff Graphical User Interface*, p. 87 for a description of the RobOff GUI), where the same results can be visualized interactively. Note that the same output files described here for the command line interface of RobOff can be generated from the GUI.

Figure 3.5. Set of output files and their equivalent GUI dialogs. Adapted and expanded from (Pouzols & Moilanen 2013).



A directory listing of a typical RobOff output directory looks like this:

```
drwxr-xr-x 2 fedemp fedemp  4096 Dec 12 13:12 .
drwxr-xr-x 3 fedemp fedemp  4096 Dec 12 13:12 ..
-rw-r--r-- 1 fedemp fedemp   176 Dec 12 13:12 0.readme.txt
-rw-r--r-- 1 fedemp fedemp   685 Dec 12 13:12 1.summary.csv
-rw-r--r-- 1 fedemp fedemp    31 Dec 12 13:12 2.log.txt
-rw-r--r-- 1 fedemp fedemp 80546 Dec 12 13:12 3.conservation_values.csv
-rw-r--r-- 1 fedemp fedemp   743 Dec 12 13:12 4.conservation_performance_ratios.csv
-rw-r--r-- 1 fedemp fedemp   222 Dec 12 13:12 5.uncertainty_analysis.csv
-rw-r--r-- 1 fedemp fedemp    48 Dec 12 13:12 6.optimization.csv
-rw-r--r-- 1 fedemp fedemp   166 Dec 12 13:12 7.optimal.allocation.csv
-rw-r--r-- 1 fedemp fedemp    63 Dec 12 13:12 8.actions_comparison.csv
```

All the output files are in plain text format. The outputs of RobOff are split into a number of simple files that are easy to process in scripts or load into spreadsheet software. The following output files are always generated:

0.readme.txt — General useful information about how the software was run, such as the time and machine where the analysis was run, RobOff version employed, computation time, etc. It also lists what results (and corresponding files) are available. This is an example readme file:

```
# Roboff results reame.txt
# RobOff 1.0.0rc4 - Warning: this is a candidate release - http://www.helsinki.fi/
bioscience/consplan/
# Generated on 20130109 at 14:38:27, on 'achernar'
# This directory contains results for the setup 'Dam_example_second_level_offset',
#   and was generated for the following setup/configuration:
# 'DamForestRiver.ro_setup'
# General results: available, see 1.summary.txt, 3.conservation_values.csv and
 4.sustainability_ratios.csv
# Uncertainty analysis results: NOT available
# Optimization results: available, see 6.optimization.csv
# Optimal allocation of resources results: available, see 7.optimal_allocation.csv
# Actions comparison results: NOT available
# Budget analysis results: NOT available
# Feature specific results: available, see the files
 10.conservation_values_per_feature.csv and 12.sustainability_ratios_per_feature.csv
# Environment specific results: available, see the
 files 11.conservation_values_per_environment.csv and
 13.sustainability_ratios_per_environment.csv
# Last but not least, don't forget to check 2.log.txt for possible warnings and
 errors.
```

1.summary.txt — Summary of results presented as a table. The table gives conservation values and sustainability ratios summarized as scalar values after applying time discounting. The following tables and listing are examples of a results summary for a problem with mandatory and optimal sets of allocations. This same output can be found in the 'summary' dialog of the results section of the RobOff GUI.

Table 3.1. Example summary of results: conservation value

		Robust	Nominal	Opportunity
Conservation value	No action	0.6884	0.7715	0.8226
	Mandatory actions	0.6012	0.7459	0.8407
	Optimal actions	0.8407	0.9462	1.017

Table 3.2. Example summary of results: conservation performance ratios

		Robust	Nominal	Opportunity
Conservation performance ratios	Weak (environments)	1.072	1.198	1.425
	Strong (environments)	0.9655	1.001	1.153
	Weak (features)	1.090	1.226	1.486
	Strong (features)	0.6863	0.9974	1.000

```
# Roboff - summary of results
Info-gap uncertainty horizon (alpha): 1 (range from  0 to 1)

 === Conserv. value (discounted, weak across features): ===
     Minimum     Nominal       Maximum
No action     0.9047      0.9126        0.9177
Mandatory actions    0.8129      0.8198        0.8243
Optimal actions     0.8395      0.8435        0.8474

 ===== Conservation performance ratios (discounted): =====
      Robust     Nominal   Opportunity
Weak for environments    0.9148      0.9242        0.9370
Strong for environments    0.8193      0.8279        0.8423
Weak for features    0.9148      0.9243        0.9370
Strong for features    0.8193      0.8279        0.8423

Mandatory Allocation:
   Environment 0 (MiddleForest) (area: 500.0):
          200.0 - Action 0 (do nothing)
          300.0 - Action 1 (Inundate)
          0.000 - Action 2 (Restore)
   Environment 1 (UpperForest) (area: 3.000e+04):
      3.000e+04 - Action 0 (do nothing)
          0.000 - Action 1 (Restore)

Best Allocation:
   Environment 0 (MiddleForest) (area: 500.0):
          0.000 - Action 0 (do nothing)
          300.0 - Action 1 (Inundate)
          200.0 - Action 2 (Restore)
   Environment 1 (UpperForest) (area: 3.000e+04):
      1.020e+04 - Action 0 (do nothing)
      1.980e+04 - Action 1 (Restore)
```

2.log.txt

This file contains a detailed log of RobOff runs. It is strongly advisable to always have a look at this file in order to make sure everything went fine. For instance, warning messages concerning possible inconsistencies in the input files are recorded here. Summaries about files found are also written into this file. An extract of an example log is shown in the following listing. Note that RobOff logs can be quite long. Important messages are usually highlighted with various symbols and lines.

```
# Roboff log
# Generated on mrg20, Tue Jun 26 17:36:55 EEST 2012 on
../setups/setup-simple-demo is a directory. Trying to append default filename:
 roboff.setup... ok.
Reading configuration file...
No discounting configuration file found. Discounting model is: 'Quasi-hyperbolic' and
 rate: 0.025%
Environments #: 4
.
.
.
Setup "../setups/setup-simple-demo" successfully loaded.
OpenMP support: there are 4  threads in parallel regions
```

3.conservation_values.csv

This file contains conservation values across the time and uncertainty dimensions. The file consists of several matrices, each of them corresponding to one of the cells in the matrix shown above for the `1.summary.csv` output file. The rows of these matrices correspond to the different time intervals considered in the planning horizon (indicated in the first column), whereas the columns correspond to the different values of the uncertainty horizon range. The following listing shows the beginning of a file of this type.

```
# Roboff results: conservation value
#
# ============= Results for *No action* =============
#
# ===== Results for 'nominal' robustness =====
#
# Uncertainty horizon (alphas)
#      0.00,0.100,0.200,0.300,0.400,0.500,0.600,0.700,0.800,0.900,1.00,
#Time
#
# Avg. conservation value (across features):
0.000,0.9145,0.9145,0.9145,0.9145,0.9145,0.9145,0.9145,0.9145,0.9145,0.9145
10.00,0.9136,0.9136,0.9136,0.9136,0.9136,0.9136,0.9136,0.9136,0.9136,0.9136
20.00,0.9128,0.9128,0.9128,0.9128,0.9128,0.9128,0.9128,0.9128,0.9128,0.9128
30.00,0.9119,0.9119,0.9119,0.9119,0.9119,0.9119,0.9119,0.9119,0.9119,0.9119
40.00,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110
50.00,0.9101,0.9101,0.9101,0.9101,0.9101,0.9101,0.9101,0.9101,0.9101,0.9101
60.00,0.9092,0.9092,0.9092,0.9092,0.9092,0.9092,0.9092,0.9092,0.9092,0.9092
70.00,0.9082,0.9082,0.9082,0.9082,0.9082,0.9082,0.9082,0.9082,0.9082,0.9082
80.00,0.9072,0.9072,0.9072,0.9072,0.9072,0.9072,0.9072,0.9072,0.9072,0.9072
90.00,0.9062,0.9062,0.9062,0.9062,0.9062,0.9062,0.9062,0.9062,0.9062,0.9062
100.0,0.9052,0.9052,0.9052,0.9052,0.9052,0.9052,0.9052,0.9052,0.9052,0.9052

# Min. conservation value (across features):
0.000,0.9142,0.9142,0.9142,0.9142,0.9142,0.9142,0.9142,0.9142,0.9142,0.9142
10.00,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110,0.9110
.
.
.
```

These matrices are generated for each different set of allocations in the following sequence: no action (no allocation of resources at all), mandatory, preset, and optimal. For each of these options, three robustness variants are considered: nominal (average), opportunity (maximal), and robust (minimal). In addition, for each robustness criterion, the matrix of average (weak) values is written first, followed by the matrix of minimum (strong) values, the matrix of discounted average values, and finally the matrix of discounted minimum values. Therefore the file can contain up to 48 (4x3x4) matrices.

Note that we use the strong/weak terms applied to conservation value in a similar sense as in the performance ratios. Weak/strong conservation value across features is equivalent to average/minimum conservation value across features. Note that this is different from the performance ratios (which are ratios between conservation values for different robustness criteria).

4.sustainability_ratios.csv

This file contains, for each of the 4 possible sets of allocations (whenever available), sustainability indices as ratios of conservation values (see Chapter 2, *Framework, Methods and Algorithms*, p. 17. The file contains up to 4 matrices of values over time of (discounted) weak and strong performance ratios across features and environments. The first lines of an

example file of this type are shown below.

```
# Roboff results: sustainability  ratios
# All values are discounted
# Robustness requirement: robust
# Info-gap alpha: 1.0
# Format (discounted): Horizon, weak_perf_feat, strong_perf_feat, weak_perf_envs,
 strong_perf_envs
#
# ===== Results for No action  =====
0,0.995537,0.994638,0.995536,0.994638
10,0.993698,0.992384,0.993697,0.992384
20,0.992099,0.9904,0.992098,0.9904
30,0.990728,0.988678,0.990727,0.988678
40,0.989568,0.987204,0.989568,0.987204
50,0.9886,0.985959,0.9886,0.985959
60,0.987802,0.984921,0.987801,0.984921
70,0.987151,0.984065,0.98715,0.984065
80,0.986626,0.983367,0.986625,0.983367
#
# ===== Results for mandatory actions  =====
0,0.893746,0.791006,0.893721,0.791006
10,0.892247,0.789214,0.892221,0.789214
20,0.890944,0.787636,0.890919,0.787636
.
.
.
```

3.4.1. Optional output files

Some additional output files are optionally generated for the following analyses: uncertainty analysis, comparison of actions, optimization of resource allocation, and budget sensitivity analysis. The standard file names and their contents are as follows.

5.uncertainty_analysis.csv

This file summarizes the results of the uncertainty analysis (`-u` or `-uncertainty` command line options). The file contains a table of values (as a function of the uncertainty horizon) for the performance indices: ratios of weak and strong sustainability across features and across environments. Average and minimum conservation values across features and environments are included as well. These results are reported for three robustness requirements: nominal, robust, and opportunity. This is the same information that can be visualized in the uncertainty analysis plot of the graphical interface. An example output uncertainty analysis file looks like this:

```
# Roboff uncertainty analysis results
# Uncertainty analysis for optimal allocation
# Format: alpha, sust_weak_features, sust_strong_features, sust_weak_environments,
 sust_strong_environments, consval_weak_features, consval_strong_features,
 consval_weak_environments, consval_strong_environments
#
# Results for 'nominal' robustness
0     ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
0.25  ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
0.5   ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
0.75  ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
1     ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
1.25  ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
1.5   ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
#
# Results for 'robust' robustness
0     ,0.90824,0.79527,0.72456,0.79527,0.85511,0.72149,0.6993,0.37766
0.25  ,0.90566,0.79178,0.72231,0.79178,0.85345,0.71938,0.6975,0.37718
.
.
.
```

6.optimization.csv

The results of the optimization of resource allocation analysis (`-p` or `--optimize-alloc` command line options) are written into this file. This file contains information about optimization criteria, the optimization algorithm used, etc.

```
# Roboff optimization results
# Method used: Exhaustive + local search
# Budget resolution: 3
# Trials calculated: 3262623
# Budget spent in optimal allocation: 1.80178e+08
# Optimal allocation: see file '7.optimal.allocation.csv' - can be visualized in the
 optimization section of the GUI
```

7.optimal_allocation.csv

This file is in the same format as the budget allocation input file (see Section 3.3.7, “Budget allocation file”, p. 65), it lists the amounts allocated to different actions, and can actually be used as an input allocation file. The file is generated if the optimization analysis is enabled in the command line. The mandatory, preset, and best allocations are reported whenever available. This type of file can be loaded in the optimization section of the GUI (see Section 4.4, “Optimization”, p. 103). The following listing shows an example of an optimal allocation output file.

```
# Roboff optimization results: optimal allocation
# RobOff saved this allocation for the setup 'Dam_example_second_level_offset'. Edit
 at your own risk!
# Format: environment, action, amount (area), equivalent money
[mandatory allocation]
MiddleForest, Inundate, 300, 0
MiddleRiver, Inundate, 20, 0
UpperRiver, Obstruct, 200, 0
LowerRiver, AlterFlow, 10, 0
[preset allocation]
[optimal allocation]
MiddleForest, Restore, 200, 1000
UpperForest, Restore, 19800, 99000
```

8.actions_comparison.csv

This file is generated if the comparison of actions analysis is enabled (`-c` or `--compare-actions` command line options). This file shows, for the range of values of the uncertainty horizon, the preference (between 0 and 1) between pairs of actions. The first column corresponds to the different values of the uncertainty horizon. The values provided in the other columns are the proportion of resources that should be allocated to the first action in every pair (in order to maximize conservation value). For every environment, as many columns as possible pairs of actions are generated. In the example shown below, there are 3 alternative actions in 2 environments (3 pairs of possible comparison within each environment).

```
# Roboff results: comparison of actions for different degrees of uncertainty
# Format: uncertainty__alpha, area_to_1st_action, money_to_1st_action,
 area_to_2nd_action, money_to_2nd_action,..., area_to_last_action,
 money_to_last_action
# These are the environment/action names:
# MiddleForest/do nothing, MiddleForest/Inundate, MiddleForest/Restore, UpperForest/
do nothing, UpperForest/Restore,
0,0,0,300,0,200,1000,10200,0,19800,99000
0.2,0,0,300,0,200,1000,10200,0,19800,99000
0.4,0,0,300,0,200,1000,10200,0,19800,99000
0.6,0,0,300,0,200,1000,10200,0,19800,99000
0.8,0,0,300,0,200,1000,10200,0,19800,99000
1,0,0,300,0,200,1000,10200,0,19800,99000
```

9.budget_analysis.csv

This file is generated if the budget analysis is enabled (`-b` or `--budget-analysis` command line options). The budget analysis can

be useful to compare the return on investment for varying budget levels. This file shows, for the budget range the following performance indices: weak and strong sustainability ratios across features and environments. The first column corresponds to the different budget levels. The values provided in the other columns are the different variants of sustainability ratios and conservation value (weak or strong, and across features or environments). This analysis requires specific options in the general settings (see Section 3.3.1, “General settings file”, p. 50).

```
# Roboff results. Budget analysis
# Allocation:
# Format: budget_level_(money), sust_weak_features, sust_strong_features,
 sust_weak_environments, sust_strong_environments, consval_weak_features,
 consval_strong_features, consval_weak_environments, consval_strong_environments
0,0.89789,0.78124,0.71544,0,0.84837,0.71286,0.69194,0.37573
20000,0.89926,0.78124,0.71624,0,0.84967,0.71286,0.69272,0.37573
40000,0.9006,0.78124,0.71703,0,0.85094,0.71286,0.69348,0.37573
60000,0.90192,0.78124,0.7178,0,0.85218,0.71286,0.69423,0.37573
80000,0.90321,0.78124,0.71856,0,0.8534,0.71286,0.69496,0.37573
1e+05,0.90447,0.78124,0.7193,0,0.8546,0.71286,0.69568,0.37573
```

10.conservation_values_per_feature.csv

This file is generated only if the command line option `-f` or `--per-feature-out` is used. It contains conservation values specific to features across time. The file consists of several matrices, with the columns corresponding to different features, and rows corresponding to the different time intervals of the planning horizon (indicated in the first column). Each of the matrices written in this file corresponds to one combination of resource allocation and robustness requirement (as in the 3.conservation_values.csv file). However, in this case conservation values are reported only for one degree of uncertainty (the preferred degree of uncertainty specified in the general settings file, see Section 3.3.1, “General settings file”, p. 50). Features are listed in the same order as they are specified in the feature-weights file (see Section 3.3.3, “Feature - weights - utility functions file”, p. 59, and their names are shown for convenience in a comment line at the beginning of the file. The following listing shows part of an example:

```
# Roboff results: feature specific conservation value
# Format: time, cons_val_feature1, cons_val_feature2, ... cons_val_last_feature
# Names of features: ForestTypeA, ForestTypeB, River,
#
# ============= Results for *No action* =============
#
# ===== Results for 'nominal' robustness =====
# conservation value, per features:
    0,0.91423,0.91468,0.99924
   20,0.9077,0.91787,0.99924
   40,0.90102,0.92102,0.99924
   60,0.89419,0.92414,0.99924
   80,0.8872,0.92723,0.99924
  100,0.88004,0.93029,0.99924

# conservation value, per features, time-discounted:
    0,0.91423,0.91468,0.99924
   20,0.91173,0.91591,0.99924
   40,0.90989,0.91679,0.99924
   60,0.90861,0.91739,0.99924
   80,0.90777,0.91778,0.99924
  100,0.90723,0.91803,0.99924

#
# ===== Results for 'opportunity' robustness =====
# conservation value, per features:
    0,0.91586,0.91631,0.99924
```

11.conservation_values_per_environment.csv

This file is generated only if the command line option `-e` or `-per-environment-out` is used. This file is analogous to the file of conservation values per feature (10.conservation_values_per_feature.csv, see above), but it contains results per environments instead of results per feature.

```
# Roboff results: environment specific conservation value
# Format: time, cons_val_environment1, cons_val_environment2, ...
 cons_val_last_environment
# Names of environments: MiddleForest, UpperForest, MiddleRiver, UpperRiver,
 LowerRiver,
#
# ============= Results for *No action* =============
#
# ===== Results for 'nominal' robustness =====
# conservation value, per environments:
    0,0.91469,0.91469,1,1,1
   10,0.91144,0.91629,1,1,1
```

12.sustainability_ratios_per_feature.csv

This file is analogous to the file 4.sustainability_ratios.csv, but contains sustainabiliy ratios for individual features (in the sequence indicated by the list of names provided in the file header). It is is generated only if the command line option `-f` or `-per-feature-out` is used. Here is the first lines of an example file:

```
# Roboff results: sustainability ratios, per feature
# All values are discounted
# Robustness requirement: robust
# Info-gap alpha: 1
# Format: time,sust_ratio_feature1,sust_ratio_feature2,... sust_ratio_last_feature
# Names of features: ForestTypeA,ForestTypeB,River,
#
# ===== Results for No action  =====
# sustainability ratio,per features:
0,1,1,1
20,1,1,1
40,1,1,1
60,1,1,1
80,1,1,1
100,1,1,1

#
# sustainability ratio,per features,time-discounted:
0,1,1,1
20,1,1,1
40,1,1,1
60,1,1,1
80,1,1,1
100,1,1,1

#
# ===== Results for mandatory actions  =====
# sustainability ratio,per features:
0,0.79527,1,0.92777
20,0.79527,1,0.92048
40,0.79527,1,0.913
60,0.79527,1,0.90534
80,0.79527,1,0.89748
100,0.79527,1,0.88941
```

13.sustainability_ratios_per_environment.csv

Similarly to the previous file (per feature sustainability ratios) this file is analogous to the file 4.sustainability_ratios.csv, but contains sustainabiliy ratios for individual environments (as listed in the file header). This file is generated only if the command line option `-e` or `-per-env-results` is used. Here is the beginning of an example file:

```
# Roboff results: sustainability ratios,per environment
# All values are discounted
# Robustness requirement: robust
# Info-gap alpha: 1
# Format: time,sust_ratio_environment1,sust_ratio_environment2,...
 sust_ratio_last_environment
# Names of environments: MiddleForest,UpperForest,MiddleRiver,UpperRiver,LowerRiver,
#
# ===== Results for No action  =====
# sustainability ratio,per environments:
0,1,1,1,1,1
20,1,1,1,1,1
40,1,1,1,1,1
60,1,1,1,1,1
80,1,1,1,1,1
100,1,1,1,1,1

#
# sustainability ratio,per environments,time-discounted:
0,1,1,1,1,1
20,1,1,1,1,1
40,1,1,1,1,1
60,1,1,1,1,1
80,1,1,1,1,1
100,1,1,1,1,1

#
# ===== Results for mandatory actions  =====
# sustainability ratio,per environments:
0,0.79527,1,0,0.94377,0.8979
20,0.79527,1,0,0.93614,0.8979
```

14.costs_over_time.csv

This file is generated only if at least one time-varying cost is defined (see Section 3.3.9, “Set of files: costs of actions”, p. 68). For the possible allocations (whenever available), it lists the costs over time of allocations to actions (only for those actions with time-dependent costs). With this information it is possible to calculate benefit-cost ratios or similar measures over time. Here is an example excerpt of costs over time output file:

```
# Format of lines: time, cost_action1,...,cost_last_action
# (only actions with time-dependent costs included)
# Names of environment/actions: MiddleForest/Restore, UpperForest/Restore,
 UpperRiver/ObstructPlusTT
#
# ===== Costs for No action  =====
0,0,0,0
20,0,0,0
40,0,0,0
60,0,0,0
80,0,0,0
100,0,0,0

# ===== Costs for mandatory actions  =====
0,50000,20000,1000
20,20000,8000,220
40,5000,5000,240
60,0,0,260
80,0,0,280
100,0,0,300
```

3.5. What RobOff does not do directly

Fixing inadequate data

RobOff cannot generate, correct, or fix your data. All that can be done is to detect and report inconsistencies in the input files. RobOff will surely produce conservation values, sustainability ratios, optimal allocations of resources, and other results no matter what data is provided as input. However, these results will be informative and relevant to the extent to which the input data is an appropriate and representative model.

Spatial planning

RobOff is not intended for explicit spatial planning. It can however be used for extracting area targets. See Section 2.7, “Dealing with connectivity”, p. 38 for some comments on how to incorporate connectivity into RobOff models and Section 5.2, “Analysis types”, p. 112 for how to use RobOff to extract area targets.

Process-based planning and dynamic interactions

RobOff is not intended for spatial dynamic planning. It does not build any dynamic interactions of feedback loops into its computations. In particular, there is no direct support for process-based planning and dynamic interactions. See Section 2.8, “Assumptions and limitations”, p. 39 for a more methodological discussion of simplifications and assumptions made in the RobOff framework.

Tip

Interactions can be represented in RobOff setups as biodiversity features, whether simple or scores. Note that scores are calculated as sums or products of simple features. This can be an effective yet simple way of modeling interactions.

Scheduling

RobOff does not implement any scheduling mechanism (here we understand scheduling as dynamic planning through space and time). Extensions might be added in the future to support simple approaches to dynamic allocation.

3.6. Implementation details about RobOff

RobOff was developed in the C++ language. There are no hard coded limits on the number of environments, features or actions that can be analyzed. The RobOff GUI was developed using the Qt libraries for cross-platform desktop (http://qt-project.org). It also uses extensively the Qwt (Qt Widgets for Technical Applications) library (http://qwt.sourceforge.net). Also, we used various versions of GCC, the GNU compiler collection (http://gcc.gnu.org). The Roboff GUI also uses icons from the open icon library (http://openiconlibrary.sourceforge.net). These free open source software projects, without which an effective implementation of RobOff would have probably been impossible, are greatly acknowledged!

In the RobOff computational core, calculations and aggregation of conservation value are done in a parallel manner in multi-processor/core shared memory systems by means of OpenMP directives (http://www.openmp.org). The command line interface and the GUI share a common library that has been designed following a modular approach. The core components perform the following tasks: construction and validation or RobOff setups, calculations of conservation value and sustainability ratios, and search of optimal allocations of resources. Additional components perform tasks such as loading and saving setups, numeric results, plots and allocations of resources.

Precompiled binaries are provided for both 32 and 64 bit systems. These are included in the software distribution, available as self-installing setup binaries or compressed packages. As of this writing precompiled binaries are provided for GNU/Linux and Windows operating systems. The software, example setups, and this manual are available from the consplan website of the Biodiversity Conservation Informatics Group at University of Helsinki: http://consplan.it.helsinki.fi/software/projects/roboff. General information can also be obtained from the group website: http://www.helsinki.fi/bioscience/consplan.

3.7. Data limitations and system requirements

RobOff fully supports multi-core 64 bits systems and imposes no hard limits on the size of setups. Processing hundreds or thousands of features and environments is no issue with current commodity PCs. Computation time increases linearly with the number of features and environments, but RobOff supports optimization strategies that can cope with large sets of actions. For such high-dimensional problems (more than 10-20 alternative actions) the 'genetic algorithm' method of optimization should be used (for more details on optimization methods see Section 2.6, “Optimizing resource allocation”, p. 35).

3.8. Troubleshooting

This section is simply a concise list of common issues and pitfalls to take into account:

- **Always check the log window (or readme and log files).** Most simple and common errors are easy to catch by carefully reading notices and warnings issued by RobOff.

- **Path of files and directories.** If RobOff complains that it cannot find certain files, check first that the relative or full paths to your input files are correct. Also use special or not so common characters in file names with extreme care (see section Section 3.3, “Input files and settings”, p. 49.

- **Computer memory capacity.** In principle, RobOff can use as much memory as available in 64 bits systems. If at some point RobOff runs too slowly, it is a good idea to start by checking how much RAM memory is actually available. If your system memory is exhausted by RobOff or other processes, RobOff may start using virtual memory (hard drive). This will result in overly poor performance and you should definitely free RAM memory or use a machine with more memory available.

- **Parallel execution.** RobOff makes intensive use of multi-core systems. If you experience significant performance degradation, check that the number of cores in use by RobOff processes (possibly running in parallel) is not higher than the number of cores available in your machine. Use the `-m` or `--max-cores` command line option conveniently, especially if you are running several RobOff processes in parallel.

- **Differences in the length of responses.** Should not be an issue as long as all the responses have a number of samples equal or greater than the length of the time or planning horizon. However, mixing responses of different length can be a very error prone practice.

- **Decimal separator and commas.** Do not use commas as decimal separators! See also the section called “Quick start”, p. 10.

Part IV. RobOff Graphical User Interface

Table of Contents

Chapter 4. RobOff Graphical User Interface

The RobOff graphical user interface (GUI) allows utilization of RobOff without the need to edit input files directly or use the command line interface. All the features of RobOff are supported by the GUI as well as the command line. If you prefer command line interfaces or need to use commands in automated scripts, see Section 3.2, “Running RobOff from the command line”, p. 46. Listed below are some of the general features of the RobOff GUI:

- Loading existing RobOff setups
- Editing and management of RobOff setups
- Saving modified setups
- Calculation and visualization of results in different windows, tables, and plots
- Tracking of the optimization process
- Browsable and searchable help/manual

The RobOff GUI is relatively self-explanatory. Probably the best way to learn how to use it is by experimenting and looking at examples. It should be possible to start using the RobOff GUI almost straight out of the box, as long as one has a working understanding of the RobOff conceptual model (i.e., how to define a RobOff setup: actions, environments, features, responses, etc.). The next sections describe the basic structure of the RobOff GUI and its main features.

4.1. Main window

From the main window, you can open an existing setup, create a new setup, save the current setup, edit a setup, check the validity (consistency), visualize results for different allocations or resources, and find optimal allocations for different criteria. These options are also available from the main toolbar at the top of the graphical interface. There is also an online help system. Most of its content is shared with this manual.

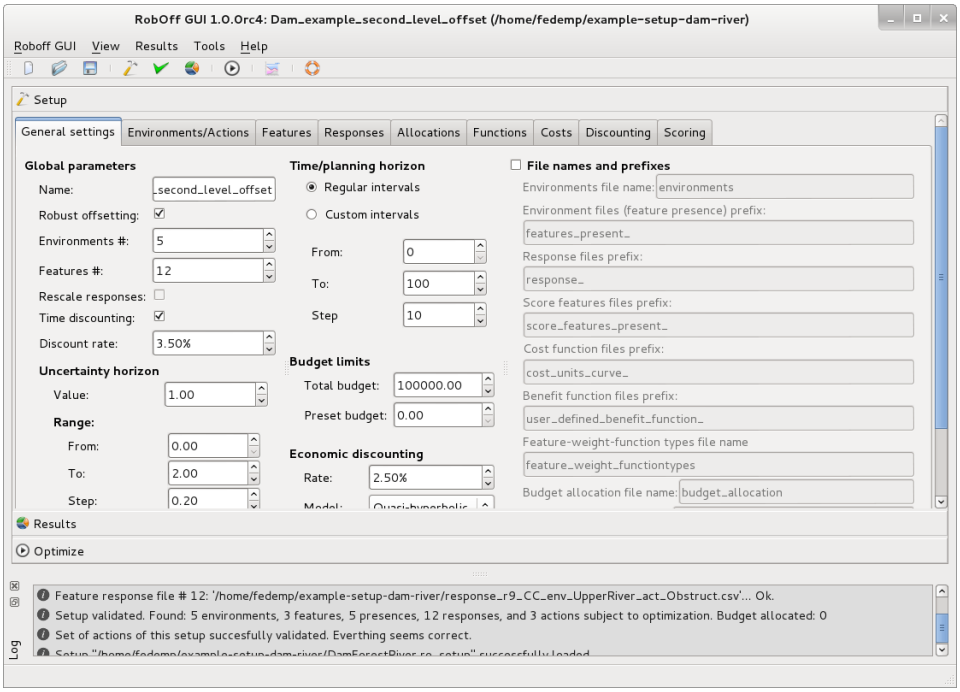
The graphical interface consists of three main blocks or sections that can be opened one at a time: *setup*, *results*, and *optimization*. These three sections are part of a toolbox widget:

- In the *setup* section, you can configure general settings and edit all the entities that are part of a RobOff setup (see Chapter 2, *Framework, Methods and Algorithms*, p. 17): environments, biodiversity features, per area unit responses of features to actions, conservation and/or development actions, costs, discounting rates, and scored features. Typically each of these types of entities are edited in separate tabs. This part of the RobOff GUI allows for defining setups without the need to edit input text files.

- In the *results* section, results can be visualized across different dimensions. Note that it is not required to perform any optimization step in order to visualize results for mandatory and preset resource allocations.

- In the *optimize* section, it is possible to find optimal allocations. A number of optimization methods and options can be selected.

Figure 4.1. RobOff main window



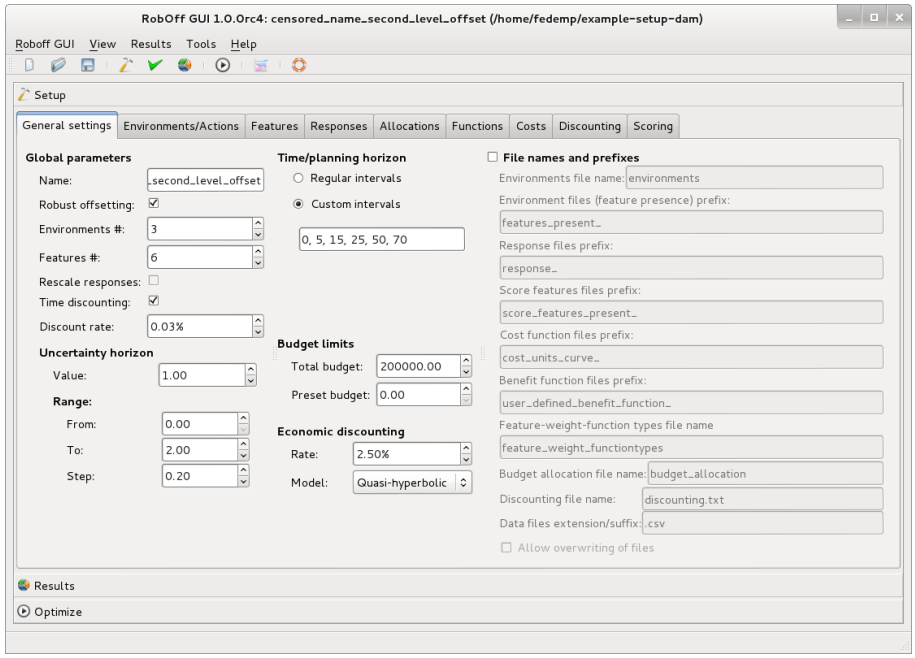
The log window is normally shown at the bottom of the graphical interface, but it can be dragged and detached. This window provides feedback about the operation of RobOff, including warnings and error messages to the user. The visibility of this window can be toggled on/off from the main menu.

4.2. Setup

In this section, RobOff setups can be defined in different tabs. Each of the tabs correspond to one of the main entities in the RobOff framework: environments, features, specific responses of features to actions, actions and their costs, time discounting, scores, etc.

General settings can be edited in the leftmost tab. This includes general parameters, such as the name or description of the setup, and values that define the context of the RobOff analysis, such as the uncertainty horizon, planning (temporal) horizon, time discount rate and budget limits. These last parameters have a strong influence on results and you will probably need to explore their impact (see more on this below, in Section 4.3, “Results”, p. 96. The economic discount model and rate are relevant only if time-dependent costs are defined. In this tab it is also possible to modify file names, even though this is not normally needed Figure 4.2, “Editing a setup (general settings)”, p. 89.

Figure 4.2. Editing a setup (general settings)



The core of a RobOff setup includes four types of entities: environments, actions, biodiversity features, and the responses of features to actions in environments (see Figure 2.1, “Basic components of a RobOff setup”, p. 20 in Section 2.2, “RobOff Setups”, p. 20). In the RobOff GUI setup section these entities can be created and modified in the following tabs:

- *Environments/Actions*: see Figure 4.3, “Editing environments and actions in the GUI”, p. 90. The information that can be edited in this section is contained in the environments file (see Section 3.3.2, “Environments file”, p. 58).

- *Features*: see Figure 4.4, “Editing biodiversity features in the GUI”, p. 91. The information that can be edited in this section is contained in the set of files of biodiversity features present (see Section 3.3.4, “Set of files: biodiversity features...”, p. 62). For each environment there is a row in the table of environment and a specific table of features present, which correspond to one file of biodiversity features present.

- *Responses*: Uncertain responses can be edited and visualized simultaneously in the 'responses' tab (see Figure 4.5, “Editing uncertain responses in the GUI”, p. 91). Each of these responses is contained in one of the files of responses of biodiversity features (see Section 3.3.5, “Set of files: responses of biodiversity features”, p. 63).

Figure 4.3. Editing environments and actions in the GUI

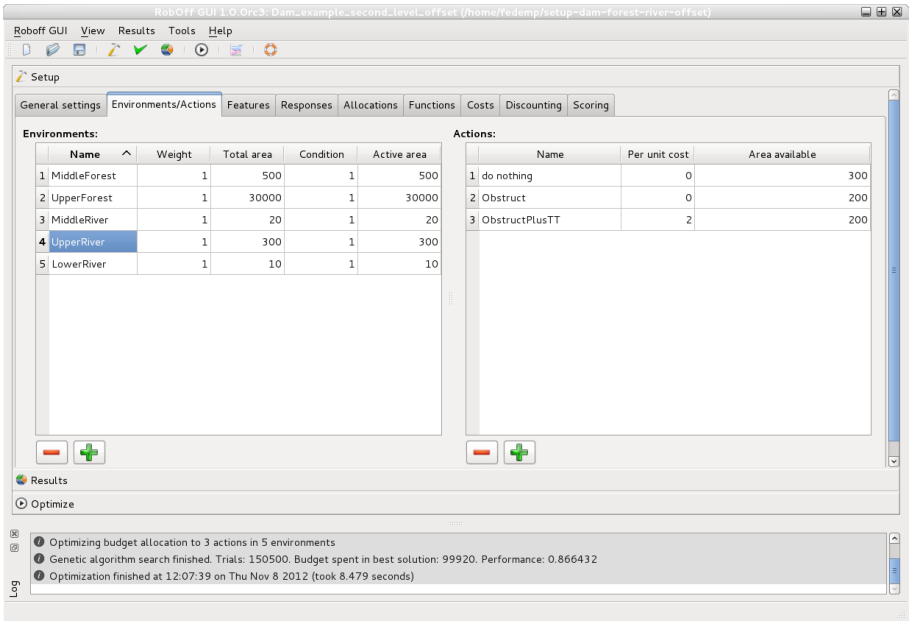


Figure 4.4. Editing biodiversity features in the GUI

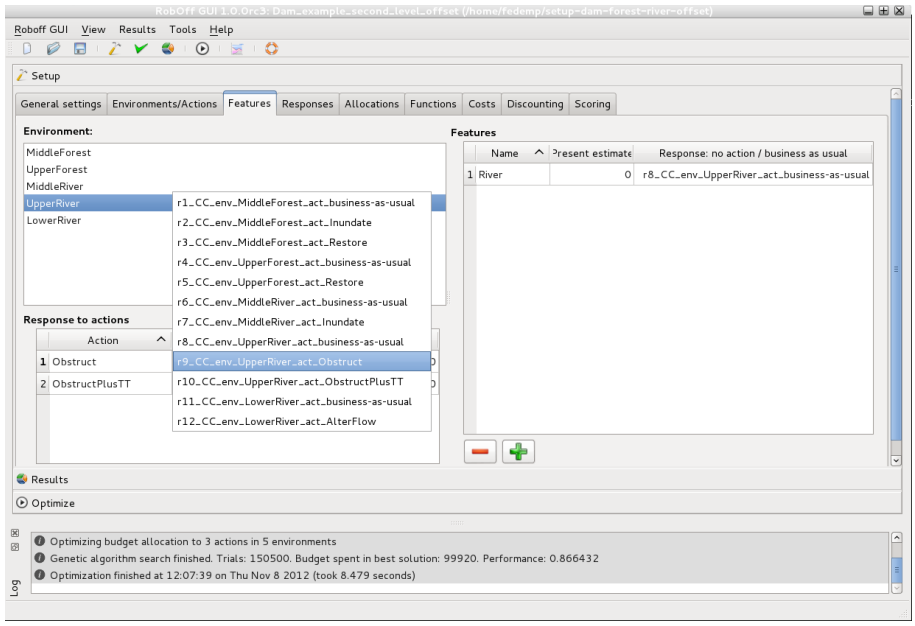
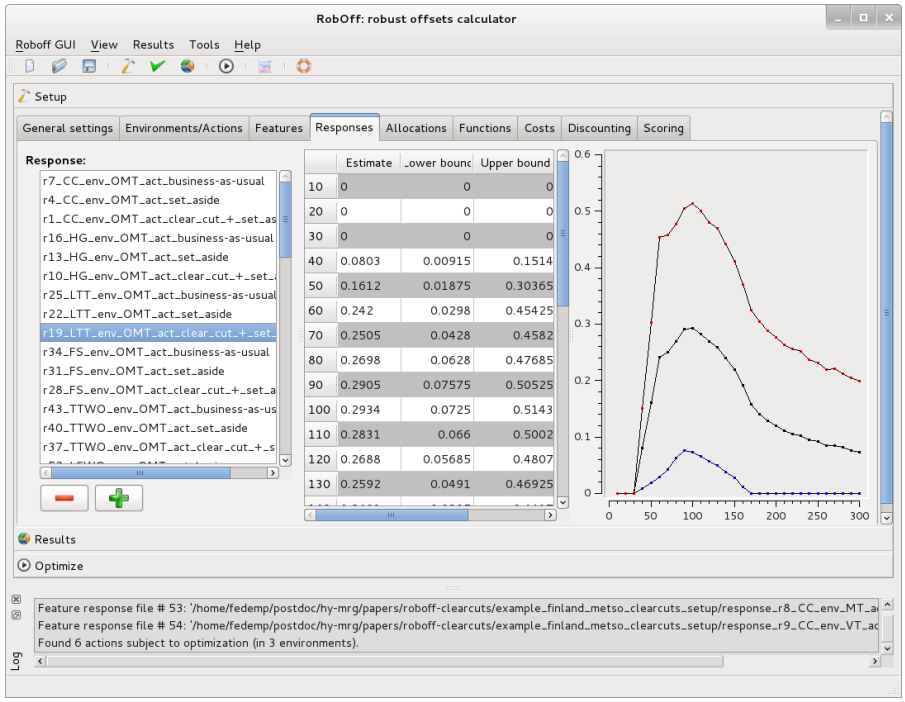


Figure 4.5. Editing uncertain responses in the GUI



All the elements described in Chapter 3, *The RobOff Software and Command Line Interface*, p. 45 can be edited via tables and various widgets. For example, it is possible to define simple and score features, as well as the time discounting model and parameters. These are the tabs where additional setup details can be defined:

- *Allocations*: see Figure 4.6, “Editing allocations in the RobOff GUI”, p. 93. The information that can be edited in this section is contained in the budget allocationen file (see Section 3.3.7, “Budget allocation file”, p. 65). Note that this tab is about predefined allocations, for how to obtain optimal allocations see Section 4.4, “Optimization”, p. 103.

- *Benefit functions*: see Figure 4.7, “Editing benefit functions in the RobOff GUI”, p. 93, where the weights and utility functions for every biodiversity feature present in any environment can be edited. In this tab every row corresponds to a different biodiversity feature. The information that can be edited in this section is contained in the file of Feature - weights - utility functions (see Section 3.3.3, “Feature - weights - utility functions file”, p. 59).

- *Costs*: see Figure 4.8, “Editing costs in the RobOff GUI”, p. 94. The information that can be edited in this section is contained in the environments file (for constant costs, see Section 3.3.2, “Environments file”, p. 58), or in the cost files (see Section 3.3.9, “Set of files: costs of actions”, p. 68) (for variable costs, either as an area-cost curve or as a time-dependent curve).

- *Time discounting model and parameters*: see Figure 4.9, “Editing time discounting model and parameters in the RobOff GUI”, p. 94. The information that can be edited in this section is contained in the time discounting file (see Section 3.3.6, “Time discounting file”, p. 65).

- *Score features*: see Figure 4.10, “Editing score features in the RobOff GUI”, p. 95. The information that can be edited in this section is contained in the files of score features (see Section 3.3.8, “Set of files: score features”, p. 66).

Figure 4.6. Editing allocations in the RobOff GUI

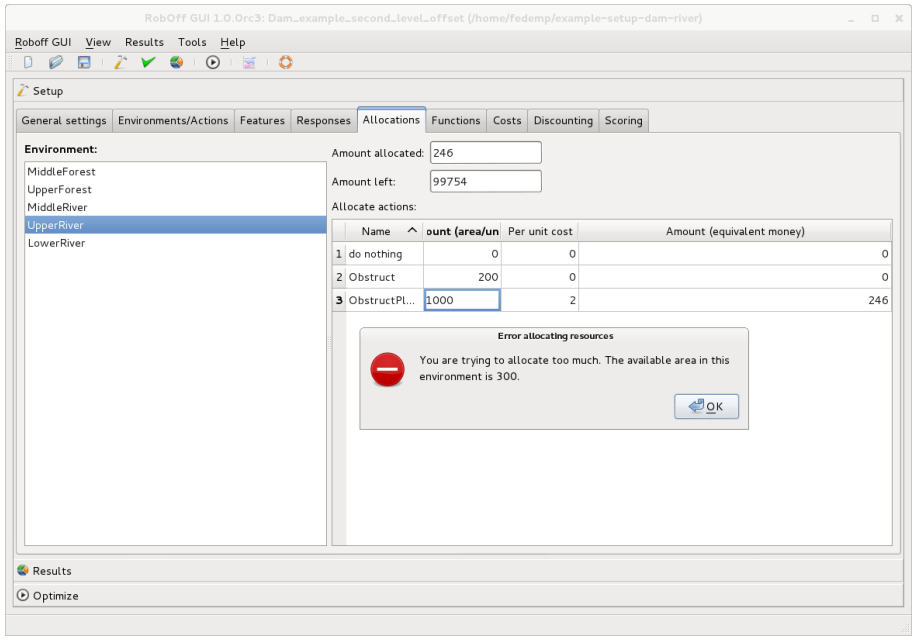


Figure 4.7. Editing benefit functions in the RobOff GUI

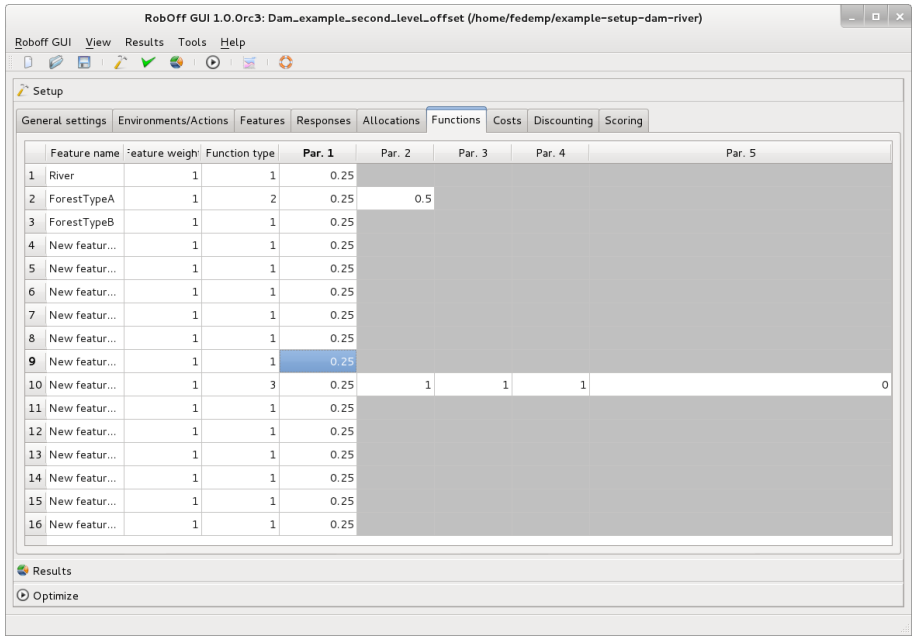


Figure 4.8. Editing costs in the RobOff GUI

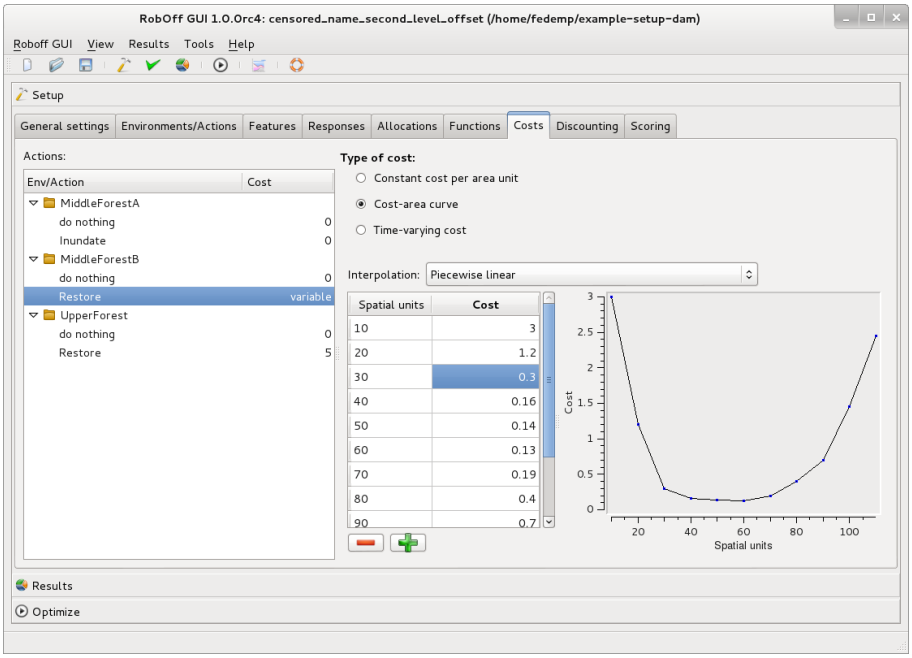


Figure 4.9. Editing time discounting model and parameters in the RobOff GUI

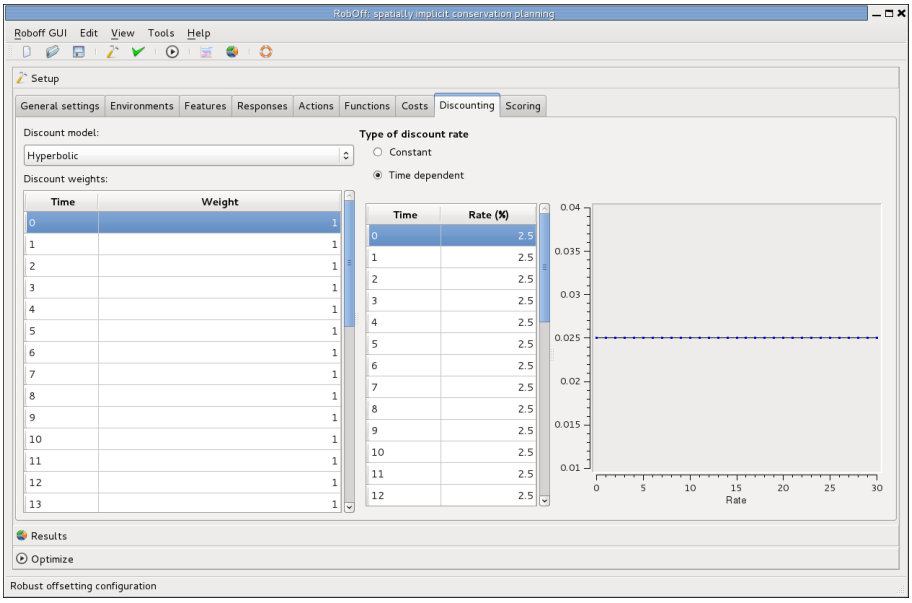
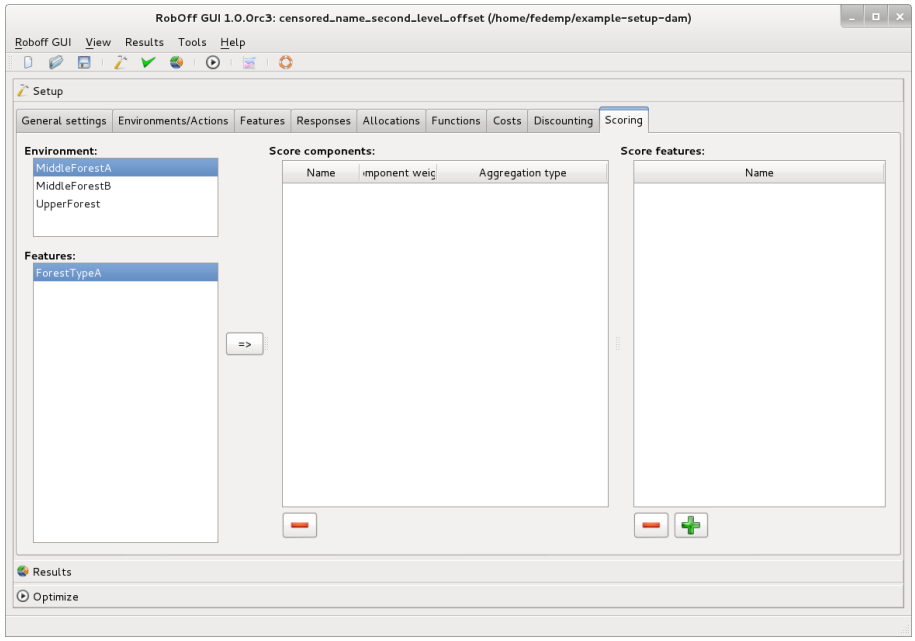


Figure 4.10. Editing score features in the RobOff GUI



4.3. Results

This section of the GUI shows global results as well as results through time and across environments, features, and actions. The first (leftmost) tab in the results section of the RobOff GUI shows a table with a summary of results. The remaining tabs allow inspecting results across different dimensions: time, environments, features, actions, etc. In addition there are tabs for specific results, such as comparisons of actions. These are described in the next sections.

As a general working principle, the list or tree on the left of each of these tabs (or visualization dialogs) allows the user to select all, groups of, or individual entities (features, actions, environments, etc.). On the right, two plots visualize results. These two plots are separated by a splitter element that can be freely shifted up and down. In general, it is possible to zoom in/out using the first/second mouse button on all the RobOff plots. The plot coordinates at any location can be obtained by clicking on the plot's canvas. Also, by right-clicking on the plots, their content can be copied into the clipboard or, alternatively, saved as an image and/or document. Common formats such as png, svg, jpg, pdf, and postscript are supported.

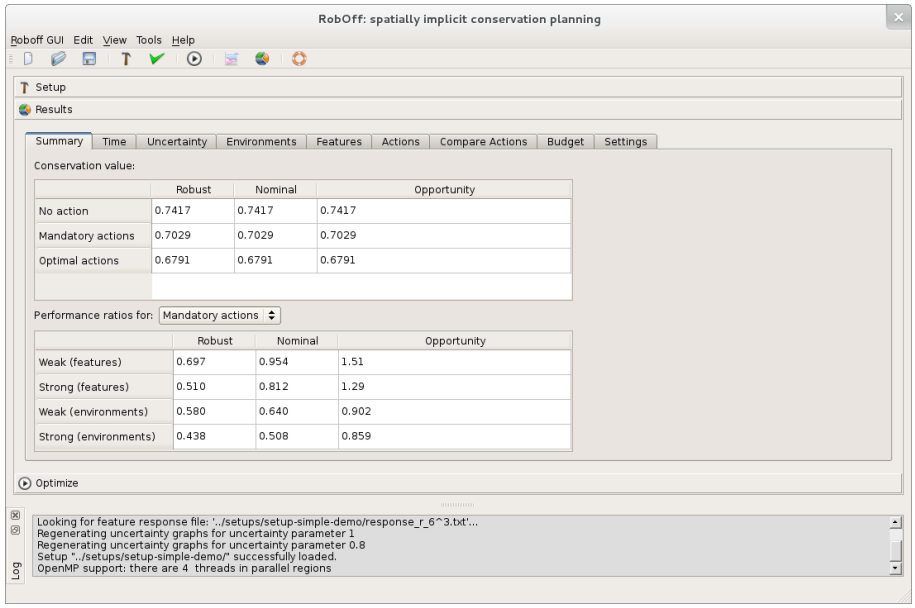
The upper plot shows conservation value whereas the lower plot shows sustainability performance indices. It is possible to select different robustness requirements: nominal, robust, or opportunity. Sustainability indices can be weak and strong. In general, the plotted values are conditioned to the entities selected on the left. Also, plots are cumulative. Plots are added one at a time, and the content of both plot panels can be cleared at any moment to start creating a new plot from scratch. For instance one can combine in the same plot the curve of conservation value over time for the whole set of features and a particular feature. As another example, it is possible to add in the same plot the weak sustainability index over time for different individual features.

The table of results summary corresponds to the results summary file in standard RobOff outputs (see Section 3.4, “Standard RobOff output”, p. 70).

Note

This section explains how to visualize results interactively. To save results into text (csv) files for later analysis you can use the option `Results - Save into text files`, from the main menu of the RobOff GUI, or the equivalent `save results` from the tool bar. Both can be normally found at the top of the main window. This will generate a set of files in the specified output directory. For details on the format of these files see Section 3.4, “Standard RobOff output”, p. 70.

Figure 4.11. Visualizing results summary



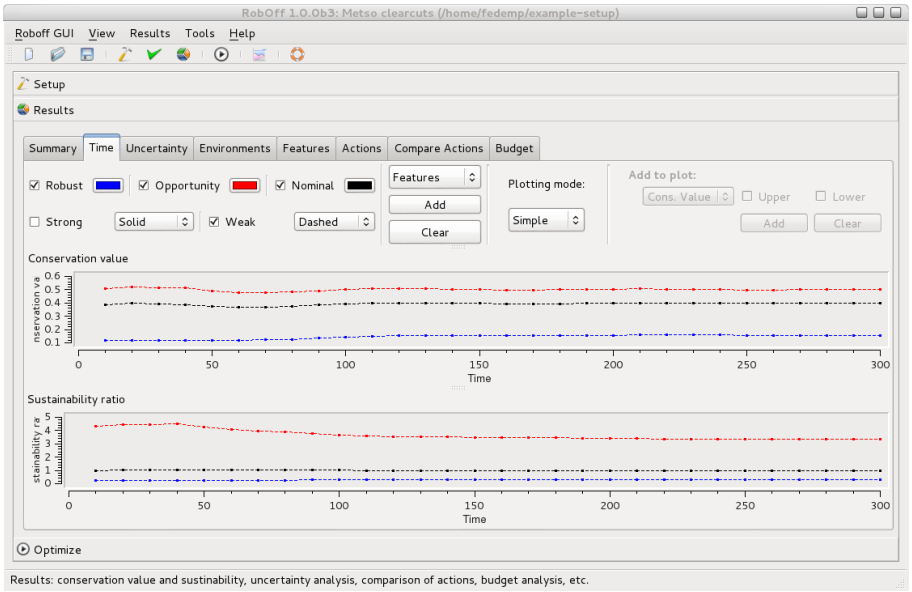
Tip

By default, the plots of the results section will be updated every time a correct setup is loaded into the GUI. The update is done by adding the new curves to the curves already displayed. By loading a sequence of setups and changing the plot properties (color and line style) you can build plots that display results for multiple setups simultaneously. You can restart the process at any time for any of the result plots by using the `clear` buttons. This way it is for example easy to generate plots of the sensitivity to uncertainty for different setups (or variations of a same setup for different resource allocation options), see next sections.

4.3.1. Visualizing results across different dimensions

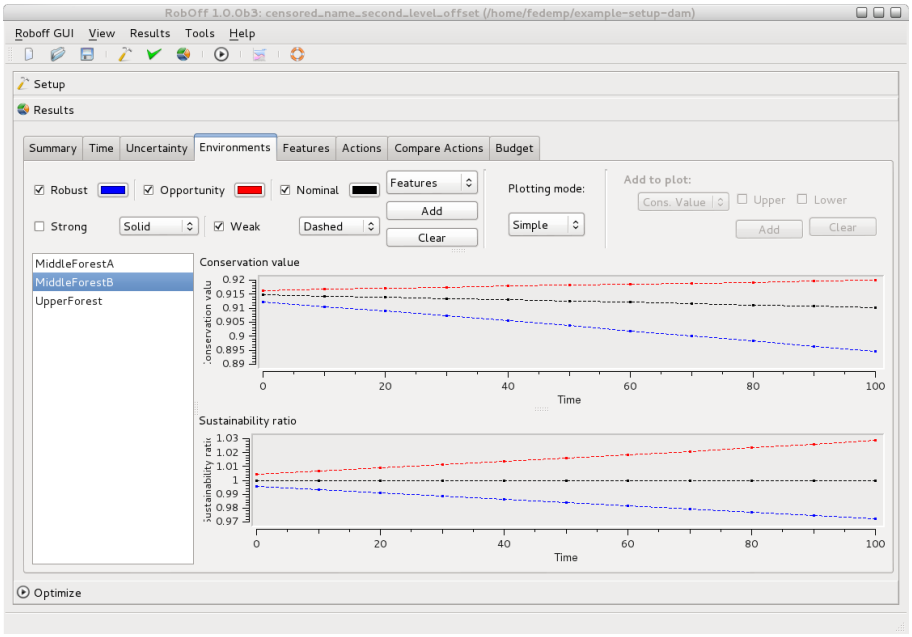
The RobOff output space can be seen as the Cartesian product of multiple and heterogeneous dimensions (see Section 2.4, “The RobOff output space”, p. 23. In the results section of the GUI it is possible to visualize the evolution of conservation value and sustainability performance ratios over time in the following tabs: 'time', 'environments', 'features', and 'actions'. On the left, a tree of environments, features, or actions (normally grouped by environment) are shown. By selecting individual entities or groups of them (using the shift and control keys) it is possible to visualize results specific to subsets of features, environments, or actions.

Figure 4.12. Visualizing results through time



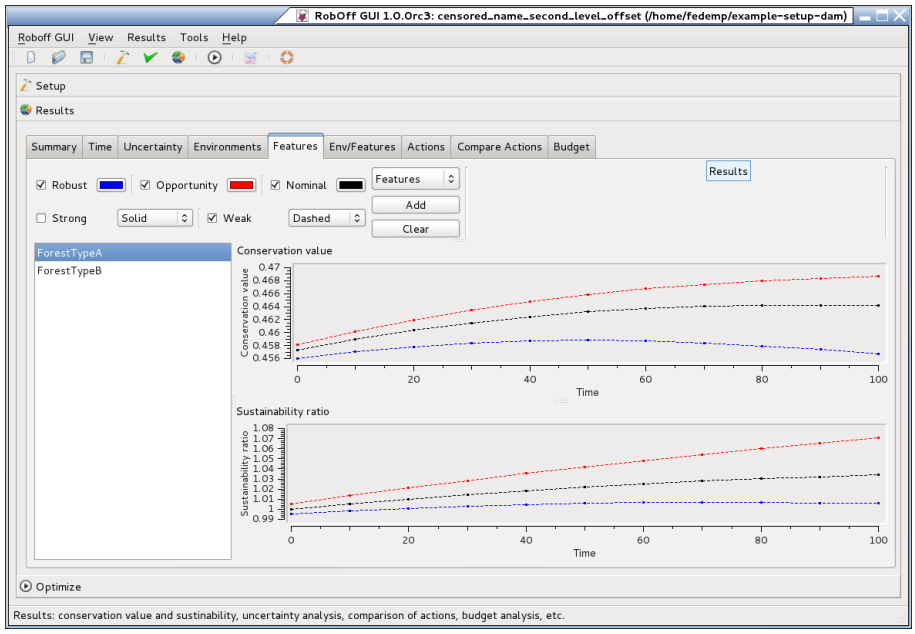
Environment specific results are conditioned to individual environments or groups of them.

Figure 4.13. Visualizing results across environments



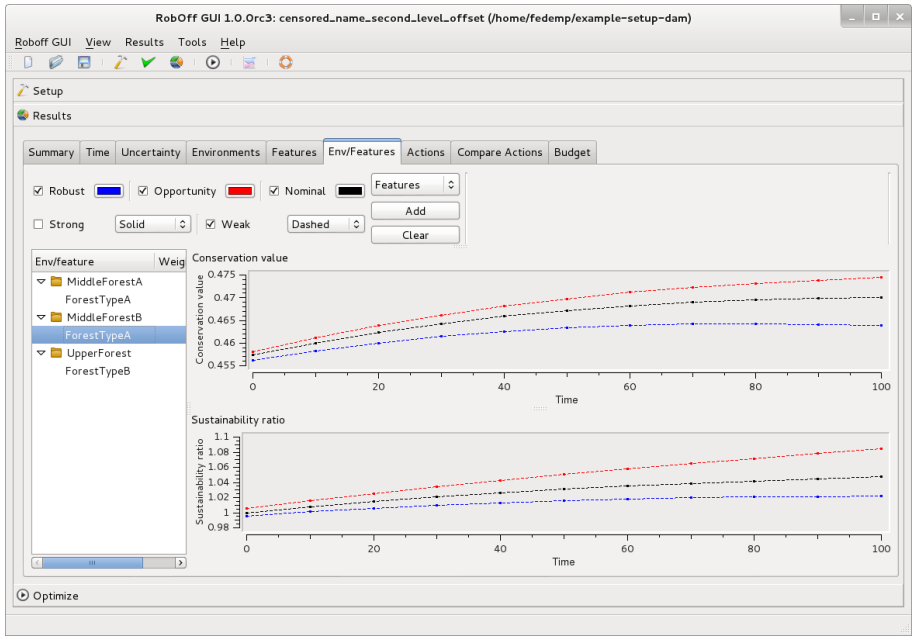
Feature specific results are conditioned to individual or groups of features (globally, across all environments) and shown in a list of features.

Figure 4.14. Visualizing results across features



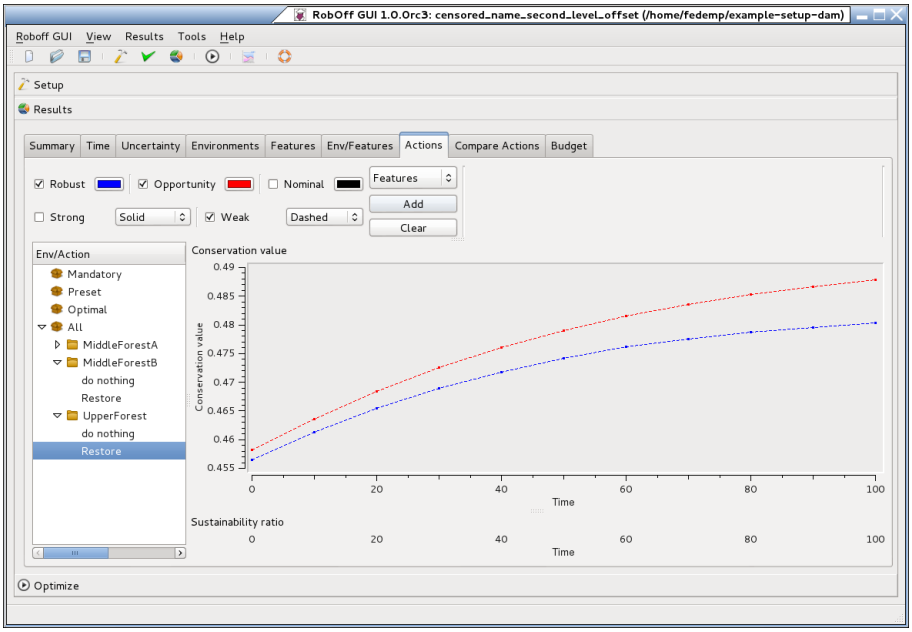
Features within environments (`Env/Features` tab) are conditioned to individual or groups of features within a single environment and shown a tree that groups features into environments where they are present.

Figure 4.15. Visualizing results across features within environments



Action specific results are conditioned to individual or groups of actions and shown in a tree that groups actions into environments where they are possible.

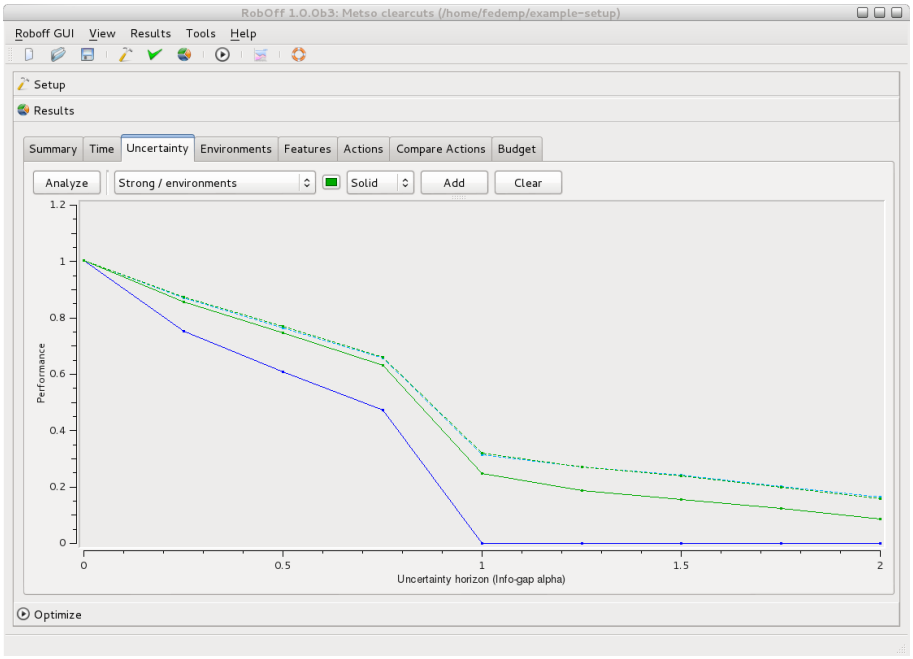
Figure 4.16. Visualizing results across actions



4.3.2. Sensitivity of results to uncertainty

In the 'uncertainty' tab it is possible to visualize results against the uncertainty horizon (or degree of uncertainty). This is the core of the anlysis type uncertainty, see Part V, “RobOff analysis setups for common planning needs”, p. 107 for more details.

Figure 4.17. Visualizing results as a function of the degree of uncertainty



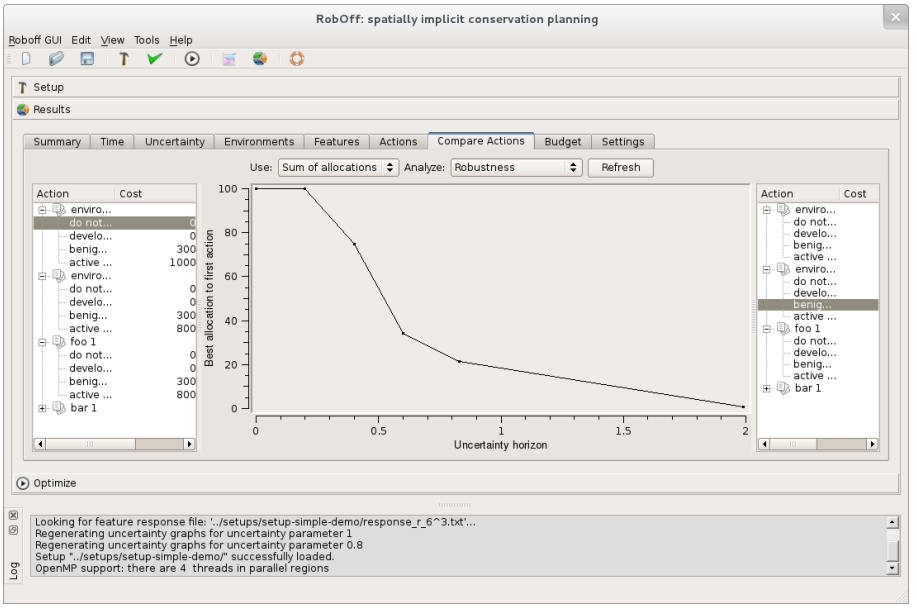
4.3.3. Comparison of alternative actions

In the 'compare actions' tab it is possible to visualize a special type of result: trade-offs between alternative actions as a function of the degree of uncertainty. Two alternative actions can be selected in the left and right trees, respectively. For convenience, these trees group actions into their respective environments.

The (pairwise) actions comparison plot shows on the vertical axis the percentage of resources allocated to the action selected on the left as compared to the amount of resources allocated to the action on the right, as a function of the degree of uncertainty or info-gap alpha value (on the horizontal axis). Gaps in the actions comparison plot denote that no resources are allocated to none of the actions.

Note that before comparing any pair of actions a specific analysis needs to be performed. This requires finding the best allocation of resources for different degrees of uncertainties (uncertainty horizon, on the x axis). This analysis process, started by pushing the 'analyze' button, can be time consuming depending on the complexity of the setup. The computation time increases linearly with the number of steps of the uncertainty range.

Figure 4.18. Comparing actions



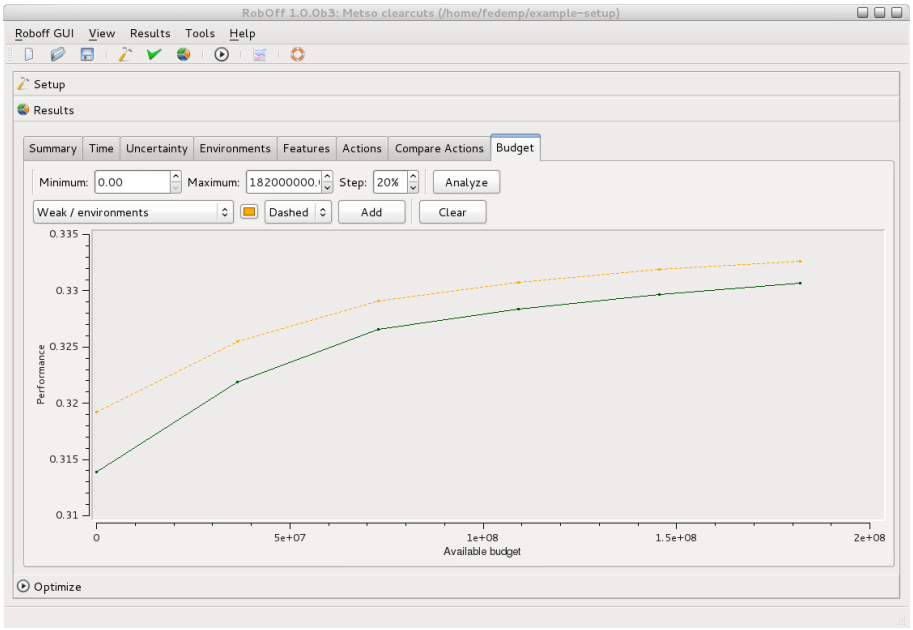
Note

The actions comparison plots will be empty if no resources are allocated to none of the selected actions for any degree of uncertainty, which can be fairly common in setups with large numbers of actions and/or limited budget. In such cases, most comparisons between pairs of actions will result in empty or partially empty plots. It is thus advisable to check first which actions are allocated some resources before starting to compare pairs of actions at random.

4.3.4. Sensitivity of results to Budget variations

Similar to the analysis of sensitivity to uncertainty, in the 'budget' tab it is possible to analyze how results vary as a function of available budget. This type of analysis can be useful to compare the return on investment for different budget levels. The range of variation of the available budget can be set into the setup general settings. Alternatively, see Chapter 3, *The RobOff Software and Command Line Interface*, p. 45 for details on how to define the range of budget variation in input files.

Figure 4.19. Visualizing results as a function of the available budget



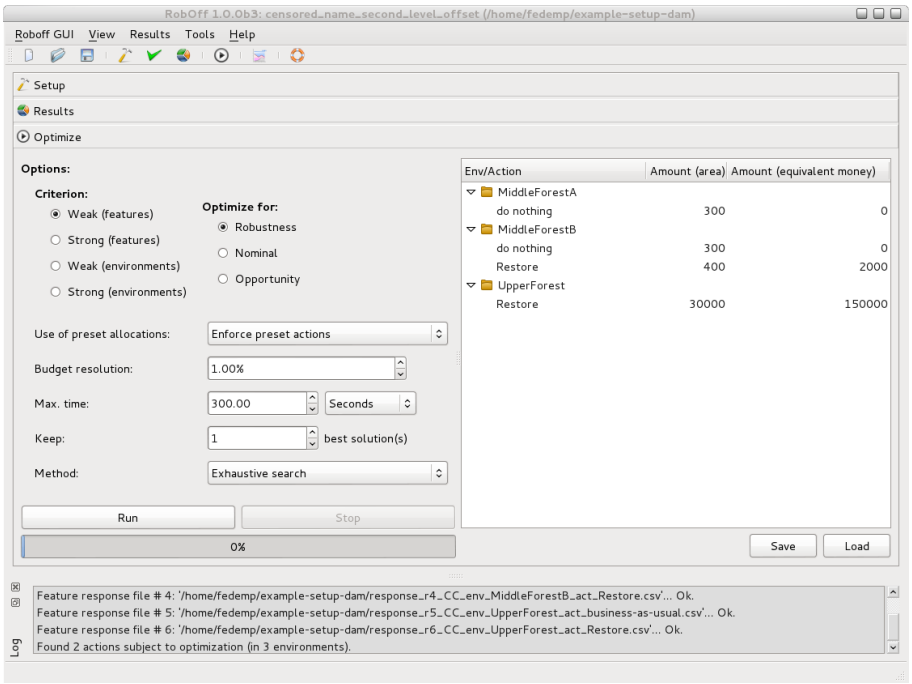
4.4. Optimization

In this section of the RobOff GUI you can find the optimal allocation of resources according to different criteria and constraints. Once you select the necessary options and run the optimization process, the optimal allocations found will be visualized on the right via a tree of actions to which resources have been allocated.

The main options that need to be selected before starting an optimization are:

- Optimization method (greedy search, exhaustive search, genetic algorithm, grid search with local operator, etc.). For details on the different optimization approaches see Section 2.6, “Optimizing resource allocation”, p. 35.
- Metric to optimize: variant of sustainability index (weak or strong performance ratio of sustainability across features or environments)
- Robustness criterion: robust, nominal, opportunity
- Additional options: budget resolution, time bounds, etc.

Figure 4.20. Setting optimization options



4.5. Preferences

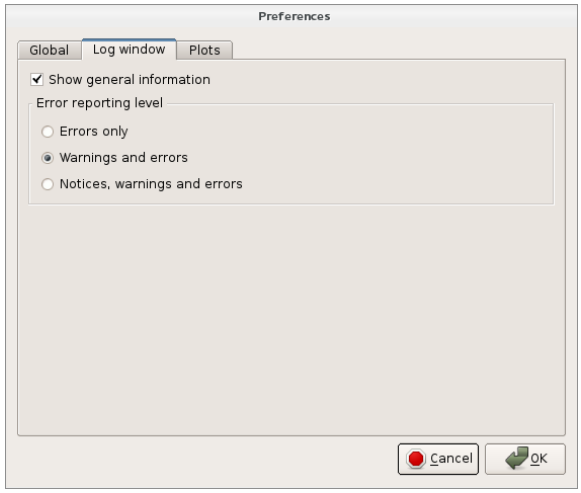
The dialog of general RobOff GUI preferences can be accessed from the main menu: Tools -> Preferences. The most relevant settings are:

- Level of verbosity in the log window: general information and error reporting level of verbosity (report only errors, or errors and warnings, or errors, warnings and notices). It is also possible to enable messages about general RobOff operation, such as consumption of resources (memory and processing time), progress of computations, etc.

- Number of cores to use in parallel calculations. By default, the maximum number of available cores will be used. This is especially relevant when optimizing resource allocations. Remember to reduce this number if you want to perform other tasks in the same computer (possibly several RobOff optimizations in parallel).

- Background color in plots. This parameter is used when saving plots or copying them into the clipboard. Its default value is white with alpha transparency (for those formats that support alpha channel transparency, such as png).

Figure 4.21. Editing RobOff GUI preferences



Part V. RobOff analysis setups for common planning needs

Table of Contents

Chapter 5. RobOff analysis setups for common planning needs

5.1. General remarks

RobOff is designed to help answer questions concerning the type of actions that should be undertaken and how much of each type. Normally this is just one (quantitative) aspect of the conservation planning process. In a more comprehensive view, the RobOff flow would be embedded or combined with general frameworks for conservation planning. See, for example, Wilson KA et al. (2007) Conserving Biodiversity Efficiently: What to Do, Where, and When. PLoS Biol 5(9): e223.

Roughly speaking, a RobOff setup consists of two parts (see Figure 2.1, “Basic components of a RobOff setup”, p. 20 for a conceptual diagram). The first part includes all the proper inputs and is more related to observations of ecosystems (e.g., environments, biodiversity features, actions that can be undertaken, and responses to actions). The second part comprises inputs that can be seen as parameters and are more related to interpretation and human decisions (e.g., degree of uncertainty, robustness requirement, discounting, costs and budget in some cases, etc.). Normally, the second part is more variable, as it essentially consists of parameters for calculations and optimizations peformed in RobOff.

For details on how to define a RobOff setup, including all the aforementioned inputs and parameters, in the RobOff graphical interface, see Chapter 4, *RobOff Graphical User Interface*, p. 87. Alternatively, Chapter 3, *The RobOff Software and Command Line Interface*, p. 45 describes how to define a RobOff setup in plain text files.

Environments, actions, biodiversity features and responses

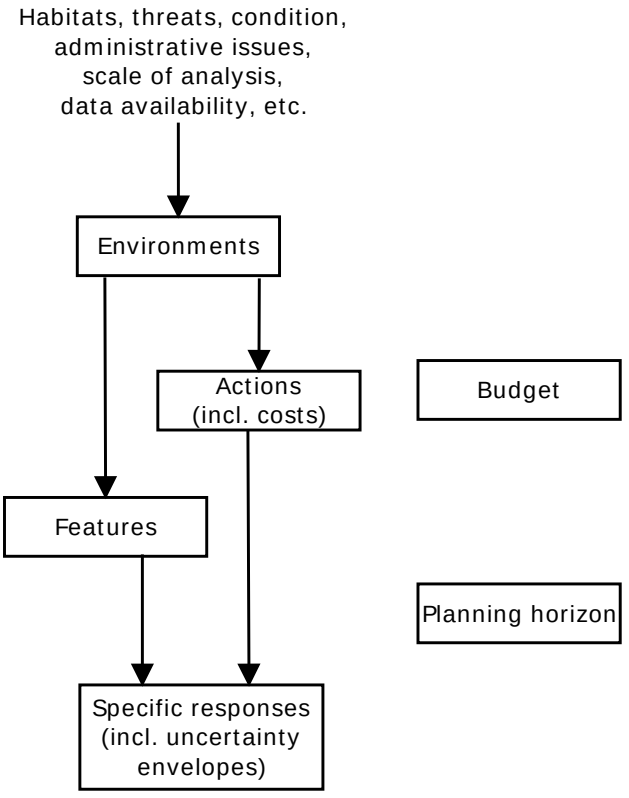
Figure 5.1, “A possible, simplified, sequence of steps required to define a RobOff setup”, p. 110 depicts a "top-down" sequence of steps to define a RobOff setup for a given planning problem (this is just one possible approach). First, environments need to be defined. To this end, one has to identify the set of habitat types relevant for planning. In principle, environments and habitat types are very closely related, but an environment is defined in RobOff as an aggregation of sites that are dealt with as a block for planning purposes. In practice, this means that when an environment is defined in a RobOff setup, it is possible to define actions, a list of biodiversity features present, and responses of features that are specific to that environment.

If planning is to be performed exactly on a habitat-by-habitat basis, then it is sensible to define an environment for each habitat. But more in general, additional factors should be considered in order to decide whether some or all of the sites

belonging to a habitat should be merged or split into different environments. Some factors that may suggest merging or splitting habitats include different conditions, levels of threat, availability of data, differences in administrative regions and/or management practices and requirements, and the scale and scope of analysis. Environments are the entities at the highest level of abstraction in RobOff setups. As such, the choice of environments will condition the definition of all the other types of entities (actions, features, responses, costs, etc.).

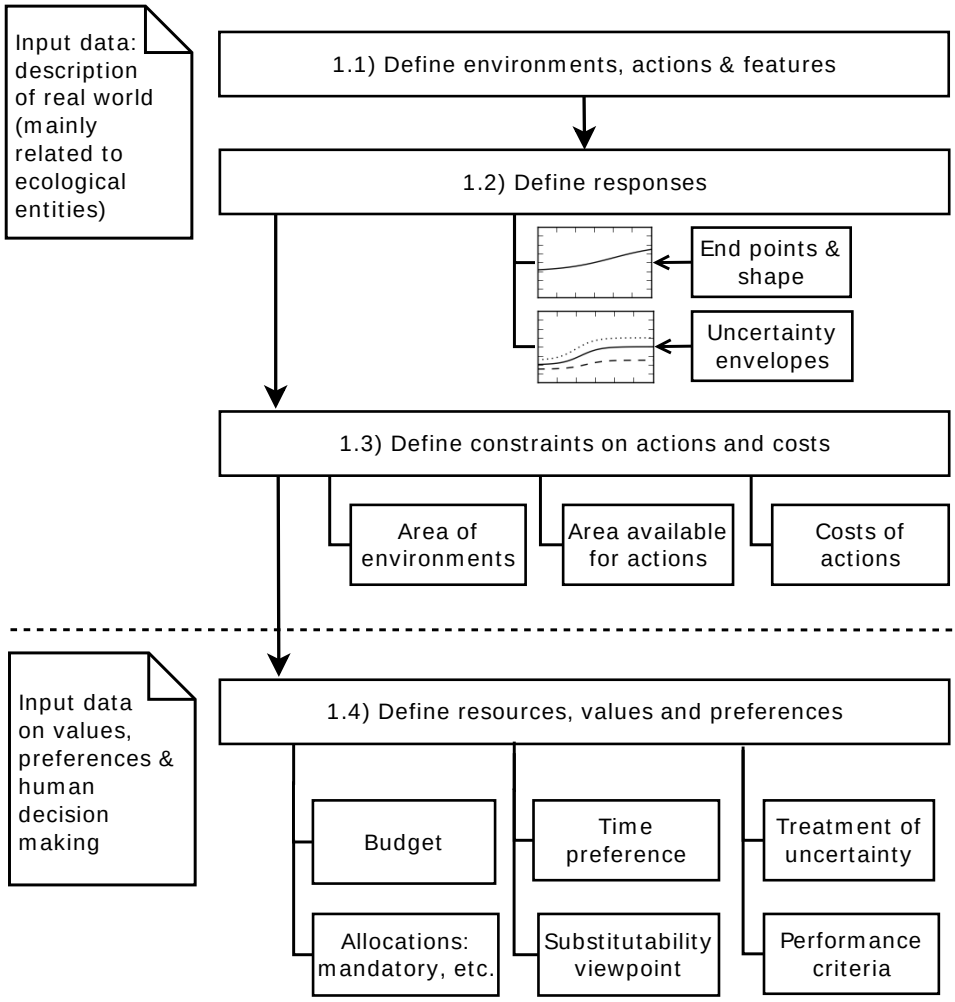
The second step consists of defining possible actions in every environment for which a cost has to be estimated. A global budget or budget range related to these costs should be estimated as well. Next, the set of biodiversity features present in each environment has to be defined. Finally, specific responses of biodiversity features to every possible action in every environment must be defined. Before defining these responses, the planning horizon that is relevant and/or viable has to be decided upon. Specific responses of features to actions are uncertain time-discrete responses, which has two implications: 1) responses should be defined for a consistent sequence of intervals (time span or planning horizon), and 2) each response is threefold, comprising best estimate, upper envelope, and lower envelope of the response.

Figure 5.1. A possible, simplified, sequence of steps required to define a RobOff setup



In a more comprehensive and strict sequence, the process of defining a RobOff setup can be split into four steps Figure 5.2, “Flow of definition of RobOff setups.”, p. 111. The definition of a setup can be seen as a first stage in the use of the RobOff framework and software, leading to a second stage (decision analysis and optimization), and a third stage (interpretation of results). For details on optimization, see (chapter methods, then command line and then GUI). For interpretation of results, the most relevant sections of this manual are Section 2.4, “The RobOff output space”, p. 23, Section 4.3, “Results”, p. 96 (for the GUI) and Section 3.4, “Standard RobOff output”, p. 70 (for the command line tool),

Figure 5.2. Flow of definition of RobOff setups. Adapted and extended from (Pouzols & Moilanen 2013).



Time preference and robustness against uncertainty

Two additional and very important aspects of any planning problem that need to be specified are time preference and requirement of robustness against uncertainty. These two aspects are particularly subject to opinion and interpretation. Once all the entities and parameters described above have been defined, it is possible (and common, even advisable) to evaluate results for different time preferences and robustness requirements. Similarly, for types of analysis that involve some form of optimization, it is possible to calculate optimal allocations of resources for different variants of time preference and robustness requirements.

Time preference is defined in RobOff by setting the following inputs: time discounting (model and rate), and time weights. See Section 2.5.4, “Time discounting”, p. 33 for methodological details.

Related to uncertainty analysis, three levels of robustness (or robustnes requirements for optimization) are considered in RobOff: nominal, robust, and opportunity. Results are provided for the three alternatives (both in the console and graphical versions of RobOff). However, each optimized solution is calculated for a particular robustness requirement. Often, one will be interested in optimal allocations for different robustness requirements and possibly different time preference options. For example, how close is the optimal robust solution to the optimal opportunity solution? How sensible is the best allocation of resources to the discounting rate?

5.2. Analysis types

The following is a brief list of different types of analyses that can be performed using the RobOff methods and software. While RobOff puts a strong focus on biodiversity offsetting, these and additional analyses are applicable to a wide range of situations involving both development and conservation actions, such as land use planning or investment in agri-environment schemes.

- *Type I - Analysis of uncertainty of conservation value*. In this analysis one investigates the uncertainty of conservation value through time for a pre-specified set of actions. One can ask, for example, how much uncertainty around the nominal value can be tolerated such that the conservation value of the landscape can be expected to not decrease from present. This analysis does not necessarily involve optimization.

- *Type II - Analysis of the time perspective*. Most studies about reserve selection and systematic planning have either explicitly or implicitly assumed static biodiversity patterns. However, in reality, some actions may result in conservation losses or gains immediately while for others, losses or gains may occur more slowly over time. In this analysis one is concerned about how time discounting influences our perception of how beneficial or detrimental some actions are. This analysis does not require optimization if assuming a pre-specified set of actions.

- *Type III - Reliability of biodiversity offsets*. In this analysis one is interested in how reliably a set of compensatory actions (offsets) can compensate for a set of ecologically damaging actions. Are the proposed offsets likely to be adequate? Here, the need for analysis follows from the fact that ecologically damaging actions frequently cause immediate and certain loss of ecological value whereas offsetting, if it is implemented via restoration, only produces uncertain gains with a time delay. There are two variants of this analysis that correspond to the concepts of weak and strong sustainability. Again, this analysis does not require any optimization if assuming a pre-specified set of actions.

- *Type IV - Optimal division of resources between conservation actions*. Conservation actions are available given constraints to the budget. Given the present state of biodiversity features, actions, and their uncertain effects through time, which set of actions would deliver the highest conservation value accounting for uncertainty, time, costs, and all other factors summarized in the sections describing the data and computational models?

- *Type V - Optimal biodiversity offsetting*. A set of environmentally damaging actions has been proposed. With which actions in what environments does one best compensate for the damage, given limitations to budgets and to the amounts of areas where different compensatory actions are feasible?

- *Type VI - Extraction of targets*. An optimal allocation of resources (output from analysis type IV) consists of amounts (areas) assigned to different alternative actions. This set of amounts of areas can be seen as a set of area targets, which are specific to particular actions in individual environments. Area targets obtained this way can then be spatially allocated by further analysis, for example using target-based site selection methods.

5.3. Uncertainty

Note

This section and the next ones are under construction and will be extended over the coming weeks. Keep an eye on our website for updates! http://www.helsinki.fi/bioscience/consplan.

The degree of uncertainty in the responses of features to actions (or info-gap uncertainty horizon parameter, or alpha value, see Section 2.5.1, “Uncertainty Analysis”, p. 25) determines how much the upper and lower envelops expand from the nominal response. In RobOff, there is an uncertainty horizon parameter that can be changed in the general settings tab of the setup section of the GUI (see Section 4.2, “Setup”, p. 89), or, alternatively, in the general settings input file (option `info-gap alpha`, see Section 3.3.1, “General settings file”, p. 50. This parameter will be used as the reference value of uncertainty, and it sets the degree of uncertainty for which most results are reported (including the summary table, plots and time series of conservation value and sustainability ratios over time).

In addition to the info-gap uncertainty horizon parameter (alpha), it is possible to define a range of variation for the info-gap uncertainty horizon. Similarly, this can be done in the general settings tab of the setup section of the GUI, or, alternatively, in the in the general settings input file (option `info-gap alpha range`, see Section 3.3.1, “General settings file”, p. 50. This range is used in the analysis of sensitivity of results to the degree of uncertainty. See Section 4.3, “Results”, p. 96 for details on how to visualize the results of this analysis in the GUI. Alternatively, see Section 3.4.1, “Optional output files”, p. 75 for details on the output files generated by the uncertainty analysis.

5.4. Time

Time preference accounts for the change of relative values over time, or how to compare values in the future to values in the present. In RobOff, the relative conservation values over time are defined by three parameters: the discounting model, the discount factor factor (which can be constant or variable over time) and weights for every time interval in the planning horizon.

In the general settings input file (see Section 3.3.1, “General settings file”, p. 50) the relevant options are: `time discounting model` and `time discounting rate`. In the GUI, these parameters can be defined in tables that can be found in the `Discounting` tab of the setup section (see Section 4.2, “Setup”, p. 89).

The influence of these parameters on the results of a RobOff analysis will depend on how immediate or delayed the effects of actions on biodiversity features are. The effects of time discounting can be observed in most of the results generated by RobOff, from the summary table to all the plots and time series of conservation value and sustainability ratios over time.

5.5. Offsetting

This section is intended to provide general guidance on how to analyze biodiversity offsets within the RobOff framework.

5.5.1. How much compensation is enough

In the context of biodiversity offsets, one often wishes to find an answer to the question of how much compensation (restoration, maintenance, etc.) is enough to compensate loss of biodiversity due to development. A robust compensation strategy often requires undertaking conservation actions in areas much larger than the ones impacted by development activities. In offsetting systems, multipliers are commonly considered in order to account for disparities between negative consequences of development and benefits from compensation, as well as risks inherent to compensation actions (e.g., uncertain benefits of restoration).

For example, in the GUI (see Chapter 4, *RobOff Graphical User Interface*, p. 87) you can follow this sequence of steps, assuming that there is a development action (`devel`) and at least a compensation action. This procedure is applicable

to cases where the amount of resources that are available for conservation can be expressed as proportional to the amount or impact of development actions (e.g. extent of urban development or amount of habitat loss).

1. Set cost of action `devel` to a negative value that represents the availability of resources to compensate a development action. This is done in the `Costs` tab of the setup section.

2. Set the total available budget to 0 (In the `General settings`) tab of the setup section.

3. Set as mandatory allocation a certain amount of area to the `devel` action. This amount may have been decided in advance in a development plan or it may be under consideration. In the latter case you might want to explore a range of amounts.

4. Go to the optimize section of the GUI and obtain an optimal allocation of resources. The budget available from the negative cost of the `devel` action will be best allocated to maximize the sustainability ratio that you select.

5. By looking at the sustainability ratios in the summary table (`Summary` tab of the results section), you can easily conclude if no-net-loss is achieved. Normally this will depend on whether the robust sustainability indices are equal or greater than 1 (or a reference value obtained for satisfactory conditions and actions).

6. To find the exact minimum amount of resources that would have been needed to achieve no-net-loss, you need to generate a budget analysis plot in the `Budget` tab of the results section. In the budget analysis plot, the x coordinate (monetary amount) at which the performance ratio attains a value of 1, is the minimum amount that answers the question of how much compensation is enough. Note that if the amount of resources available was not enough, you will need to increase the maximum of the budget range for which the budget analysis is performed.

5.6. Interactions

The RobOff conceptual model does not account for process-based planning and dynamic interactions between features through time. However, it is for instance possible to enter interactions as simple biodiversity features or score features which are sums or products of several interacting factors. See Section 3.5, “What RobOff does not do directly”, p. 82 for more details on the limitations of RobOff regarding interactions, and how to incorporate simple models of interactions in RobOff setups.

Part VI. Tutorial and examples

Table of Contents

Chapter 6. Tutorial and examples

6.1. Aim

This tutorial illustrates the practical use of RobOff in different planning contexts. Some of the example setups included in this tutorial (and the RobOff distribution) can serve as a starting point when preparing new setups for modeling your own planning problems.

The following examples illustrate different types of planning problems. Typically different variants of a same base case need to be explored. This involves many different aspects or dimensions, including ecological uncertainty and uncertainty related to human decisions. More specifically, dimensions that need to be considered include: uncertainty horizon (in responses of biodiversity features), time preference (different discounting rates and different models) range of budget availability, range of costs, different schemes for weighting environments and species, and alternative robustness requirements. These trade-offs are illustrated as far as possible but it is far beyond the scope of this tutorial to explore all the possible variants.

Note

This tutorial is currently work in progress. Several conservation planning projects are currently ongoing and new material is expected soon. Do not hesitate to contact us for further information or if you find any issue. Keep an eye on our website for updates! http://www.helsinki.fi/bioscience/consplan.

6.2. A first contact with the RobOff GUI

The aim of this example is to learn how to visualize a setup and the results obtained, and how to find optimzal allocations of resources.

If you prefer not to use the graphical interface of RobOff just skip this section. See Section 6.3, “A first contact with the RobOff command line interface”, p. 120 for a first contact with the RobOff command line.

To start using the RobOff GUI, follow these steps:

- Find the 'first-contact-gui' directory in the example setups included with the RobOff distribution and open the setup that can be found inside (file first-contact-gui.ro_setup). This setup is already complete and consistent. You do not need to know how to define it or change its core components.
- Once you have loaded the setup, you can inspect the environments, features, actions, responses and other entities included in the different tabs of the setup section. The `General settings` tab includes important parameters such as the uncertainty horizon, planning horizon, and available budget. See Section 4.2, “Setup”, p. 89 for details on what are the components of the setup section of the GUI.
- You can visualize a summary of results and plots of conservation value and sustainability ratios over time in the results section. different types of analyses (like sensitivity to uncertainty, pairwise comparison of actions, or sensitivity to available budget) can be performed in the different tabs of the results section. See Section 4.3, “Results”, p. 96 for details on what results can be visualized in the results section.
- To obtain an optimal allocation of resources, go to the `optimize` section of the GUI. You can use the default settings or select a different performance criterion, robustness requirement level, and optimization method. Try different options and compare the solutions that you obtain on the optimal allocation tree, where you can visualize the amounts allocated to each action.

6.3. A first contact with the RobOff command line interface

The aim of this example is to learn how to edit the files that make up a RobOff setup, as well as the output files generated by RobOff. To learn how to find optimal allocations of resources by using the command line interface of RobOff.

If you prefer not to use the command line interface of RobOff just skip this section. See Section 6.2, “A first contact with the RobOff GUI”, p. 120 for a first contact with the RobOff graphical interface.

Follow these steps:

- Find the 'first-contact-command' directory in the RobOff distribution. If your current working directory is the root of the RobOff installation, the general

settings file (main file) for this setup is 'example-setups/first-contact-command/first-contact-command.ro_setup'. This setup is already complete and consistent. You do not need to know how to define it or change its core components. If you want to inspect the setup contents you can do so by editing the general settings file with a plain text editor. From this file you can see what other files are included in the setup.

- Run the RobOff command line tool by using the `-s` or `--setup` option and the general settings file of the first contact setup.
- Inspect the results generated in the output directory. Several files are generated, see Section 3.4, “Standard RobOff output”, p. 70 for details. Some results are only generated if certain command line options are provided to the command line tool. This includes the uncertainty analysis, optimized allocation of resources, pairwise comparison of actions, and the sensitivity to budget availability. See Section 3.4.1, “Optional output files”, p. 75 for details on the available options and the optional files that are produced.
- To obtain an optimal allocation of resources, use the `--optmize-alloc` or `-p` option. You will obtain the corresponding optional file (7.optimzal.allocation.csv, see Section 3.4, “Standard RobOff output”, p. 70 for details). This will use the optimization options defined in the setup. Try different optimization alternatives by modifying the optimization options of the general settings file of this setup. For details on these options, such as `optimization criterion` and `optimization method`

6.4. A minimal example

This section describes what are the components required to build a minimal RobOff setup. The set of input files presented is not intended to model any actual planning case, but rather to a) illustrate what are the strictly necessary inputs, and b) serve as a starting point to develop your own setups.

Note

This setup can be found in the RobOff software distribution under the folder/directory `setup-minimal-toy-offsetting`.

6.4.1. Minimal set of input files

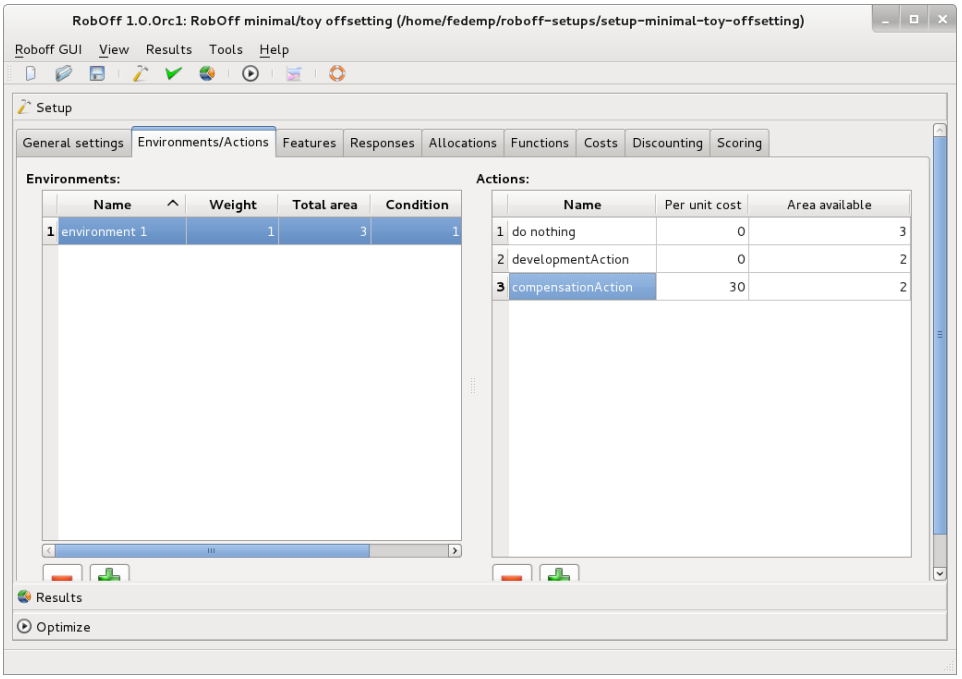
This example setup includes the following files:

- `toy-offsetting.ro_setup`: General settings file. Defines options such as the planning horizon, uncertainty horizon, time discounting parameter, names of files, etc.
- `environments.csv`: Environments file. Define the list of environments relevant to this problem.

- `features_present_environment 1.csv`: Feature presence file. Defines what features are present in an environment (environment 1 in this case, which is the only environment).
- `response_donothing.csv`: Response of the only biodiversity feature considered to the action "do nothing" or business-as-usual scenario. Two additional response files are included: `response_development.csv` and `response_compensation.csv`.
- `feature_weight_functiontypes.csv`: File of feature-weight-function types, where the feature specific weights and benefit functions are defined.
- `budget_allocation.csv`: Budget allocation file, containing a list of amounts allocated to different actions.

Setups can also be defined and modified interactively in the RobOff GUI. If you load this setup into the GUI you can see the contents of this setup in the different tables included in the tabs of the setup section. For example, the environments and actions defined are shown in two tables in the `Environments/actions` tab of the setup section of the GUI (see Figure 6.1, “Defining environments and actions in the setup section of the RobOff GUI”, p. 122)

Figure 6.1. Defining environments and actions in the setup section of the RobOff GUI



6.4.2. Output obtained

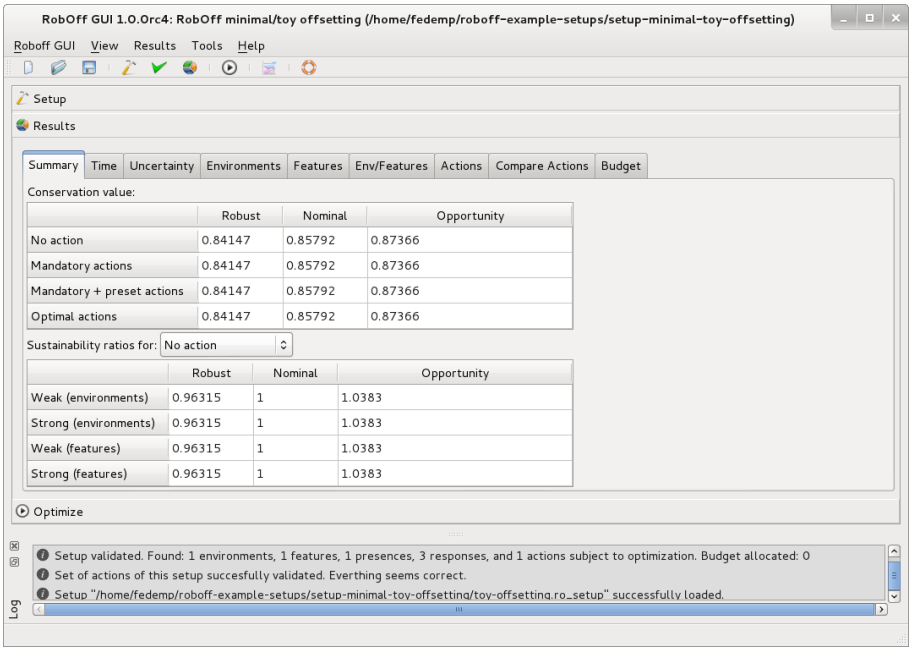
By running a command line, the summary shown below is obtained. A similar summary table is shown in the `Summary` tab of the results section of the GUI (Figure 6.2, “Summary of results for the minimal setup in the RobOff GUI”, p. 124). You can visualize different results in the multiple tabs of the results section. See Section 3.4, “Standard RobOff output”, p. 70 for details on the output files that are generated by the RobOff command line (or by saving results in the GUI). Alternatively, see Section 4.3, “Results”, p. 96 for details on the visualization functionality of the results section of the GUI.

```
   RobOff - software for allocation of conservation effort with multiple actions.
Loading configuration...
Reading RobOff setup file..toy-offsetting.ro_setup
Processing RobOff setup file...
No discounting configuration file found. Discounting model is: 'Quasi-hyperbolic' and
 rate: 0%
Reading environments file (./environments.csv). Environments #: 1
Environment 'environment 1', weight: 1, total area: 3, active area: 3, condition: 1
No budget allocation file name specified, skipping loading of budget allocations.
Reading features (weights, function types) file: ./feature_weight_functiontypes.csv
Reading specific responses:
reading file:./features_present_environment 1.csv
Looking for 3 feature response files, starting from 'compensation'...
Feature response file # 1: './response_compensation.csv'... Ok.
Feature response file # 2: './response_development.csv'... Ok.
Feature response file # 3: './response_donothing.csv'... Ok.
Setup validated. Found: 1 environments, 1 features, 1 presences, 3 responses, and 1
 actions subject to optimization. Budget allocated: 0
Set of actions of this setup succesfully validated. Everthing seems correct.
Starting core calculations of conservation value and sustainability...

=================  Summary of results:  ===================
Info-gap uncertainty horizon (alpha): 0.5 (range from  0 to 2)

 === Conserv. value (discounted, weak across features): ===
              Minimum     Nominal       Maximum
No action         0.8415      0.8579        0.8737

 ===== Conservation performance ratios (discounted): =====
                   Robust     Nominal   Opportunity
Weak for environments     0.9631       1.000         1.038
Strong for environments   0.9631       1.000         1.038
Weak for features         0.9631       1.000         1.038
Strong for features       0.9631       1.000         1.038

=================     End of output     ===================
Roboff finished ok.
```

To obtain the output above this command was used:

```
roboff --setup toy-offsetting.ro_setup --output example-output
```

Figure 6.2. Summary of results for the minimal setup in the RobOff GUI



6.5. A simple biodiversity offsetting example: Dam-Forest

Note

This setup can be found in the RobOff software distribution under the folder/directory `setup-dam-forest-offset`.

This setup considers 2 environments, 2 biodiversity features (1 per environment 2 environments, 1 feature, and 2 actions per environment.

The general settings file, `DamForest.ro_setup` contains the following definitions and options:

```
# RobOff setup saved automatically. Edit at your own risk!

name = Dam_example_offset_analysis

robust offsetting = 1

environment types = 2

# allow overwriting =  0

planning horizons = 0:10:100

info-gap alpha = 1

info-gap alpha range = 0:0.1:1

time discounting model = quasi-hyperbolic

time discounting rate = 3.5

feature responses = 5

environments file = environments
environments files prefix = features_present_

budget allocation file = budget_allocation

# These ones are not needed unless you don't like the default names
# response files prefix = response_
# feature weight function type file = feature_weight_functiontypes
# score features files prefix = score_features_
# benefit function files prefix = custom_benefit_function_

budget = 100000
mandatory budget = 0
preset budget = 0
budget resolution = .5

# optimization method = exhaustive

# optimization criterion = weak (features)

# robust (default) OR opportunity OR nominal

# optimization robustness requirement = robust
```

The environments file, `environments.csv`, contains two lines where environments are added to the setup:

```
# Environments file. Saved automatically by RobOff. Edit at your own risk!
# Environment types file, containing a table/list of environment types.
# Format:
# env_name, weight, total_area, area_no_action, condition,
 list_of_triplets(action_name, pau_cost, area_available)
#   (pau = per-area-unit cost)
MiddleForest, 1,  500.0,  0, 1, Inundate,  0,   300.0, Restore, 5.0, 200.0
UpperForest, 1, 30000.0,  0, 1, Restore, 5.0, 20000.0
```

The only feature present in the MiddleForest environment is defined in the presence file, `features_present_MiddleForest.csv`. In this file it is also

defined what are the responses of the present features to the actions that can be undertaken in this environment:

```
# RobOff - file of biodiversity features present in an environment: MT
# feature_name, present_estimate_not_used, response_no_action,
 list_of_quartets(action, response, init_value, env_value)
#
# dam construction will result in complete loss of 300 ha of total forest area of
 1000 ha
# management to control introduced pests could be applied in balance of forest
#
ForestTypeA, 0, r1_CC_env_MiddleForest_act_business-as-usual,
 Inundate, r2_CC_env_MiddleForest_act_Inundate, 0, 0, Restore,
 r3_CC_env_MiddleForest_act_Restore, 0, 0
```

As an example response, the response of the UpperForest environment to the Restoration action, `response_r5_CC_env_UpperForest_act_Restore.csv`:

```
# Feature specific response file. Saved automatically by RobOff. Edit at your own
 risk!
# assumes more rapid increase in ecological integrity as a result of intensive
 control of introduced
# browsers and predators
#
0.7,0.695,0.705
0.765,0.75505,0.77495
0.82,0.8051,0.8349
0.865,0.84515,0.88485
0.9,0.8752,0.9248
0.925,0.89525,0.95475
0.94,0.9053,0.9747
0.945,0.90535,0.98465
0.945,0.90535,0.9896
0.945,0.90535,0.99455
0.945,0.90535,0.9995
```

The feature-weight-function file defines equal weights for both features (1), and functions of time concave increase with diminishing return. This file, `feature_weight_functiontypes.csv`:

```
# Feature-weight-function types file. Saved automatically by RobOff. Edit at your own
 risk!
# Format:
#  feature_name, weight, benefit_function, comma_separated_list_of_parameters
ForestTypeA, 1.00, 1, 0.25
ForestTypeB, 1.00, 1, 0.25
```

Finally, a budget allocation file can be used to enforce certain actions and to constrain the optimization process. The following is an example, `budget_allocation.csv`:

```
[mandatory allocation]
MiddleForest, Inundate, 300
#MiddleForest, Restore, 200
#[preset allocation]
#MiddleForest, Restore, 200
```

6.6. A more elaborated biodiversity offsetting example: Dam-Forest-River

Note

This setup can be found in the RobOff software distribution under the folder/directory `setup-dam-forest-river-offset`.

This setup is an extension of the one presented in the previous section. It includes 5 environments, and 3 biodiversity features. The third feature, River, is present in the three environments that have been added: LowerRiver, MiddleRiver, and UpperRiver. 7 different actions are included.

The general settings file, `DamForestRiver.ro_setup` is similar to the one shown in the previous setup example. The environments file now looks like this. Note that in some of the new environments only one action (related to development) is possible:

```
# Environments file. Saved automatically by RobOff. Edit at your own risk!
# Environment types file, containing a table/list of environment types.
# Format:
# env_name, weight, total_area, area_no_action, condition,
 list_of_triplets(action_name, pau_cost, area_available)
#   (pau = per-area-unit cost)
MiddleForest, 1,  500.0,  0, 1, Inundate,  0,   300.0, Restore, 5.0, 200.0
UpperForest, 1, 30000.0,  0, 1, Restore, 5.0, 30000.0
MiddleRiver,  1,   20.0,  0, 1, Inundate,  0,   20.0
UpperRiver,  1,   300.0,  0, 1, Obstruct,  0,   200.0, ObstructPlusTT, 2.0, 200.0
LowerRiver, 1,   10.0,  0, 1, AlterFlow, 0,   10.0
```

The feature presence files for the river environments are as follows:

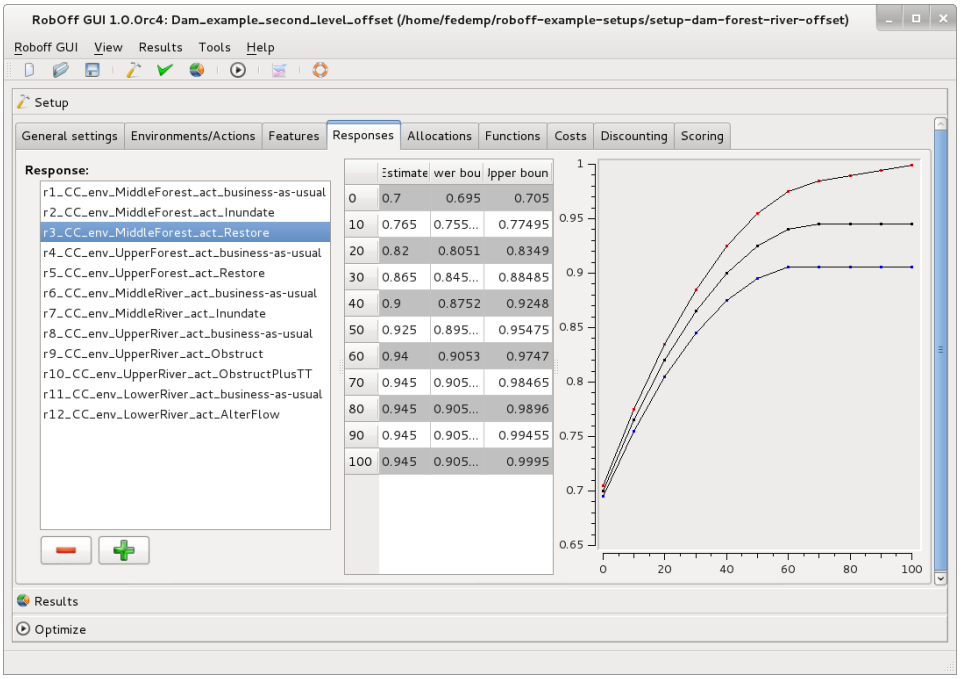
```
# RobOff - file of biodiversity features present in an environment: MT
# feature_name, present_estimate_not_used, response_no_action,
 list_of_quartets(action, response, init_value, env_value)
#
# accounts for effects of increased flow variability downstream of the dam due to
 fluctuating generation
#
River, 0, r11_CC_env_LowerRiver_act_business-as-usual,  AlterFlow,
 r12_CC_env_LowerRiver_act_AlterFlow, 0, 0
```

```
# RobOff - file of biodiversity features present in an environment: MT
# feature_name, present_estimate_not_used, response_no_action,
 list_of_quartets(action, response, init_value, env_value)
River, 0, r6_CC_env_MiddleRiver_act_business-as-usual, Inundate,
 r7_CC_env_MiddleRiver_act_Inundate, 0, 0
```

```
# feature_name, present_estimate_not_used, response_no_action,
 list_of_quartets(action, response, init_value, env_value)
#
River, 0, r8_CC_env_UpperRiver_act_business-as-usual, Obstruct,
 r9_CC_env_UpperRiver_act_Obstruct, 0, 0, ObstructPlusTT,
 r10_CC_env_UpperRiver_act_ObstructPlusTT, 0, 0
```

Remember that the responses of features to actions, and all the entities required to define a RobOff setup, can be created from scratch and edited in the RobOff GUI. The following screenshot shows the responses of the Dam-Forest-River example in the setup section of the GUI (for more details about defining setups and visualizing results with the GUI, see Chapter 4, *RobOff Graphical User Interface*, p. 87).

Figure 6.3. Feature responses of the Dam-Forest-River example in the RobOff GUI



Index

Symbols

A

B

C

D

E

F

G

H

I

S

T

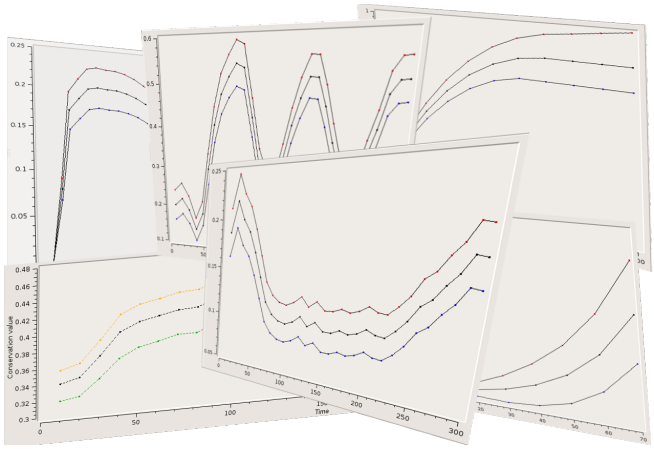
U

V

W

Z

Please check our website for the latest version and news:
www.helsinki.fi/bioscience/consplan



The following partners have supported the development and implementation of RobOff

European Research Council

UNIVERSITY OF HELSINKI

METSÄHALLITUS

ACADEMY OF FINLAND

Contact information:
Biodiversity Conservation Informatics Group
Finnish Centre of Excellence in Metapopulation Biology
University of Helsinki

Department of Biosciences
P.O. Box 65 (Viikinkaari 1)
Fi00014 University of Helsinki
Finland

ISBN: 978-952-10-8720-2 (paperback)
ISBN: 978-952-10-8721-9 (PDF)